

User's Manual

Renesas e² studio 2023 -10 or Higher Quick Start Guide

User's Manual

RA Family Renesas MCU

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

Renesas Electronics

Rev.1.01 Jul 2024

Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of the use of the area in the design of your product or system.
- these circuits, software, or information. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or other 2
- 3. others.
- 4
- 5.
- 6.

- Buildment, including but not be observed to the product data, returning, entends, programs, adjoint and springer of express.
 Bolicense, express, implied or otherwise, is granted hereby under any patients, copyrights or other intellectual property rights of Renesas Electronics or others.
 You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
 You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics or oducts are classified according to the following two quality grades. "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 "Standard": Computers; office equipment; communications equipment; industrial robots; etc..
 "High Quality": Transportation equipment; dautomobiles, trains, ships, etc.): traffic control (traffic lights); large-scale communication equipment; etc.
 Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics shall have absoluted or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; key plant systems; millary equipment; etc.). Renesas Electronics product, threads ender, user's manual or other Renesas Electronics product that is inconsistent with any Renesas Electronics shall have absolutely ave absolutely ave absolutely or security preserves inclusing but not limited to any unauthorized access to or use of a Renesas Electronics product, Renesas Electronics shall have absolutely not on 7.
- 8 specified ranges.
- specified ranges. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of 9
- Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
 It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
 This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics. Electronics products. 10

- Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

Trademarks

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

Contact information

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

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (Max.) and VIH (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (Max.) and VIH (Min.). Prohibition of access to reserved addresses

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

8. Differences between products

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



RA Family Renesas e² studio 2023 – 10 or Higher

Contents

1.	Overview	
1.1	System Configuration	
1.2	System Requirements	
1.2		
1.2	-1 5	
1.3	Supported Toolchains	
1.4	Supported Emulator Devices	
1.5	Outline of a RA Project Development	5
2.	Installation	6
2.1	Installing the FSP with e ² studio Installer	6
2.2	Installing e ² studio and FSP Independently	14
2.2		
2.2	•	
2.2		
2.3	Updating e ² studio	
2.4	Updating FSP	
2.5	Uninstalling e ² studio	
2.6	Installing RA SC for Keil MDK and IAR EWARM	
-		
3.	Project Generation	
3.1	Generating a New RA Project for a Non-TrustZone device	
3.2	Generating a New RA Project for a TrustZone device	
3.2		
3.3	Importing an Existing RA Project	
3.4	Generating and Using a RA Static Library	
3.4	5 , ,	
3.4		
3.5	RA Project Configuration Editor	47
3.5	.1 Summary Page	47
3.5	.2 BSP Page	48
3.5	.3 Clocks Configuration Page	49
3.5	.4 Pin Configuration Page	51
3.5	.5 Stacks Configuration Page	57
3.5		
3.5		
3.5		
3.6	Editor hover	
4.	Building	
4.1	Build Configurations	
4.2	Building a Sample Project	
4.3	Saving the Build Settings Report	72
5.	Debugging	73
5.1	Changing an Existing Debug Configuration	
5.2	Creating New Debug Configurations	
5.3	Basic Debugging Features	77



5.3	.1	Debug View	
5.3	.2	Breakpoints View	80
5.3	.3	Expressions View	82
5.3	.4	Registers View	84
5.3	.5	Memory View	85
5.3	.6	Memory Usage View	88
5.3	.7	Disassembly View	90
5.3	.8	Variables View	91
5.3	.9	IO Registers View	
5.3	.10	Eventpoints View	94
5.3	.11	Trace View	97
5.3	.12	Fault Status View	100
5.3	.13	Run Break Timer	101
6.	Settin	g up a FreeRTOS Application	
6.1	Gene	al Purpose Timer Example in FreeRTOS	102
6.2	Creat	ng the Sample Project	103
7.	Settin	g up an Azure RTOS Application	
7.1		al Purpose Timer Example in Azure RTOS	
7.2	Creat	ng the Sample Project	109
8.	Help.		114
Revisi	on His	ory	115



1. Overview

Renesas e² studio is the Integrated Development Environment for Renesas embedded microcontrollers. e² studio is based on the industry-standard open-source Eclipse IDE framework and the C/C++ Development Tooling (CDT) project, covering build (editor, compiler, and linker control) and debug phases with an extended GNU Debug (GDB) interface support.

The e² studio IDE supports the Renesas Flexible Software Package (FSP), an optimized software package designed to provide easy-to-use, scalable, high-quality software for embedded system design. The primary goal of FSP is to provide lightweight, efficient drivers that meet common use cases in embedded systems.

The e^2 studio IDE includes multiple Graphical User Interface (GUI) wizards for auto-generating code, including and configuring existing drivers, configuring build and debug options, and running the applications you create. Driver documentation is integrated in the form of tooltips, which are available in the code editor view.

The Renesas FSP support is included in e² studio releases 2022-07 (64-bit) and higher. Multiple views and editors are available to support specifically Renesas RA microcontrollers and the open-source GNU Arm Embedded Toolchain.

This user manual targets "Non-TrustZone device" and "Flat (Non-TrustZone) Project in TrustZone device."

The e² studio IDE also supports Arm[®] TrustZone[®] technology. Arm TrustZone technology divides the system and the application into Secure and Non-Secure partitions. e² studio helps users set up new TrustZoneenabled projects and provides debugging features for secure and non-secure applications on Renesas devices with Arm TrustZone technology. Refer to this link for more information about RA Arm TrustZone tools: <u>https://www.renesas.com/sg/en/document/apn/ra-arm-trustZone-tooling-primer</u>.

When using a 3rd-party IDE and toolchain, you can use the Renesas RA Smart Configurator to configure the software system (BSP, drivers, RTOS, and middleware) for a Renesas RA microcontroller.

Note: The contents displayed on the screen may differ slightly depending on the e² studio, the device used, and the FSP version.



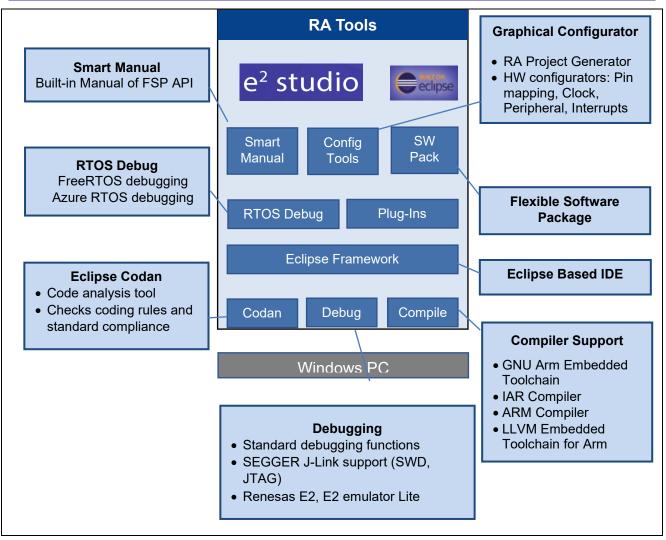


Figure 1-1. Renesas RA In e² Studio

1.1 System Configuration

A typical system configuration includes a host machine and a target board, as shown below.

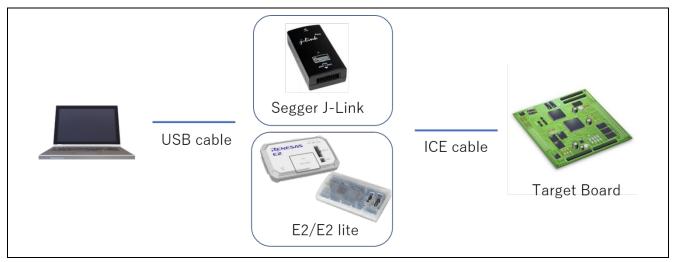


Figure 1-2. System Configuration



1.2 System Requirements

1.2.1 Hardware Environment:

- System: x64 based processor, 2 GHz or faster, CPU has dual cores or more
- Memory capacity: We recommend 8 GB or more. At least 4 GB.
- Capacity of hard disk: At least 2 GB of free space.
- Display: Graphics resolution should be at least 1024 x 768, and the mode should display at least 65,536 colors
- Interface: USB 2.0.
- Microsoft Visual C++ 2010 SP1 runtime library *1
- Microsoft Visual C++ 2015-2019 runtime library *1
- *1. This software will be installed at the same time as the e² studio

1.2.2 Operating Environment:

Architecture	OS	e² studio
64-bit version	Windows 11	2023-10
	Windows 10	
x64 based processor	Linux	2024-01.1
Apple Silicon (AArch64-based)	MacOS	2023-01

Note: 64-bit OS is required for e^2 studio 2022-07 and higher versions.

1.3 Supported Toolchains

- GNU Arm Embedded Toolchain (version: 12.2 or 13.2-Rel1)
- LLVM Embedded Toolchain for Arm (version: 17.0.1)
- IAR Compiler 9.40.1 or later
- Arm Compiler (version: 6.19 or later)

1.4 Supported Emulator Devices

• SEGGER J-Link, E2, E2 emulator Lite

1.5 Outline of a RA Project Development

This document provides detailed instructions on how to start developing with Renesas RA. The main steps are outlined below. By understanding the main steps below, readers can relate better to the procedures described in Chapters 3 and 4.

- Generating a RA project
- Configuring the RA project to fit hardware specifications such as clock, ICU, and pin functions
- Configuring the FreeRTOS.
- Configuring the Azure RTOS.
- Configuring the BSP (selecting HAL driver models)
- Adding user code
- Building the project
- Configuring the debugger and launching debugging



2. Installation

The development tools can be installed using either the "FSP with e^2 studio Installer" or the standard e^2 studio Installer.

2.1 Installing the FSP with e² studio Installer

The FSP with e² studio Installer includes the e² studio tool, FSP packs, GCC and LLVM toolchain, and other tools required to use this software. To download and install the FSP with e² studio Installer, follow the steps below:

Visit the GitHub page of Flexible Software Package (FSP) for Renesas RA MCU Family: <u>https://github.com/renesas/fsp/releases</u>

Select FSP with e² studio Installer (for example, setup_fsp<version>_e2s_<version >.exe) and click on the link to download directly.

Release No	otes
Flexible Softwar	e Package (FSP) for Renesas RA MCU Family, version 5.0.0.
Minimum e2 stu	udio version for FSP 5.0.0 is e2 studio 2023-10
Download the F	SP with e2 studio Windows installer for this release, setup_fsp_v5_0_0_e2s_v2023-10.exe, from <u>here</u> .
Refer to https://	SP with e2 studio Linux AppImage for this release, setup_fsp_v5_0_0_e2s_v2023-10.AppImage, from <u>here</u> . <u>(en-support.renesas.com/knowledgeBase/19934358</u> for information on installing e2 studio and related onents in a Linux PC.
lf using IAR or K 10.exe, from <u>he</u> l	Keil MDK, download the Renesas Advanced Smart Configurator for this release, setup_fsp_v5_0_0_rasc_v202 r <u>e</u> .
All installers are	available in the Assets section of this release.
Refer to the <u>REA</u>	ADME.md in the FSP root folder for setup instructions, hardware details, and related links.
Tools	

Figure 2-1. Installation – Download the FSP Package

Run the installation file.

On the **Select Install Type** page, if you would like to customize the components to be installed, choose **Custom Install**, then click on **Next**.

It is recommended that new users select the **Quick Install** option to minimize the configuration steps. This option will install e² studio, FSP, and GCC ARM Embedded by default. The last step will not be shown if the user selects Quick Install.



Renesas RA Flexible Software Package (FSP) v5.0.0 with e ² studio 2023-10 Setup					
Install Type					
Select Install Type:					
Quick Install Default installation of e ² studio, FSP, Arm GNU Toolchain & LLVM Embedded Toolchair	o for Arm				
Custom Install Custom installation of e ² studio, FSP, Arm GNU Toolchain & LLVM Embedded Toolchai					
v202310202041 User: All Users < <u>B</u> ack <u>N</u> ext > Inst	all	Cance	I		

Figure 2-2. Installation – Select Install Type

You may use the default folder or change it on the welcome page by clicking on [Change...]. Click on Next to continue.

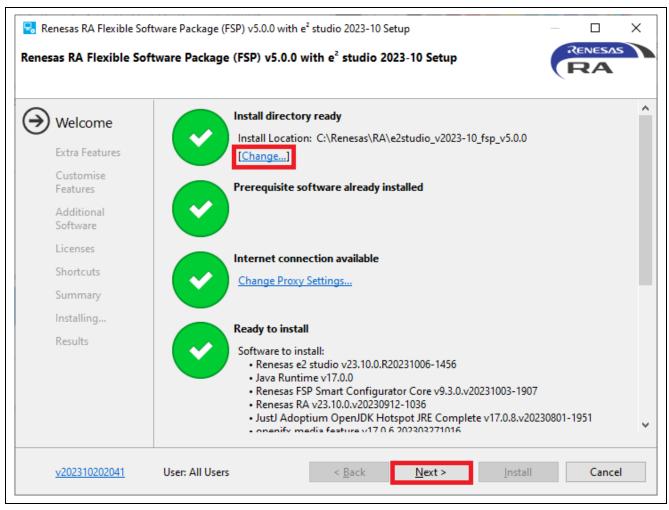


Figure 2-3. Installation – Welcome Page



On the Extra Features page, click on the required functions, then click on Next.

This page will not be shown if you select **Quick Install**.

Renesas RA Flexible Sof	Renesas RA Flexible Software Package (FSP) v5.0.0 with e ² studio 2023-10 Setup Renesas RA Flexible Software Package (FSP) v5.0.0 with e ² studio 2023-10 Setup Select the extra features you wish to install						
Welcome		P	Japanese Language Support				
Customise Features			Chinese (Simplified) Language Support				
Additional Software Licenses			Chinese (Traditional) Language Support				
Shortcuts Summary			Git Integration Git SCM Support				
Installing Results		P	Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet				
	Select A	AII					
<u>v202310202041</u>	User: All l	Jsers	< <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel				

Figure 2-4. Installation - Extra Features



Select the components you want to install on the Customize Features page and click on the **Next** button to continue. This page will not be shown if you select **Quick Install**.

🗟 Renesas RA Flexible Sc	oftware Package (FSP) v5.0.0 with e² studio 2023-10 Setup —	×					
Renesas RA Flexible So	Renesas RA Flexible Software Package (FSP) v5.0.0 with e ² studio 2023-10 Setup						
Welcome Extra Features	Renesas e2 studio (23.10.0.R20231006-1456)						
Oustomise Features	OpenJDK & OpenJFX Renesas FSP Smart Configurator Core (9.3.0.v20231003-1907) Common components for Renesas FSP Smart Configurator						
Additional Software	Renesas RA (23.10.0.v20230912-1036) Renesas RA Development Environment						
Licenses	Renesas RA Debug Support Files (23.10.0.v20231003-0928) Renesas RA Debug Support Files						
Shortcuts	Renesas e2 studio Common Components (23.10.0.R20231006-1456) Common components for Renesas e2 studio						
Summary Installing	Renesas e2 studio Common Components for ARM Devices (23.10.0.R20231006-1456) Common components for Renesas e2 studio						
Results							
	Git integration for Eclipse (6.7.0.202309050840-r)	*					
	Select All Optional Deselect All Optional Size of install: 741.2 MB						
<u>v202310202041</u>	User: All Users < <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel						

Figure 2-5. Installation - Customize Features

On the **Additional Software** page, select the "GNU ARM Embedded 12.2. MPACCBTI-Rel1" or "GNU ARM Embedded 13.2. Rel1", "LLVM Embedded Toolchain for Arm 17.0.1" and other necessary software to be installed, then click on **Next**. This page will not be shown if you select **Quick Install**.



	tware Package (FSP) v5.0.0 with e ² studio 2023-10 Setup tware Package (FSP) v5.0.0 with e ² studio 2023-10 Setup	_		×
Select the additional softwork Welcome Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results	are you wish to install Renesas RA Renesas FSP v5.0.0 GNU ARM Embedded 12.2.MPACBTI-Rel1 12.2-MPACBTI.Rel1 LLVM Embedded Toolchain for Arm 17.0.1 It.UVM Embedded Toolchain for Arm 17.0.1 Renesas QE QE for AFE QE for Motor 1.2.0 QE for Capacitive Touch 3.3.0 QE for BLE[RA,RE,RX] 1.6.0 QE for Display[RX,RA] 3.2.0 ✓ Renesas Reality Al for RA 23.10.0 Q Renesas Reality Al Data Storage Tool 1.1.0			
<u>v202310202041</u>	User: All Users < <u>B</u> ack <u>N</u> ext > <u>I</u> nsta		lownload re Cance	

Figure 2-6. Installation – Select Additional Software

Tick the checkbox to accept the license agreement, then click on **Next** to continue.

Welcome	Please read and accept the	following Software Agreements
Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing	Renesas e2 studio OpenJDK License Agreem ARM DS-5 Toolchain Integ Renesas FSP v5.0.0 GNU ARM Embedded 12.2 LLVM Embedded Toolchai Renesas Common Library	("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS. The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is
Results	✓ I accept the terms of the	Software Agreements Print

Figure 2-7. Installation – Software Agreements



Select the shortcut name for the start menu on the Shortcuts page and click the **Next** button to continue.

Note: If you already have installed e² studio in another location, it is recommended that you rename this installation to distinguish it from the other e² studio(s).

				Chartouto				
	<u>v202310202041</u>	User: All Users	< <u>B</u> ack	<u>N</u> ext >	<u>I</u> nstall		Cance	1
	Results							
	Installing							
	Summary							
()	Shortcuts							
	Licenses							
	Additional Software							
	Features					8	<u>R</u> estore De	fault
	Customise	🗹 ln start menu group:	Renesas RA v5.0.	0				
	Extra Features							
	Welcome	Shortcuts to important prog	rams and files wil	I be created in the	following location	ons:		
Kene:	sas KA Flexible Sof	tware Package (FSP) VS.V.V	with e studio	2023-10 Setup			RA	
Ponor	rac PA Elevible Sof	tware Package (FSP) v5.0.0	with o ² studio	2022 10 Setup		-	ENESAS	
号 Re	enesas RA Flexible Sof	tware Package (FSP) v5.0.0 with	e ² studio 2023-10	Setup		_		×

Figure 2-8. Installation – Shortcuts



Check the **Summary** and click on **Install** to continue.

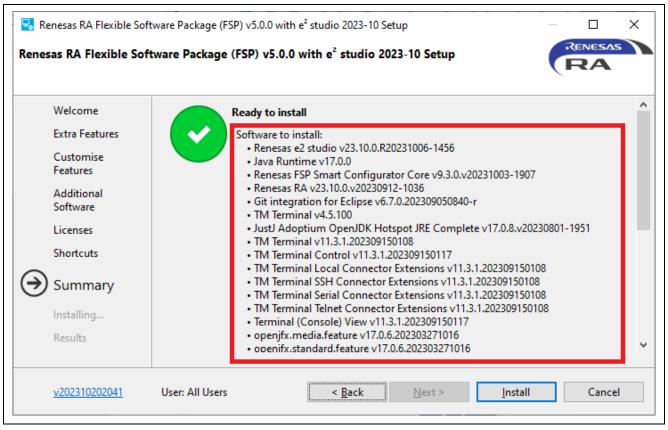


Figure 2-9. Installation – Summary

Click on **OK** to finish the installation.



🗟 Renesas RA Flexible S	Renesas RA Flexible Software Package (FSP) v5.0.0 with e ² studio 2023-10 Setup — 🛛 🗙					
Renesas RA Flexible S	oftware Package (FSP) v5.0.0 with e ² studio 2023-10 Setup					
Welcome Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing	Installation of Renesas RA Flexible Software Package (FSP) v5.0.0 with e² studio 2023-10 is complete. Please click OK to close. □ Launch e2 studio? ☑ View Release Notes? ☑ View What's New? ☑ View Renesas FSP User Manual? Useful Links: Renesas FSP: C:\Renesas\RA\e2studio v2023-10 fsp v5.0.0 Process FSP User Manual?					
<u>v202310202041</u>	User: All Users < <u>B</u> ack <u>N</u> ext > OK Cancel					

Figure 2-10. Installation – Complete Installation



2.2 Installing e² studio and FSP Independently

This section describes the independent installation of the following components.

- e² studio IDE
- GCC ARM Embedded Compiler
- Renesas Flexible Software Package (FSP)

2.2.1 Installing e² studio

To install e² studio for RA, follow these steps:

- 1. Download e² studio 2023-10 (64-bit version) offline installer from https://www.renesas.com/e2studio
- 2. Unzip the download file and run the e^2 studio installer to invoke the e^2 studio installation wizard page.
- 3. If an e² studio is installed on your PC, the options to modify, remove the existing version, and install the e² studio in a different location will be shown. It is possible to install multiple versions of e² studio by selecting **Install to a different location**. Click on the **Next** button to continue.

🛃 Renesas e² studio 2023-10 Setup	— 🗆 X			
Renesas e ² studio 2023-10 Setup	RENESAS			
e2 studio version 23.1.0.R20230106-1556 is already installed.				
What do you want to do?				
Upgrade Upgrade to version 23.10.0.R202310 Location: C:\Renesas\e2_studio	006-1456.			
Install Install to a different location.				
v202310061604 User: All Users < <u>B</u> ack	<u>N</u> ext > <u>Install</u> Cancel			

Figure 2-11. Install Multiple Versions of e² Studio



4. Install Type page:

Select the type of installation.

This page offers the [Lite Install] (installation in the Lite mode), [Standard Install] (installation in the advanced mode), and [Custom Install] (installation in the custom mode) options.

We recommend that you select [Lite Install]; however, [Custom Install] is selected here for the sake of explanation. Click on the **Next** button to continue.

🗟 Renesas e² studio 2023-10 Setup	—	×
Renesas e ² studio 2023-10 Setup	ΝΕ <mark>Σ</mark> Δ	S
Install Type		
Please select the e ² studio installation type. <u>Click here</u> for help selecting a type and to see what feature	es are included.	
Select Install Type:		
Lite Install (Recommended) This installs e ² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & debugging with	only important features	
Standard Install This installs e ² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features		
Custom Install Custom installation of e ² studio This mode is allows you to select which features are installed		
v202310061604 User: All Users < Back Next >	Install Cance	1

Figure 2-12. Installation – Install Type Page

5. On the Welcome page, the default installation location is set to C:\Renesas\e2_studio. You can click on [Change...] to modify it. Click on the Next button to continue.



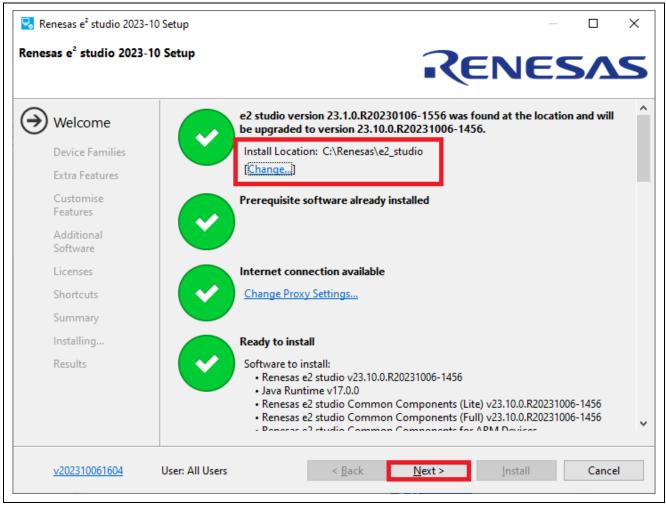


Figure 2-13. Installation – Welcome Page

6. Device Families page:

Check the checkbox for **RA**. Checkboxes of other device families are optional. Click on the **Next** button to continue.



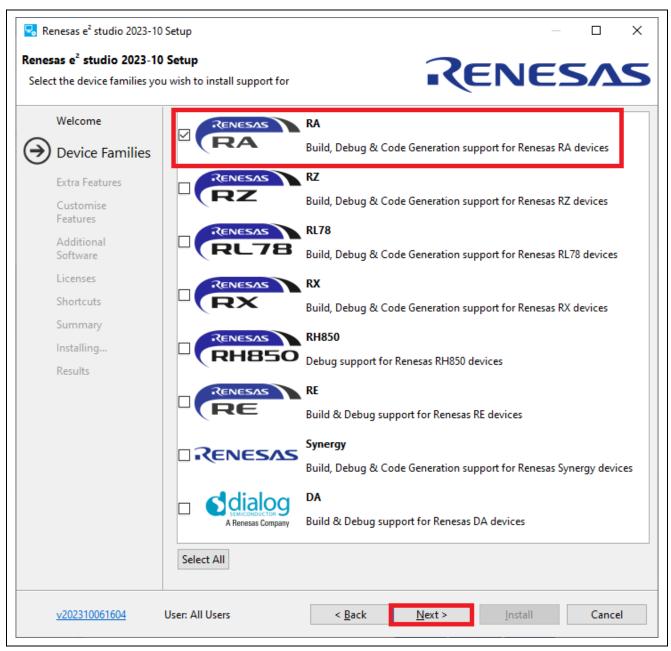


Figure 2-14. Installation – Device Families Page



7. Extra Features page:

Select **Extra Features** (that is, Language support, Git Integration, RTOS support, and so on) to install. For non-English language users, select the language to support at this step. Click on the **Next** button to continue.

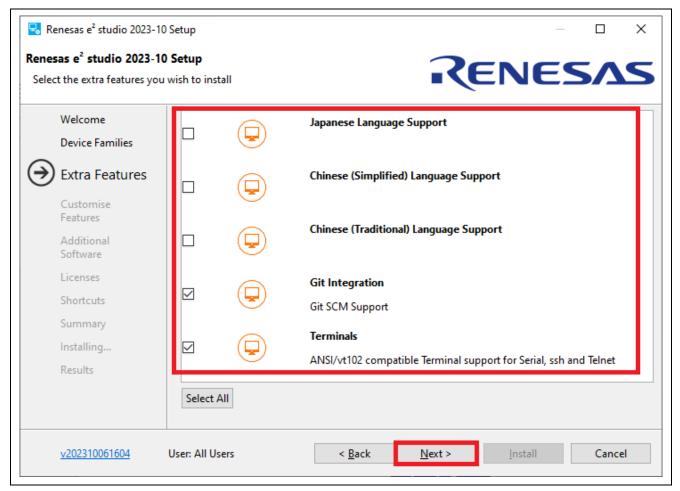


Figure 2-15. Installation – Extra Features Page



8. Customize Features page:

Ensure that **Renesas RA Family Support** is checked.

Click on the Next button to continue.

🛃 Renesas e² studio 2023-	10 Setup — 🗆	×
Renesas e ² studio 2023-		5
Welcome Device Families	Select the components you want to install.	^
Extra Features Customise Features	 ⁽¹⁾ Renesas e2 studio Common Components (Full) (23.10.0.R20231006-1456) Common components for Renesas e2 studio ⁽²⁾ Renesas e2 studio Common Components for ARM Devices (23.10.0.R20231006-1456) Common components for Renesas e2 studio 	
Additional Software Licenses	 	
Shortcuts Summary	Renesas RZ Family Support (23.10.0.R20231006-1456) Renesas RZ family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.	¥
Installing Results	Size of install: 741.2 MB Temporary space required: 1 Available: 120.9 GB	I.3 GB
<u>v202310061604</u>	User: All Users < <u>B</u> ack <u>Next</u> > <u>Install</u> Cancel	

Figure 2-16. Installation – Customize Features Page



9. Additional Software page:

Select the GCC Toolchains & Utilities tab and check the GNU Arm Embedded 12.2. MPACBTI-Rel1 or "GNU ARM Embedded 13.2. Rel1", "LLVM Embedded Toolchain for Arm 17.0.1" check box to install the GNU Arm Embedded toolchain.

Click on the Next button to continue.

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

😼 Re	enesas e ² studio 2023	-10 Setup						×
	as e² studio 2023 t the additional softw		to install		EN	ES	5	S
	Welcome Device Families			hecking Program V1.02.00 If-Checking Program V1.00.00 Utilities	v1.02.00 v1.00.00	1000 B 1000 B		^
	Extra Features			Ided 12.2.MPACBTI-Rel1	12.2.1.20230214	1000 B		
	Customise Features		GNU AKM Embed GNU ARM Embed GNU ARM Embed	ded 9.3.1 2020q2	10.3.0.202110 9.3.1.2020q2	1000 B 1000 B 1000 B		
(\rightarrow)	Additional		LLVM Embedded	Toolchain for Arm 17.0.1	9.2.1.2019q4 17.0.1	937.8 MB	1	
\smile	Software Licenses		Renesas FSP Renesas FSP v5.1.(Renesas FSP v5.0.)		5.1.0 5.0.1	117.7 MB 114.9 MB		J
	Shortcuts					1.2 MB dow	nload re	quired
	Summary v202310061604	User: All Us	sers	< <u>B</u> ack <u>N</u> ext >	<u>I</u> nstal		Cance	I

Figure 2-17. Installation – Additional Software Page

10. Licenses

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

Please note that you must accept the license agreement, or installation cannot proceed.

11. Shortcuts

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

12. Summary

Click on the Install button to install Renesas e² studio.

13. Installing...

The installation will start. Depending on the items selected in the "Addition Software" dialog, new dialogs may open to proceed with the installation of these software packages.



2.2.2 Setting Up the GNU Arm Embedded Toolchain

The GNU Arm Embedded Toolchain can be installed during e² studio installation. Alternatively, after e² studio has been installed, the GNU Arm Embedded Toolchain can be installed separately.

To install the GNU Arm Embedded Toolchain, follow these steps:

- 1. Download version 12.2.rel1 of the GNU Arm Embedded Toolchain supported by Renesas RA (arm-gnutoolchain-12.2.rel1-mingw-w64-i686-arm-none-eabi.exe) from <u>https://developer.arm.com/downloads/-</u> /arm-gnu-toolchain-downloads
- 2. Run the installer to install the GNU Arm Embedded Toolchain on the host machine.
- 3. Select the installation language. Click on **[Yes]** in the installation confirmation dialog.
- 4. Keep all default settings in the installation wizard.
- 5. When the **Install wizard Complete** dialog appears, check the box **Add path to an environment variable** and click on **[Finish]** to complete the installation.

2.2.3 Installing the Renesas RA Flexible Software Package (FSP)

To install the FSP, follow these steps:

- 1. Visit the GitHub page of Flexible Software Package (FSP) for Renesas RA MCU Family: https://github.com/renesas/fsp/releases
- 2. Find and download the latest FSP packs installer (for example, FSP_Packs_<version>.exe). The FSP Package Installer includes the driver library, HTML User's Manual, and a readme file.

• FSP_Packs_v5.0.0.zip 669b14f09baddf3f124bc22f8cd11a84		
• FSP_Packs_v5.0.0.exe e6c10f42bfa6a2bfaa59a5b4abfc4e12		
• fsp_documentation_v5.0.0.zip cbc95d60af9955fdc161af048	8a026268	
 setup_fsp_v5_0_0_e2s_v2023-10.exe e287c88e5ac12f7b829 	96a1a2af9cf6f	
• setup_fsp_v5_0_0_e2s_v2023-10.AppImage 7801a3a269e86	639200f9ab66bc3734ab	
• setup_fsp_v5_0_0_rasc_v2023-10.exe 27c30bd4707d97fd79	079230c6cbe3b28	
 setup_fsp_v5_0_0_rasc_v2023-10.AppImage 47bc45b2ca83 	88d9b77d82aa56499f6a	
MDK_Device_Packs_v5.0.0.zip bac7c614a09d2718becc62b1	17e760adb	
MDK_Device_Packs_v5.0.0.zip bac/c614a09d2/18becc62b1 Assets 10	17e760adb	
	17e760adb 23.2 MB	4 days ago
Assets 10		4 days ago 4 days ago
Assets 10 Øfsp_documentation_v5.0.0.zip	23.2 MB	, ,
Assets 10 Øfsp_documentation_v5.0.0.zip ØFSP_Packs_v5.0.0.exe	23.2 MB 137 MB	4 days ago
Assets 10 Øfsp_documentation_v5.0.0.zip ØFSP_Packs_v5.0.0.exe ØFSP_Packs_v5.0.0.zip	23.2 MB 137 MB 115 MB	4 days ago 4 days ago

Figure 2-18. Installation – Download the FSP Packs



- 3. Ensure that a compatible e² studio was installed and closed during this installation.
- 4. Run the FSP packs installer and click on **Next** to continue.
- 5. Click on I Agree to accept the agreement.
- 6. Browse to the folder where e² studio is installed (for example, C:\Renesas\e2_studio) and click on Install.

C	Renesas FSP v5.0.0 Setup			_		×
_		Choose Install				
-	CENESVS	Choose the fold	er in which to inst	tall Renesas FSP \	/5.0.0.	
	The installation path must po C:¥Renesas¥e2_studio). Ple					
	Browse to folder where e2	studio is installed				
	C:¥Renesas¥e2_studio¥			B <u>r</u> ov	vse	
	Space required: 190.9 MB					
	Space available: 571.1 GB					
			< <u>B</u> ack	Install	Cano	el
			- Endr	2. 10 com	Curre	

Figure 2-19. Choose Install Location

7. Click on **Finish** to finish the installation.

2.3 Updating e² studio

To update e^2 studio, run the new version of e^2 studio installer (either **FSP with e^2 studio installer** or standard e^2 studio installer). Download the installer according to Chapter 2.

Please note that you should not overwrite an existing installation. Before the IDE upgrade, users must uninstall the old version of e^2 studio. However, to keep both old and new e^2 studio versions, you can create a new folder as an installation destination for the new e^2 studio version.

2.4 Updating FSP

To update FSP, run the new version of FSP installer. Please download the installer according to Chapter 2.2.3.

2.5 Uninstalling e² studio

Users can uninstall e² studio by following the typical steps to uninstall a program on the Windows 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 on **Uninstall** to confirm the deletion in the **Uninstall** dialog.



At the end of the uninstallation, e^2 studio will be deleted from the installed location, and the shortcut in the Windows menu will be removed.

Note: If you have installed e² studio at multiple locations, you may not be able to find the uninstaller in **Apps** & features of the Control Panel. In such cases, launch the e² studio uninstaller located at: {e2 studio installed folder}/uninstall.exe.



2.6 Installing RA SC for Keil MDK and IAR EWARM

The RA Smart Configurator (RA SC) is a desktop application designed to configure device hardware, such as clock setup and pin assignment, as well as initialization of FSP software components for a Renesas RA microcontroller project when using a 3rd-party IDE (Keil MDK and IAR EWARM) and toolchain.

To download and install the RA SC Installer, follow the steps below:

1. Visit the GitHub page of Flexible Software Package (FSP) for Renesas RA MCU Family: https://github.com/renesas/fsp/releases

Search for the RA SC installer and download it (for example, setup_fsp<version>_rasc_<version</pre>>.exe).

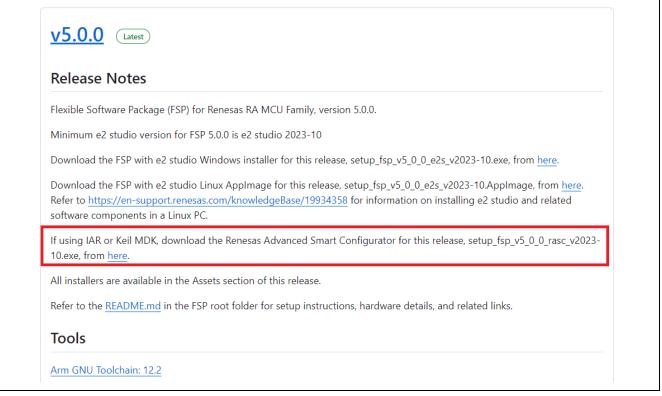


Figure 2-20. Installation – Download the RA SC Package

2. Run the installation file.

On the welcome page, you may use the default folder or change it by clicking on **[Change...]**. Click on **Next** to continue.



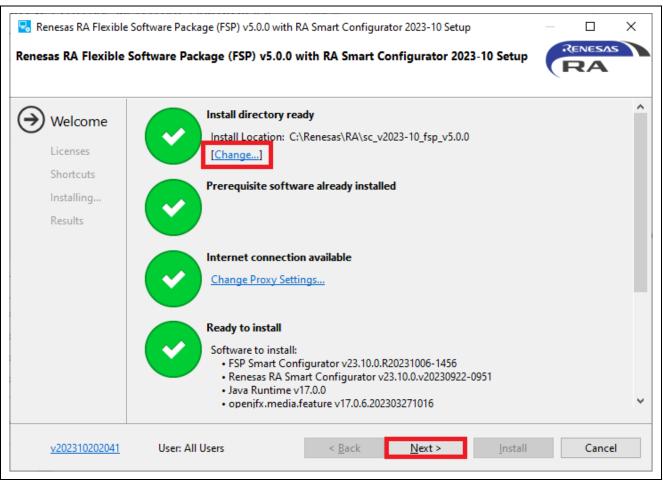


Figure 2-21. Installation – Welcome Page

3. Tick the checkbox to accept the license agreement and click on **Next** to continue.



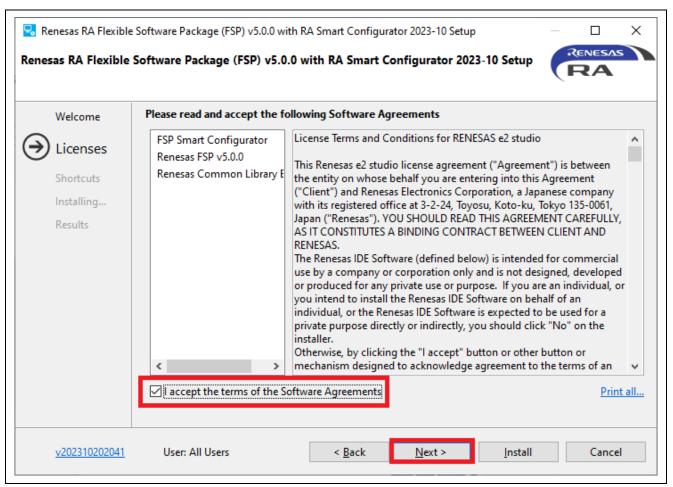


Figure 2-22. Installation – Software Agreements

4. Check the Shortcuts and click on **Install** to continue.

		0 with RA Smart Configurator 2023-10 Setup 5.0.0 with RA Smart Configurator 2023-10 Setup		×
Welcome Licenses Shortcuts Installing Results	Shortcuts to important pro	grams and files will be created in the following locations Renesas Electronics Smart Configurator	<u>R</u> estore De	fault
<u>v202310202041</u>	User: All Users	< <u>B</u> ack <u>N</u> ext > <u>Install</u>	Cance	ł

Figure 2-23. Installation – Shortcut

5. Click on **OK** to finish the installation.



	Software Package (FSP) v5.0.0 with RA Smart Configurator 2023-10 Setup – \Box × Software Package (FSP) v5.0.0 with RA Smart Configurator 2023-10 Setup
Welcome Licenses Shortcuts Installing Results	Installation of Renesas RA Flexible Software Package (FSP) v5.0.0 with RA Smart Configurator 2023-10 is complete. Please click OK to close. Open Quick Start Guide? View Renesas FSP User Manual?
<u>v202310202041</u>	User: All Users < <u>B</u> ack <u>N</u> ext > OK Cancel

Figure 2-24. Installation – Complete Installation

- **Note:** Refer to "Renesas RA Smart Configurator Quick Start Guide" for more information on using RASC with Keil MDK and IAR EWARM.
 - You can refer to this document through the following steps:
 - (1) After installing RASC, open "rasc_quick_start.html" in the installed eclipse folder.
 - (2) Open from the RASC Help menu.

3. Project Generation

This chapter describes the creation of a new RA project. The e² studio includes a wizard to help quickly create a new RA project. This is achieved by the ability of the wizard to match the project to a particular RA device and board.

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

As a prerequisite, the FSP and the toolchain must be installed on the host machine as described in Chapter 0.

3.1 Generating a New RA Project for a Non-TrustZone device

This chapter describes how to generate an RA project for a non-TrustZone device. For a TrustZone device, please refer to Chapter 0

A simple project generation wizard is available in e² studio to generate a new RA project with a project name and the associated device and board, including board-level drivers.

Start the e² studio application and choose a workspace folder in the Workspace Launcher. To configure a new RA project, follow these steps:



1. Select File \rightarrow New \rightarrow Renesas C/C++ Project \rightarrow Renesas RA.

File	Edit Source Refactor Navigate	Search Projec	t Re	enesas Views Run Window Help			
	New	Alt+Shift+N >		Renesas C/C++ Project	>	Renesas Debug	
	Open File		C +	Makefile Project with Existing Code		Renesas RA	
\square	Open Projects from File System		C	C/C++ Project	Ī	, 🔤 U 🗸 👻	9 - 14 - 14
	Recent Files	>		Project			
	Close Editor	Ctrl+W	C++	Convert to a C/C++ Project (Adds C/C++ Nature)			E Out
	Close All Editors	Ctrl+Shift+W	69		- 1		There is

Figure 3-1. Project Generation – New Project Creation

2. Select Renesas RA: Renesas RA C/C++ Project template. Click on Next to continue.

📴 New C/C++ Projec	:t	- D X	
Templates for Rene	sas RA Project		
All C/C++	Renesas RA C/C++ Project Create an executable or static library C/C	C++ project for Renesas RA.	
?	< Back Next >	Finish Cancel	

Figure 3-2. Project Generation – Select RA Project

- 3. In the project generation wizard, enter the following project information:
 - Project name: enter a name, for example, **RA_Tutorial**
 - Use default location: Checked. If you want to create a project in a different location, uncheck this checkbox and enter a new location.
 - Click on **Next** to continue.

🖪 Renesas RA C/C++ Project			—	×
Renesas RA C/C++ Project Project Name and Location				Ź
Project name RA_Tutorial				
Use default location			Bro	owse
Choose file system: default 🗸				
You can download more Renesas packs here				
?	< Back	Next > Fin	nish Ca	ancel

Figure 3-3. Project Generation – New RA Project Generation Wizard



- 4. In the device selection dialog, enter device and tool information:
 - Board: EK-RA6M3
 - Toolchain version: Latest GNU Arm Embedded Toolchain approved for use with Renesas RA (for example, GCC ARM Embedded 12.2.1.arm-12-mpcbt)
 - Debugger: J-Link (ARM)
 - Keep all other fields as default.
 - Click on **Next** to continue.

Device Selection FSP Version: Board: EK-RA6M3 Device: R7FA6M3AH3CFC Core: CM4 Language: © C ◯ C++	Board Description Evaluation kit for RA6M3 MCU Group Visit https://www.renesas.com/ra/ek-ra6m3 to get kit user's manual, quick start guide, errata, design package, example projects, etc.
Toolchains	
GNU ARM Embedded ARM Compiler 6.15 ARM Compiler 6.16 ARM Compiler 6.12 ARM Compiler 6.17	Debugger J-Link ARM ~
10.3.1.20210824	

Figure 3-4. Project Generation – Device Selection

5. Build Artifact Selection: Executable RTOS Selection: No RTOS

📴 Renesas RA C/C++ Project		– 🗆 X
Renesas RA C/C++ Project Build Artifact and RTOS Selection		Ď
Build Artifact Selection	RTOS Selection	
 Executable Project builds to an executable file 	No RTOS	~
 Static Library Project builds to a static library file 		
Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static library project		
0	< <u>B</u> ack <u>N</u> ext >	Einish Cancel

Figure 3-5. Artifact And RTOS Selection



6. In the project template dialog, select a project template, for example, **Blinky**.

Project Template Selection			1
Project Template Selection			
Bare Metal - Blinky Bare metal FSP project that i the C runtime environment. [Renesas.RA.]] pack]	ncludes BSP and will blink LEDs if available. This project will initialize cloc	ts, pins, stacks,	and
			_
Bare Metal - Minima Bare metal FSP project that i [Renesas.RA	ncludes BSR This project will initialize clocks, pins, stacks, and the C runtir	ne environmen	t.
Bare metal FSP project that i		ne environmen	t

Figure 3-6. Project Generation – Project Template

7. Click on the **Finish** button to create a new project.

You may be prompted to open the **FSP Configuration** perspective. Click on **Yes** to open the perspective.

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

e² studio creates a new project with various views. Among them are the Project Explorer View, the RA Project Configuration Editor, and the Visualization View.



	roject Renesas Views Run Window Help				
🐔 🔅 🔳 🎋 Debug		A = A = A = A = A = A = A = A = A =	- 🖻 😳 🔆 🐐	s v 🤷 v	
\$\$. ▼ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$			ର୍ 🖻 🖬 ସ	/C++ 🔅 FSP C	onfiguration
Project Explorer 😒	□ Image: Big (RA_Tutorial) FSP Configuration S □ Image: Big (RA_Tutorial) FSP Configuration S □ Image: Big (RA_Tutorial) FSP Configuration S □ Image: Big (RA_Tutorial) FSP Configuration S	Generate Project Content	FSP Visualizatio		
> 🔊 Includes > 🔗 ra Proj	Select Pin Configuration	Export to CSV file 🛛 🖺 Configure Pin Driver Warnings			₽ >>
 > Zaragen Explo > src > a_stg > script Script RTFA6M3AH3CFC.pincft, RA_Tutorial Debug.Flat. > ⑦ Developer Assistance 	$ \begin{array}{ c c c c c } \hline & & & & & & & & & & & & & & & & & & $	Manage configurations RA Project Configuration Editor Value V	ર		
	Pin Function Pin Number Summary BSP Clocks Pins Interrupts	Event Links Stacks Components	▶ Legend		
Properties 🔀 🧖 Problems	Smart Browser	□ Pin Conflicts 🔀		Ŷ	8
Property	Value	0 items Description	Module	Pin	Location
		<			>

Figure 3-7. Project Generation – New Project Creation View

3.2 Generating a New RA Project for a TrustZone device

This chapter guides users to generate an RA project for a TrustZone device. For a non-TrustZone device, please refer to chapter 3.1.

Arm TrustZone technology allows you to create projects for both secure and non-secure applications. For more information about the RA Arm TrustZone tool, please see the link below.

https://www.renesas.com/document/apn/ra-arm-trustzone-tooling-primer.

Flat projects (projects that are not TrustZone enabled) create a self-contained ELF executable without a security partition, making it suitable for immediate execution on a target device.

3.2.1 Flat (Non-TrustZone) Project

To create a new Flat (Non-TrustZone) project, follow these steps:

- 1. From the menu, select File \rightarrow New \rightarrow Renesas C/C++ Project \rightarrow Renesas RA.
- 2. Select Renesas RA: Renesas RA C/C++ Project template. Click on Next to continue.
- 3. In the project generation wizard, enter the following project information:
 - Project name: Enter a name, for example, **RA_Flat**.
 - Use default location: Checked. If you want to create a project in a different location, uncheck this checkbox and enter a new location.
 - Click on the **Next** button to continue.



📴 Renesas RA C/C++ Project	– 🗆 X
Renesas RA C/C++ Project	-
Project Name and Location	
Project name	
RA_Flat	
Use default location	
Location: C:¥workspace¥RA_Flat	Browse
Choose file system: default 🗸	
You can download more Renesas packs here	
? < Back Next > Finis	sh Cancel

Figure 3-8. Project Generation – New RA Project Generation Wizard



- 4. In the device selection dialog, enter device and tool information:
 - Board: EK-RA6M4
 - Toolchain version: Latest GNU Arm Embedded Toolchain approved for use with Renesas RA (for example, GCC ARM Embedded 12.2.1.arm-12-mpacbti-3)
 - Debugger: J-Link (ARM)
 - Keep all other fields as default.
 - Click on **Next** to continue.

Device Select	tion		
FSP Version	-	Board Description	^
Board:	EK-RA6M4	Evaluation kit for RA6M4 MCU Group Visit https://www.enesas.com/ra/ek-ra6m4 to get kit user's	
Device	RTFAGMAAF3CF8	manual, quick start guide, errata, design package, example projects, etc.	
Core	CM33 ~	Device Details	-
Language	€c Oc++	Inustione Ves Pins 144 ¢	, , ,
Toolchains		Debugger	
GNU ARM I ARM Comp ARM Comp ARM Comp ARM Comp	iler 6.15 iler 6.16 iler 6.12 iler 6.17	J-Link ARM	~

Figure 3-9. Project Generation – Device Selection

5. Project type selection: Flat (Non-TrustZone) Project

Project Type Selection	
 Flat (Non-TrustZone) Project Renesas RA device project without TrustZone separation All code, data and peripheral settings will be configured in this project Renesas RA device will remain in secure mode EDMAC RAM buffers will automatically be placed in non-secure RAM TrustZone Secure Project Renesas RA device project for TrustZone secure execution All code, data and peripherals placed in this project will be initialized as secure Secure project settings such as TrustZone partitions, linker maps and a list of secure peripherals will be passed to a selected non-secure project After initialization, a call to the non-secure startup handler will be made TrustZone Non-secure Project Renesas RA device project for TrustZone non-secure execution All code, data and peripherals placed in this project will be initialized as secure Secure project settings call to the non-secure startup handler will be made TrustZone Non-secure Project Renesas RA device project for TrustZone non-secure execution All code, data and peripherals placed in this project will be initialized as non-secure Must be associated with a secure project or secure bundle Non-secure startup handler will be called after secure code initialization 	
 Must be associated with a secure project or secure bundle 	

Figure 3-10. Project Type Selection



6. Build Artifact Selection: **Executable** RTOS Selection: **No RTOS**

Build Artifact Selection	RTOS Sel	ection
 Executable Project builds to an executable file 	No RTOS	5
 Static Library Project builds to a static library file 		
 Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static libra 	y project	

Figure 3-11. Artifact and RTOS Selection

7. In the project template dialog, select a project template, for example, **Blinky**.

Kenesas KA C/	C++ Project	-
Project Templat	e Selection	4
Project Templa	te Selection	
•	Bare Metal - Blinky	
	Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, ar the C runtime environment.	d
	[Renesas.RA.iiiiiiii pack]	
	Bare Metal - Minimal	
	Bare Metal - Minimal Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas:RA.IIIIIII pack]	
Code Generatio	Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RA	
	Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RA.IIIIIII pack]	
	Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RA	
	Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RA	

Figure 3-12. Project Generation – Project Template



8. Right-click on the project name and select **Build Project**. The project is built without error.

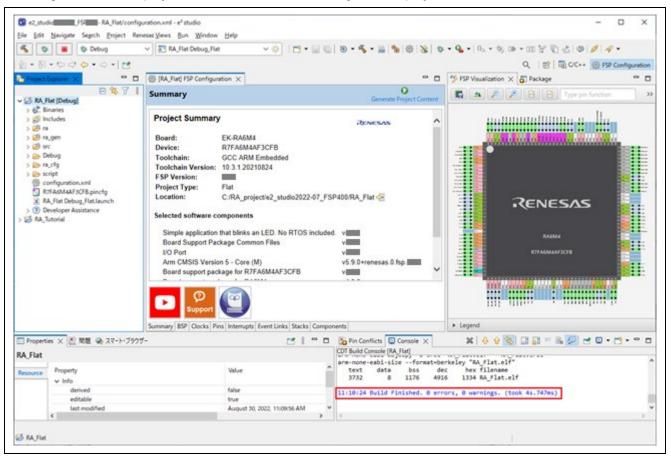


Figure 3-13. Project is Built Successfully



3.3 Importing an Existing RA Project

To import an existing RA Project, please follow the below steps.

1. Click on **File** \rightarrow **Import**.

File	Edit	Navigat	te S	Search	Project	t F	Renesas Vie	ws F
	New	F 11-					Alt+Shif	t+N>
~		File						
		Projects	from	File Sy	stem			
	Recer	nt Files						>
	Close	Editor					Ctr	+W
	Close	All Edito	rs				Ctrl+Shift	+W
	Save						Ct	rl+S
	Save	As						
B	Save	All					Ctrl+Shi	it+S
	Rever	t						
	Move							
2	Renar	me						F2
8	Refree	sh						F5
	Conv	ert Line D	elimi)	iters To				>
Ð	Print.						Ct	rl+P
è	Impo	rt						
4	Expor	t						
	Prope	erties					Alt+E	nter

Figure 3-14. Import Project

In the Import dialog, select General → Existing Projects into Workspace. Click on Next.
 Note: To rename the project to be imported, select General → Rename & Import Existing Projects into Workspace instead.

Import -	
Select	r ^y s
Create new projects from an archive file or directory.	
Select an import wizard:	
type filter text	
 General Archive File CMSIS Pack Existing Projects into Workspace File System Preferences Projects from Folder or Archive Rename & Import Existing C/C++ Project into Workspace Renesas CS+ Project for CA78K0R/CA78K0 Renesas CS+ Project for CC-RX and CC-RL 	^ ~
⑦ < Back Next > Einish	Cancel

Figure 3-15. Select the Type of Import

3. In the **Import Projects** dialog, select **Select archive file**, then **Browse...** to browse to the compressed file (.zip) containing the project.

If the existing project is stored in a folder, then **Select root directory** should be selected.



4. Select the project to import and click on **Finish**.

💽 Import		_	
Import Projects Select a directory to searc	ch for existing Eclipse projects.		
O Select root directory:	C:\Users\Renesas\e2_studio\workspac	۶/	B <u>r</u> owse
• Select <u>a</u> rchive file:	C:\Users\Vinh Loi\Desktop\RA_Tutoria	.zip ~ [B <u>r</u> owse
Projects:			
RA_Tutorial (RA_Tut	torial/)		Select All
			Deselect All
			Refresh
Working sets			
Add projec <u>t</u> to workin	ng sets		Ne <u>w</u>
Working sets:		~	S <u>e</u> lect
3	< Back Next >	<u>F</u> inish	Cancel

Figure 3-16. Select the Project in the Compressed File

5. The project will be imported to e^2 studio.

 ✓ ➢ RA_Tutorial [Debug] > ➢ Includes > ➢ ra > ➢ ra_gen > ➢ src > ➢ Debug > ➢ ra_cfg > ➢ script ※ configuration.xml ℝ7FA6M3AH3CFC.pincfg
ra_cfg.txtRA_Tutorial Debug_Flat.launch

Figure 3-17. The Imported Project

3.4 Generating and Using a RA Static Library

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

3.4.1 Creating the Static Library Project

The following steps show an example of creating an RA static library project.



- 1. Select File \rightarrow New \rightarrow Renesas C/C++ Project \rightarrow Renesas RA.
- 2. Select Renesas RA C/C++ Project template. Click on Next to continue.

New C/C++ I Templates for I	Project Renesas RA Project	_		×
All C/C++	Renesas RA C/C++ Project Create an executable or static libro project for Renesas RA.	ary C/C++		
?	< Back Next > F	inish	Canc	el

Figure 3-18. Project Generation – Select Library Project Template

3. On the project details page, enter a name for the static lib project (for example, RA_Lib) and click Next.

🕲 Renesas RA C/C++ Project	– 🗆 X
Renesas RA C/C++ Project Project Name and Location	
Project name RA_Lib	
Use default location Location: C:\workspace\RA_Lib Choose file system: default ~	Browse
You can download more Renesas packs here	
(?)	< Back Next > Finish Cancel

Figure 3-19. Library Project Configuration



4. In the **Device and Tool Selection** dialog, select a board (here, we will use EK-RA6M3). Keep everything else as default and click on **Next**.

Renesas RA C/C++ Project	
Device and Tools Selection	
Device Selection	
FSP Version:	Board Description
Board: EK-RA6M3	Evaluation kit for RA6M3 MCU Group
	Visit https://www.renesas.com/ra/ek-ra6m3 to get kit user's manual, quick start guide, errata, design package, example
Device: R7FA6M3AH3CFC	m projects, etc.
Core: CM4	Device Details
Language: C C++	TrustZone No ^
	Pins 176 🗸
	>
Toolchains	Debugger
GNU ARM Embedded	J-Link ARM V
ARM Compiler 6.15 ARM Compiler 6.16	1
ARM Compiler 6.16 ARM Compiler 6.12	
ARM Compiler 6.17	
10.3,1,20210824 ~	

Figure 3-20. Select Device and Toolchain

5. Build Artifact Selection: **Static Library** RTOS Selection: **No RTOS**

CIICK	on r	lext	10 (contil	nue.	

Build Artifact and RTOS Selection Build Artifact Selection	RTOS Selection
Executable Project builds to an executable file	No RTOS
 Static Library Project builds to a static library file 	
 Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static library project 	

Figure 3-21. Artifact and RTOS Selection



6. In the project template dialog, select **Bare Metal - Blinky**, then click **Finish** to create the project.

Renesas RA C/	C++ Project		_	4
Project Templat	e Selection			4
Project Templa	te Selection			_
•	Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project v the C runtime environment. [Renesas.RA.	vill initialize clocks, pins	, stacks, an	d
° 👩	Bare Metal - Minimal Bare metal FSP project that includes BSR This project will initialize clocks, pins, stacks, [Renesas.RA.	, and the C runtime envi	ronment.	
Code Generatio	· · ·			

Figure 3-22. Select Project Template for Library

- 7. The e² studio may prompt you to switch to the FSP Configuration perspective. Click on **Open Perspective** to open it.
- 8. Click on Generate Project Content.

e2_studic FSP A.Lib/confi File Edit Navigate Search Project R			×
🐔 🔯 🔳 🍁 Debug		000-5-0000	\$ • \$ • 1. • \$ = • m ¥ \$ & & Ø Ø Ø •
1-0-000-0-10			Q, 👔 🖬 C/C++ 🛞 FSP Configuration
Project Explorer ×	(RA_Flat) FSP Configuration (RA_Lib) FSP Configu	ration X	YSP Visualization × Package
E & FA		O Generate Project Cont	
SALLE (Debug) SB Includes SB incl	Project Summary Board: EK-RA6M3 Device: R7FA6M3AH3CFC Toolchain: GCC ARM Embedded Toolchain: GCC ARM Embedded Toolchain: GCC ARM Embedded Toolchain: GCC ARM Embedded Toolchain: CC ARM Embedded Project Type: Flat Location: C.TRA_project/e2_studio2022- Selected software components Simple application that blinks an LED. No RTOS is Board Support Package Common Files I/O Port Arm CMSIS Version 5 - Core (M) RA6M3-EK Board Support Files Device: Exerct Links Stacks	ncluded. v v v5.9.0+renesas.0.fsp. v	A Legend
	- Internet and the second seco		
🗌 Properties 🗙 💽 開題 🧠 スマート・ブラ RA Lib	19- 🖸 🖁 '	CDT Build Console (RA_Lib)	X 8 9 8 3 3 = 8 8 2 € 0 • 3 • • •
Property	Value	^	
Resource Property	wase		
derived	false		
editable	true		
last modified	August 30, 2022, 11:36:09 AM	× .	
<		> <	,

Figure 3-23. Generate Library Project Content



RA Family

9. From the Project Explorer window, open hal_entry.c under RA_Lib\src\.

lentry.c ⊠		
2	* * Copyright [2020] Renesas Electronics Corporation and/or its affiliates.	Al 💊
20		
21	<pre>#include "hal data.h"</pre>	
22	-	
23	<pre>void R_BSP_WarmStart(bsp_warm_start_event_t event);</pre>	
24		
25	<pre>extern bsp_leds_t g_bsp_leds;</pre>	
26		
28	• * @brief Blinky example application	
33	<pre> •void hal_entry (void) </pre>	
34	{	
35	<pre> #if BSP_TZ_SECURE_BUILD </pre>	
36	/* Entre and state */	
37	/* Enter non-secure code */	
38 39	<pre>R_BSP_NonSecureEnter(); #endif</pre>	
40	#eliuli	
41	/* Define the units to be used with the software delay function */	
42	<pre>const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS;</pre>	
43		
44	/* Set the blink frequency (must be <= bsp delay units */	
45	<pre>const uint32_t freq_in_hz = 2;</pre>	
46		
47	<pre>/* Calculate the delay in terms of bsp_delay_units */</pre>	
48	<pre>const uint32_t delay = bsp_delay_units / freq_in_hz;</pre>	\sim
	٢	>

Figure 3-24. Old hal_entry.c

Then rename the function <code>hal_entry()</code> to <code>hal_entry_lib()</code>, and add a declaration for <code>hal_entry_lib()</code>.

lal_entry.c ≈		- 0
2	• * Copyright [2020] Renesas Electronics Corporation and/or its affiliates.	A] 🔨
20		
21	<pre>#include "hal_data.h"</pre>	
22		
23	<pre>_void R BSP WarmStart(bsp warm_start_event_t event);</pre>	
24	<pre>void hal_entry_lib();</pre>	
25	extern bsp_leds_t g_bsp_leds;	
26		
28	<pre>* @brief Blinky example application.</pre>	
33	<pre>void hal_entry_lib (void)</pre>	
34	í.	
35	<pre> @#if BSP_TZ_SECURE_BUILD </pre>	
36		
37	/* Enter non-secure code */	
38	R_BSP_NonSecureEnter();	
39	#endif	
40		
41	/* Define the units to be used with the software delay function */	
42	<pre>const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS;</pre>	
43		
44	<pre>/* Set the blink frequency (must be <= bsp_delay_units */</pre>	
45	<pre>const uint32_t freq_in_hz = 2;</pre>	
46		
47	/* Calculate the delay in terms of bsp_delay_units */	
48	<pre>const uint32_t delay = bsp_delay_units / freq_in_hz;</pre>	~
	<	>

Figure 3-25. New hal_entry.c



Build the Library Project. The build outputs a static library file libRA_Lib.a.

 Project Explorer ≈ S RA_Lib [Debug] ▼ ■ Archives
 archives
> 🔯 libRA_Lib.a
> 🔊 Includes
> 🐸 ra
> 🐸 ra_gen
Y 🐸 src
> 🖻 hal_entry.c
> 🗁 Debug
> 🗁 ra_cfg
> 🗁 script
🏶 configuration.xml
R7FA6M3AH3CFC.pincfg
RA6M3-EK.pincfg
> ⑦ Developer Assistance

Figure 3-26. The Built Static Library



3.4.2 Using Static Library in Executable Project

This chapter shows how to use the static library created in the previous chapter (3.4.1) in an RA executable project by performing the following steps:

- Create an RA C executable project.
- Modify the source code to call a function "hal_entry_lib()" declared in the static library project.
- Build and run the RA C executable project.

Follow the following steps:

- 1. Select File \rightarrow New \rightarrow Renesas C/C++ Project \rightarrow Renesas RA.
- 2. Select Renesas RA C/C++ Project template. Click on Next to continue.
- 3. Enter the project name RA_App and click on Next to continue.
- 4. In the **Device and Tool Selection** dialog, select a board (here, we'll use EK-RA6M3). Keep everything else as default and click on **Next**.
- 5. On the **Build Artifact and RTOS Selection** page, select **Executable**. In RTOS selection, select **No RTOS**. Click on **Next**.

Note. The Build Artifact and RTOS Selection page, Unavailable is **"Executable Using an RA Static Library".**

Renesas RA C/C++ Project Build Artifact and RTOS Selection		
Build Artifact Selection	RTOS Selection	
 Executable Project builds to an executable file 	No RTOS	~
 Static Library Project builds to a static library file 		
 Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static library project 		

Figure 3-27. Artifact and RTOS Selection



6. In the project template dialog, select **Bare Metal - Minimal**. Click on the **Finish** button to create a project.

Renesas RA C/C++ Project	
Project Template Selection	
Project Template Selection	
Bare Metal - Blink	sy .
Bare metal FSP project the C runtime environme	hat includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and ent.
[Renesas.RA. pack]	
Bare Metal - Minin Bare metal FSP project ti [Renesas.RA.IIII pack]	hat includes BSR This project will initialize clocks, pins, stacks, and the C runtime environment.
Bare metal FSP project th	hat includes BSR This project will initialize clocks, pins, stacks, and the C runtime environment.
Bare metal FSP project ti [Renesas.RA.IIII pack]	hat includes BSR This project will initialize clocks, pins, stacks, and the C runtime environment.
Bare metal FSP project ti [Renesas.RA. pack] Code Generation Settings	hat includes BSR This project will initialize clocks, pins, stacks, and the C runtime environment.
Bare metal FSP project ti [Renesas.RA. pack] Code Generation Settings	hat includes BSR This project will initialize clocks, pins, stacks, and the C runtime environment.

Figure 3-28. Project Generation – Project Template



- 7. Add the Existing RA Library.
 - Set the Project Properties.
 - Select and add Settings \rightarrow Libraries \rightarrow Libraries (-I) $\stackrel{\textcircled{}}{42} \rightarrow$ RA_Lib.
 - Select and add Settings → Libraries → Libraries search path (-L) $\stackrel{\bigoplus}{=}$ → "\${workspace_loc:/RA_Lib/Debug}".
 - Select Apply and Close.

Properties for RA_App		$ \Box$ >	<
type filter text	Settings	← → ⇒ →	80
 > Resource Builders > C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor > C/C++ General > MCU Project Natures Project References Renesas QE Run/Debug Settings Task Tags > Validation 	 Tool Settings Toolchain P Build Steps Puild Artifact B Binar Target Processor Optimization Warnings Preprocessor Includes Warnings Miscellaneous Soft ARM Cross C Compiler Preprocessor Includes Optimization Warnings Miscellaneous Soft ARM Cross C Linker General Soft ARM Cross Print Size Soft ARM Cross Print Print	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	~
?	A	pply and Close Cancel	

Figure 3-29. Setting the Project Properties

8. From the Project Explorer window, open hal_entry.c under RA_App\src\.

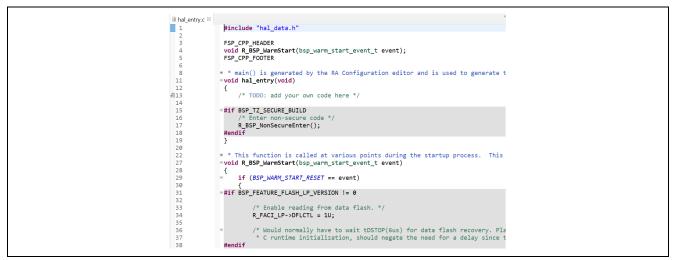


Figure 3-30. Old hal_entry.c



Remove function "R_BSP_WarmStart()" and its declaration.

Add codes to call the LED blinking library function "hal_entry_lib()" in the "hal_entry()" function and add a declaration for the library function.

🗟 hal_entry.c 🛛	
1	<pre>#include "hal data.h"</pre>
2	
3	ESP CPP HEADER
4	extern void hal_entry_lib()
5	FSP CPP FOOTER
6	
8	* main() is generated by the RA Configuration editor and is used to generate t
11	<pre>ovoid hal entry(void)</pre>
12	
@ 13	/* TODO: add your own code here */
14	hal_entry_lib()
15	#if BSP_1Z_SECURE_BUILD
16	/* Enter non-secure code */
17	R_BSP_NonSecureEnter();
18	#endif
19	}
20	
21	
22	#if BSP_TZ_SECURE_BUILD
23	
24	<pre>BSP_CMSE_NONSECURE_ENTRY void template_nonsecure_callable ();</pre>
25	
26	/* Trustzone Secure Projects require at least one nonsecure callable function i
27	BSP_CMSE_NONSECURE_ENTRY void template_nonsecure_callable ()
28	(
29	
30	}
31	#endif
20	

Figure 3-31. New hal_entry.c

- 9. Build the application project.
- 10. Set a breakpoint where the library function "hal_entry_lib()" is called. Run the RA_App project. When the program stops at the breakpoint, resume it. Confirm that the library function that blinks the LEDs (for example, "hal_entry_lib()") is executed.

P Threa arm-non	elf [1] [cores: 0] d #1 1 (single core) [core: 0] (Running) ie-eabi-gdb (7.8.2) GDB server (Host)	<			∴ >
 Anal_entry.c ≥ Anal_entry.c ≥<!--</th--><th><pre>#include "hal_data.h" FSP_CPP_HEADER extern void hal_entry_lib(); FSP_CPP_FOOTER * * main() is generated by th void hal_entry(void) { /* TODO: add your own co hal_entry_lib(); }</pre></th><th>e RA Configuration ed</th><th>itor :</th><th> Outline Correct Ex Project Ex RA_App [Debug] Stanaries Includes Debug Excrete Src App Debug,jlin RA_App Debug,jlin </th><th>6.4</th>	<pre>#include "hal_data.h" FSP_CPP_HEADER extern void hal_entry_lib(); FSP_CPP_FOOTER * * main() is generated by th void hal_entry(void) { /* TODO: add your own co hal_entry_lib(); }</pre>	e RA Configuration ed	itor :	 Outline Correct Ex Project Ex RA_App [Debug] Stanaries Includes Debug Excrete Src App Debug,jlin RA_App Debug,jlin 	6.4
RA_App Debug [F Hardware brea Hardware brea	Kks Problems Smart Browser Mem Renesas GDB Hardware Debugging] akpoint set at address Øxblc akpoint set at address Øx600 akpoint set at address Øxblc	nory			

Figure 3-32. Application Project Executing Library Function



3.5 RA Project Configuration Editor

The RA Project Configuration editor view displays the current project configuration settings. The settings are saved in the file "configuration.xml". The project configuration settings are grouped into multiple pages that allow you to set several configurable aspects of the project, such as how pins and clocks are set up and which drivers are included. Drivers can range from simple hardware-level drivers to RTOS-aware applications. Multi-thread specific components like mutexes, semaphores, and events can be configured.

To edit the project configuration, make sure that:

 FSP Configuration perspective is selected in the upper right-hand corner of the e² studio window or click on Window → [Perspective] → [Open Perspective] → Other... → [FSP Configuration] and the "configuration.xml" file is opened.

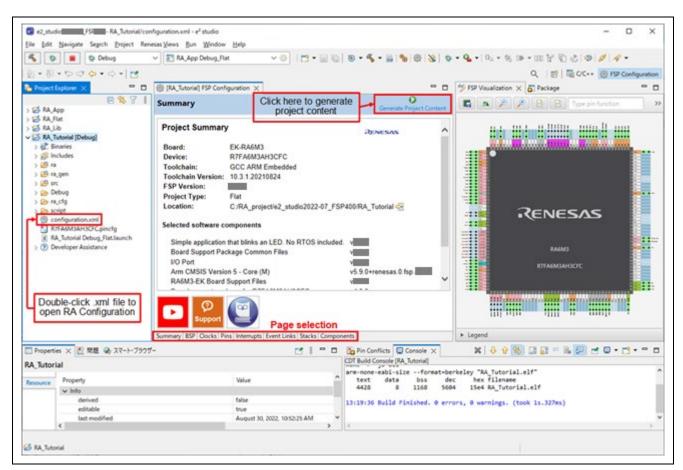


Figure 3-33. RA Project Configuration – RA Project Configuration View

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

The **Summary** page contains project-specific summary information.

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

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

3.5.1 Summary Page

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



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

Summary		Generate	Project Content	
Project Summary	/	RENESAS	^	
Board: Device: Toolchain: Toolchain Version: FSP Version: Project Type: Location:	EK-RA6M3 R7FA6M3AH3CFC GCC ARM Embedded 10.3.1.20210824 Flat C:/RA_project/e2_studio	_FSF /RA_Tutorial 😔		Information about board device, toolchain and FS
	that blinks an LED. No RTOS i kage Common Files on 5 - Core (M)	included. v v v v5.9.0+renesas.0. v	fsp	Software component included in the project
D P Support	Useful links			

Figure 3-34. Summary Page

3.5.2 BSP Page

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

Board Supp	ort Package	Configuration		Gener	Tate Project Content
		Click here to select	FSP version	1	💀 Restore Defaults
Device Selection	on		_	-	
FSP version:		_	Board Details	Click here to import CMSIS board	
Board:	EK-RA6M3	~ 🔤 *		information	to get kit user's
Device:	R7FA6M3AH3CFC		manual, quick st projects, etc.	art guide, errata, design paci	kage, example
Core:	CM4	~ ~		Click here to select	device
RTOS:	No RTOS	~		ARM core type	
	Click here to	select board			
ummary BSP (Clocks Pins Interr	upts Event Links Stacks Co	omponents		

Figure 3-35. RA Project Configuration – BSP Page



When the **BSP** page is selected, the e² studio **Properties** view will display the available properties for the selected board. According to the project's requirements, these properties may be modified in the Properties view as necessary.

	8471			0	
5 (5 RA)	App	Board Support Package Configur	ition	Gamerate Project Content	
> 15 RA. > 15 RA.	Ub			Rentore Defaults	
- SRA		Device Selection			
	a a_get	FS venion	Board Details fiveluation kit for RAV dat Visit https://www.ner	GMS MCU Group Interactory/ra/ek-colm3 to get kit user's	
0 🕮 s 0 👜 s	Debug	Device: R/FA6MSAH3CFC		uide, erista, design package, example	
200		Cove CM4			
8	configuration.xml UFA5M3AH3CFC.gincfg IA_Tutorial Debug_RatJaunch	RTOS No RTOS			
Proper	Developer Assistance tex: X 記 期間 @) スマート・ブラウ	Summary BSP Cocks Pins Interrupts Event L	inis Stacks Components		
EK-RA6N	M3		txtracting a	opport files * Incremental Build of configuration	
Settings	Property w RMAXMANOCIC part, number mm, size, bytes mm, size, bytes data, fisch, size, hytes data, fisch, size, hytes package, pins w RAGM3 Family > OFS0 register settings > OFS0 register settings	Nature R71 A0/USAH3CFC 2007/152 60536 60536 LOFF 176 6	make -r -j8 - arm-tone-esb text d 4428	all i-sizeformat-berkeley "RA_Tutori	

Figure 3-36. Board Properties

3.5.3 Clocks Configuration Page

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

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

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

Note: If the value goes out of range, it will turn red, and the "!" mark will be displayed.

				0
Clocks Configuration				Generate Project Content
				Restore Defaults
XTAL 24MHz			> ICLK Div /2	✓ → ICLK 120MHz
×	PLL Src: XTAL		PCLKA Div /2	✓ → PCLKA 120MHz
	PLL Div /2 ~		PCLKB Div /4	✓ → PCLKB 60MHz
	PLL Mul x20.0		PCLKC Div /4	✓ → PCLKC 60MHz
V USBMCLK 24MHz	PLL 240MHz	CIOCK DIG. TEE	✓ + PCLKD Div /2	✓ → PCLKD 120MHz
HOCO 20MHz V		Clock Src: HOCO Clock Src: MOCO	SDCLKout On	✓ → SDCLKout 120MH
LOCO 32768Hz		Clock Src: LOCO Clock Src: XTAL	∱ → BCLK Div /2	✓ → BCLK 120MHz
MOCO 8MHz		Clock Src: SUBCLK Clock Src: PLL	↓ BCLK/2	$\sim \rightarrow$ BCLKout 60MHz
SUBCLK 32768Hz			⇒UCLK Div /5	✓ → UCLK 48MHz
<			1	×

Figure 3-37. RA Project Configuration – Clocks Configuration



[RA_Lib] FSP Configuration	lahal_entry.c	Image: Image: # The second	uration [∞]] =
Clocks Configuration				Generate Project Conten
				🗟 Restore Default
XTAL 24MHz			ICLK Div /1	V → ICLK 240MHz
	> PLL Src: XTAL	~	→ PCLKA Div /2	✓ → PCLKA 120MHz
	PLL Div /2	~	→ PCLKB Div /4	✓ → PCLKB 60MHz
	PLL Mul x20.0	~	→ PCLKC Div /4	$\sim \longrightarrow$ PCLKC 60MHz
↓ USBMCLK 24MHz	PLL 240MHz	→ Clock Src: PLL	✓ ↔ PCLKD Div /2	$\sim \rightarrow$ PCLKD 120MHz
HOCO 20MHz v			SDCLKout On	$\sim \rightarrow$ SDCLKout 120MHz
LOCO 32768Hz			⇒ BCLK Div /2	$\sim \longrightarrow$ BCLK 120MHz
MOCO 8MHz			↓ BCLK/2	$\vee \longrightarrow BCLKout 60MHz$
SUBCLK 32768Hz			⇒ UCLK Div /5	✓ → UCLK 48MHz
			⇒ FCLK Div /4	✓ → FCLK 60MHz

Figure 3-38. Value Goes Out of Range

- 2. Save the Project Configuration Settings, for example, using the **Ctrl-S** shortcut.
- 3. Click on the Generate Project Content button.

Clocks Configurat	ion	Generate Project Content
XTAL 24MHz	PLL Src: XTAL ∨ PLL Div /2 ∨ PLL Mul x20.0 ∨	Restore Defaults ICLK Div /2 PCLKA Div /2 PCLKB Div /4 PCLKC Div /4
USBMCLK 24MHz HOCO 20MHz	PLL 240MHz Clock Src: PL	
LOCO 32768Hz		→ BCLK Div /2
MOCO 8MHz SUBCLK 32768Hz		BCLK/2

Figure 3-39. Generate Project Content



4. The file "bsp_clock_cfg.h" is updated with the selected clock configuration.

Project Explorer 🛛 🖻 🥵 🐨 🔨		帶 [RA_Tutorial	I] RA Configuration Bsp_clock_cfg.h ≅	- 0
😕 RA_Tutorial [Debug]	^	1	/* generated configuration header file - do not edit */	^
> 🗱 Binaries		2	<pre>#ifndef BSP_CLOCK_CFG_H_</pre>	
> 🔊 Includes	1.00	3	#define BSP_CLOCK_CFG_H_	
> 🥴 ra		4	#define BSP_CFG_XTAL_HZ (2400000) /* XTAL 24000000Hz */	
		5	<pre>#define BSP_CFG_PLL_SOURCE (BSP_CLOCKS_SOURCE_CLOCK_MAIN_OSC) /* PLL Src: XTAL */</pre>	
Y ❷ ra_gen		6	#define BSP_CFG_HOCO_FREQUENCY (2) /* HOCO 20MHz */	
> bsp_clock_cfg.h		/	#define BSP_CFG_PLL_DIV (BSP_CLOCKS_PLL_DIV_2) /* PLL_Div_/2 */	
> B bsp_pin_cfg.h		8	#define BSP_CFG_PLL_MUL BSP_CLOCKS_PLL_MUL_20_0 /* PLL Mul x20.0 */	
> 🗟 common data.c		9 10	#define BSP_CFG_CLOCK_SOURCE (BSP_CLOCKS_SOURCE_CLOCK_PLL) /* Clock Src: PLL */	
> 🖻 common data.h		10	#define BSP_CFG_ICLK_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_2) /* ICLK_Div /2 */	
		12	#define BSP_CFG_PCLKA_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_2) /* PCLKA_Div /2 */	
> 🖻 hal_data.c		13	#define BSP_CFG_PCLKB_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_4) /* PCLKB_Div_/4 */	
> hal_data.h		14	#define BSP_CFG_PCLKC_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_4) /* PCLKC_Div /4 */	
> 🖻 main.c		15	<pre>#define BSP_CFG_PCLKD_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_2) /* PCLKD Div /2 */ #define BSP_CFG_SDCLK_OUTPUT (1) /* SDCLKout On */</pre>	
> 🖻 pin data.c		16	#define BSP_CFG_BCLK_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_2) /* BCLK_Div_/2 */	
> 🖻 vector_data.c		17	#define BSP_CFG_BCLK_DIV (BSP_CLOCKS_STS_CLOCK_DIV_2) / BCLK DIV /2 // #define BSP_CFG_BCLK_OUTPUT (2) /* BCLK/2 */	
the second s		18	#define BSP_CFG_UCK_DIV (BSP_CLOCKS_USB_CLOCK_DIV_5) /* UCLK Div /5 */	
> 🖻 vector_data.h		19	#define BSP_CFG_FCLK_DIV (BSP_CLOCKS_OSD_CLOCK_DIV_S) / OCCK_DIV_/S //	
RA6M3-PK.csv		20	#define BSP_CFG_CLKOUT_SOURCE (BSP_CLOCKS_CLKOUT_DISABLED) /* CLKOUT Disabled */	
Y 🤒 SIC		21	#define BSP_CFG_CLKOUT_DIV (BSP_CLOCKS_SYS_CLOCK_DIV_1) /* CLKOUT Div /1 */	
> 🖻 hal entry.c		22	#endif /* BSP CLOCK CFG H */	
> Chebug	~	23		~
	>		<	>
Properties 🛛 🖹 Problems 🗣 S	man and Dave		📑 👻 🖶 🌆 Pin Conflicts 📮 Console 😫	

Figure 3-40. bsp_clock_cfg.h is Updated

3.5.4 Pin Configuration Page

The **Pin Configuration** page provides a graphical user interface for generating the pin configuration settings for the project.

🌼 [RA_Tutorial] FSP Configuration 🔀			=	· 🗆
Pin Configuration			Generate Project Cor	itent
Select Pin Configuration			🔚 Export to CSV file 🛛 Configure Pin Driver Warning	s
RA6M3-EK.pincfg	✓ Manage configurations	(1)	Generate data: g_bsp_pin_cfg	
Pin Selection $\textcircled{\blacksquare} = \downarrow^a_{z}$	Pin Configuration		😲 Cycle Pin Gro	up
Type filter text	Name	Value	Link	
> ♥ Peripherals > ♥ Other Pins		(3)		
(2)	<			>
Pin Function Pin Number				
Summary BSP Clocks Pins Interrupts	Event Links Stacks Components			

Figure 3-41. RA Project Configuration – Pin Configuration GUI

The Pin Configuration window consists of 3 parts:

1. **Select pin configuration**: Select the pin configuration file and specify the name for the associated data structure. Clicking on [Manage configurations] allows the duplication or removal of the pin configuration.



Multiple Pin Configuration Management Modify pin configuration list or import/export external file RA6M3-EK.pincfg (Current) R7FA6M3AH3CFC.pincfg Remove Rename Duplicate Merge to Import Export	Manage Pin Configurations	— 🗆 X
R7FA6M3AH3CFC.pincfg Remove Rename Duplicate Merge to Import		
Export		Remove Rename Duplicate Merge to

Figure 3-42. RA Project Configuration – Merge Pin Configurations

[Add]: Adds a new pin configuration file.

[Remove]: Removes the selected pin configuration file.

[Rename]: Changes the name of the pin configuration file.

[Duplicate]: Duplicates the selected pin configuration file.

[Merge to]: Merges the peripheral channels and pin numbers into the selected pin configuration file. [Import]: Imports a pin configuration file to be used in another configuration file.

[import]: imports a pin conliguration life to be used in another conliguration life.

[Export]: Exports the selected pin configuration file into another configuration file.

- 2. **Pin Selection**: Select a pin or peripheral that will be set up.
- 3. **Pin Configuration**: Sets up for function/property of the selected pin / peripheral.

The best way to configure pins is to configure the peripherals to be used in the project using the steps below:

- 1. Select a peripheral in the **Pin Selection** pane, for example, **Connectivity: SCI** → **SCI4**. The configuration for this peripheral will be shown in the **Pin Configuration** pane.
- 2. Select an Operation Mode for the peripheral, for example, Simple SPI.
- 3. Select the pins you would like to use for the input/output functions of the selected peripheral in the selected mode.

Note: You can jump to the port setting by clicking on the arrow on the right side of the pin.



IRA_Tutorial] FSP Configuration ⊗ Pin Configuration					0
Select Pin Configuration		📑 Export	to CSV file		Generate Project Content ure Pin Driver Warnings
RA6M3-EK.pincfg	✓ Manage configurations		Generate data	a: g_bsp_	pin_cfg
Pin Selection $\textcircled{\blacksquare} \textcircled{\downarrow}^{a}_{\mathbf{Z}}$	Pin Configuration				😲 Cycle Pin Group
Type filter text Analog:CMP Analog:DAC12 Connectivity:CAN Connectivity:ETHERC Connectivity:IIC Connectivity:SCI SCI0 SCI1 SCI2 SCI3 (1) SCI4	Name Pin Group Selection Operation Mode V Input/Output TXD_MOSI RXD_MISO SCK CTS_RTS_SS SDA SCL	Value Mixed Simple SPI P900 P315 P901 None None None	Lock (2) (3) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1	Link	
SCI5 SCI6 SCI7 SCI8 SCI9 Pin Function Pin Number		ole I2C mode, ensure port p between I2C and other mo			open drain.
Summary BSP Clocks Pins Interrupts B	event Links Stacks Components				

Figure 3-43. RA Project Configuration – Pin Configuration Setting (By Peripheral)



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

- 1. Select a pin in the **Pin Selection** pane, for example, **Ports** → **P0** → **P003**. The configuration for this pin will be shown in the **Pin Configuration** pane.
- 2. Enter properties for this pin, for example:

in Configuration						Generate Project Conter
elect Pin Configuration				📑 Export t	o CSV file 🖺 C	Configure Pin Driver Warnings
RA6M3-EK.pincfg	✓ Ma	nage configuratio	ons	G	enerate data: g	_bsp_pin_cfg
Pin Selection 🕀 🖻	↓ <mark>a</mark> Pin Configu	iration				📢 Cycle Pin Group
Type filter text	Name		Value		Link	
✓ ✓ Ports	Symbo	lic Name				
V V PO	Comm	ent		(2)		
P000	Mode		Disabled	(2)		
🛩 P001	✓ Input/0					
(4) P002	P00	3	None		\rightarrow	
(1) P003						
✓ P004						
P005						
P006						
P007 ✓ P008						
✓ P008 ✓ P009						
P010	<					3
✓ P014						
P015	Module n					
> 🗸 P1	Port Capa		AN07 PGAVSS0			
> 🗸 P2	×	ADC0.1	OAV550			

Figure 3-44. RA Project Configuration – Pin Configuration Setting (By Single Pin)

3. The **Visualization** view shows this pin change.



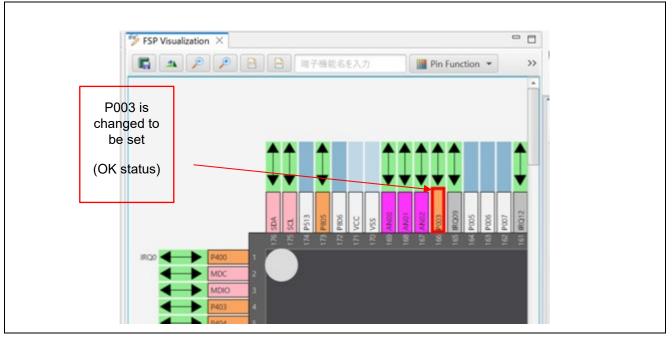


Figure 3-45. RA Project Configuration – Package View (Connection Status)

It is possible to migrate a pin configuration from one device to another device on this page. Use the **Import a pin configuration** button on the toolbar to perform this migration. This function allows the migration of the pin configuration to the new device while retaining user setup.

To import an existing pin configuration to the current project, click **Manage configurations...** \rightarrow **Import** and select the pin configuration file to import.



🌼 [RA_Tutorial] FSP Configuration 🔀				- 8	🌮 FSP Visualization
Pin Configuration			Generate P	Project Content	"
Select Pin Configuration		📑 Export to CS	SV file 🛛 Configure Pin Dri	ver Warnings	
RA6M3-EK.pincfg	✓ Manage configurations	🗹 Gene	rate data: g_bsp_pin_cfg		
Pin Selection $\textcircled{\blacksquare} \boxdot \textcircled{a}_{z}$	Pin Configuration		😲 Cyc	cle Pin Group	
Type filter text Y Ports > Y P0 > Y P1 > Y P2 > Y P3 > Y P4 > Y P5 Y P6 Y P7 Pin Function Pin Number Summary BSP Clocks Pins Interrupts E Image Pin Configurations E	×	Value Value Import Configuration Pin Configuration Imp		>	► Legend
Multiple Pin Configuration Manage Modify pin configuration list or import/		Import pin configuration			
RA6M3-EK.pincfg (Current) R7FA6M3AH3CFC.pincfg	Add Remove	From Configuration file : Pin configurations:			Browse
	Rename				Select All
	Duplicate Merge to Import Export	Merge symbolic name Merge port comment(Deselect All
	ОК			Finish	Cancel

Figure 3-46. Import An Existing Pin Configuration to The Current Project

The import function might prompt the user about conflicts and provide the following options for the user:

- Cancel the import operation
- Ignore the conflicts and import the conflicting settings anyway
- Continue the import operation without importing the conflicting settings.



3.5.5 Stacks Configuration Page

The Stack Configuration page allows you to:

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

Threads	🕄 New Thread 🖹 Remove 😑	HAL/Common Stacks	🕄 New Stack > 🚔 Extend Stack > 🔊 Remove
	L ⁱ coport I/O Port Driver on r_ioport (1) € New Object > € Remove (3)	 [∉] g_ioport I/O Port Driver on r_ioport (2) 	
		Stacks Components	
Summary	BSP Clocks Pins Interrupts Event Links	tata componento	
Propert	es 🛙 🖺 Problems 🤏 Smart Browser	and a second and a second a se	1
Propert			t < □

Figure 3-47. RA Project Configuration – Stacks Configuration GUI

The Stacks Configuration page consists of 3 panes:

- 1. Threads pane: Add/remove threads. Details are explained in Chapter 6.
- 2. **Stacks** pane: Add/remove FSP module instances, that is, IO port, SCI, UART, etc.
- 3. **Objects** pane: Add/remove kernel objects. Details are explained in Chapter 6.

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



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

- 1. Select a thread, such as **HAL/Common**. The modules and objects in this thread are shown.
- 2. In the Stacks pane, click on 1 New Stack to add a module to the thread, that is, New Stack \rightarrow Monitoring \rightarrow Clock Accuracy Circuit (r_cac).
- 3. Click the **Generate Project Content** Generate Project Content button to generate the source code content.
- 4. The **Properties** view shows the properties of the selected module. Users can change them according to their requirements.

Stacks Configuration Threads	(r_ioport)	9_cac0 Clock Accuracy Circuit (r_cac)	(3) Generate Project Content ick > ∴ Extend Stack > Remove (2)
✓ AL/Common (1)	g_ioport I/O Port (r_ioport)	g_cac0 Clock Accuracy	
Objects 🕢 New Object > 🗐 Remove		0	
Summary BSP Clocks Pins Interrupts Event Links Sta	cks Components		
Properties × g_cac0 Clock Accuracy Circuit (r_cac)	(4)		2 🖬 🖓 🗔 🔗 🕴 🗖
Settings Property		Value	^
API Info V Common		D (1) (DCD)	
Parameter Checking Module g_cac0 Clock Accuracy Circuit (r_ 	cac)	Default (BSP)	
Name		g_cac0	
Reference clock divider		32	
Reference clock source		Main Oscillator	
<			>

Figure 3-48. RA Project Configuration – Add New Module to Thread

Note: For another example, refer to chapter 6.1 which describes the procedure for adding the General Purpose Timer (GPT) module to the Blinky Thread.



An added module may require dependent modules or configuration settings. Necessary dependent modules will be added automatically. Optional dependent modules are suggested to be added manually by the user. In this case, users should click on the suggested modules to add and configure their properties (for example, Add New Stack \rightarrow Connectivity \rightarrow USB PCDC (r_usb_pcdc).

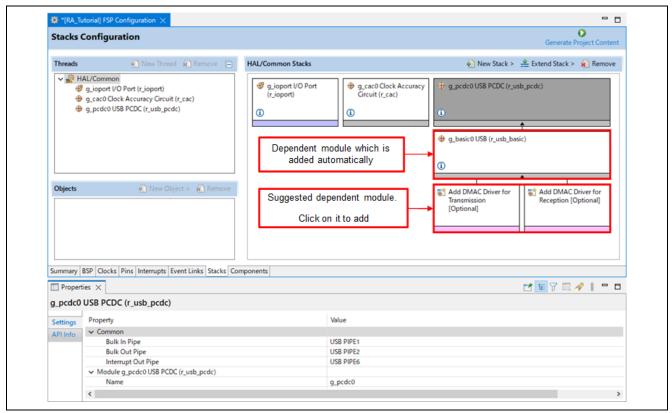


Figure 3-49. RA Project Configuration – Dependent Module



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

Stacks Configuration		Generate Project Content
Threads 🐑 New Thread 🔬 Remove 😑	HAL/Common Stacks	🀑 New Stack > 🔮 Extend Stack > 截 Remove
 ✓ ALL/Common ⊕ g_joport I/O Port (r_joport) ⊕ g_cac0 Olock Accuracy Circuit (r_cac) ⊕ g_pede0 US8 PCDC (r_usb_pede) ⊕ g_uart0 UART (r_sci_uart) 	g_pcdc0 US8 PCDC (r_usb_pcdc)	Image: g_wart0 UART (r sci wart) Team Image: g_wart0 UART (r sci wart) Resource Configurations Image: g_wart0 UART (r sci wart)
	g_basic0 US8 (r_usb_basic)	Add Dr C Dr Copy Transe Copy Copy Copy Convex option X Delete
Objects	Add DMAC Driver for Transmission [Optional] [Optional]	r Non-secure Callable import 2 Import 2 Export 3 Module Resources
Summary BSP Clocks Pins Interrupts Event Links Stacks	< Components	Run As Debug As Compare With Replace With
Stacks Configuration Threads		Generate Project Content
Stacks Configuration Threads	g_pcdc0 USB PCDC (r_usb_pcdc)	Generate Project Content
Stacks Configuration Threads New Thread Remove E	g_pcdc0 US8 PCDC (r_usb_pcdc)	Generate Project Content
Stacks Configuration Threads Item Thread Remove E Image: Stacks Configuration Image: Stacks Configuration E E Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration E Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration Image: Stacks Configuration	g_pcdc0 USB PCDC (r_usb_pcdc)	Generate Project Content
Stacks Configuration Threads	<pre> g_pcdc0 US8 PCDC (r_usb_pcdc) g_basic0 US8 (r_usb_basic) g_basic0 US8 (r_usb_basic) </pre>	Generate Project Content
Stacks Configuration Threads	g_pcdc0 USB PCDC (r_usb_pcdc) g_basic0 USB (r_usb_basic) g_basic0 USB (r_usb_basic) for Transmission Transmission	Generate Project Content New Stack >
Stacks Configuration Threads Investigation Image: Stacks Configuration Image: Stacks	g_pcdc0 USB PCDC (r_usb_pcdc) g_basic0 USB (r_usb_basic) f g_basic0 USB (r_usb_basic) f Add DMAC Driver for Transmission [Optional] C	Generate Project Content Cont
	g_pcdc0 USB PCDC (r_usb_pcdc) g_basic0 USB (r_usb_basic) f g_basic0 USB (r_usb_basic) f Add DMAC Driver for Transmission [Optional] C	Cenerate Project Content Cenerate Project Cenerate Cenerate P

Figure 3-50. Copy and Paste Operation



There will be a name conflict between the old module instance and the new one. Renaming one of the module instances will solve the problem.

Note that only the copy-and-paste of g_ioport will have no name conflict.

Stacks Configu	ıration				Generate Project Content
Threads	🕢 New Thread 👔 Remove 📄	HAL/Common Stacks		New Stack >	🔮 Extend Stack > 😰 Remove
⊕ g_cac0 C ⊕ g_pcdc0 ∯ g_uart0 L	on I/O Port (r_ioport) Iock Accuracy Circuit (r_cac) USB PCDC (r_usb_pcdc) JART (r_sci_uart) JART (r_sci_uart)	g_uarto UART (r_sc	_uart)	ල්∲ g_uart0 UART (r_sci_uart	•
gy g_date (Ann (Lastania)	Add DTC Driver for Transmission [Recommended bu optional]	Reception [Not	Add DTC Driver for Transmission [Recommended but optional]	Add DTC Driver for Reception [Not recommended]
Objects	new Object > 🔬 Remove		Ν	lame conflict after "copy and paste" operation]
		<			>
	ks Pins Interrupts Event Links 🔇 Stacks	Components			
Summary BSP Clock					
Summary BSP Clock					📑 🐨 🗔 🔗 🕴 🗖
	_sci_uart)				📑 🔚 🏹 🗔 🛷 🕴 🧮 🗖
Properties × g_uart0 UART (r Settings Property APLInfo × Mod	/ ule g_uart0 UART (r_sci_uart)		Value		- <u>-</u>
Properties × g_uart0 UART (r Settings Property APLInfo × Mod	/		Runto Re	ename one of the module stances to solve the conflict	
Properties × g_uart0 UART (r Settings Property APLInfo × Mod	/ ule g_uart0 UART (r_sci_uart) eneral Name		g_uart0	ename one of the module	

Figure 3-51. Module Instance Name Conflict



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

g_gcdc0 US8 PCDC (r_usb_pcdc) g_uart0 UART (r_sci_uart) g_uart1 UART (r_sci_uart) g_gart1 g_uart1 UART (r_sci_uart) g_gart1 g_uart1 UART (r_sci_uart) g_gart1 g_uart1
Add DMAC Driver for Transission [Optional] Optional] Paste Chrick Paste Chrick Paste Chrick Paste Chrick Paste Chrick Paste Chrick Chrick Chrick Paste Chrick Chric
Export Stack Export Export a stack to a configuration fragment on the local file system. To file: CWUsersWTEMPWuart.xml Browse. Browse.
Options



Concerte Configuration Intread @ Remove Image: Second Construction Image: Second Construction	Threads New Thread Remove MAL/Common Mal/Common Stacks New Stack >
Objects New Object >	Objects New Object > Remove Expont Expont Summary BSP Clocks Pins Interrupts Event Links Stacks Company Replace Stack Import Import Import X Stack Import Import Import X Stack Import Import Import X Stack Import Import stacks from a configuration fragment on the local file system. From file: CHUSers/NTEMP¥uart.xml Browse Stacks: X
	Import stacks from a configuration fragment on the local file system. From file: C+Users#TEMP¥uart.xml Stacks:

Figure 3-53. Import the RA Stack



3.5.6 Components Configuration Page

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

Modules common to all RA projects are preselected (for example, HAL Drivers \rightarrow all \rightarrow r_cac).

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

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

Components Configuration		Gener	orate Project Co	ntent
		Group by: Vendor V Filter: All V	Search	
Component	Version	Description	Variant	^
V 🖗 Board				- 12
Custom		Custom Board Support Files		
ra2a1_ek	100	RA2A1-EK Board Support Files		
ra2e1_ek		RA2E1-EK Board Support Files		
ra2e1_fpb		RA2E1-FPB Board Support Files		
ra2e2_ek	1.1	RA2E2-EK Board Support Files		
ra2e2_fpb		RA2E2-FPB Board Support Files		
ra2l1_ek		RA2L1-EK Board Support Files		
ra2l1_rssk	1 N	RA2L1-RSSK Board Support Files		
ra4e1_fpb		RA4E1-FPB Board Support Files		
ra4m1_ek	100	RA4M1-EK Board Support Files		
ra4m2_ek		RA4M2-EK Board Support Files		
ma4m3_ek		RA4M3-EK Board Support Files		
ra4w1_ek		EK-RA4W1 Board Support Files		
ra6e1_fpb		RA6E1-FPB Board Support Files		
ra6m1_ek		RA6M1-EK Board Support Files		
ra6m2_ek	1.1	RA6M2-EK Board Support Files		
☑ ra6m3_ek		RA6M3-EK Board Support Files		
📄 ra6m3g_ek		RA6M3G-EK Board Support Files		
📰 ra6m4_ek		RA6M4-EK Board Support Files		
ra6m5 ck	Sector Sector	RA6MS-CK Board Support Files		V

Figure 3-54. RA Project Configuration – Components Configuration



3.5.7 Interrupt Configuration Page

The **Interrupt** page allows the management of Events (interrupts) and ISRs (Interrupt Service Routines) for use with the RA interrupt framework.

The Interrupt page consists of 2 panes:

- 1. The User Events pane shows a list of events created manually by a user.
- 2. The **Allocations** pane shows a list of events that have been provided by instantiated RA modules in the section 3.5.5.

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

nterrupts Co	onfiguration			Generate Project Conte
Jser Events				🕙 New User Event > 🔊 Remov
Event			ISR	
		(1)		
		(')		
llocations				
	Event		ISR	
nterrupt	event			
nterrupt)	USBFS INT (USBFS interrupt)		usbfs_interrupt_handler	
nterrupt) 1		(2)	usbfs_interrupt_handler usbfs_resume_handler	
nterrupt D 1 2	USBFS INT (USBFS interrupt)	(2)		
) [2	USBFS INT (USBFS interrupt) USBFS RESUME (USBFS resume interrupt)	(2)	usbfs_resume_handler	
Interrupt 0 1 2 3 4	USBFS INT (USBFS interrupt) USBFS RESUME (USBFS resume interrupt) USBFS FIFO 0 (DMA transfer request 0)	(2)	usbfs_resume_handler usbfs_d0fifo_handler	

Figure 3-55. RA Project Configuration – Interrupt Page

A user event and its ISR can be created manually by clicking the New User Event button and then selecting an event to create.

🔅 [RA_Tutori	al] FSP Configuration $ imes$				🌮 FSP Visualiz	xation imes	📮 Package
Interrupts	s Configuration		Generate Project (Content		۶	Type pin function
User Events			New Use ADC	>			
		100	AGT	>			
Event		ISR	CAC	>			
			CAN	>			
			CGC	>		NG1	•
			COMP	>	KON TRAJN	NON I	
			CTSU	>	1/21		
			DMAC	>	DMACO	>	DMAC0 INT (DMAC transfer end 0)
Allocations			DOC	. >	DMAC1	· ·	
	Event	ISR	DRW	Ś	DMAC2	Ś	
Interrupt							~
0	USBFS INT (USBFS interrupt)	usbfs_inter		>	DMAC3	>	RENESA
1	USBFS RESUME (USBFS resume interrupt)	usbfs_resu		>	DMAC4	>	- (
2	USBFS FIFO 0 (DMA transfer request 0)	usbfs_d0fif	-	>	DMAC5	>	
3	USBFS FIFO 1 (DMA transfer request 1)	usbfs_d1fif	- FDIDC	>	DMAC6	>	
4	USBHS USB INT RESUME (USBHS interrupt)	New User Event	runt ha			×	
Summary BSF	² Clocks Pins Interrupts Event Links Stacks Com	Enter the name of the ISR fo	or the new user event:	ОК	Canc		RA6M3

Figure 3-56. Interrupt Page – Adding A New User Event



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

in [in] in the	ial] FSP Configuration ⊠		- 6
Interrup	ts Configuration	Generate Proje	ect Conten
User Event	s	New User Event > *	D Remove
Event		ISR	
DMAC0 IN	IT (DMAC transfer end 0)	MyISR	
Allocation	5		
Interrupt	Event	ISR	^
	USBFS INT (USBFS interrupt)	and for the second data allow	
0	USBES INT (USBES Interrupt)	usbfs_interrupt_handler	
0	USBFS RESUME (USBFS resume interrupt)	usbfs_resume_handler	
0 1 2			
1	USBFS RESUME (USBFS resume interrupt)	usbfs_resume_handler	1
1 2	USBFS RESUME (USBFS resume interrupt) USBFS FIFO 0 (DMA transfer request 0)	usbfs_resume_handler usbfs_d0fifo_handler	~

Figure 3-57. Interrupt Page – Generate Source Code

To remove a user event, select the event and click on the solution in the **User Events** pane (events added by instantiated RA modules in the **Allocations** pane cannot be removed).

nterrupt	s Configuration	Generate	Project Content
Jser Events		🗟 New User Ever	nt > 🔊 Remove
Event		ISR	De-regis
DMAC0 IN	(DMAC transfer end 0)	myISR	
llocations			
	Event	ISR	^
	Event USBFS INT (USBFS interrupt)	ISR usbfs_interrupt_handler	^
Interrupt			^
Interrupt	USBFS INT (USBFS interrupt)	usbfs_interrupt_handler	^
0	USBFS INT (USBFS interrupt) USBFS RESUME (USBFS resume interrupt)	usbfs_interrupt_handler usbfs_resume_handler	

Figure 3-58. Remove User Events



3.5.8 Event Links Configuration Page

The **Event Links** page lets users specify how non-FSP drivers within an RA project use the Event Link Controller (ELC). The UI allows the user to declare that such a driver might produce a set of ELC events or consume a set of ELC events via a set of peripheral functions.

Event Links	Configuration		Generate Project Content
User Events	New User Event >	User Events Consumed	🙆 New User Event 🖷 Remove
Produced	Remove Per	Peripheral Function	Event
Event			
Allocations			
Peripheral Fun	ction	Event	^
GPT (A)		No allocation	
GPT (B)		No allocation	
GPT (C)		No allocation	
GPT (D)		No allocation	
GPT (E)		No allocation	
GPT (F)		No allocation	×

Figure 3-59. Event Links Page



To declare a user-produced event:

- 1. Select the **New User Event** button on the **User Events Produced** table.
- 2. A cascading menu appears containing all the ELC events supported by the selected RA device.
- 3. The **User Events Produced** list will be updated with the event selected from the menu.

A Trate dell	RA Configuration 🛛				
	_	CGC	>	0	
Event Links	Configuration	COMP	>	Generate Project Cor	ntent
		CTSU	>		
User Events Produced	🕙 New User Event >	DMAC	> isumed	🕙 New User Event 📧 Rem	ove
Produced	NJ Remove	DOC	> ion	Event	
Event		DRW	>		- 1
		DTC	>		
		EDMAC	>		
		ELC	> ELC S	OFTWARE EVENT 0 (Software event 0)	
		EPTPC	> ELC S	OFTWARE EVENT 1 (Software event 1)	
		FCU	>		
Allocations		GLCDC	>		
Peripheral Fund	ction	GPT	>		~
GPT (A)		ICU	> ocation		
GPT (B)		IIC	> ocation		
GPT (C)		IOPORT	> ocation		-
GPT (D)		IWDT	> >cation		
GPT (E)		JPEG	> ocation		-
			and an		\sim
GPT (F)		K F Y	> pcation		
GPT (F) Summary BSP C	Clocks Pins Interrupts Ev	KEY LPM	> pents		
GPT (F) Summary BSP (C	Clocks Pins Interrupts Ev			Generate Project Co	
GPT (F) Summary BSP (C	RA Configuration ⊠	LPM			ontent
GPT (F) Summary BSP (C) * (RA_Tutorial) Event Links	RA Configuration 없 Configuration 원 New User Event >	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) *(RA_Tutorial) Event Links User Events Produced	RA Configuration	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration ⊠ Configuration New User Event >	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration 없 Configuration 원 New User Event >	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration ⊠ Configuration New User Event >	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration ⊠ Configuration New User Event >	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration ⊠ Configuration	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) * *[RA_Tutorial] I Event Links User Events Produced Event	RA Configuration ⊠ Configuration	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) we *[RA_Tutorial] I Event Links User Events Produced Event ELC SOFTWARI Allocations	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	> nents	Generate Project Co	ontent
GPT (F) Summary BSP (C) Summary BSP (C) Summary BSP (C) Summary BSP (C) Event Links User Events Produced Event ELC SOFTWARI Allocations Peripheral Functions	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	ents Consumed	Generate Project Co	ntent
GPT (F) Summary BSP (C) Summary BSP (C) Summary BSP (C) Event Links User Events Produced Event ELC SOFTWARI Allocations Peripheral Func GPT (A)	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	ents Consumed al Function	Generate Project Co	ntent
GPT (F) Summary BSP (C) Summary BSP (C) Summary BSP (C) Event Links User Events Produced Event ELC SOFTWARI Allocations Peripheral Func GPT (A) GPT (B)	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	> nents ents Consumed ral Function Event No allocation No allocation	Generate Project Co	ntent
GPT (F) Summary BSP (C) Summar	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	ents Consumed al Function	Generate Project Co	ntent
GPT (F) Summary BSP (C) Summary BSP (C) Event Links User Events Produced Event ELC SOFTWARE Allocations Peripheral Func GPT (A) GPT (B) GPT (C) GPT (D)	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	rents r	Generate Project Co	ntent
GPT (F) Summary BSP (C) Summary BSP (C) Summary BSP (C) Event Links User Events Produced Event ELC SOFTWARI Allocations Peripheral Func GPT (A) GPT (B)	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	> nents	Generate Project Co	nove
GPT (F) Summary BSP (C) Summar	RA Configuration 적 Configuration 인 New User Event > Remove	LPM	ents Consumed al Function Event No allocation No allocation No allocation	Generate Project Co	ontent

Figure 3-60. Declare User-Produced Event



To declare a user-consumed event:

- 1. Select the **New User Event** button on the **User Events Consumed** table
- 2. A **New User Event** dialog should open with a combo box containing the available peripheral functions and available ELC events. Select a **Peripheral Function** and **Event**, then click on **OK**.
- 3. The selected event will be allocated to the selected peripheral function within the RA configuration. It should be visible within the **User Events Consumed** list and the **Allocations** list on the **Event Links** page.

® *[RA_Tutorial]	RA Configuration 🛛			
Event Links	Configuration			oject Content
User Events	🗟 New User Event >	User Events Consumed	🗟 New User Even	t D Remove
Produced	🛍 Remove	Peripheral Function	E	
Event				
ELC SOFTWAR	E EVENT 0 (Software event 0)			
	e ² New User Event		×	
	Peripheral Function:	GPT (A)	~	
	Event:		~ _	
Allocations		ELC SOFTWARE EVENT 0 (Softwar	e event 0)	
Peripheral Fun		IOPORT EVENT 1 (Port 1 event)		^
GPT (A)		IOPORT EVENT 2 (Port 2 event) IOPORT EVENT 3 (Port 3 event)		
GPT (B)		IOPORT EVENT 4 (Port 4 event)		
GPT (C)		USBFS FIFO 0 (DMA transfer requ	est 0)	
GPT (D)		USBFS FIFO 1 (DMA transfer requ		
GPT (E)		USBHS FIFO 0 (DMA transfer requ		
011(0)		USBHS FIFO 1 (DMA transfer requ		
GPT (F) Summary BSP	Clocks Pins Interrupts Event L		est 1)	¥
GPT (F) Summary BSP *[RA_Tutorial] Event Links	Clocks Pins Interrupts Event L RA Configuration 22 Configuration	inks Stacks Components	Generate P	□ □ oroject Content
GPT (F) Summary BSP (Clocks Pins Interrupts Event L RA Configuration 23 Configuration	inks Stacks Components		□ □ roject Content
GPT (F) Summary BSP *[RA_Tutorial] Event Links Vser Events Produced	Clocks Pins Interrupts Event L RA Configuration 22 Configuration	inks Stacks Components User Events Consumed Peripheral Function Event	Generate P € New User Ever	ा व roject Content nt 🔊 Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Produced Event	Clocks Pins Interrupts Event L RA Configuration 23 Configuration	inks Stacks Components User Events Consumed Peripheral Function Event	Generate P	ा व roject Content nt 🔊 Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Produced Event	Clocks Pins Interrupts Event L RA Configuration 22 Configuration 인 New User Event > 요 Remove	inks Stacks Components User Events Consumed Peripheral Function Event	Generate P € New User Ever	् । roject Content nt 🔊 Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Produced Event	Clocks Pins Interrupts Event L RA Configuration 22 Configuration 인 New User Event > 요 Remove	inks Stacks Components User Events Consumed Peripheral Function Event	Generate P € New User Ever	ा । roject Content nt ¥ो Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Produced Event ELC SOFTWAR	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	inks Stacks Components User Events Consumed Peripheral Function Event	Generate P € New User Ever	ा । roject Content nt ¥ो Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Vser Events Produced Event ELC SOFTWAR Allocations	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	Inks Stacks Components User Events Consumed Peripheral Function Event Event Event	Generate P € New User Ever	् roject Content nt २० Remove
GPT (F) Summary BSP *[RA_Tutorial] Event Links Vser Events Produced Event ELC SOFTWAR Allocations Peripheral Fun	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	Inks Stacks Components User Events Consumed Peripheral Function Event Event Event	Generate P € New User Even OFTWARE EVENT 0 (Softv	ତ roject Content nt % Remove vare event 0)
GPT (F) Summary BSP *[RA_Tutorial] Event Links Voser Events Produced Event ELC SOFTWAR Allocations Peripheral Fun GPT (A)	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	Inks Stacks Components User Events Consumed Peripheral Function EVENt GPT (A) ELC SO Event ELC SOFTWARE EVEPT	Generate P € New User Even OFTWARE EVENT 0 (Softv	् roject Content nt २० Remove
GPT (F) Summary BSP * (RA_Tutorial) Event Links Produced Event ELC SOFTWAR Allocations Peripheral Fun GPT (A) GPT (B)	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	User Events Components User Events Consumed Peripheral Function EVENT EVENT EVENT EVENT EVENT EVENT EVENT EVENT EVENT ELC SOFTWARE EVEN No allocation	Generate P € New User Even OFTWARE EVENT 0 (Softv	ତ roject Content nt % Remove vare event 0)
GPT (F) Summary BSP * (RA_Tutorial) Event Links Voduced Event ELC SOFTWAR Allocations Peripheral Fun GPT (A) GPT (B) GPT (C)	Clocks Pins Interrupts Event L RA Configuration ²² Configuration New User Event >	Inks Stacks Components User Events Consumed Peripheral Function ELC So Event ELC SOFTWARE EVEN No allocation No allocation	Generate P € New User Even OFTWARE EVENT 0 (Softv	● □ roject Content nt 和 Remove

Figure 3-61. Declare A User-Consumed Event



3.6 Editor hover

 e^2 studio supports hovers in the textual editor. This function can be enabled/disabled via **Window** \rightarrow **Preferences** \rightarrow **C/C++** \rightarrow **Editor** \rightarrow **Hovers**.

type filter text		Hovers		← ▼ ⇒ ▼
 Code Style Core Build Toolchains Debug 	^	Expand vertical ruler icons up Text <u>H</u> over key modifier prefere	oon hovering (does not affect op nces:	en editors)
✓ Editor		Text Hover Name	Pressed Key Modifier While Ho	vering
> Content Assist		✓ Combined Hover		
Encrypted Files		Debugger		
Folding		Renesas IO Register Help		
> Hovers		RenesasCDocHover		
Mark Occurrences		Problem Description		
Save Actions				
Scalability		Macro Expansion		
Syntax Coloring		Source	Shift	
Templates		Annotation Description		
Typing				
File Types				
Indexer		Pressed key <u>m</u> odifier while hove	ering:	
Language Mappings		Description:		
New C/C++ Project Wizard		Tries the hovers in the sequence	e listed below and uses the one v	vhich fits best
Property Pages Settings		for the selected element and the	ne current context.	
> Renesas				
Task Tags			Restore Defaults	Apply
Template Default Values	~			

Figure 3-62. Hover Settings

To enable hover, check **Combined Hover**. To disable it, uncheck it. This function is enabled by default.

The hover function allows the user to view detailed information about any identifiers in the source code: hover the mouse over an identifier and check the pop-up.

@ [RA_Tutorial] RA	Configuration	i li hal entry.c ⊠	- 0	■ Package 🛛	- 0
82	5	/" loggie this pin. "/			
83		<pre>uint32 t pin = leds.p leds[i];</pre>	^	A package view is not available for the active editor.	1
84		R_PFS->PORT[pin >> 8].PIN[pin & 0xFF].PmnPFS_b.PODR	^=		
85		}			
86		,			
87		R PMISC->PWPR b.PFSWE = 0; ///< Writing to PmnPFS re	ais		
88		R PMISC->PWPR b.BOWI = 1U; ///< Writing to PFSWE bit			
89		K_FMISC->FWFK_D.BOWI = 10, /// Writing to From Dit	uı		
90		/* Delav */			
90					
		R_BSP_SoftwareDelay_delay, bsp_delay_units);	_		
92	}	<pre>void R_BSP_SoftwareDelay (uint32_t delay, bsp_delay_unit</pre>	s_t un	hits)	
93	}	{			
94		uint32_t iclk_hz;			
96	● * This	<pre>uint32_t cycles_requested;</pre>			
101	⊖void R_E	uint32 t ns per cycle;			
102	{	uint32 t loops required = 0;			
103	⊖ if (<pre>uint32_t total_us = (delay * units);</pre>		<pre>/** Convert the requested time</pre>	
104	{	uint64 t ns 64bits;			
105					
106		<pre>iclk_hz = SystemCoreClock;</pre>		/** Get the system clock frequ	
107				, det the system ersen frequ	
108		/* Running on the Sub-clock (32768 Hz) there are 305	17 ns/	cycle This means one cycle takes 31	
109	}	/ Rumining on the Sub-CIOCK (S2708 HZ) there are Sub-	1/ 115/	Dross 'E2' for focu	
110	}		_	Press 'E2' for focu	
111			\checkmark		
	<		>		
				1	1

Figure 3-63. Information from Hover Function



4. Building

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

4.1 Build Configurations

When a project is created, the default build option is generated. It can usually be used to build the project.

However, if changing build options is necessary (for example, toolchain version or optimization options), please follow the following steps before building the project.

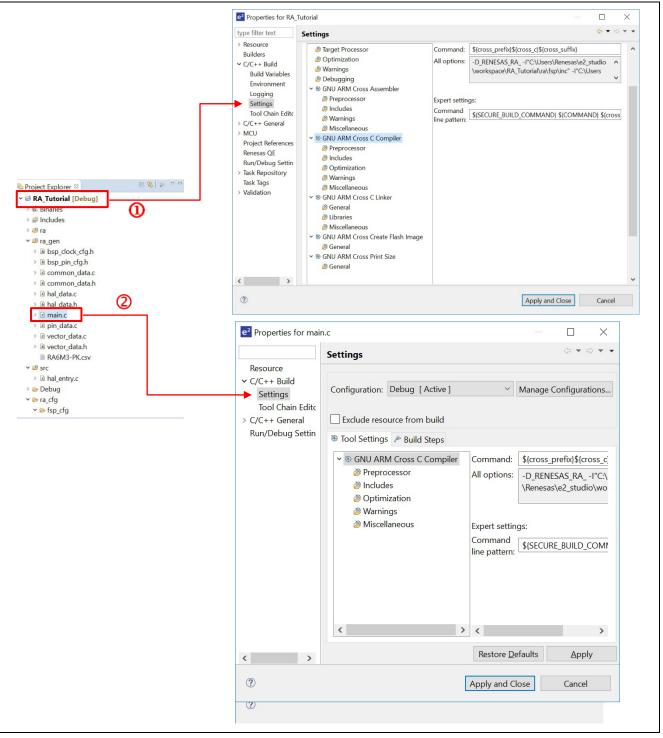


Figure 4-1. Build – Properties for RA Project And main.c Source File



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

- 1. \bigcirc Set the focus at the project name or \bigcirc set the focus at the source file name.
- 2. Right-click to select Properties or use shortcut keys [Alt] + [Enter] to open the properties dialog.
- 3. Click on C/C++ Build \rightarrow Settings option to view or edit the configuration settings.

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

4.2 Building a Sample Project

Follow the steps below to build the project.

- 1. In **Project Explorer**, click on the RA project to bring it into focus.
- 2. Click on **Project** \rightarrow **Build Project** or the 4 icon to build this project.
- 3. Confirm that there are no errors after the build is finished.

File Edit Navigate Search	Proj	ect Renesas Views Run Window	Help		
🐔 🔯 🔳 🕸 Debug		Open Project		🖻 🗕 🖬 👘 😽 💊	- 🖬 📮 🔅
		Close Project			
Project Explorer 🛛		Open FSP Configuration		figuration ⊠	
✓ S RA_Tutorial [Debug]	010	Build All Ctrl+A	Alt+B		
> Includes	_	Build Configuration Build Project	>	tion	
> 📴 ra		Build Project	trl+B		
> 😕 ra_gen		Build Working Set	>	read 🔊 Remove 🛛 📄	HAL/Commor
> 🐸 src		Clean			
> 🗁 Debug		Build Automatically		Port Driver on r_ioport	∉ g_ioport I
> 🗁 ra_cfg		C/C++ Index	>	iver on r_rtc	Driver on
> 🗁 script	e ²	Update All Dependencies A	Alt+D		(i)
configuration.xml		Change Device			
R7FA6M3AH3CFC.pinc	8	C/C++ Project Settings Ctrl+A	Alt+P		
📄 ra_cfg.txt		Properties			
🖹 RA Tutorial Debug Fla	u au				

Figure 4-2. Build – Building A Sample Project



4.3 Saving the Build Settings Report

Project build settings in e² studio IDE can be saved to a file using the **Project Reporter** feature.

- 1. Right-click in the **Project Explorer** view to pop up the context menu.
- 2. Select Save build settings report to save the build settings report.

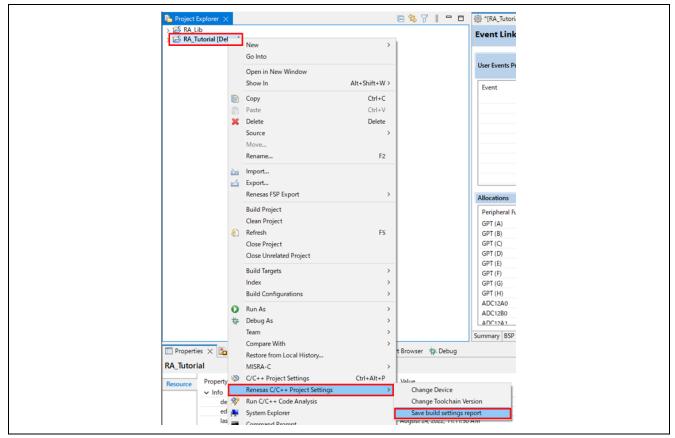


Figure 4-3. Build – Saving the Build Settings Report



5. Debugging

This chapter describes using debug configuration and key debugging features for e² studio. The following illustration refers to the RA project built in Chapter 4.2 Building a Sample Project and based on the following hardware configuration: J-link ARM emulator and RA6M3 EK board.

Right-click on any perspective icon and select **Show Text** to show the name of each icon. Depending on the version of e^2 studio, text may be displayed by default.

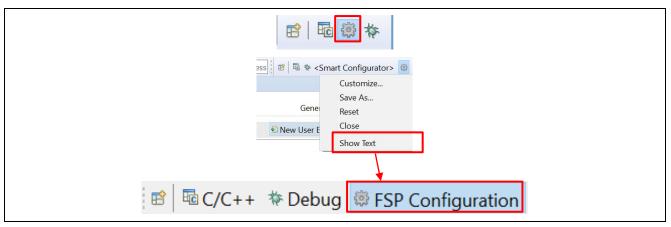


Figure 5-1. Debug – Switch to Debug Perspective

Open the RA project in e² studio and click **Debug** to switch to the **Debug** perspective.

As discussed earlier, a **Perspective** in Eclipse defines the layout of panes and views in the **Workbench** window. Each perspective consists of a combination of views, menus, and toolbars that enable the user to perform a specific task. For instance,

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

If a user attempts to connect the debugger when not in the **Debug** perspective, e^2 studio will prompt the user to switch to the **Debug** perspective.

One or more perspectives can exist in a single Workbench setup. Users can customize them or add new perspectives.

5.1 Changing an Existing Debug Configuration

A default debug configuration is automatically created the first time a specific RA project is built. An existing debug configuration can be changed as follows.

- 1. Click on the project name in the **Project Explorer** view to set focus.
- 2. Click on Run → Debug Configurations... or the ^{***} icon (downward arrow) → Debug Configurations... to open the Debug Configurations window.



File Edit Navigate Search Project Renesa			
Image: Second system Image: Second system Image: Second system Im		Renesas Debug Tools Resume Suspend Terminate Disconnect Step Into Step Into Selection Step Over	>
> 28 ra_gen > 28 src > 25 Debug > 25 ra_cfg > 25 script	n (← (←	Step Return Run to Line Use Step Filters	Shift+F5
configuration.xml	Q	Run	Ctrl+F11 F11
 R7FA6M3AH3CFC.pincfg ra_cfg.txt RA_Tutorial Debug_Flat.launch Oveloper Assistance 	**	Debug Run History Run As Run Configurations	> >
	蓉	Debug History Debug As	>
		Debug Configurations	

Figure 5-2. Debug – Opening the Debug Configurations Window



- **RA** Family
- 3. In the **Debug Configurations** windows, expand the **Renesas GDB Hardware Debugging** debug configuration and click on the existing debug configuration (for example, **RA_Tutorial Debug_Flat**).
- 4. Go to the **Main** tab and browse to add the load module (that is, **RA_Tutorial.elf**) located in the project build folder.

Image:

Figure 5-3. Debug – Selecting the Load Module

- 5. Switch to the **Debugger** tab and set J-Link ARM and R7FA6M3AH as the target device.
 - Debug Hardware: J-link ARM
 - Target Device: **R7FA6M3AH**
- 6. Click on the **Apply** button to confirm the settings.
- 7. Click the **Debug** button to execute the debug launch configuration to connect to the J-Link and the RA board.

Debug hardware: J-Link ARM E2 (ARM) E2 Lite (ARM) Debug Tool Settings GDB Settings Cd J-Link ARM Debug Tool Settings GDB Connection Settings Image: Autostart local GDB server O Connect to remote GDB server GDB port number: GDB port number: G1234 Connection timeout (s):	📄 Main <mark> Startup</mark> 🕞 Startup	Common 🤤 Source	
GDB Settings Cc E2 Lite (ARM) Debug Tool Settings GDB Connection Settings Image: Colling		Target Device: R7FA6M3A	н
O Connect to remote GDB server GDB port number: 61234	GDB Settings Cc 2-Lite (ARM)	Debug Tool Settings	
GDB port number: 61234	Autostart local GDB server	Host name or IP address:	localhost
Connection timeout (s): 30 ~	Connect to remote GDB server	GDB port number:	61234
		Connection timeout (s):	30 ~

Figure 5-4. Debug – Changing the Connection Settings

8. For a successful connection, the **Debug** view shows the target debugging information in a tree hierarchy. The program entry point is set at Reset_Handler() in startup.c.

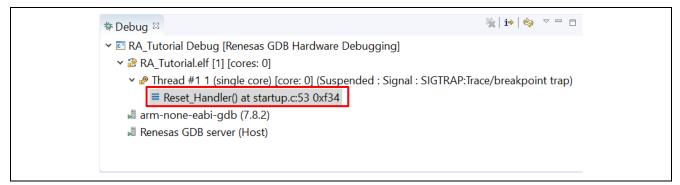


Figure 5-5. Debug – User Target Connection In The Debug View

5.2 Creating New Debug Configurations

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

- 1. Open the **Debug Configuration** window (refer to Figure 5-2. Debug Opening the Debug Configurations Window).
- 2. In the **Debug Configurations** window, select a debug configuration (for example, **RA_Tutorial Debug**) and click on the icon (which duplicates the currently selected launch configuration). A new debug launch configuration (for example, **RA_Tutorial Debug (1)**) is created.
- 3. The new debug configuration can be configured as described in chapter 0.

Debug Configurations		-	X
Create, manage, and run configurat	ions		Ť
	Name: RA_Tutorial Debug_Flat (1)		
type filter text	📄 Main 🕸 Debugger 🕟 Startup 🍫	Source 🔲 Common	
C/C++ Application	Project:		
C/C++ Remote Application	RA_Tutorial		<u>B</u> rowse
GDB Hardware Debugging	C/C++ Application:		
C GDB OpenOCD Debugging C GDB Simulator Debugging (RH8	Debug/RA_Tutorial.elf		
Java Applet		<u>V</u> ariables Searc <u>h</u> Project	B <u>r</u> owse
Java Application Launch Group	Build (if required) before launching		
Remote Java Application	Build Configuration: Use Active		~
✓ C [™] Renesas GDB Hardware Debugg	O Enable auto build	O Disable auto build	
RA_Tutorial Debug_Flat RA_Tutorial Debug_Flat (1)	Use workspace settings	Configure Workspace Settings	
c* Renesas Simulator Debugging (F			
< >		Revert	Apply
Filter matched 14 of 16 items		<u>ver</u> eit	марих
?		Debug	Close

Figure 5-6. Debug – Duplicating A Selected Debug Launch Configuration



5.3 Basic Debugging Features

This section explains the typical Debug views supported in e² studio.

- Standard GDB Debug (supported by Eclipse IDE framework): Breakpoints, Expressions, Registers, Memory, Disassembly, and Variables (MMU view is not supported in RA).
- Renesas Extension to Standard GDB Debug: IO Registers, Eventpoints, Trace and Fault Status.

The following are some useful toolbars in the **Debug** view:

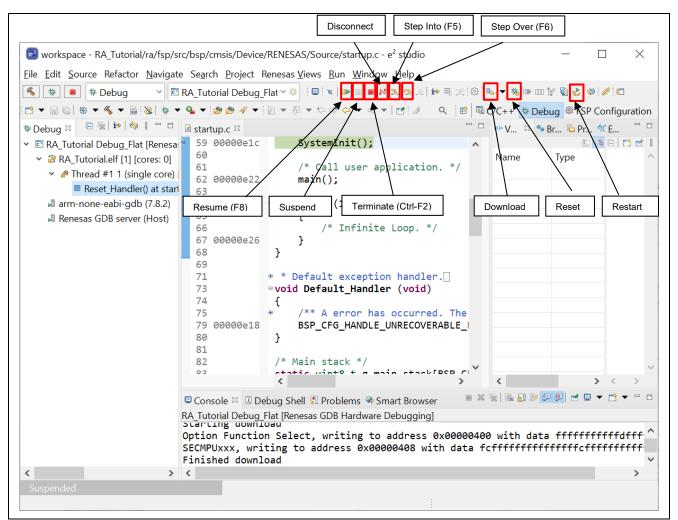


Figure 5-7. Debug – Useful Toolbars In Debug Views

Run the program by clicking on the Ib button or pressing **F8**.

The program execution can be suspended by a breakpoint or by clicking the III button. When the program execution is suspended, you can perform the following operations:

- Not the security of the stepping into the next method call at the currently executing line of code.
- button or F6 can be used for stepping over the next method call (executing but without entering it) at the currently executing line of code.
- Image: button can be clicked again to resume program execution.

To stop the debugging process, the local button can be clicked to end the selected debug session and/or process, or the selected process.



The other operations are as follows:

- The Solution can be clicked to reset and run the program. It may stop at main() if the breakpoint is configured in the Debug configuration.
- The 🏂 button can be clicked to reset the program to its entry point at the PowerOn Reset.
- The 🤽 button is used for re-downloading the binary file to the target system.

The Launch Bar, which provides the Build and Debug buttons, is located in the toolbar area of e^2 studio's main window. It is hidden by default. However, it can be displayed by clicking on **Window** \rightarrow [Settings] \rightarrow [Run/Debug] \rightarrow [Launching] \rightarrow [Launch Bar].

Note: When the debug toolbar is displayed, the 🌌 button is shown. To show the debug toolbar, select **Show Debug Toolbar** from the icon indicated below.

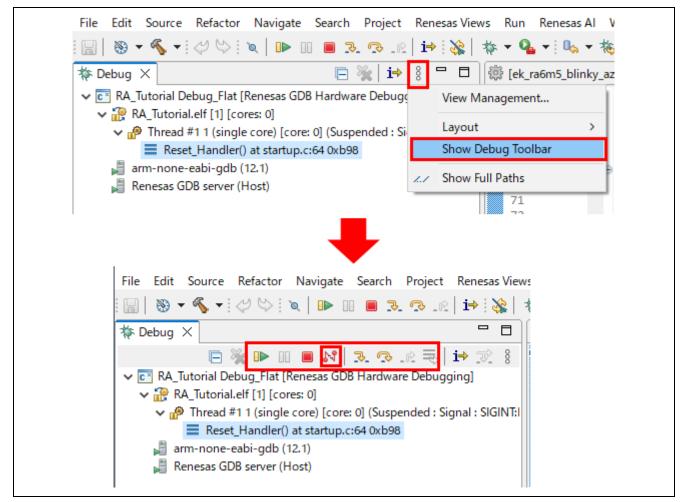


Figure 5-8. Debug – Toolbar in the [Debug] View

5.3.1 Debug View

The Debug View shows executed functions per thread.



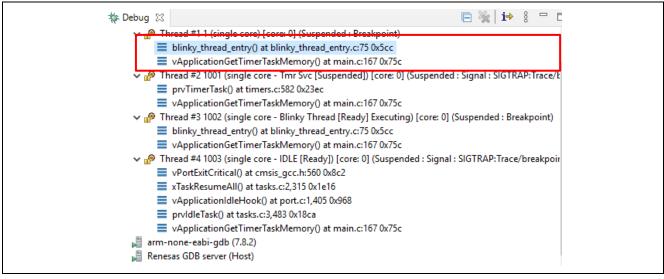


Figure 5-9. Debug – [Debug] View



Setting the Display of Executed Functions in the Debug View feature is set in the Debug Configurations dialog box.

- 1. Select the **Run > Debug Configurations** menu and open the **Debug Configurations** dialog box.
- 2. Select the **Debugger** tab and the **Debug Tool Settings** tab.
- 3. If you will be using RTOS, set **RTOS Integration in Debug View** to **Yes**. If you select No, this feature will not be available. This selection is set to Yes by default.

			200
C 🖻 ᅍ 🗎 🗙 🕒 🏹 👻	Name: blinky_with_thread Debug_Flat		
type filter text	📄 Main 🕸 Debugger 🍉 Startup 🔲 Common 🍹	∕ Source	
C C/C++ Application C C/C++ Remote Application E EASE Script	Debug hardware: J-Link ARM V Target Device	R7FA6M3AH	
C GDB Hardware Debugging	GDB Settings Connection Settings Debug Tool Se	ettings	
GDB OpenOCD Debugging	Use CFI-Flash	No	¥ ^
💽 GDB Simulator Debugging (RH850)	CFI Start	0x0	
🜌 Java Applet	CFI End	0x0	
Java Application	✓ Semihosting		
🚭 Launch Group 🖳 Remote Java Application	Semihosting breakpoint address		
 Remote Java Application C³ Renesas GDB Hardware Debugging 	→ RTOS		
Referes of the reference of the refer	RTOS Integration in Debug View	Yes	×
Renesas Simulator Debugging (RX, RL78)	RTOS Debugging - Large Number of Threads.	No	×
e nenesas sinnalator b es agging (ro, nero)	System Allow caching of flash contents	Yes	×
	Allow caching of flash contents V Time Measurement	Tes	¥
	Run Break Time Measurement	Yes	v
	Count Every Core Cycle	Yes	
	Operating Frequency [MHz]		
			×
Filter matched 13 of 15 items		Ren	vert Apply

Figure 5-10. Debug – [Debug Tool Settings]

5.3.2 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 is suspended before that line of code executes. e² studio allows software and hardware breakpoints to be set explicitly in the IDE. Any breakpoints added by double-clicking on the marker bar are, by default, hardware breakpoints. If the hardware resources are not there, then the breakpoint setting will fail. If a hardware breakpoint setting fails, an error message will prompt the user to switch to a software breakpoint.

To select a Hardware or Software breakpoint:

 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. As an example, in startup.c at line 62, double-click on the marker bar located in the left margin of the C/C++ Editor pane to set a breakpoint. A dot 20 (Hardware breakpoint) or 20 (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 on the marker bar to choose **Toggle Hardware Breakpoint** or **Toggle Software Breakpoint** to set a hardware breakpoint and or a software breakpoint set.



3. Click on Windows → Show View → Breakpoints or icon ^{So} (or use shortcut key ALT+Shift+Q, B) to open the Breakpoints view to view the corresponding breakpoints set. Breakpoints can be enabled and disabled in the Breakpoints view.

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

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



Figure 5-11. Debug – [Breakpoints] View



Note: If the following error occurs when setting a software breakpoint in the internal code flash area, "Cannot Insert breakpoint xx in function name at source file name: number of lines: Remote failure reply: FFFFFFF add software breakpoint"

Please refer to the following document to enable flash breakpoints." E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RA Devices)"–"3.3.2. Notes on a Debugging Operation that Involves Reprogramming of Flash Memory"-"(7) Conditions for using software breaks in flash memory" and enable flash breakpoints. E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RA

E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RA Devices) (renesas.com)

5.3.3 Expressions View

The Expressions view monitors the value of global variables, static variables, or local variables during debugging.

Follow the steps below to watch a variable:

- 1. Click on Windows \rightarrow Show View \rightarrow Expressions or icon $\frac{6}{2}$ to open the Expressions view.
- 2. Drag and drop a variable (for example, g_fsp_version in bsp_common.c) to the Expressions view.
- 3. Alternatively, right-click on the variable to select the **Add Watch Expression...** menu item to add it to the **Expressions** view.

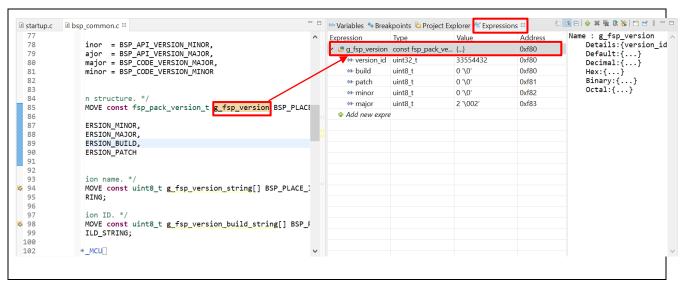


Figure 5-12. Debug - [Expressions] View

An item selected in the **Expressions** view can be set as an Eventpoint.

1. Right-click on the target item (e.g., version_id) and select **Set Event Break** from **Renesas Eventpoints**. When the dialog box is displayed, set the conditions and click the **OK** button.



RA Family

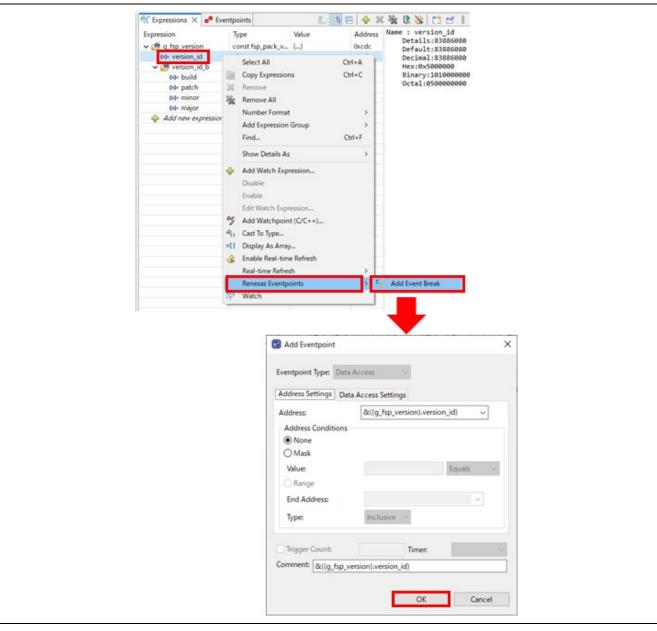


Figure 5-13. Debug – Setting an Item as an Eventpoint

2. An event break is set in the **Eventpoints** view.

ype	Address	Data	Count	Timer	Handle	Comment
🗌 🎁 Trace Start						
🗌 🞬 Trace Stop						
🔄 🧉 Trace Record						
C Event Break						
OR OR	&((g_fsp_version).versi	Read/Write Byte			4	&((g_fsp_version).version_id)

Figure 5-14. Debug – [Eventpoints] View



5.3.4 Registers View

The **Registers** view lists information about the general registers in RA. When the program stops, changed values are highlighted.

- 1. Click on Windows \rightarrow Show View \rightarrow Registers or icon ¹⁰⁰ to open the Registers view.
- 2. Click on the name of a register to view the values in different radix format.

When the program stops, changed values are highlighted (for example, in yellow) in the **Registers** view.

me	Value	Description	
General Registers		General Purpose and FPU Register Group	
1010 rO	0x0		
1010 r1	0x11f		
1000 r2	0x13a2		
1000 r3	0x60		
1010 r4	0x1ffe08fc		
1889 r5	0x1ffe0954		
1010 r6	0x1ffe0954		
1818 r7	0x0		
⁸⁸⁸ r8	0x13b4		
iiii r9	0x1		
^{ዘዝ} r10	0x40040d00		
1010 r11	ΩxΩ		
			>
me : r0 Hex:0x0 Decimal:0 Octal:0 Binary:0 Float:0 Default:0			

Figure 5-15. Debug – [Registers] View



5.3.5 Memory View

The **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 (for example, Hex integer, signed integer, unsigned integer, or ASCII image).

To view the memory of a variable (for example, g_fsp_version_build_string):

- 1. Click on Windows \rightarrow Show View \rightarrow Memory to open the Memory view.
- 2. Click on the icon to open the Monitor Memory dialog box. Enter the address of the variable &g fsp version build string.

e ² Monitor Memory Enter address or express &g_fsp_version_build_st		is pr	•	memory ren	version_buil	- 0	
? ОК	Cancel						
Console @ Tasks 🕆 Proble O Execut.							≇ Renesa [□] □ N 🕫 🖩 😫 🕷 🔻 🔻
	&g fsp. ersion build st Address	ring : 0x11E0 < 0 - 3	Hex Integer>	≅	enderings C - F		≄ Kenesa —
Monitors	80 fsp. ersion build st Address 00000000000011F0	ring : 0x11E0 < 0 - 3 53462323	: <u>Hex Integer> :</u> 4 - 7 45565F50	A → New Re 8 - B 4F495352	cnderings C - F 55425F4E		* Kenesa –
Monitors & &g_fsp_version_build_string	8 80 fsp. ersion build st Address 00000000000011F0 0000000000001200	ring : 0x11F0 < 0 - 3 53462323 5F444C49	Hex Integer> 4 - 7 45565F50 49525453	23	enderings C - F 55425F4E 00000000		* Kenesa
Monitors & &g_fsp_version_build_string Memory Monitor for	80 fsp. ersion build st Address 00000000000011F0	ring : 0x11E0 < 0 - 3 53462323 5F444C49 53462323	: <u>Hex Integer> :</u> 4 - 7 45565F50	A → New Re 8 - B 4F495352	cnderings C - F 55425F4E		* Kenesa
Monitors &g_fsp_version_build_string Memory Monitor for g_fsp_version_build_string" is	8 80 fsp. ersion build st Address 00000000000011F0 0000000000001200 0000000000	ring:0x11F0 < 0 - 3 53462323 5F444C49 53462323 00000000	Hex Integer> 4 - 7 45565F50 49525453 45565F50	 New Re 8 - B 4F495352 2323474E 4F495352 	C - F 55425F4E 0000000 0023234E		* Kenesa
Monitors	8 80 fsp. ersion build st Address 00000000000011F0 0000000000001210 0000000000	ring:0x11F0 < 0 - 3 53462323 5F444C49 53462323 00000000 BF00B5F8	Hex Integer> 4 - 7 45565F50 49525453 45565F50 BF00B5F8	 New Re 8 - B 4F495352 2323474E 4F495352 BC08BCF8 	C - F 55425F4E 0000000 0023234E 4770469E 4770469E		* Kenesa

Figure 5-16. Debug – Memory View



To add a new rendering format (for example, ASCII) for the variable g fsp version build string:

Click on the **Wew Renderings...** tab to select **ASCII** to add the rendering. This creates a new tab named **&g_fsp_version_build_string <ASCII>** next to the tab **&g_fsp_version_build_string <Hex Integer>**.

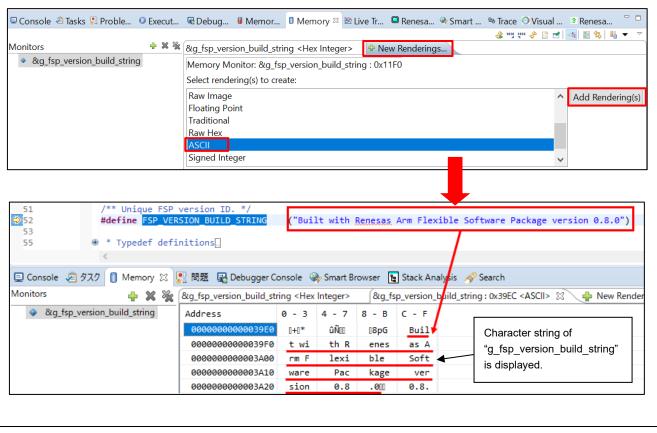


Figure 5-17. Debug – New Rendering in Memory View

An item selected in the Memory view can be set as an Eventpoint.

1. Right-click on the target address (e.g., 0x00000C90) and select **Set Event Break** from **Renesas Eventpoints**. When the dialog box is displayed, set the conditions and click the **OK** button.



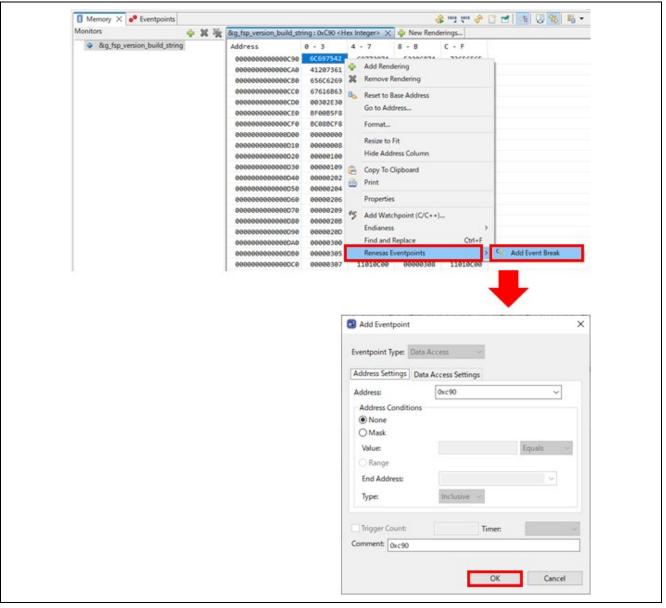


Figure 5-18. Debug – Setting an Item as an Eventpoint

2. An event break is set in the **Eventpoints** view.

Туре	Address	Data	Count	Timer	Handle	Comment
🗌 🎁 Trace Start						
🗌 🗃 Trace Stop						
🔄 🧉 Trace Record						
V C Event Break						
OR OR	0xc90	Read/Write Byte			16	0xc90
1. Timor Start						
Timer Stop						

Figure 5-19. Debug – [Eventpoints] View



5.3.6 Memory Usage View

Memory Usage will be used to get information from a (*.map) file or library list file (,) from a project. This will list the total memory size, ROM and RAM ratio usage, and detailed information on sections, objects, symbols, modules, vectors, and cross references used in the project.

From version 7.3, e² studio supports the graphical view to show usage in the ROM and RAM memory areas.

To show the Memory Usage view, click Window \rightarrow Show View \rightarrow Other... \rightarrow C/C++. In the Show View dialog, select Memory Usage and click on Open.

e ² Show View		×
type filter text ◆ Fault Status ☐ IO Registers ≧ Live Trace Console ① Memory 		
 Memory Browser Memory Usage MMU Modules OS Resources Performance Analysis Peripherals Profile 		
 [®] Real-time Chart [™] Registers ³ Renesas Coverage 	Cancel	~

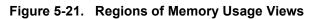
Figure 5-20. Show Memory Usage View



The **Memory Usage** view has three regions: (1) Group Size region, (2) Memory Region Usage region (Device Memory Usage region is not supported yet), and (3) Detail table region.

Note: When the selected project does not contain the linker script file or no region defined in the linker script file, the Memory Region Usage region will display a warning message: "Linker script file is invalid."

Size:						Memory Reg	ion Usage Device N	1emory Usage	
Program	ר:			5056 by	te(s)	Memory Re	gion Usage:		
Constan	t:			0 by	te(s)	Linker script	file is invalid.		
Initialize	d Data:			8 by	te(s)				
Uninitial	ized Data:	(1)	1536 by	te(s)				(2)
Data:				24 by	te(s)				(4)
Stack:				1024 by	te(s)				
Others:				16 by	te(s)				
ection Object	Symbol								
Section	Group	Start addres	End address	Size (byte)	Align	Attribute	Load addre		
.vfp11_veneer	Constant	0x000013C0		0					
.v4_bx	Constant	0x000013C0		0					
	Program	0x00000000	0x000013BF	5056					
.text	Stack	0x1FFE0620	0x1FFE0A1F	1024				(2)	
	Uninitialized D	0x90000000		0				(3)	
.stack_dum .sdram		0x0100A160		0					
.stack_dum	Constant			0			0x000013C8		
.stack_dum .sdram .rel.dyn .qspi_non_re	Constant Others	0x60000000							
.stack_dum .sdram .rel.dyn	Constant	0x60000000 0x60000000		0					



The following operations are supported in the Memory Usage view:

- . Choose a map or library list file for Memory Usage display.
- *: Refresh all information in the Memory Usage view.
- Export data for all tabs in the **Detail** table region
- E: Open *.map or *.lbp file in Editor (there is no library list file in the RA library project).
- Some the Map file output page of the selected project.
- 4. Open the Section page of the selected project.



5.3.7 Disassembly View

The **Disassembly** view shows the loaded program as assembler instructions mixed with the source code for comparison. The currently executing line is highlighted by an arrow marker in the view. In the **Disassembly** view, users can set breakpoints at assembler instructions, 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 on Windows \rightarrow Show View \rightarrow Disassembly to open the Disassembly view.
- 2. Click on the 🔄 icon to enable synchronization between the assembly source and the C source (active debug context).
- 3. In the **Disassembly** view, right-click at the address column to select **Show Opcodes** and **Show Function Offsets**.

	startup.c 52							
	52		oid Reset_Ha	dlan (vo	id)			
	⇒ 53	f34 {	JIG RESEC_Hai		10)		^	
	54		/* Initia	liza svet	em using BSP	*/		
	55	f36	SystemInit		en using bor	• /		
	56	150	Systematia	-(/)				
	57		/* Call us	ser annli	cation. */			
	58	f3a	<pre>main();</pre>		,			
	59							
	60	0	while (1)					
	61		{					
	62		•	finite Lo	op. */			
	63		}		•			
	64	}						
	65						\checkmark	
		<					>	
Disassembly	1 2					Enter location h	e 🗸 🎗 🖞 🕏 📴 🖻	~ - 6
				Reset_Har				^
00000 1 34 55	: 0x0000	08b5 Reset	_Handler+0		<pre>{r3, lr} emInit();</pre>			
00000f36	: 0x00f0	03f8 Reset	_Handler+2	bl	0xf40 <syst< td=""><td>:emInit></td><td></td><td></td></syst<>	:emInit>		
58				main				
			_Handler+6	bl	0x648 <main< td=""><td></td><td></td><td></td></main<>			
00000f3e	: 0x0000	fee7 Reset	_Handler+10	b.n		et_Handler+10>		
				SystemIn				
		244a Syste		ldr			SystemInit+148>)	
		8830 Syste			r3, [r2, #1		0	
	: 0X43†4	7003 Syste	201117+6	orr.w	rs, rs, #15	;728640	0xf00000	
105 00000f4a	: 0x0000	70b5 Syste	emInit+10	{ push	{r4, r5, r6	5, lr}		~
	Opco	odes Fu	Inction Offsets					

Figure 5-22. Debug – Disassembly View



5.3.8 Variables View

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

To observe a local variable (for example, leds for function hal_entry ()):

- 1. Click on Windows \rightarrow Show View \rightarrow Variables to open the Variables view.
- 2. Step into the function hal_entry () to view the local variable timeout value.

lame	Туре	Value	Name : leds	
🖻 leds	bsp. leds. t	{}	Details:{led_count = 3, p_leds = 0x	<pre>(16c4 <g_bsp_prv_leds>}</g_bsp_prv_leds></pre>
⊯ led coun	t uint16 t	3	Default:{} Decimal:{}	
→ p_leds	const uint16_t *	* 0x16c4 <g_bsp_prv_le< td=""><td></td><td></td></g_bsp_prv_le<>		
			> <	>
[RA_Tutorial] RA	Configuration 🛛 🚨 l	hal_entry.c 🛛 🖻 startup.c	🖻 main.c	-
32 33 34 35 36 37 38 39 40 41	<pre>const bsp /* Set th const uin /* Calcul const uin /* LED ty</pre>	<pre>the units to be use p_delay_units_t bsp_d ne blink frequency (m nt32_t freq_in_hz = 2 late the delay in ter nt32_t delay = bsp_de ype structure */</pre>	<pre>d with the software delay function */ elay_units = BSP_DELAY_UNITS_MILLISECONDS; ust be <= bsp_delay_units */ ; ms of bsp_delay_units */ lay_units / freq_in_hz;</pre>	
42 566	bsp_leds_	_t <mark>leds</mark> = g_bsp_leds;		
43 44	if (0 == {	is board has no LEDs leds.led_count)	then trap here */	
45 574 46 47 48 49 50	<pre>while { ; }</pre>		// There are no LEDs on this board	

Figure 5-23. Debug – Variables View



5.3.9 IO Registers View

The IO Registers are also known as the Special Function Registers (SFRs). The **IO Registers** view displays all the registers defined in a target-specific IO file. Users can further customize the **IO Registers** view by adding specific IO registers to the **Selected Registers** pane.

To view selected IO registers:

- 1. Click on the **IO Registers** view (icon) or select the **Window** menu → **Show** View → **Other** to open the Show View window, and then select **Debug** → **IO Registers** to open the view.
- 2. Under the **All Registers** tab, locate a module (for example, CAC) in the **IO Registers** view. Expand its IO register list.
- 3. Drag and drop its registers (the CAICR and CASTR) to the **Selected Registers** pane. A green dot next to the IO register indicates the status of being a selected register.
- 4. Switch to the Selected Registers tab to view the selected IO Registers.

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

lame	Value (Hex)	Value (Bin)	Address	Access	
<pre> CAC </pre>					
> CACR0	0x00	0000000	0x40044600	RW	
> CACR1	0x00	0000000	0x40044601	RW	
> CACR2	0x00	0000000	0x40044602	RW	
> CAICR	0x00	0000000	0x40044603	RW	
> CASTR	0x00	0000000	0x40044604	R	
> CAULVR	0x0000	00000000	0x40044606	RW	
> CALLVR	0x0000	00000000	0x40044608	RW	
> CACNTBR	0x0000	00000000	0x4004460a	R	
Il Registers Selected	Registers				

Figure 5-24	Debug – IO Registers View
1 iguie 3-24.	Debug - IO Registers view

00000000 0x40044603

0000000 0x40044604

RW

R

All Registers Selected Registers

CACCAICR

> • CASTR

0x00

0x00



An item selected in the IO Registers view can be set as an Eventpoint.

1. Right-click on the target I/O register (e.g. CACR0) and select **Set Event Break** from **Renesas Eventpoints**. When the dialog box is displayed, set the conditions and click on the **OK** button.

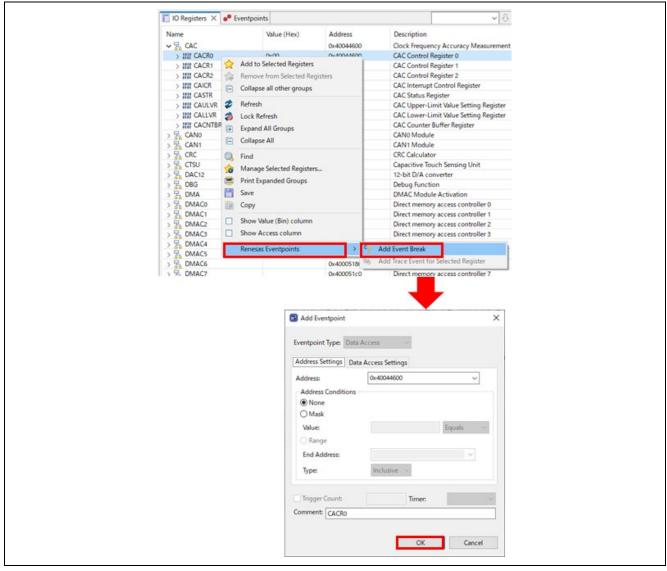


Figure 5-25. Debug – Setting an Item as an Eventpoint

2. An event break is set in the **Eventpoints** view.

📄 IO Registers 🧧 Eventp	points X				i	🔌 🎽	~~~~
Туре	Address	Data	Count	Timer	Handle	Commen	t
🗌 🎁 Trace Start							
🔄 🞬 Trace Stop							
🔄 🥤 👕 Trace Record							
~ 🔽 🕄 Erent Dreak							
✓ OR	0x40044600	Read/Write Byte			19	CACR0	
C. Timor Start			_				
🗌 🕐 Timer Stop							

Figure 5-26. Debug – [Eventpoints] View



5.3.10 Eventpoints View

An 'event' refers to a combination of conditions set for executing break or trace features during program execution. The **Eventpoints** view enables users to set up or view defined events of different categories, for example, trace start, trace stop, or event break.

Data access event break is supported for RA projects. The emulator detects access under a specified condition to a specified address or a specified address range, allowing complex address and data-matching criteria to be set up.

The e^2 studio 2023-01 and later versions support a vector catch feature that stops a program when an exception occurs. This feature enables stopping a program when an exception occurs without using a hardware breakpoint.

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

 Table 5-1.
 Event combination

Event combination	Explanation
OR	The condition is met when any one of the specified events occurs.
AND (cumulative)	The condition is met when all of the specified events occur, regardless of the timing.
Sequential	The condition is met when the specified events occur in a specified order.

To set an event break for a global variable when address/data is matched (for example, when g_bsp_leds is accessed):

Click on Renesas Views \rightarrow Debug \rightarrow Eventpoints to open the Eventpoints view.

Double-click on the Event Break option to open the Edit Event Break dialog box.

Click on the Add... button to continue.

Туре		Addre	SS	C	ata	Cou.	. Tin	ner l	Han	Comment	
	Trace Start	C Edit Event Br	eak								
	Trace Record									Trigger: OR	•
	Event Break Timer Start Timer Stop	Туре	Ad	dress	Data		Count	Timer	Channel	Comment	
<											
Project	Saved Templates	_									
		Add E	dit	Delete PC	:: 0/0 OA: 0/4 A	All: 0/4				ОК	Cancel

Figure 5-27 Debug – [Eventpoints] View (1/2)



Select the **Data Access** eventpoint type.

Go to the Address Settings tab and click on the [...] icon to browse for the symbol g_bsp_leds . (The address of this global variable is g_bsp_leds .)

Next, switch to the Data Access Settings tab and set the Read/Write selection to Read.

Click on **OK** to proceed.

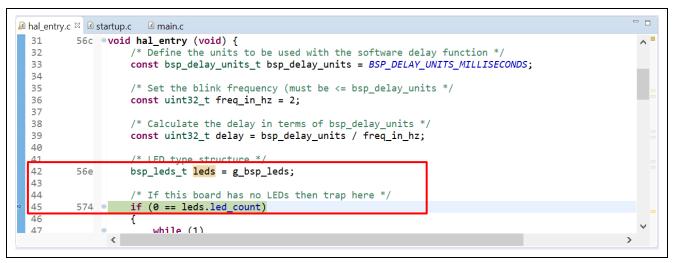
Add Eventpo	pint	× State Add Eve	entpoint		×
	s: Current Control Con	Address Data Set Read/Wi Size: Bus Mas	rite: ter: pare Settings: e: lue: ison: r Count:		el
● Eventpoints 🛛		X	× ¾ 🖻	PC: 1/6	0A: 1/4 🗞 🖻 🗖
Type □ [™] Trace Start □ [™] Trace Stop	Address	Data	Cou Time		
Trace Record					
v ✓ ^c i Event Break	&g_bsp_leds (0	Read Byte		12	
⊙ Timer Start ⊙ Timer Stop					
< Project Saved Templates					>

Figure 5-28 Debug – Eventpoints View (2/2)



Perform a reset to execute the program from the start.

The figure below shows that when the variable g bsp leds is accessed (read), the program stops.





To use the feature for stopping a program without using a breakpoint (e^2 studio 2023-01 and later versions) when an exception occurs, click on the button shown below to show the **Vector Catch** window.

Type	Address	Data	Count	Timer	Handle	Comment
Trace Start	Address	Data	count	miner	rianaic	comment
Trace Stop						
Trace Record						
C Event Break						
🗌 😥 Timer Start						
🗌 🕐 Timer Stop						
Vector Catch						×
Vector Catch Settin	qs					
VC_CORERESET	Enable Reset Vector Catch.					
VC_MMERR	Enable debug event on a N	lemory Management except	ion.			
VC_MMERR				ocessor.		
	Enable debug event on a L	JsageFault caused by an acce	ss to a copr		error or divi	de-by-zero trap.
	Enable debug event on a L Enable debug event on a L		ss to a copr by an alignr	ment check		de-by-zero trap.
VC_NOCPERR	Enable debug event on a U Enable debug event on a U Enable debug event on a U	JsageFault caused by an acce JsageFault exception caused JsageFault exception caused	ss to a copr by an alignr	ment check		de-by-zero trap.
VC_NOCPERR	Enable debug event on a L Enable debug event on a L Enable debug event on a L Enable debug event on a B	JsageFault caused by an acce JsageFault exception caused JsageFault exception caused JusFault exception.	ss to a copr by an alignr by a state in	nent check formation (error.	de-by-zero trap.
VC_NOCPERR	Enable debug event on a L Enable debug event on a L Enable debug event on a L Enable debug event on a B Enable debug event on a fa	JsageFault caused by an acce JsageFault exception caused JsageFault exception caused AusFault exception. ault occurring during except	ss to a copr by an alignr by a state in	nent check formation (error.	de-by-zero trap.
VC_NOCPERR	Enable debug event on a L Enable debug event on a L Enable debug event on a L Enable debug event on a B	JsageFault caused by an acce JsageFault exception caused JsageFault exception caused JusFault exception. ault occurring during except lardFault exception.	ss to a copr by an alignr by a state in	nent check formation (error.	de-by-zero trap.
 VC_NOCPERR VC_CHKERR VC_STATERR VC_BUSERR VC_INTERR VC_HARDERR 	Enable debug event on a L Enable debug event on a L Enable debug event on a L Enable debug event on a B Enable debug event on a f Enable debug event on a F	JsageFault caused by an acce JsageFault exception caused JsageFault exception caused JusFault exception. ault occurring during except lardFault exception.	ss to a copr by an alignr by a state in	nent check formation (error.	de-by-zero trap.

Figure 5-30. Debug – Break with the Use of the Vector Catch Feature



5.3.11 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 users to track the program execution flow to search for and examine the points where problems arise.

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

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

- 1. Click on **Renesas Views** \rightarrow **Debug** \rightarrow **Trace** to open the **Trace** view.
- 2. Turn on the **Trace** view by selecting the *icon*.

🖳 Co	🗷 Tasks	🖹 Pro	Exe	🖄 Liv	🏶 Sm	Ren		🎭 Trace 🛛						
							3 🕨 🛛	‡ Q §	2 😂 🖛	🚖 🖄 🔖	₩ 🗸 🕼	% U %	1	$\overline{\nabla}$
No recor	rds													
Record	Label	Addr	Source											

Figure 5-31. Debug – Turn on Trace View



- 3. Execute the program and stop program execution by using a breakpoint or by pressing the **Suspend** button on the **Debug** toolbar. The content stored in trace memory at that point in time is displayed as a trace result.
- 4. Select the display mode by clicking on the corresponding button. The following figure shows the trace result before the main() function is executed.

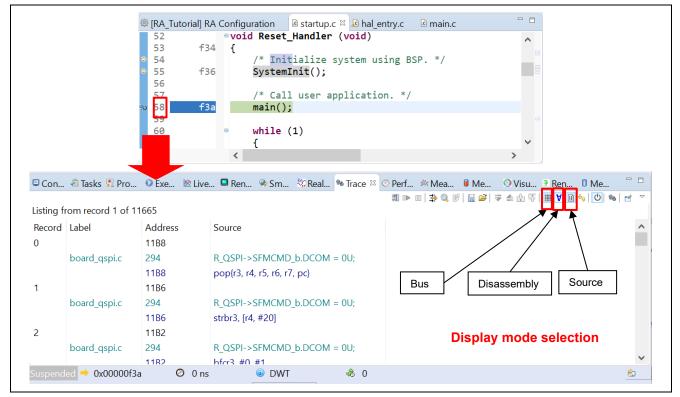


Figure 5-32. Debug – Select Display Mode in Trace View

The trace records are displayed from the oldest data to the latest data by default. The display order can be changed by clicking on the solution.

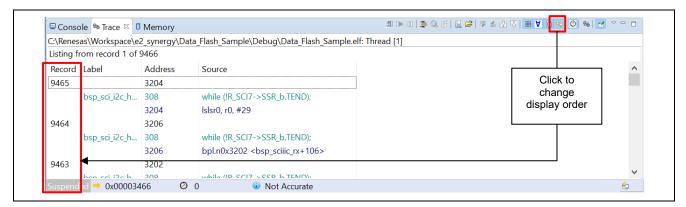


Figure 5-33. The Display Order is Changed



The trace result can be filtered by clicking on the \Rightarrow button. You can select filtering by **Record** and/or **Address**.

	e [™] Ren [©] Me [◆] Fau [□] □ [™] ▲ ☆ [™] [™] [™] [™] [™] [™] [™] [™]
e ² Filter	Address 0 Range 0 Exclude

Figure 5-34. Debug – Filter Trace Result

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

🗣 Trace 🛱 🖏 R	ea 🔋 Me	■ Ren 0 Me	. 🚸 Fau	- 8
3 🕨 🖉		≜ 🖞 🖏 🗰 🗸 🖾) 🍫 🕛 🕹 🛛	₹ ▽
Save trace res	ult L	oad trace result		
				\mathbf{A}

Figure 5-35. Debug – Save And Load Trace Result



5.3.12 Fault Status View

The **Fault Status** view shows the bit status of several fault status registers and the value of the key register to the user when a hardware fault crash occurs. When a hardware fault occurs, the bits of the register related to the cause of the fault are checked, and the r0, r1, r2, r3, r12, lr, pc, and psr register values are displayed. This is shown in the figure below. This function is available in e^2 studio v5.2 and above.

🖻 Console 🧔 Tasks 🖹 Problems 🖄 Live Trace Console 🛸 Trace 🔶 Fault Status 🖾 🔋 Memory Usage 🏾 Memory	i - 🗆
+ HFSR	
• MMFSR	
• BFSR	
• UFSR	
Registers	



✓ HFSR 0x4000000	
VECTTBL [1] FORCED [30] DEBUGEVT [31]	
MMFSR 0x0	
► BFSR 0x0	
▼ UFSR 0x1	
UNDEFINSTR [0]	
INVSTATE [The processor has attempted to execute an undefined instruction	
When mouse hovers over the bit name,	
the dependent on of the hit is discussed	
▼ Registers	
✓ Registers 1000 rO	0x1ffe093c
888 rO 888 r1	0x4001e3fe
8189 r0 8189 r1 8189 r2	0x4001e3fe 0x0
6161 rO 6060 r1 6060 r2 6060 r3	0x4001e3fe 0x0 0x4001e0a0
888 r0 888 r1 888 r2 888 r3 888 r3	0x4001e3fe 0x0 0x4001e0a0 0x1ffe08b0
6161 rO 6060 r1 6060 r2 6060 r3	0x4001e3fe 0x0 0x4001e0a0

Figure 5-37. Fault Status Hardware Fault Occurred



5.3.13 Run Break Timer

The **Run Break Timer** feature allows the user to see the last execution performance on the status bar. When the program is suspended, the user can check the current program counter (PC), the last execution timing either in time or CPU cycles, and the accuracy or measurement method used.

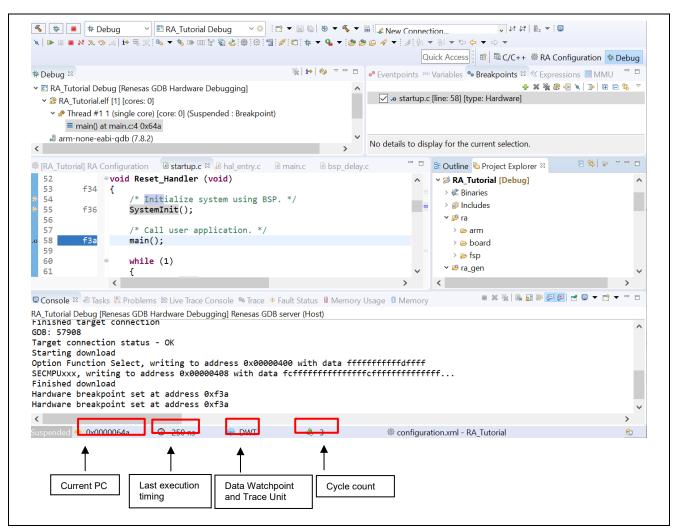


Figure 5-38. Run Break Timer Shows the Last Execution Performance

The following table shows the support of the Run Break Timer feature available for various RA devices.

Table 5-2.	Support for Run Break Timer
------------	-----------------------------

Device Debugger		Support
RA2 Series (Cortex-M23)	J-Link	System Time
RA4, RA6 Series	J-Link	Data Watchpoint and Trace Unit (DWT) – Cycle Count and number of overflows calculated using the System Time

The **Run Break Timer** feature is supported in e^2 studio v7.3.0 and higher versions. For updates in the specification, refer to the e^2 studio release note at this link: <u>https://www.renesas.com/e2studio</u>.



6. Setting up a FreeRTOS Application

This example shows how to generate and build an RA project that includes FreeRTOS objects and a General Purpose Timer (GPT) module using the project template **FreeRTOS – Blinky – Static Allocation**.

6.1 General Purpose Timer Example in FreeRTOS

In the **FreeRTOS – Blinky – Static Allocation** RA project from **Project Template Selection**, LEDs are blinked by putting the task to delay for a while before toggling the LEDs state.

In this example, instead of a delay, the Blinky Thread waits for a semaphore and a timer interrupt (generated by GPT), which puts this semaphore every 1 second so that the thread can resume.

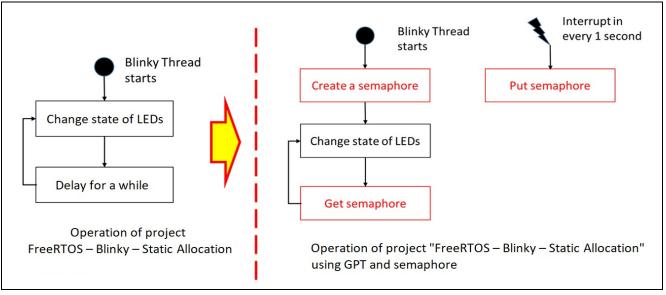


Figure 6-1. Setting Up a FreeRTOS Application – Introduction



6.2 Creating the Sample Project

To create a sample FreeRTOS project with GPT and semaphore, configure the RA project as follows:

 Invoke the New Project editor and follow the steps in Chapter 3.1 (Generating a New RA Project for a Non-TrustZone device) to generate a new project. However, in the Build Artifact and RTOS Selection dialog, select FreeRTOS, and in the Project Template dialog, select FreeRTOS – Blinky – Static Allocation.

Renesas RA C/C++ Project	
Build Artifact and RTOS Selection	
Build Artifact Selection	RTOS Selection
 Executable Project builds to an executable file 	FreeRTOS (v10.4.6+fsp.
 Static Library Project builds to a static library file 	
 Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static library project 	
	< Back Next > Einish Cancel
Paparas PA C/C++ Project	– – ×
Renesas RA C/C++ Project	
	2
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Alloca FreeRTOS FSP project that includes BSP and	ation I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Alloca FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCI LEDs will be started. [Renesas.RA.]]pack]	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Alloca FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCI LEDs will be started. [Renesas.RApack] FreeRTOS - Minimal - Static Allo	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Alloca FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCI LEDs will be started. [Renesas.RApack] FreeRTOS - Minimal - Static Allo Empty FreeRTOS FSP project with no threads	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the Decation
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Allocal FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCU LEDs will be started. [Renesas.RApack] FreeRTOS - Minimal - Static Allo Empty FreeRTOS FSP project with no threads initialize the MCU using the BSR	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the Decation
Project Template Selection Project Template Selection FreeRTOS - Blinky - Static Allocal FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCU LEDs will be started. [Renesas.RApack] FreeRTOS - Minimal - Static Allo Empty FreeRTOS FSP project with no threads initialize the MCU using the BSP. [Renesas.RApack]	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the Decation
 FreeRTOS - Blinky - Static Allocation FreeRTOS FSP project that includes BSP and allocation. This project will initialize the MCU LEDs will be started. [Renesas.RA. pack] FreeRTOS - Minimal - Static Allocation FreeRTOS FSP project with no threads initialize the MCU using the BSR [Renesas.RA. pack] Code Generation Settings 	I will blink LEDs if available. FreeRTOS is pre-configured for static memory U using the BSR FreeRTOS will also be initialized and a single thread to blink the Decation

Figure 6-2. Setting Up a FreeRTOS Application - Create New Project

- 2. Open the **Stacks** page in the **RA Project Configuration**. Please refer to chapter 3.5.5: Stacks Configuration Page.
- 3. Add the GPT module to the Blinky Thread by selecting **Blinky Thread** in the **Threads** panel and selecting **⁴ ⁴ New Stack** → **Timers** → **Timer, General PWM(r_gpt))** in the **Stacks** panel.

[RA_Tutori	ial] FSP Configuration	c hal_entry.c	🖻 startup.c	🖻 main.c	💼 system.c	[RA_FreeRTOS] F:	SP Configuration $ imes$	- 6]	🌮 FSP Visualization	🔓 Package	🔜 Disas
tacks Co	onfiguration						Generate Projec					Enter loo
	-						Generate Projec	ct Content	_	No debug context		
Threads	🔄 New Thread 🧃	Remove	Blinky Thread Sta	cks		💽 New Stack >	🐣 Extend Stack > 🛛 🔊	Remove				
V 💣 HAL	/Common		Add stag	ks to the selecte	d thread by using	the 'New Stack' tool	Analog	>				
 g_ioport I/O Port (r_ioport) FreeRTOS Port (rm_freertos_port) Blinky Thread 	here from the clipboard.				Artificial Intelligence	>						
	s_port)					Audio	>					
						Bootloader	>					
							CapTouch	>				
							Connectivity	>				
							DSP	>				
							Graphics	>				
							Input	>				
							Monitoring	>				
							Motor	>				
							Networking	>				
							Power	>				
bjects	New Object >	Remove					RTOS	>				
-,		- <u></u>					Security	>				
							Sensor	>				
							Storage	>				
							System	>				
							Timers	> (€	Port Output Enable	for GPT (r_poeg	g)
							Transfer	> (€	Realtime Clock (r_rtc)	
						4	Search	•	€	Three-Phase PWM (_gpt_three_ph	ase)
						~	Jearenni		€	Timer, General PWM	(r_gpt)	
								4	Ð	Timer, Low-Power (r	agt)	-

Figure 6-3. Setting Up a FreeRTOS Application – Adding the GPT Module



- 4. Configure the GPT module as follows.
 - Name: g_timer
 - Mode: Periodic
 - Period: 1
 - Period Unit: Seconds
 - Callback: gpt_callback
 - Overflow/Crest Interrupt priority: Priority 2

寧 *[RA_Free	eRTOS] RA Configuration ⊠						- 6		
Stacks O	Configuration					Generate Projec	ct Content		
Threads	🗟 New Thread 🔹 Rem	ove 🖻 g_timer Timer Dr	gpt Stacks	Stacks 🔊 New Stack > 🛎 Extend					
Y ⊜ Blink ⊕ g_	cy Thread timer Timer Driver on r_gpt	◆ g_timer Time ↓ r_gpt	r Driver or	h					
Objects	😢 New Object > 🖻 R	emove		_					
Summary [BSP Clocks Pins Interrupts Event Links	Stacks Components							
Propertie	es 🛿 🕄 Problems 👒 Smart Browser		~	Bein Conflicts ∞		1	⇒		
				0 items					
g_umer	Timer Driver on r_gpt	Disablea		Description	^	Module	Pin		
Settings	✓ Module g_timer Timer Driver on r_g	pt	^	boonpaon		inoutic			
API Info	✓ General								
	Name	g_timer							
	Channel	0							
	Mode	Periodic							
	Period	1							
	Period Unit	Seconds	_						
	> Output	,							
	> Input								
	✓ Interrupts								
	Callback	gpt_callback							
	Overflow Interrupt Priority	Priority 2							
	Capture A Interrupt Priority	Disabled	~						
	<		>	<			2		

Figure 6-4. Setting Up a FreeRTOS Application – GPT Module Configuration



5. Add a semaphore object to the **Blinky Thread** by selecting the **Blinky Thread** in the **Threads** panel and selecting **New Object** → **Binary Semaphore** in the **Objects** panel.

Stacks Confi	i] RA Configuration ⊠					Generate Projec	t Conten
Threads	🗐 New Thread 🔹 Remo	ove 😑 Blinky Thre	ad Stacks		🗟 New Stack	> ≜ Extend Stack > €) Remove
👻 🛛 🖉 👻 👻	t I/O Port Driver on r_ioport		Timer Driver or	1			
Objects		nting Semaphore					
	Ever	nt Group					
	• Mes	nt Group sage Buffer					
🗆 Properties 😣 통	locks Pins Interrupts Eve Mes Problems & Smart Bro Stree	sage Buffer ex		the Pin Conflicts ≅		1	<u>↓</u> ∨ □ [
□ Properties ¤ 🖁	locks Pins Interrupts Eve Mut Problems & Smart Bro d	sage Buffer ex ue am Buffer		0 items	^		
□ Properties 🛛 🖁 Blinky Thread	locks Pins Interrupts Eve Mut Problems Smart Bro d HOOKS	sage Buffer ex ue am Buffer			^	3 Module	⇒ ⊽ ⊏ Pin
Properties 🛛 🖁 Blinky Thread Settings → S	Iocks Pins Interrupts Eve Mut Problems & Smart Bro Cue Stree Time Stats	sage Buffer ex ue am Buffer		0 items	^		
□ Properties ≈ Blinky Thread Settings > S	locks Pins Interrupts Eve Mut Problems Smart Bro d HOOKS	sage Buffer ex ue am Buffer		0 items	^		
Properties Properties Settings	Iocks Pins Interrupts Eve Mut Problems Smart Bro Stree Time Stats Memory Allocation	sage Buffer ex ue am Buffer		0 items	^		
Properties Blinky Thread Settings Strings St	Iocks Pins Interrupts Eve Mut Problems Smart Bro d Stats Memory Allocation Timers	sage Buffer ex ue am Buffer		0 items	^		
Properties Blinky Thread Settings Strings St	A Mess A Mess A Mess A Mess A Mut A Mut	sage Buffer ex ue am Buffer		0 items	^		
Properties 🛛 🖗 Blinky Thread Settings > 5 > 1 > 1 > 0 > 6 • Thr	A Mess A Mess A Mess A Mess A Mut A Mut	sage Buffer ex ue am Buffer		0 items	^		

Figure 6-5. Setting Up a FreeRTOS Application – Adding A Semaphore Object

6. Configure this newly created semaphore as follows:

Name: Blinky Semaphore

Symbol: g_blinky_semaphore

Stacks Conf	iguration					Generate Proje	ect Conten
Threads	🗟 New Thread 🔞 Remov	ve 🖻 Blink	y Thread Stacks		🗟 Nev	v Stack > ≗ Extend Stack > ∦	🗈 Remove
 Blinky Three g_timer 	t I/O Port Driver on r_ioport	r	9_timer Timer Driver or _gpt				
Summary BSP C	locks Pins Interrupts Event Links S	itacks Compo		R.R. G. Hunn			*2 ▽ □ 1
Summary BSP C □ Properties ≈ 3	llocks Pins Interrupts Event Links S		nents	Be Pin Conflicts ≅			≱ ⊽ ⊟ (
Summary BSP C	locks Pins Interrupts Event Links S			0 items	^		
Summary BSP ⊂ □ Properties ¤ g_new_binary	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema				^	Module	‡> ⊽ ⊟ t
Summary BSP C Properties × g_new_binary Settings Prop	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema	aphore		0 items	^		
Summary BSP C Properties & g g_new_binary Settings Prop Sy	locks Pins Interrupts Event Links S Problems Smart Browser y_semaphore0 Binary Sema perty	phore Value		0 items	^		
Summary BSP C Properties & g g_new_binary Settings Prop Sy	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema verty mbol	aphore Value g_blinky_sen		0 items	^		
Summary BSP C Properties & g g_new_binary Settings Prop Sy	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema verty mbol	aphore Value g_blinky_sen		0 items	^		
Summary BSP C Properties & g g_new_binary Settings Prop Sy	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema verty mbol	aphore Value g_blinky_sen		0 items	^		
Summary BSP C Properties & g g_new_binary Settings Prop Sy	locks Pins Interrupts Event Links S Problems & Smart Browser y_semaphore0 Binary Sema verty mbol	aphore Value g_blinky_sen		0 items	*		

Figure 6-6. Setting Up a FreeRTOS Application – Semaphore Object Configuration



0

- 7. Press **Ctrl + S** to save the setting and click the **Generate Project Content** Generate Project Content button to generate source code content.
- 8. Open src\blinky_thread_entry.c and implement the following contents:
 - Add source code to initialize the GPT module before the "while(1)" loop in blinky_thread_entry().

```
g_timer.p_api->open(g_timer.p_ctrl, g_timer.p_cfg);
```

```
g_timer.p_api->start(g_timer.p_ctrl);
```

- Delete the task delay instruction and add code to wait for the semaphore in blinky_thread_entry(). xSemaphoreTake(g_blinky_semaphore, portMAX_DELAY);
- Implement the gpt_callback() function to signal the semaphore for the Blinky thread.

```
void gpt_callback(timer_callback_args_t *p_args) {
```

```
(void)p_args;
```

static signed portBASE_TYPE xHigherPriorityTaskWoken;

```
xSemaphoreGiveFromISR(g blinky semaphore, &xHigherPriorityTaskWoken);
```

```
}
```

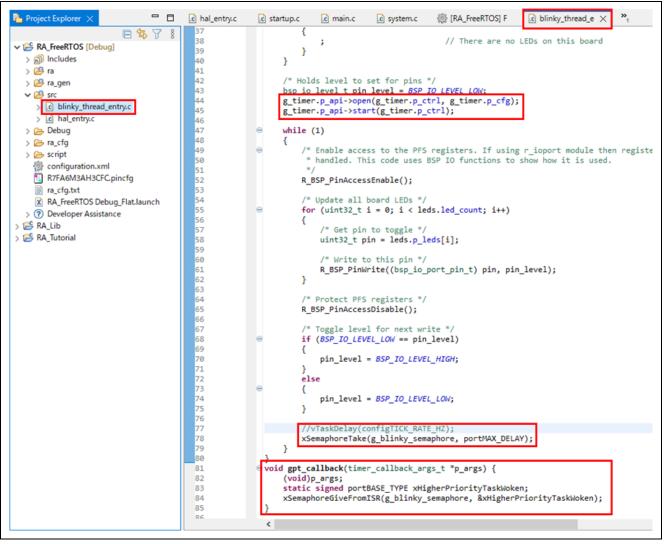


Figure 6-7. Setting Up an FreeRTOS Application – Adding User Source Code

9. Build and run the project on the EK-RA6M3 board. Confirm that the LEDs are turned ON/OFF every 1 second.

7. Setting up an Azure RTOS Application

This example shows how to generate and build an RA project that includes Azure RTOS objects and a "General Purpose Timer" (GPT) module using the project template Azure RTOS ThreadX—**Blinky**.

7.1 General Purpose Timer Example in Azure RTOS

In the **Azure RTOS ThreadX – Blinky** RA project from **Project Template Selection**, LEDs are blinked by putting the task to delay for a while before toggling the LEDs state.

In this example, instead of a delay, the Blinky Thread waits for a semaphore and a timer interrupt (generated by GPT), which puts this semaphore every 1 second so that the thread can resume.

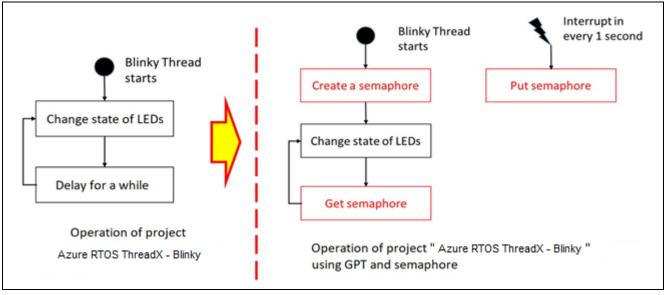


Figure 7-1. Setting up an Azure RTOS Application – Introduction



7.2 Creating the Sample Project

To create a sample Azure RTOS project with GPT and semaphore, configure the RA project as follows:

 Invoke the New Project editor and follow the steps in Chapter 3.1 (Generating a New RA Project for a Non-TrustZone device) to generate a new project. However, in the Build Artifact and RTOS Selection dialog, select Azure RTOS ThreadX, and in the Project Template dialog, select Azure RTOS ThreadX – Blinky.

Renesas RA C/C++ Project Build Artifact and RTOS Selection	
Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file Executable Using an RA Static Library Project builds to an executable file Project uses an existing RA static library project	RTOS Selection Azure RTOS ThreadX (v6.1.11+fsp.
?	< <u>Back Next > Einish</u> Cancel
	- • ×
	: BSP and will blink LEDs if available. This project will initialize the MCU using the
Renesas RA C/C++ Project Project Template Selection Project Template Selection Azure RTOS ThreadX - Blinky Azure RTOS ThreadX project that includes BSR ThreadX will also be initialized and a se [Renesas.RA] pack] Azure RTOS ThreadX - Minima	BSP and will blink LEDs if available. This project will initialize the MCU using the single thread to blink the LEDs will be started.
Renesas RA C/C++ Project Project Template Selection Project Template Selection Azure RTOS ThreadX - Blinky Azure RTOS ThreadX project that includes BSR ThreadX will also be initialized and a [Renesas.RA] pack] Azure RTOS ThreadX - Minima Empty ThreadX FSP project with no thread	BSP and will blink LEDs if available. This project will initialize the MCU using the single thread to blink the LEDs will be started.

Figure 7-2. Setting up an Azure RTOS Application - Create New Project



- 2. Open the **Stacks** page in the **RA Project Configuration**. Please refer to chapter 3.5.5: Stacks Configuration Page.
- 3. Add the GPT module to the Blinky Thread by selecting **Blinky Thread** in the **Threads** panel and selecting **[€] New Stack** → **Timers** → **Timer, General PWM(r_gpt)** in the **Stacks** panel.

💼 hal_entry.c 💼 startup.c 🌼 [RA	FreeRTOS] F 💽 blinky_thread_e	[RA_Azure_RTOS] >	< »3	- 8	🌮 FSP Visualization	👩 Package 🔛 Disassemb
Stacks Configuration			Generate Project (lo debug context	
			Generate Project (Lontent		
New Thread 🙀 Remove	Blinky Thread Stacks	🔄 New Stack > 🚔	Extend Stack > 🛛 🐔 R	emove		
	Add stacks to the selected thread by using the or by pasting here from the clipboard.		Analog	>		
V 🖉 HAL/Common		e clipboard.	Artificial Intelligence	>		
g_ioport I/O Port (r_ioport) Azure RTOS ThreadX Port (rm_thr			Audio	>		
Blinky Thread			Bootloader	>		
			CapTouch	>		
			Connectivity	>		
			DSP	>		
			Graphics	2		
			Input Monitoring	2		
			Motor	Ś		
			Networking	Ś		
			Power	Ś		
			Security	,		
			Sensor	>		
			Storage	,		
< >			System	>		
Objects 🔄 New Object > 🔊 Remove			Timers	> 🕀	Port Output Enabl	e for GPT (r_poeg)
			Transfer	> 🕀	Realtime Clock (r_r	tc)
		A	Search	+		(r_gpt_three_phase)
						1.000
				+	Timer, Low-Power	(r_agt)
1						

Figure 7-3. Setting up an Azure RTOS Application – Adding the GPT Module



- 4. Configure the GPT module as follows.
 - Name: g_timer
 - Mode: Periodic
 - Period: 1
 - Period Unit: Seconds
 - Callback: gpt_callback
 - Overflow/Crest Interrupt priority: Priority 2

戀 *[RA_A	zure_RTOS] FSP Configuration 🔀			
Stacks (Configuration			Generate Project Content
Threads	🔄 New Thread 🛛 🙀 Remove 📄	g_timer0 Timer Driver on r_gpt Stacks	🗿 New Stack >	🚢 Extend Stack > 📓 Remove
•	inky Thread ∧ g_timer Timer Driver on r_gpt ∨	 g_timer Timer Driver on r_gpt 		^
Objects	🛃 New Object > 🔬 Remove			~
Summary	BSP Clocks Pins Interrupts Event Links S	itacks Components		
	ties 🔀 🎇 Problems 🏽 🏶 Smart Browser		28 🗆 🗖	Pin Conflicts
			i o u	0 items
g_timer '	Timer Driver on r_gpt			^
	D			Description
Settings	Property	Value		
API Info	> Common			
	 Module g_timer Timer Driver on r_gpt 			
	✓ General			
	Name	g_timer		
	Channel	0		
	Mode	Periodic		
	Period	1		
	Period Unit	Seconds		
	> Output			
	> Input			
	✓ Interrupts			
	Callback	gpt_callback		
	Overflow/Crest Interrupt Priority	Priority 2		

Figure 7-4. Setting up an Azure RTOS Application – GPT Module Configuration

5. Add a semaphore object to the **Blinky Thread** by selecting the **Blinky Thread** in the **Threads** panel and selecting **New Object** → **Semaphore** in the **Objects** panel.

∰ *[RA_Azure_RTOS] FSP Configuration ⊠	
Stacks Configuration	Generate Project Content
Threads New Thread Remove Guide Blinky Thread Guide Guid	pre

Figure 7-5. Setting up an Azure RTOS Application – Adding A Semaphore Object

6. Configure this newly created semaphore as follows:



Name: Blinky Semaphore

Symbol: g_blinky_semaphore

#[RA_Azure_RTOS] FSP Configuration X log bl Stacks Configuration	inky_thread_entry.c Generate Project Content
Threads New Thread Remove g_timer Timer Driver on r_gpt g_timer Timer Driver on r_gpt g_binky_semaphore Objects New Object > Remove g_blinky_semaphore Semaphore Summary BSP Clocks Pins Interrupts Event Links Properties X Problems Smart Browse 	
g_new_semaphore Semaphore	
Settings Property Value Name Blinky Semaphore Symbol g_blinky_semaphore Initial count 0	re

Figure 7-6. Setting up an Azure RTOS Application – Semaphore Object Configuration





Figure 7-7. Setting up an Azure RTOS Application – Adding User Source Code

9. Build and run the project on the EK-RA6M3 board. Confirm that the LEDs are turned ON/OFF every 1 second.



8. Help

The help system allows users to browse, search, bookmark, and print help documentation from a separate **Help** window or **Help** view within the workbench. From here, users can also access an online forum dedicated to the e^2 studio.

Click on the Help tab to open the Help menu.

<u>H</u> elp			
3	Welcome (1)		n 🗉 N 3. 3- 10 🗟 73 🕸 🗞
?	Help Contents (2)		-
89	Search		re_RTOS] × », □ □ Ø
	Show Context Help (3)		
	Show Active Keybindings	Ctrl+Shift+L	Generate Project Content
	Cheat Sheets		
	Renesas Help	>	🔏 RenesasRulz Community Forum (4)
	CMSIS Packs Management	>	Renesas Helpdesk
Ø	Add Renesas Toolchains		Renesas e2 studio feedback
۹	Eclipse User Storage	>	
2	Perform Setup Tasks		
<i>e</i>	Check for Updates		
6 -	Install New Software		
\$	Eclipse Marketplace		
	Install New Device Family Support		
	IAR Embedded Workbench plugin manager.		
0	About e ² studio		

Figure 8-1. Help – Help Menu

Quick Help Tips:

- 1. Click on **Welcome** for an overview of the e² studio and to view Release Notes.
- 2. Click on **Help Contents** to open a separate Help window with a search function.
- 3. Click on **Show Context Help** to open the Help view within the workbench.
- 4. Click on **RenesasRulz Community Forum** to go to an online forum that is dedicated to topics and discussions related to the e² studio (an Internet connection is required).

Under the Help Contents window, there are many useful topics, such as:

The Debugging Projects topic provides useful information such as debug configuration, supported number of breakpoints, etc.

It can be launched by clicking on the Help menu \rightarrow Help Contents \rightarrow e² studio User Guide \rightarrow Debugging Projects.

The **RA Contents** topic provides information about RA project creation using the RA Configuration Editor and FAQs.

It can be launched by clicking on the Help menu \rightarrow Help Contents \rightarrow RA Contents.



Revision History

		Description		
Rev.	Date	Page	Summary	
1.00	Sep.22.22	—	First release document	
			(Updated from "Renesas e ² studio 2021-04 or higher User's Manual: Quick Start Guide")	
1.01	Jul.16.24	_	Updated from "Renesas e ² studio 2022-07 or higher User's Manual: Quick Start Guide".	



Renesas e^2 studio 2023-10 or higher – Quick Start Guide

Publication Date: Jul.16.24

Published by: Renesas Electronics Corporation

Renesas e² studio 2023-10 or higher – Quick Start Guide

