User's Manual

e² studio

Integrated Development Environment

Additional Document for Quick Start Guide
Multicore Project

Renesas MCU
RA Family

Renesas Electronics
www.renesas.com

Rev.1.00 Feb 2026

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

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/.

Trademarks

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

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 reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Contents

1. Overview

This document describes how to generate, build, and debug projects for multicore devices in the RA Family of microcontrollers. Read “Renesas RA Family e² studio 2023-10 or Higher Quick Start Guide” before reading this document.

Note: The contents displayed on the screen may be slightly different from those in the figures of this document, depending on the terminal you are using and the versions of the e² studio and FSP.

1.1 Operating Environment

The elements of the operating environment used in confirming operation for this document are listed below.

IDE: e² studio 2025-10
FSP: FSP v6.2.0
Evaluation board: EK-RA8P1
Emulator: J-Link OB (mounted on the evaluation board), E2 emulator, or E2 emulator Lite

2. Generating a Multicore Project

This chapter describes how to generate a new RA project for a multicore device. The e² studio includes a wizard to help quickly generate a new project. The wizard is capable of generating a project that is suitable for the RA device and user board you are using.

The project generator for RA MCUs can set up the pin configuration, interrupts, clock configuration, and the necessary driver software.

As a prerequisite for project generation, FSP v6.0.0 or later versions and the toolchain must be installed on the host machine as described in Chapter 2, Installation, in "Renesas RA Family e² studio 2023-10 or Higher Quick Start Guide".

Note: FSP 6.2.0 and earlier versions do not support the generation of multicore solutions that leverage Arm® TrustZone®. Only flat multicore solutions can be generated.

2.1 Generating a Solution Project

This section describes how to generate a solution project for use in the easy creation of projects for a multicore device. Invoking the project wizard once will generate two projects, one for CPU0 and one for CPU1, to be handled as multicore projects and a solution project as a container to bundle them together.

Note: FSP v6.2.0 and earlier versions do not support the operation of a multicore device that supports Arm® TrustZone®.

Invoke the e² studio and choose a workspace folder. To generate a new RA project, perform the following steps.

1. Click on [File] → [New] → [Renesas C/C++ Project] → [Renesas RA].

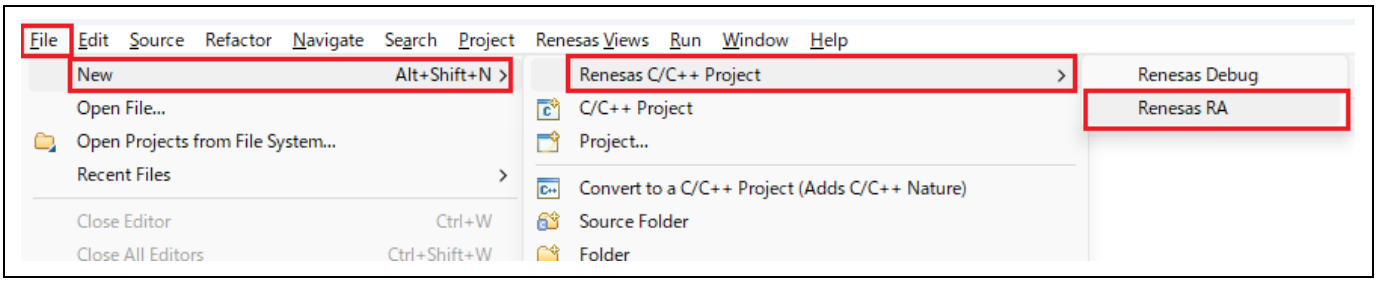

Figure 2-1 Generating a New Project

2. Select the “Renesas RA FSP Solution” template. Click on [Next] to proceed.

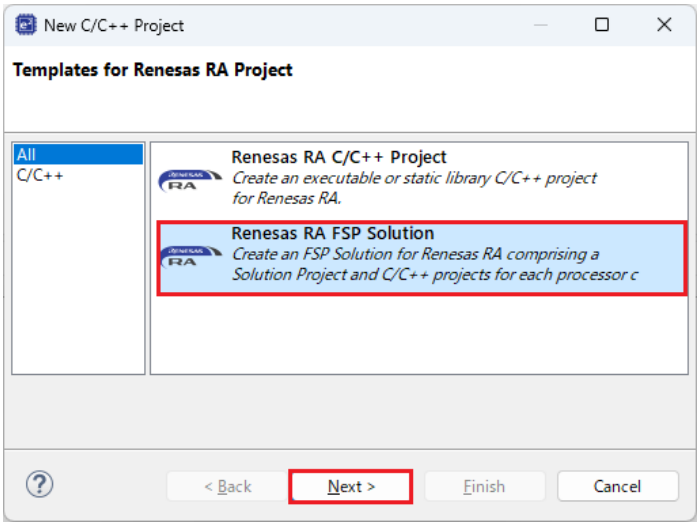

Figure 2-2 Selecting an RA Solution Project

3. In the solution project generation wizard, enter the following project information.
   Project name: Enter a project name (example: RA_Multicore).
   Use default location: Ensure this is ticked to create the solution project inside the current workspace. If you want to create a project at a different location, enter a new location below it instead of ticking this checkbox.
   Click on [Next] to proceed.

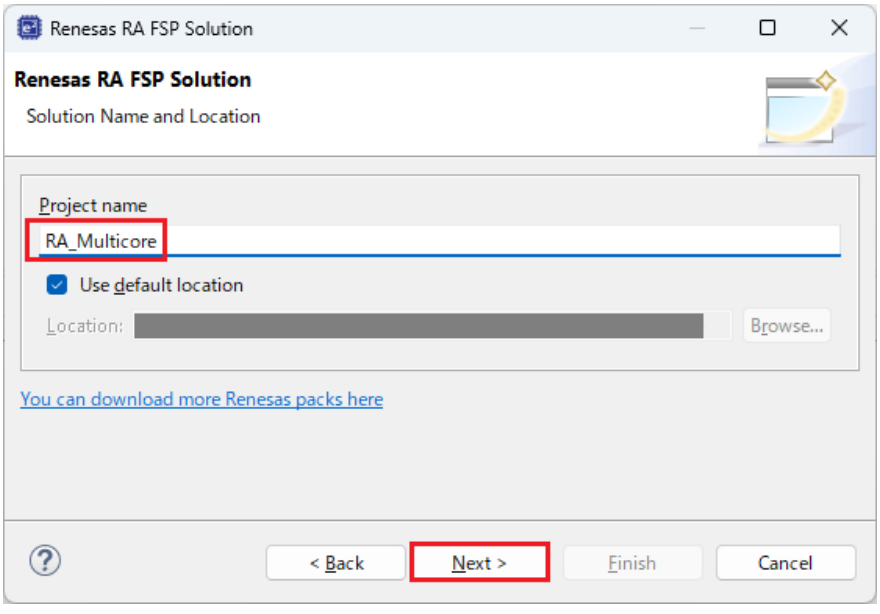

Figure 2-3 New RA Project Generation Wizard

4. In the window for selecting a device, enter information on the device, tool, and debugger you are using.
   Board: EK-RA8P1
   Solution Template Selection: Select Multicore > Flat > Bare Metal > Blinky.
   IDE Project Type: e² studio managed build
   Toolchains: Select the latest version of the LLVM Embedded Toolchain for Arm that is recognized for use with devices of the Renesas RA Family (example: 18.1.3).
   Debugger: J-Link ARM
   Keep all other fields at the default settings.

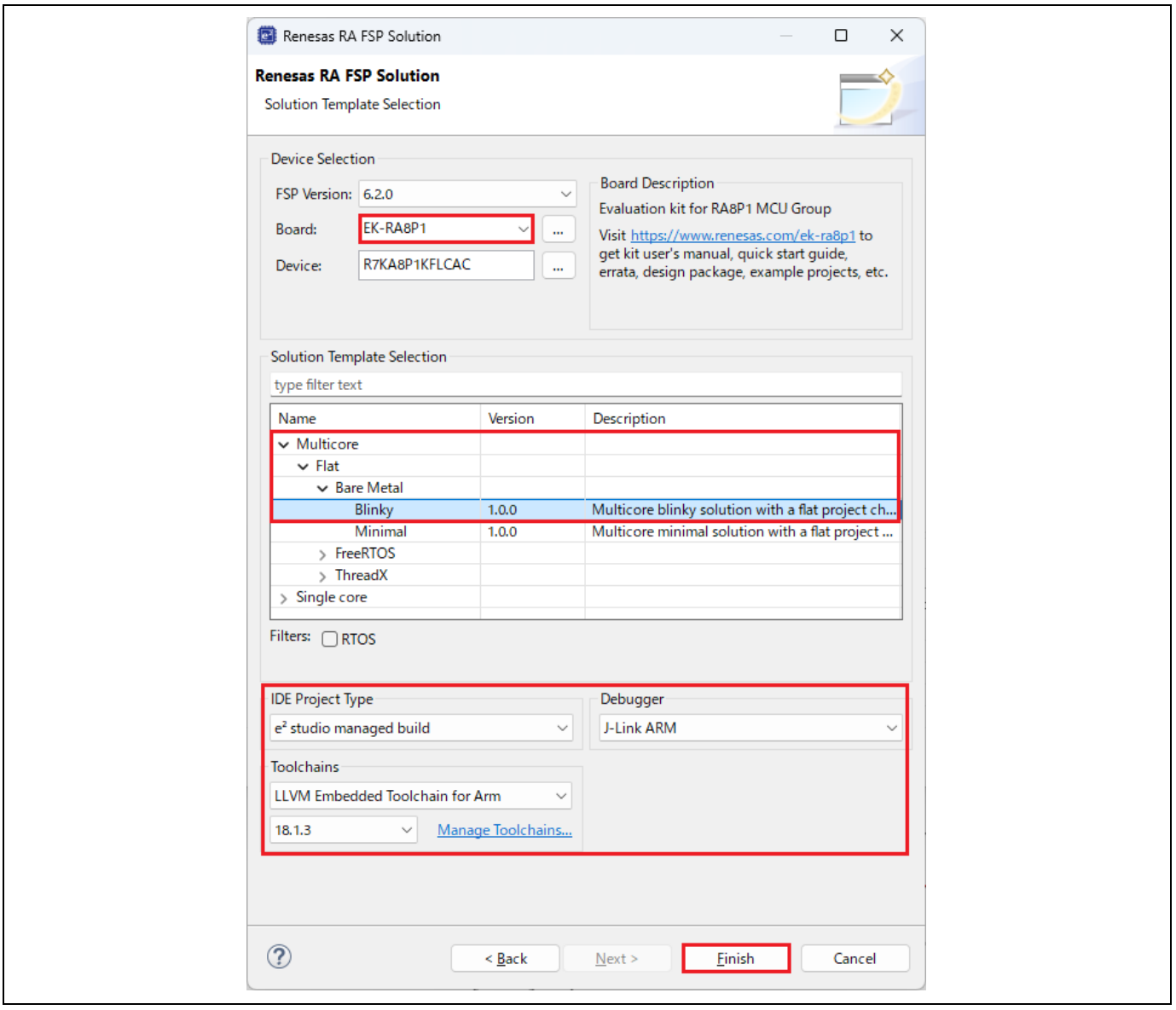

Figure 2-4 Selecting the Device and Other Fields

5. Clicking on the [Finish] button automatically creates and then builds the individual projects for each of the CPUs, followed by input of the configuration data for saving in the solution project.

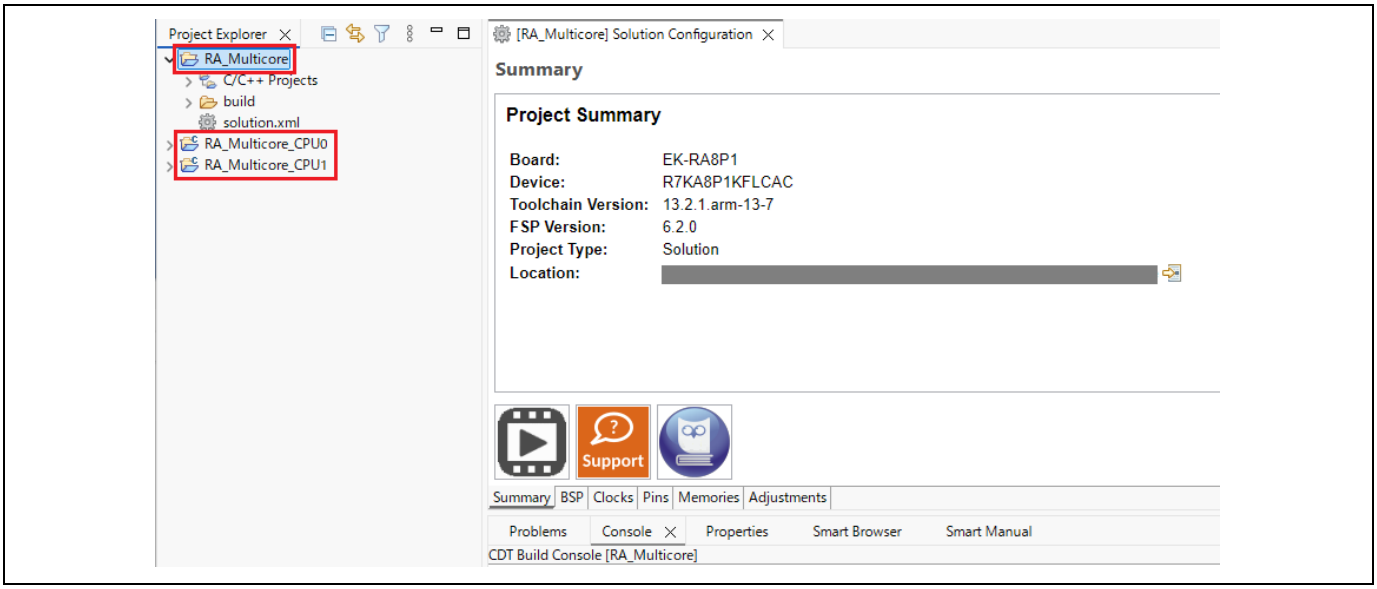

Figure 2-5 View of New Projects

The solution project stores and controls the overall configuration of the CPU projects. Its role is referred to as that of a “container”.

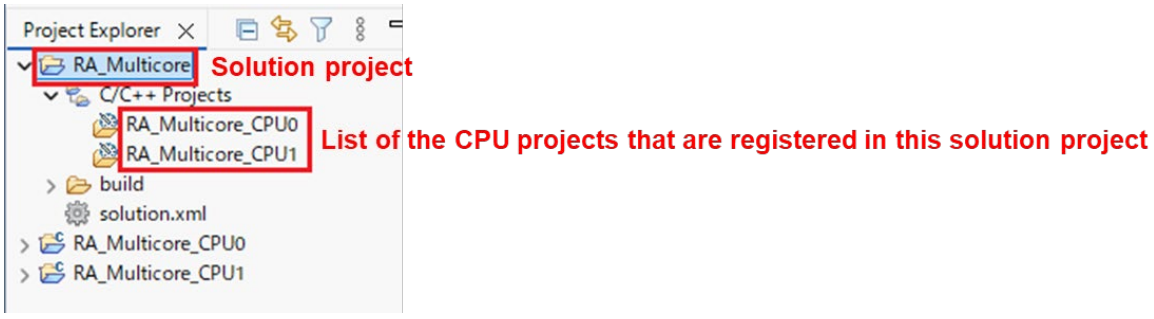

Figure 2-6 Configuration of a Solution Project

The two “CPU” projects are for each core and are independent of each other. The entire multicore application for downloading to the target MCU is generated by combining these projects.

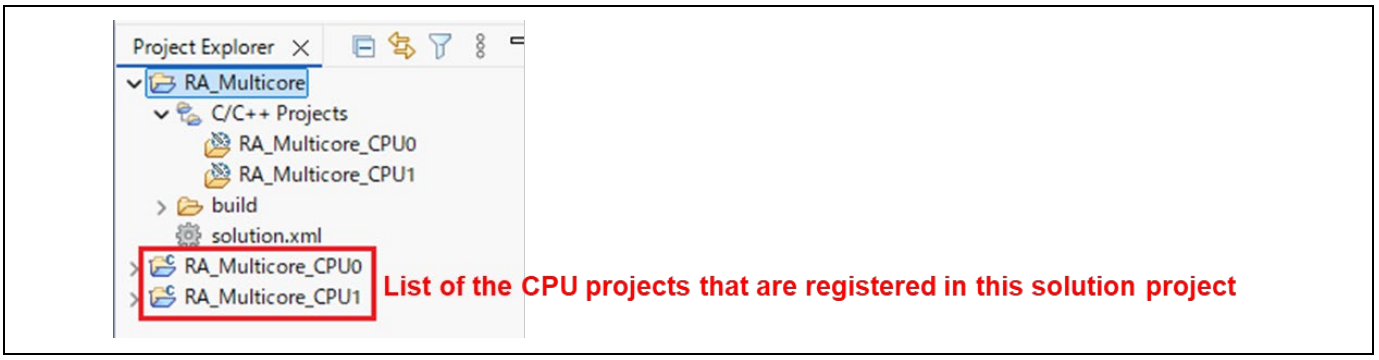

Figure 2-7 CPU Projects in the Solution Project

2.2 RA Solution Editor

The [RA Solution Editor] view shows the contents displayed when the solution project, “solution.xml”, is opened.

The configurations are saved in the “solution.xml” file. The project configuration settings are grouped on multiple pages that allow you to set several configurable aspects of the project, such as how pins and clocks are set up and the memory areas.

To edit the project configuration, make sure of the following.

- Confirm that the [FSP Configuration] perspective is selected in the upper right-hand corner of the e² studio window. If it is not selected, click on the Windows menu → [Perspective] → [Open Perspective] → [Others…] → [FSP Configuration].
- Confirm that the “solution.xml” file is open.

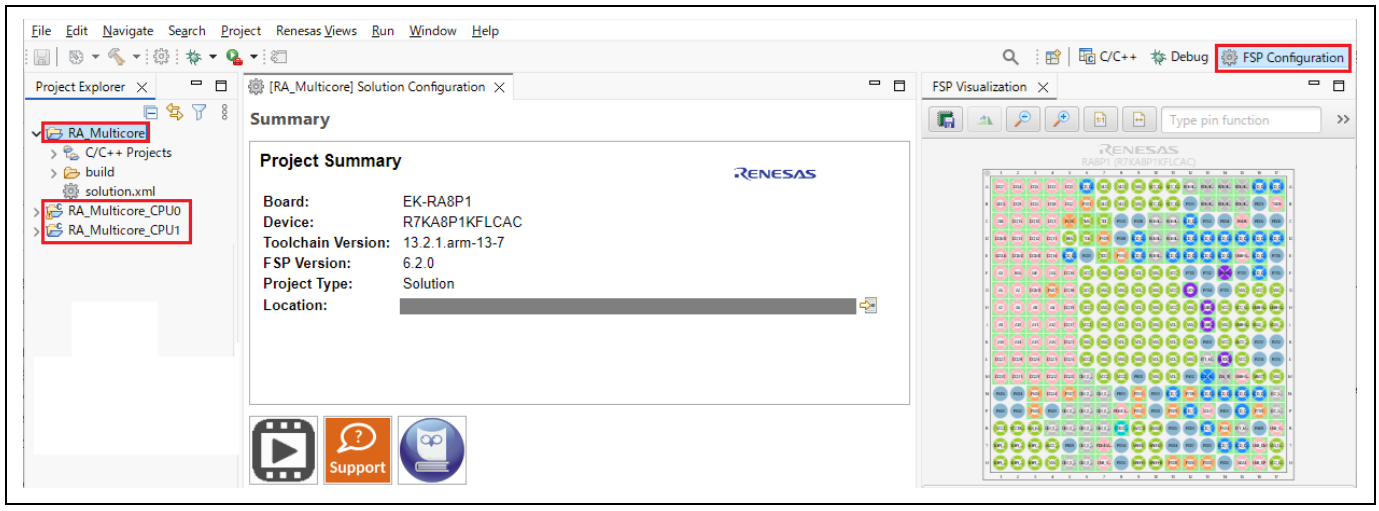

Figure 2-8 Solution Editor Window

2.2.1 [Summary], [BSP], [Clocks], and [Pins] Pages

For the [Summary], [BSP], [Clocks], and [Pins] pages, refer to the respective subsections in section 3.5, RA Project Configuration Editor, in “Renesas RA Family e² studio 2023-10 or Higher Quick Start Guide”.

Note: There is currently an issue with the solution project behavior when passing configuration information from the solution down the project chain. See below for details.

RA Flexible Software Package Documentation: e2 Studio 2025-12 Pin Configuration Workarounds

2.2.2 [Memories] Page

The [Memories] page will display detailed information about memory resources and memory partitions, such as name, start address, size, core, and security.

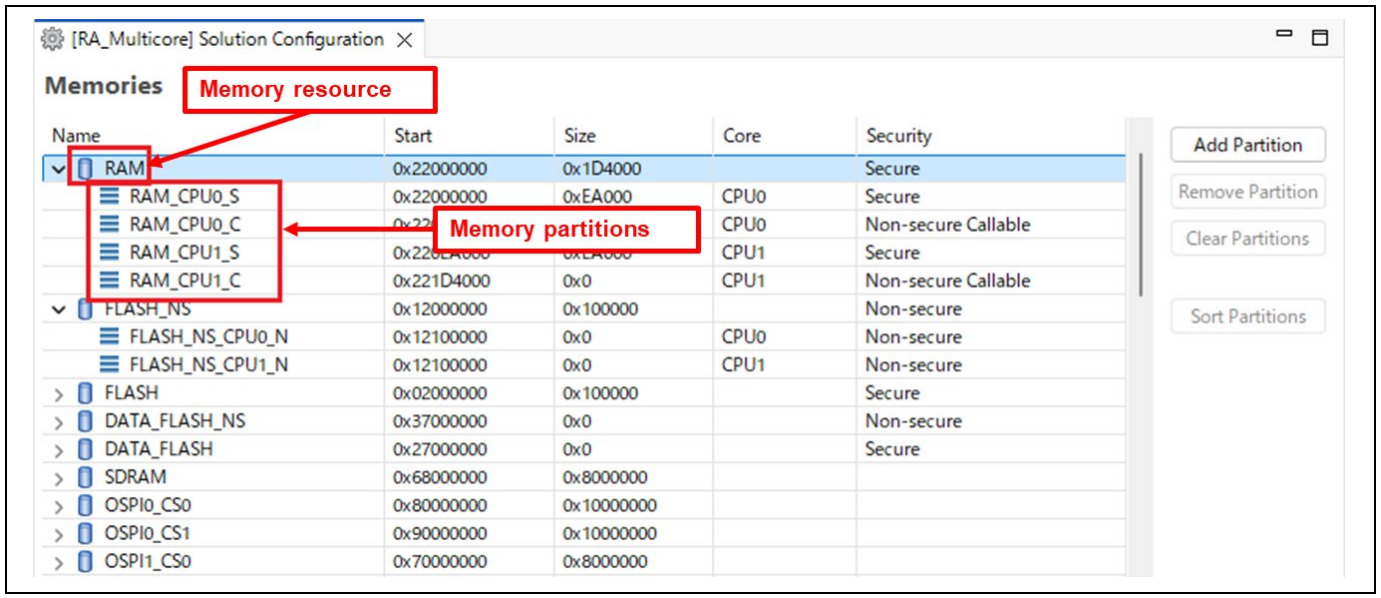

Figure 2-9 [Memories] Page

The address and size of a memory partition can be changed by clicking on the name of the memory partition.

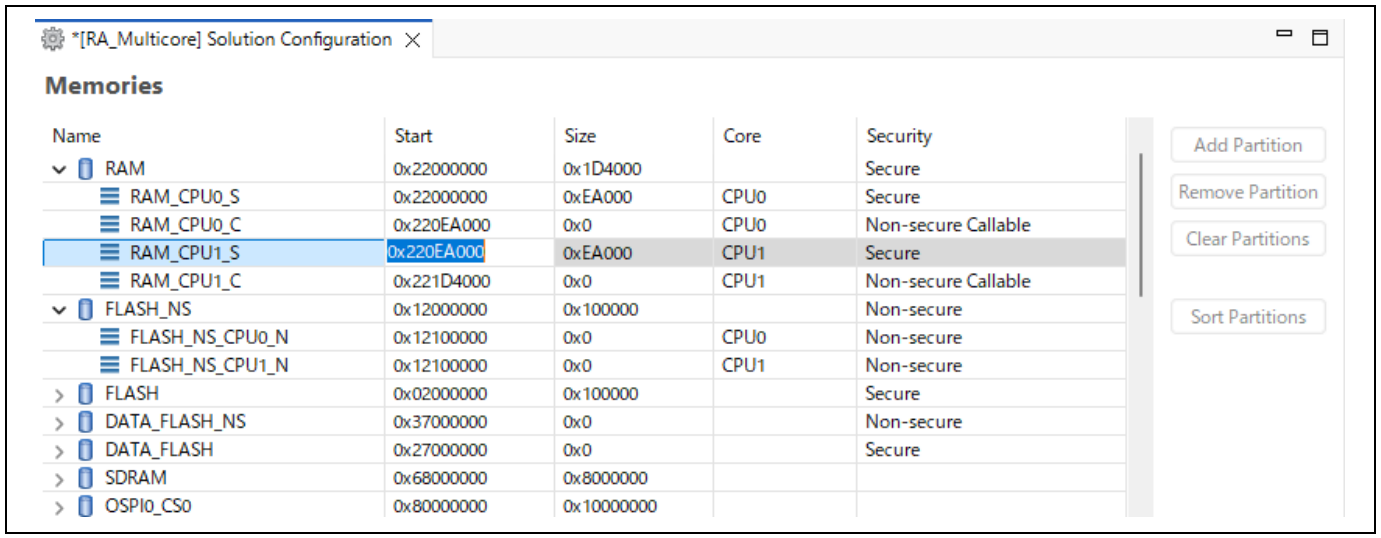

Figure 2-10 Selecting a Memory Partition

Note: The address and size of memory resources cannot be changed.

Adding, removing, and clearing are the available functions for editing the user-defined memory partitions. The buttons on the right side of the page are used to perform these functions. Each of the functions is described below and on the following pages.

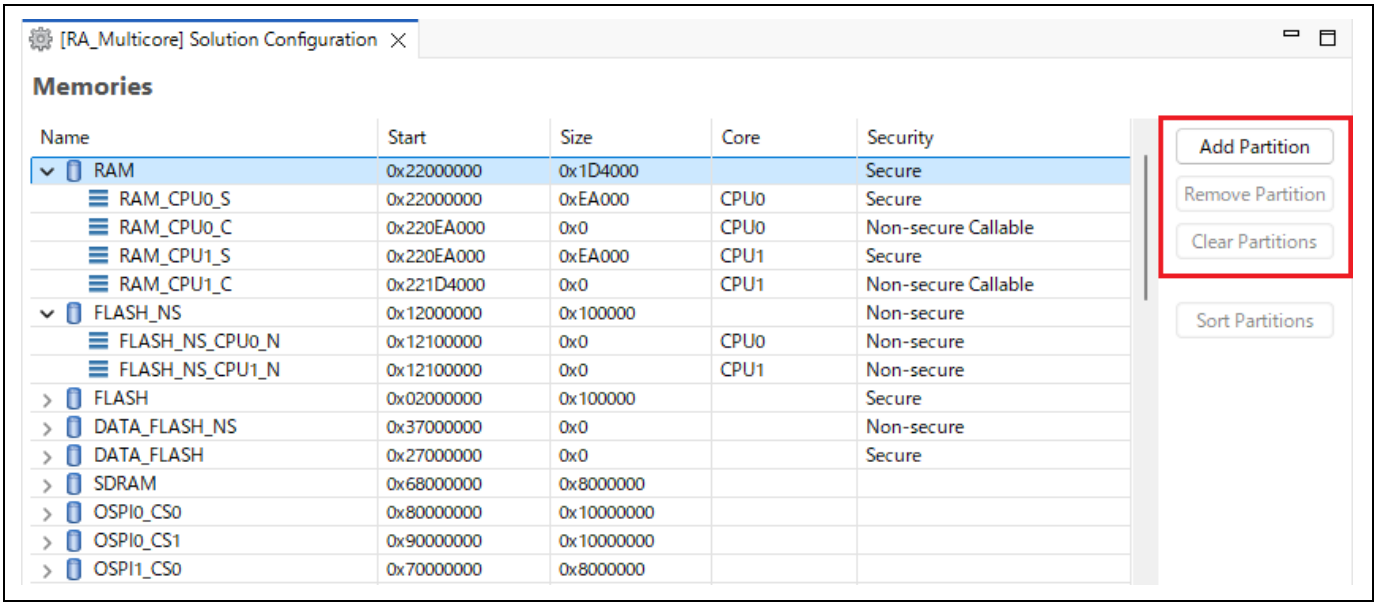

Figure 2-11 Buttons for Use in Editing

[Add Partition] Button:

This button is used to create a new user-defined memory partition in the selected memory resource. This button is only enabled when a memory resource is selected.

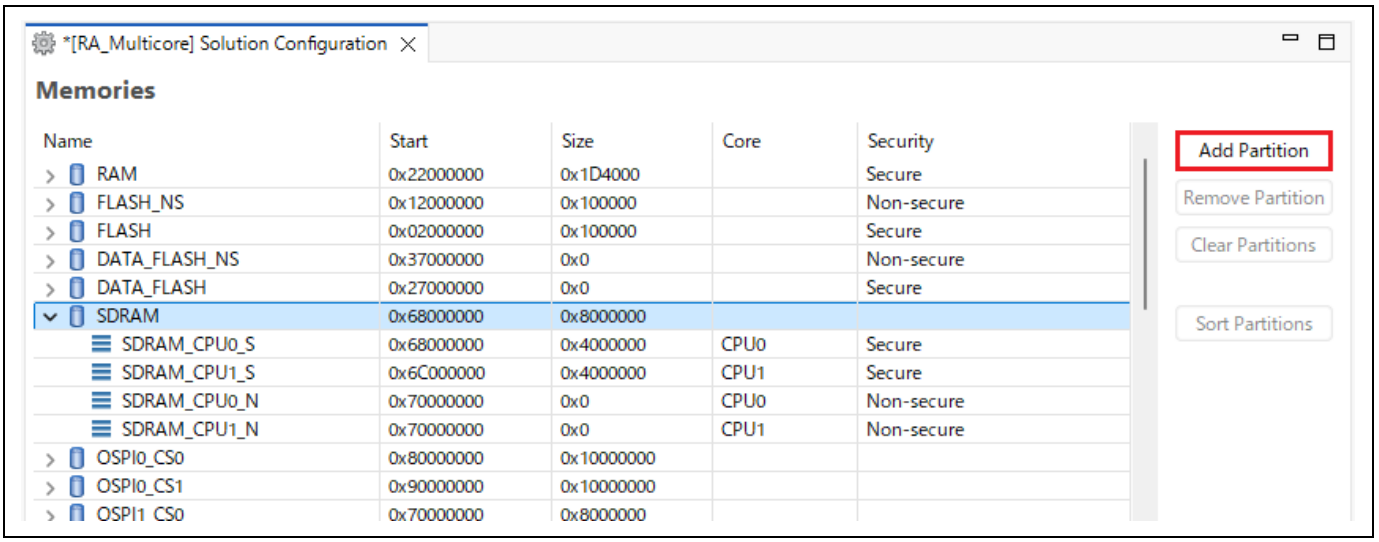

Figure 2-12 [Add Partition] Button

Clicking on the [Add Partition] button opens the [New Partition] dialog box.

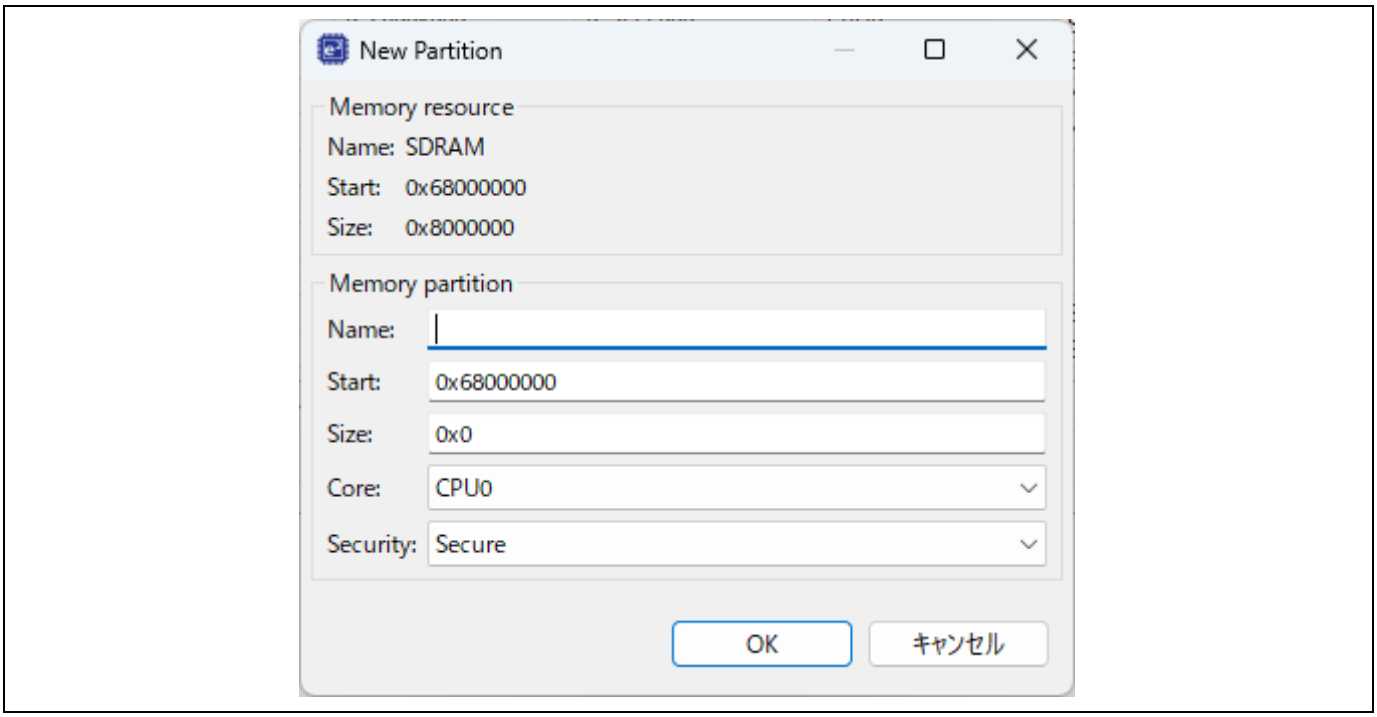

Figure 2-13 Setting a New Memory Partition

This dialog box is used to create a new memory partition in the specified memory resource. Set the partition name, start address, size, core, and security. The name, start address, and size of the selected memory resource are also displayed in the dialog box for reference. The default settings are the address in [Start] of the parent memory resource as the address where the partition starts, “0x0” as the size of the partition, the name of the first applicable core as the core, and the first applicable setting for the security. If the memory resource is core-specific, only the name of that core will be available in the [Core] combo box. Otherwise, all cores of the device are displayed, and the first core is selected by default. If the device does not have TrustZone, the [Security] combo box will only contain “Undefined”. If the device has TrustZone and the memory resource is security-specific, the [Security] combo box will only contain the given security setting. Otherwise, it will contain “Secure” or “Non-secure”, with “Secure” being selected by default. If the partition name is invalid, the start address or size of the partition is not a valid number, or the range of the partition extends beyond the boundaries of the resource at the time of clicking on the [OK] button, an error dialog box with a corresponding error message will be opened.

[Remove Partition] Button:

This button is used to remove a selected user-defined memory partition from its memory resource. This button is only enabled when a user-defined memory partition is selected. Clicking on this button opens a confirmation dialog box for the user to confirm the removal.

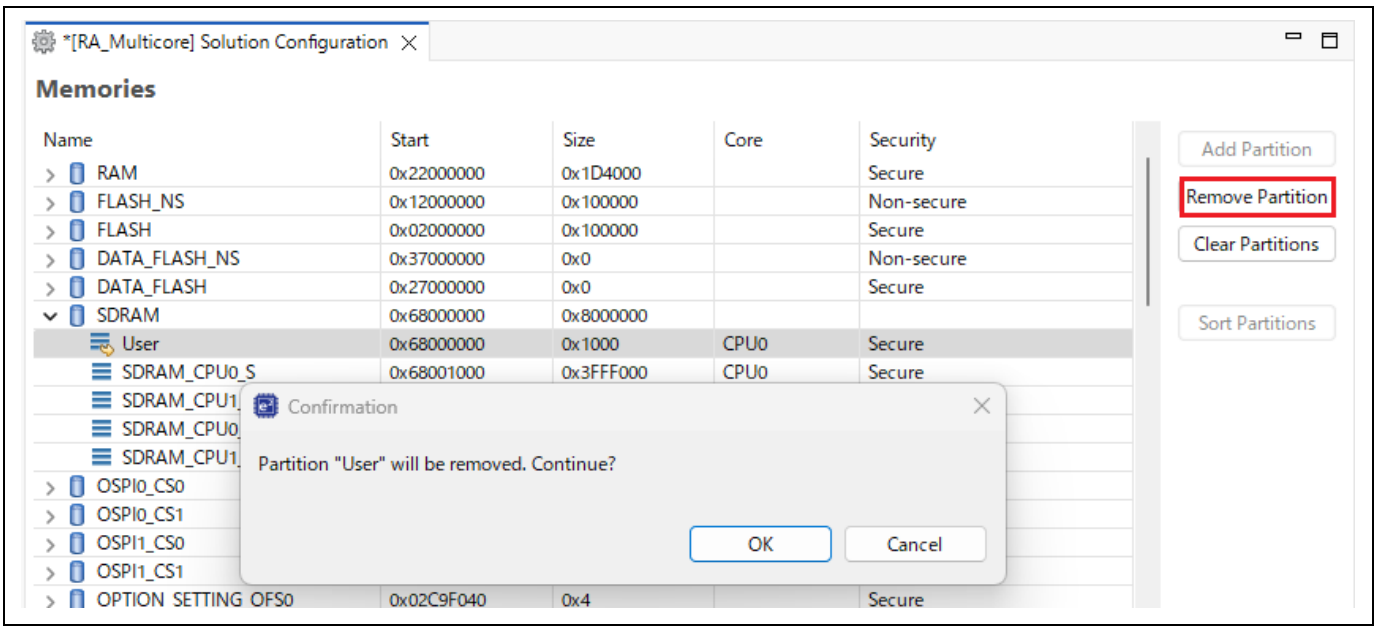

Figure 2-14 [Remove Partition] Button

[Clear Partitions] Button:

This button is used to remove all user-defined memory partitions from all memory resources. This button is enabled whenever user-defined memory partitions exist. Clicking on this button opens a confirmation dialog box for the user to confirm the removal.

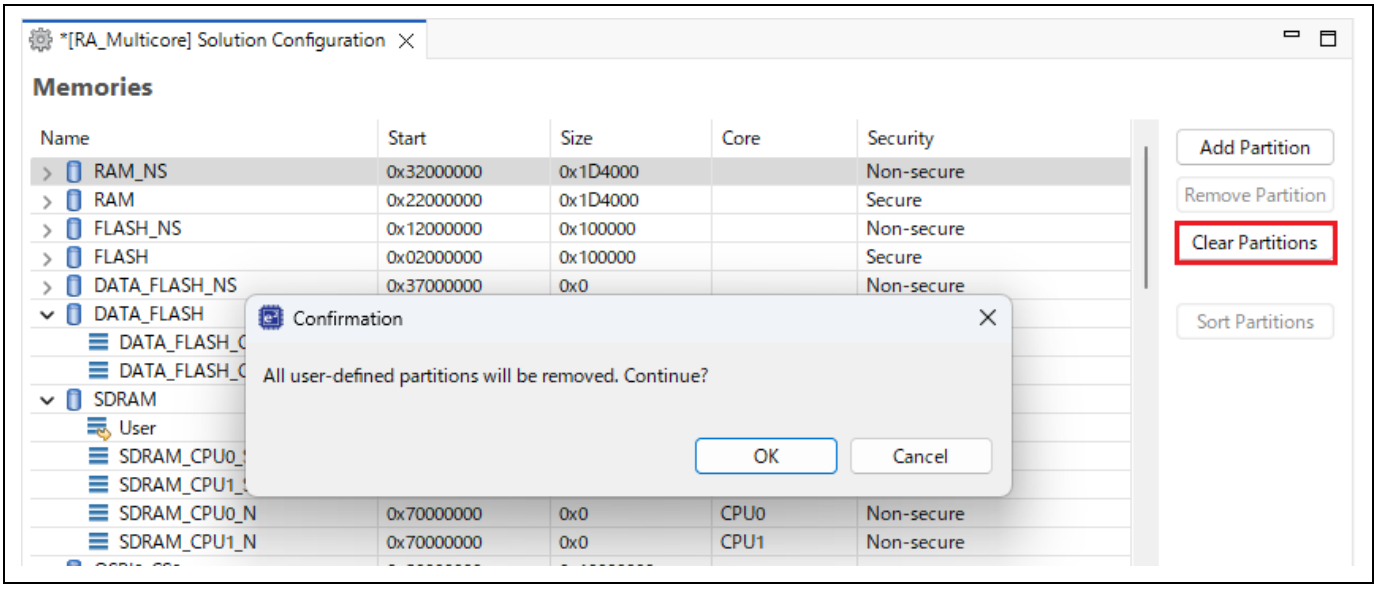

Figure 2-15 [Clear Partitions] Button

2.2.3 [Adjustments] Page

This function is for future support of TrustZone projects. It displays nothing with FSP v6.2.0.

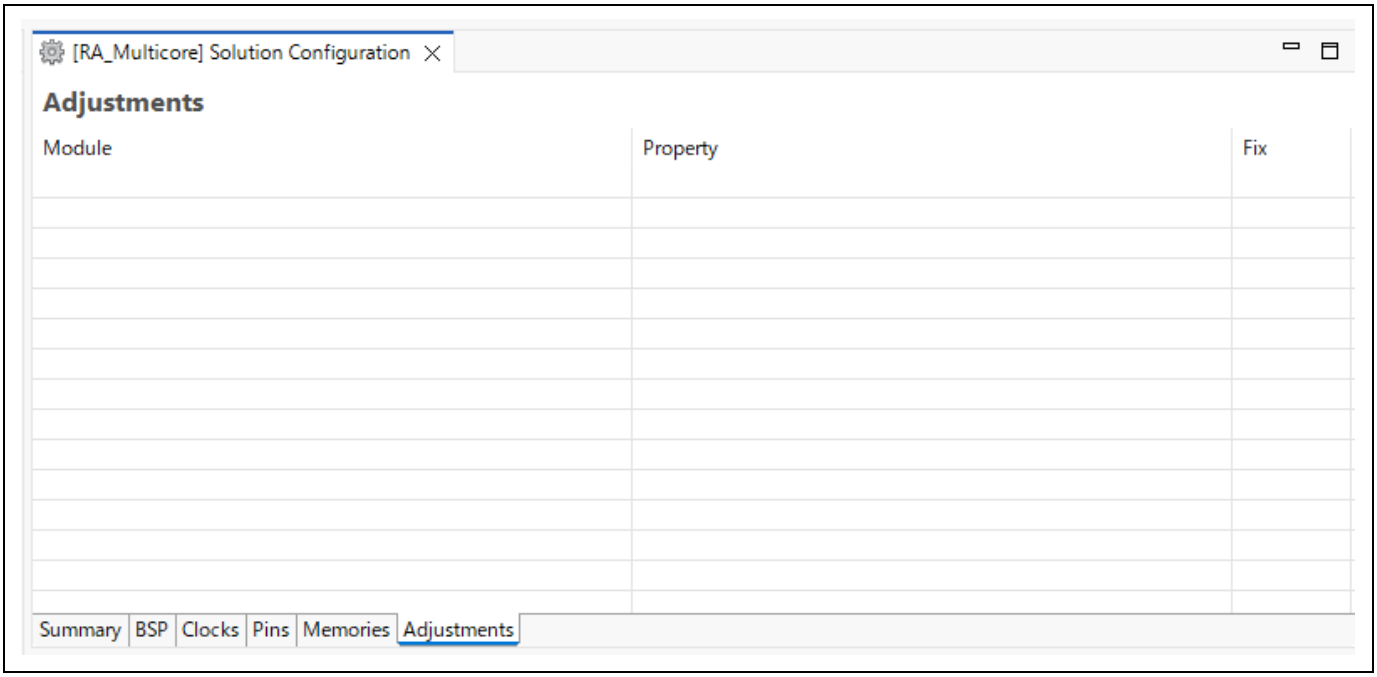

Figure 2-16 [Adjustments] Page

3. Building a Solution Project

Follow the steps below to build a solution project. The CPU projects associated with the solution project will be built at the same time.

1. In the [Project Explorer] view, click on the name of the solution project to bring it into focus.
2. Click on [Project] → [Build Project] to build the solution project. All of the associated CPU projects will also be built.

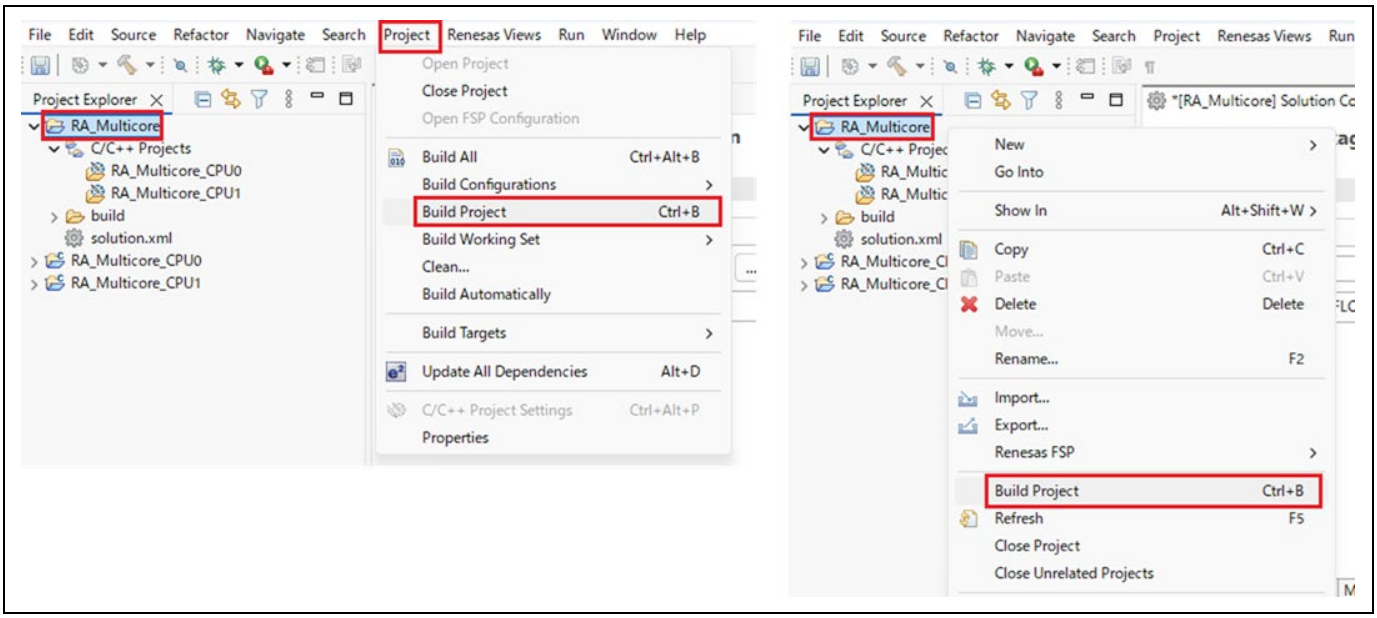

Figure 3-1 Build – Building a Solution Project

3. Confirm that there are no errors in any CPU project after building has finished.

To confirm the result of building each CPU project, select the name of each CPU project in the [Project Explorer] view and confirm the results of building in the [Console] view.

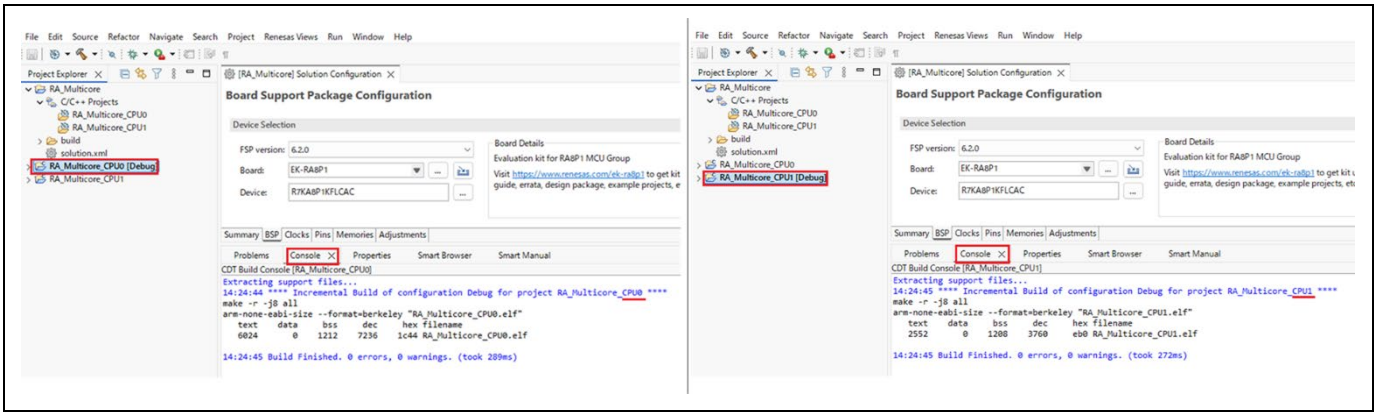

Figure 3-2 Build – Confirming the Results of Building the CPU Projects

4. Debugging

This chapter describes the configurations for the debugging of multicore projects with the E2 emulator or E2 emulator Lite (collectively referred to as E2/E2 Lite) and J-Link emulator (referred to as J-Link), and how to start up the debugger.

4.1 Debug Configurations and Starting the Debugger

The contents of the configurations for debugging a solution project differ between the E2/E2 Lite and J-Link. Since the CPU projects associated with the solution project are closely linked to the solution project, be careful when changing the debug configurations.

4.1.1 Common method for starting the debugger for J-Link and E2/E2 Lite

In e² studio 2025-10 and later, you can start the debugger by selecting the solution project in Project Explorer and clicking the debug icon on the toolbar.

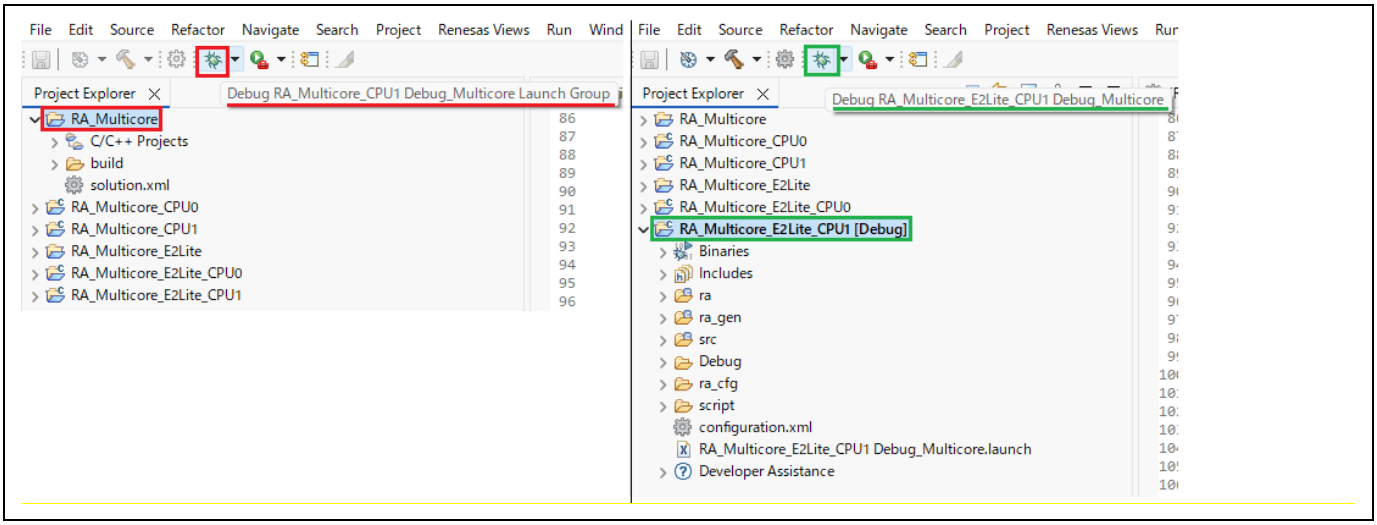

Figure 4-1 Debug – Common debugger startup method for J-Link and E2/E2 Lite

4.1.2 Debug Configurations of the J-Link and Starting the Debugger

For the J-Link, four debug configurations in total are created; three consisting of one each for the CPU0, CPU1, and a multicore project, and one for the launch group used to start the debugger. Start the debugger from the launch group.

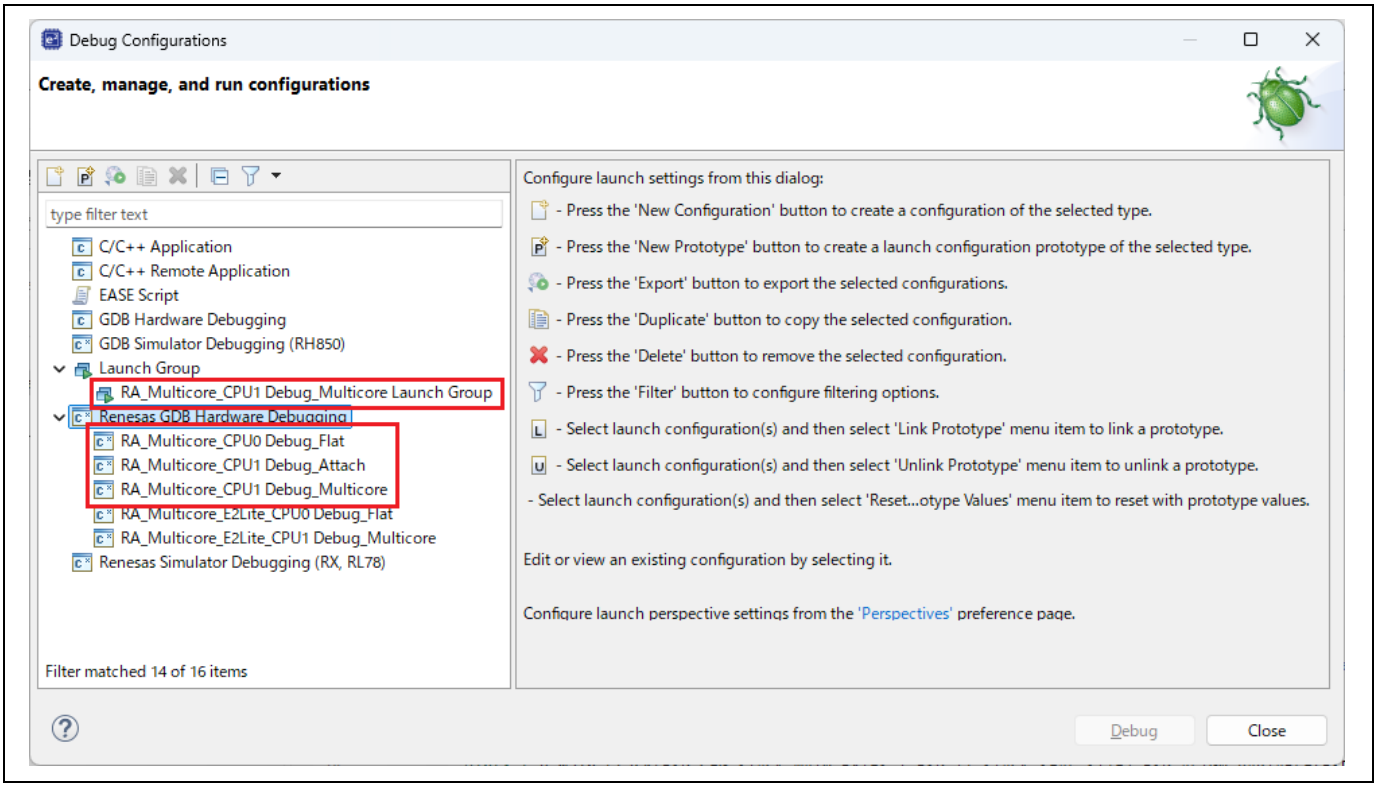

Figure 4-2 Debug – Debug Configurations of the J-Link

After confirming that a setting has been made to download the CPU0 and CPU1 object files, click on the [Debug] button. The debug launch configuration is executed, and the J-Link is connected to the RA board.

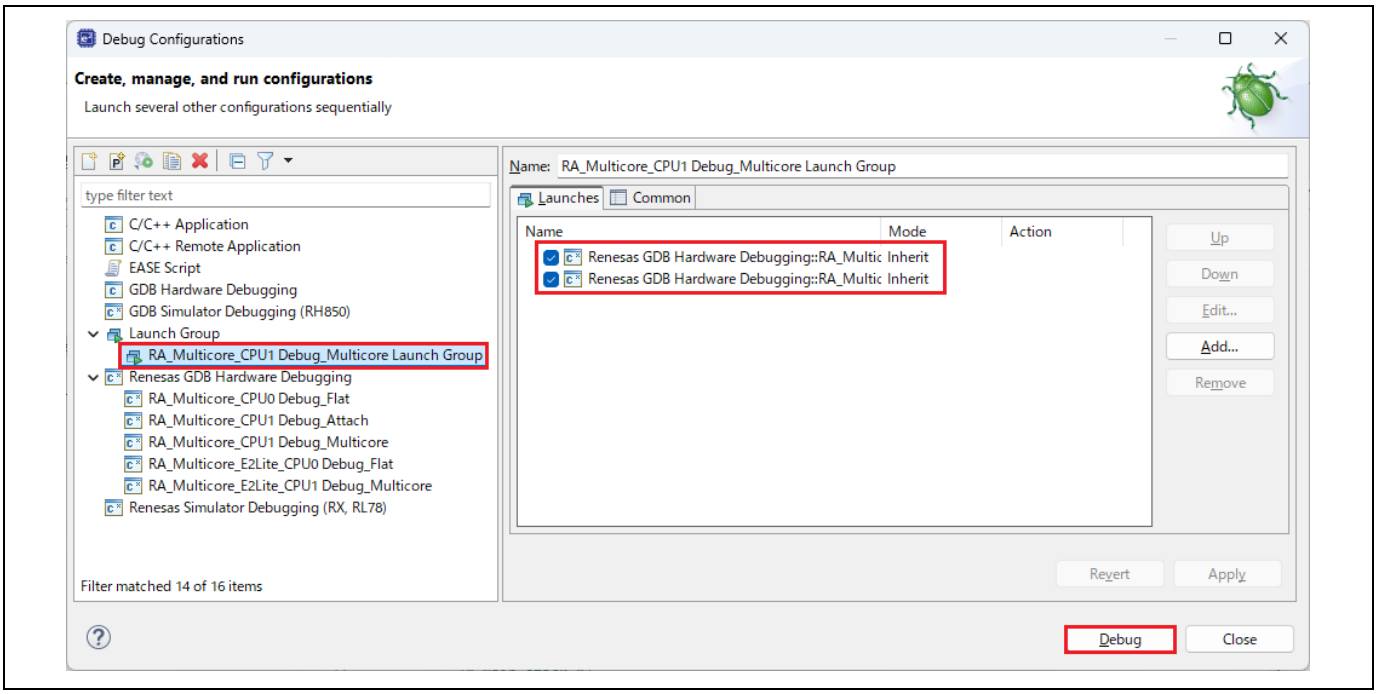

Figure 4-3 Debug – Settings for Downloading to the J-Link

When the connection is successful, the [Debug] view window, as shown in the figure below, is displayed. After connection, program execution is temporarily suspended at the entry point for CPU0 (example: Reset_Handler() in startup.c), and the code for CPU1 is kept in the state of waiting (indicated by “Running”) to be invoked from CPU0.

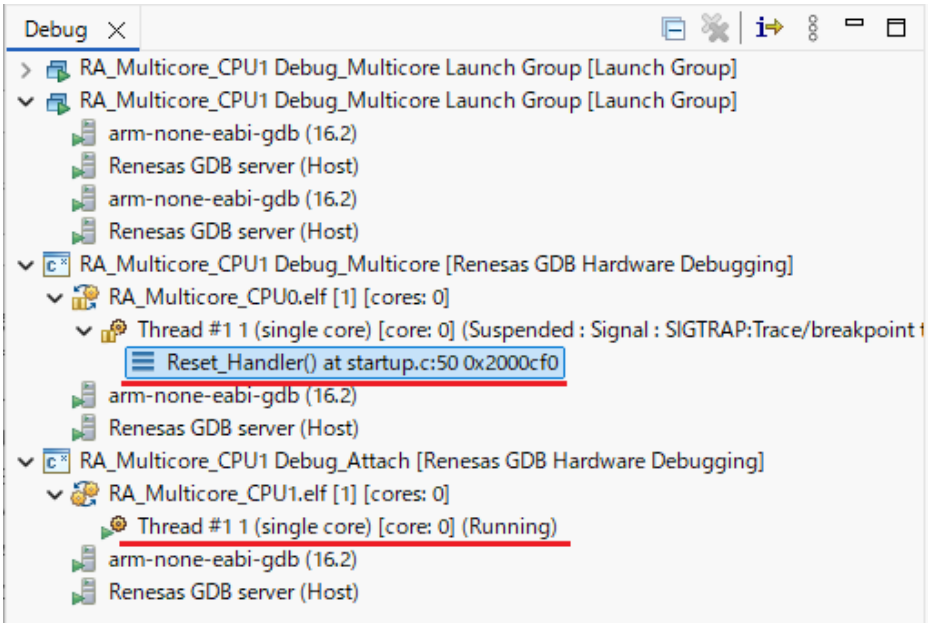

Figure 4-4 Debug – [Debug] View after Connecting the Target to the J-Link

4.1.3 Debug Configurations of the E2/E2 Lite and Starting the Debugger

For the E2/E2 Lite, two debug configurations are created: one for the CPU0 project and one for the multicore project. Start the debugger from the debug configuration that has _Multicore at the end of its name.

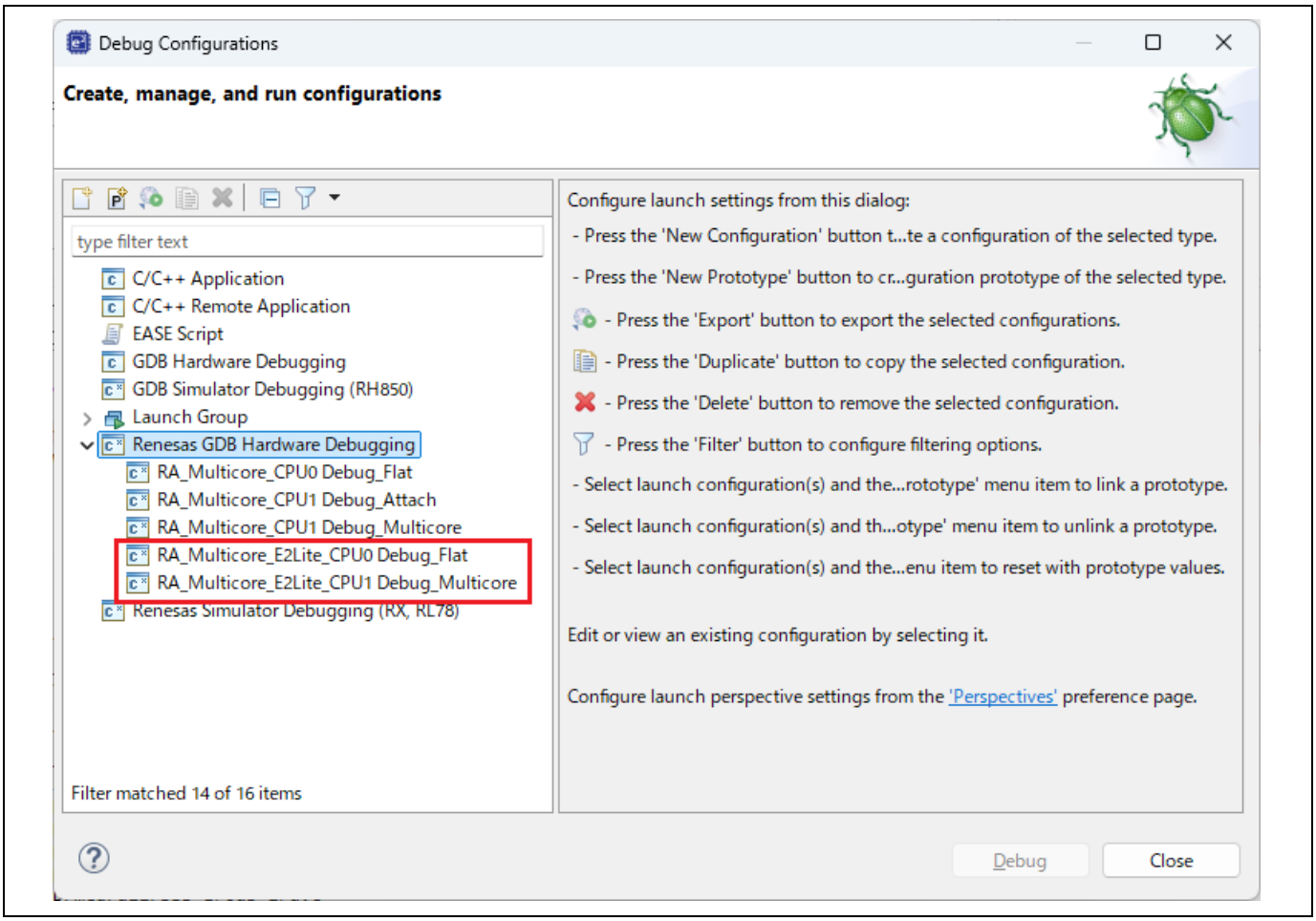

Figure 4-5 Debug – Debug Configurations of the E2/E2 Lite

After confirming that a setting has been made to download the CPU0 and CPU1 object files, click on the [Debug] button. The debug launch configuration is executed, and the E2/E2 Lite is connected to the RA board.

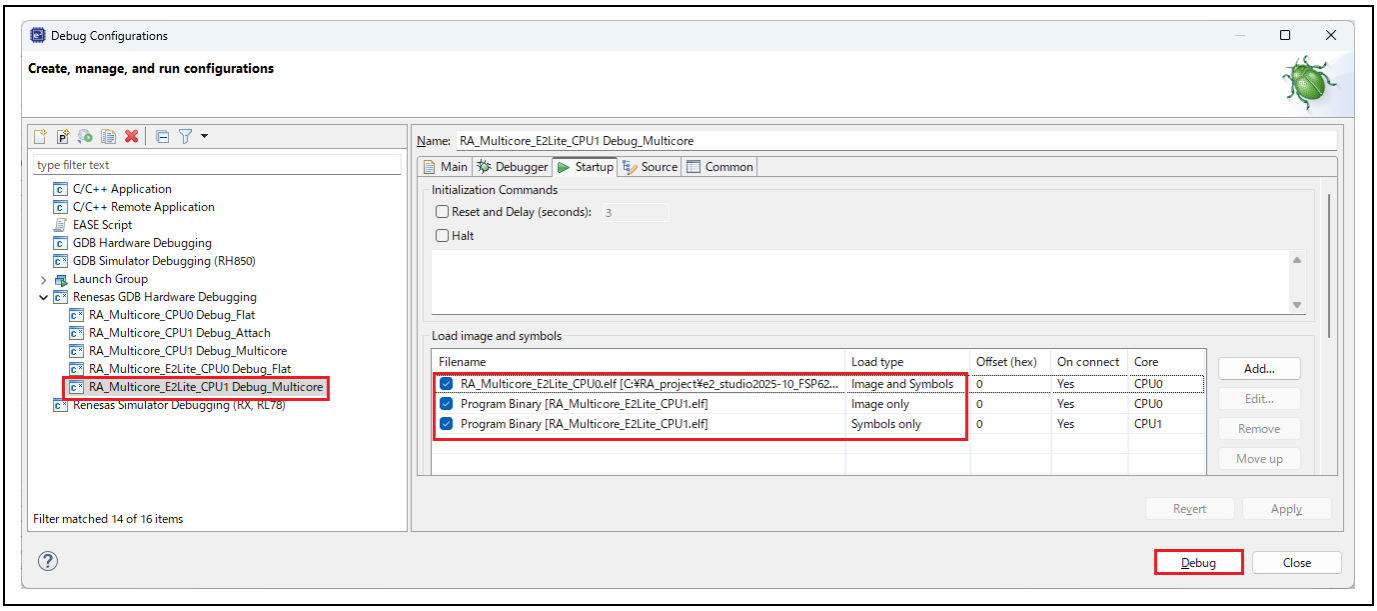

Figure 4-6 Debug – Settings for Downloading to the E2/E2 Lite

When the connection is successful, the [Debug] view window, as shown in the figure below, is displayed. After connection, program execution is temporarily suspended at the entry point for CPU0 (example: Reset_Handler() in startup.c), and the code for CPU1 is kept in the state of waiting (indicated by “Inactive state”) to be invoked from CPU0.

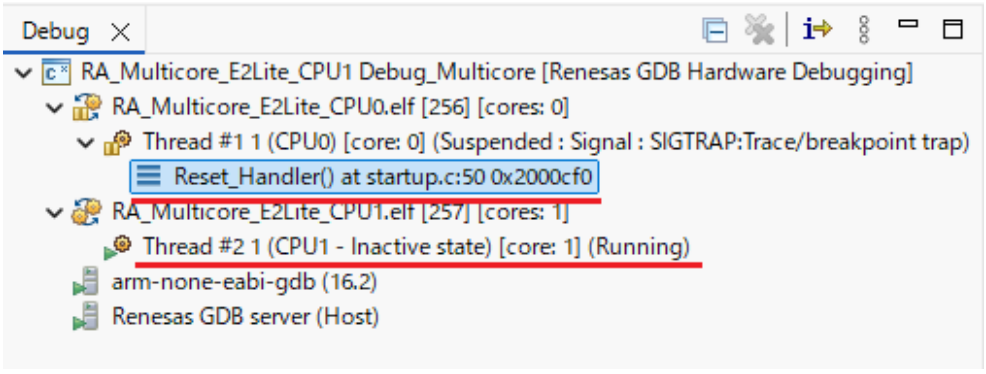

Figure 4-7 Debug – [Debug] View after Connecting the Target to the E2/E2 Lite

4.2 Functions Unique to Multicore Debugging

This section explains the functions that are unique to multicore debugging.

4.2.1 Toolbar

The toolbar dedicated to multicore debugging is explained.

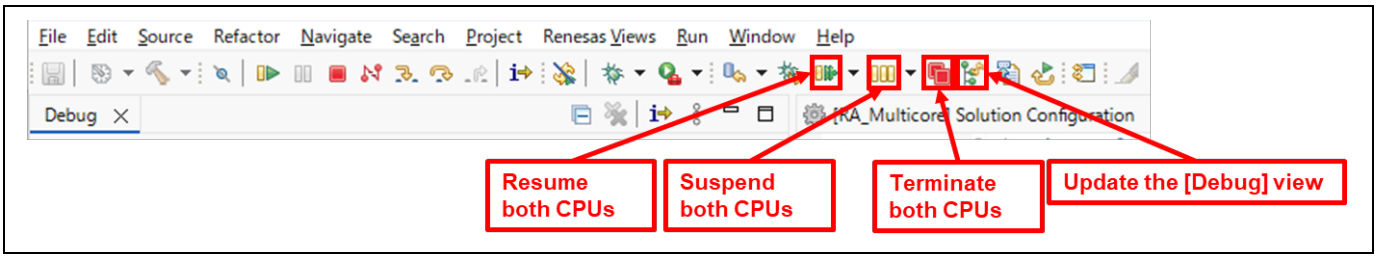

Figure 4-8 Debug – Debugging Functions Unique to Multicore Debugging

- The button is used to simultaneously start execution by both CPUs. (J-Link works with e² studio 2025-12 and later)
- The button is used to simultaneously suspend execution by both CPUs. (J-Link works with e² studio 2025-12 and later)
- The button is used to end the debug sessions of both CPUs.
- The button is used to update the [Debug] view.

4.2.2 Differences in Operation in Simultaneous Execution and on the Occurrence of Breaks

The differences in operation in simultaneous execution and on the occurrence of breaks at breakpoints in the E2/E2 Lite and J-Link are explained.

Simultaneous Execution	
E2/E2 Lite	There is hardly any time lag because simultaneous execution is on the hardware level.
J-Link	There is a very small time lag because simultaneous execution is on the software level, and execution by the CPUs is started/paused in sequence. However, this lag is generally not noticeable in real use.

A Break Occurring at a Breakpoint after the Start of Simultaneous Execution	
E2/E2 Lite	When program execution stops at a breakpoint for either of the CPUs, a break will occur for both CPUs.
J-Link	A break will only occur for the CPU with code that has the breakpoint.

4.2.3 View Displays for Each of the CPUs

The views of registers, memories, and breakpoints can be displayed independently for each of the selected CPUs. This section describes the procedure, taking the [Registers] view as an example.

When the [Registers] view is opened, the registers of the CPU that is selected in the [Debug] view are displayed.

Note: When using J-Link, pausing two cores at the same time will slow down the pausing of the second core. This issue will be fixed in e² studio 2026-04.

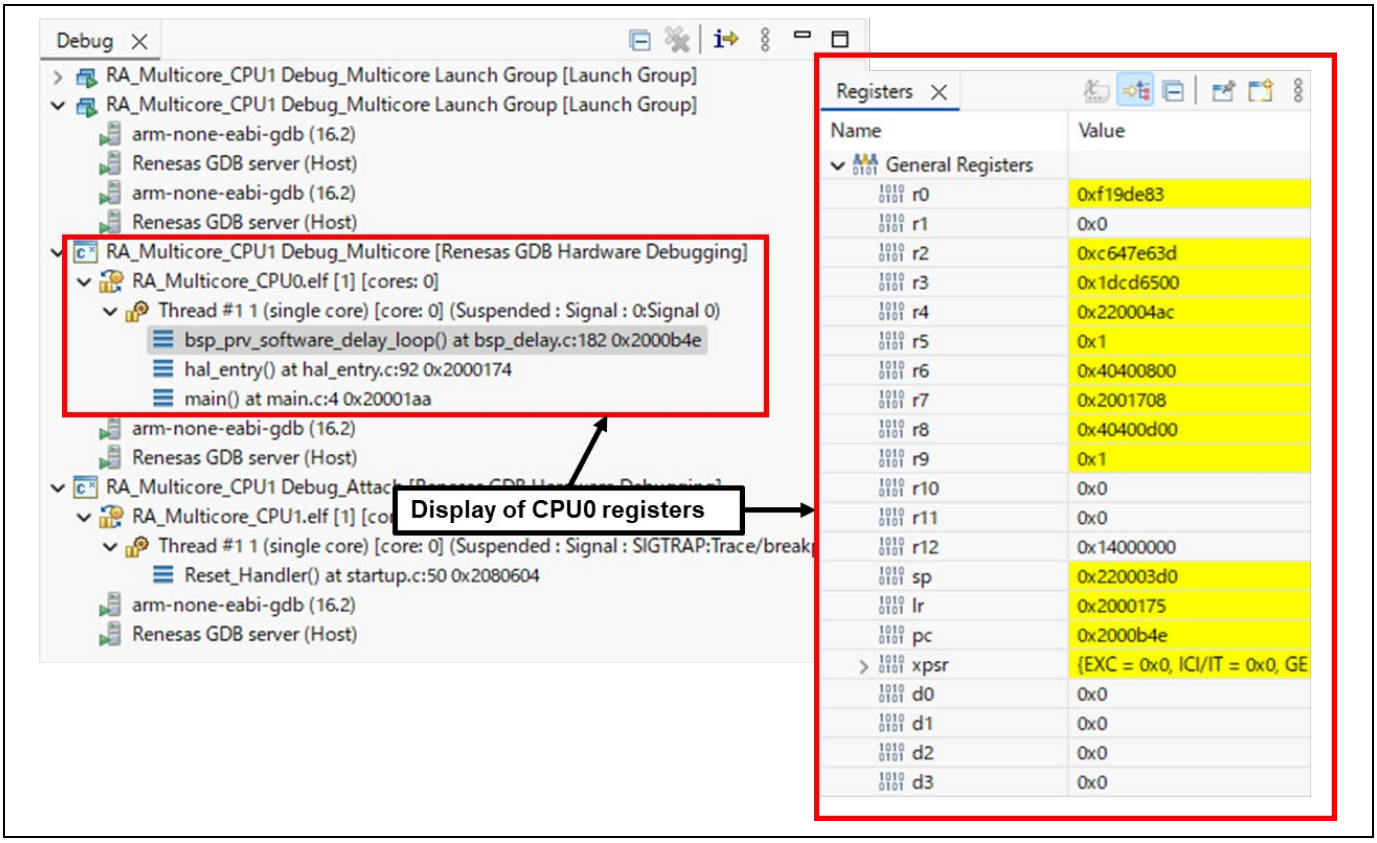

Figure 4-9 Debug – Initial Display of the [Registers] View

Next, click on [Open New View] in the [Registers] view.

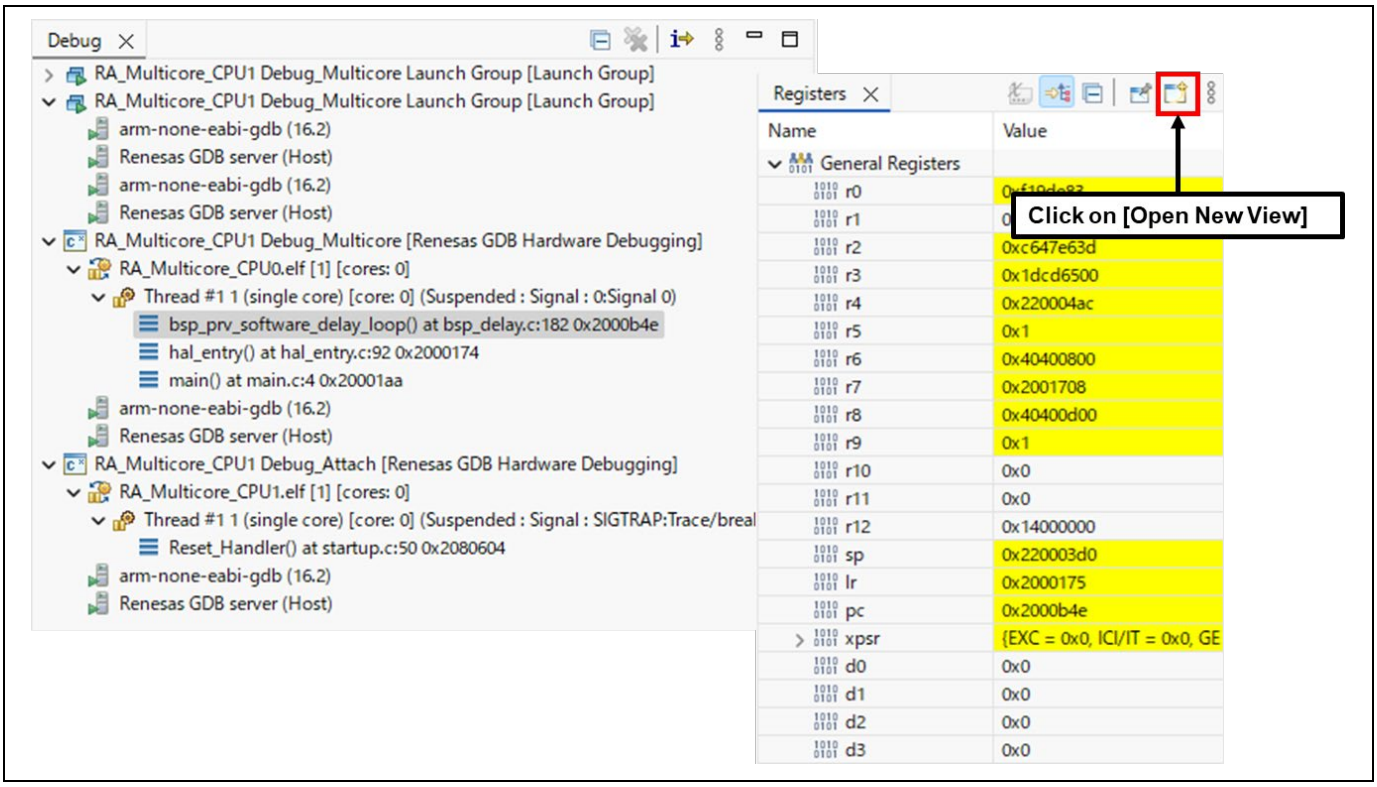

Figure 4-10 Debug – New View of the [Registers] View

A new [Registers] view showing the registers of CPU0 is opened.

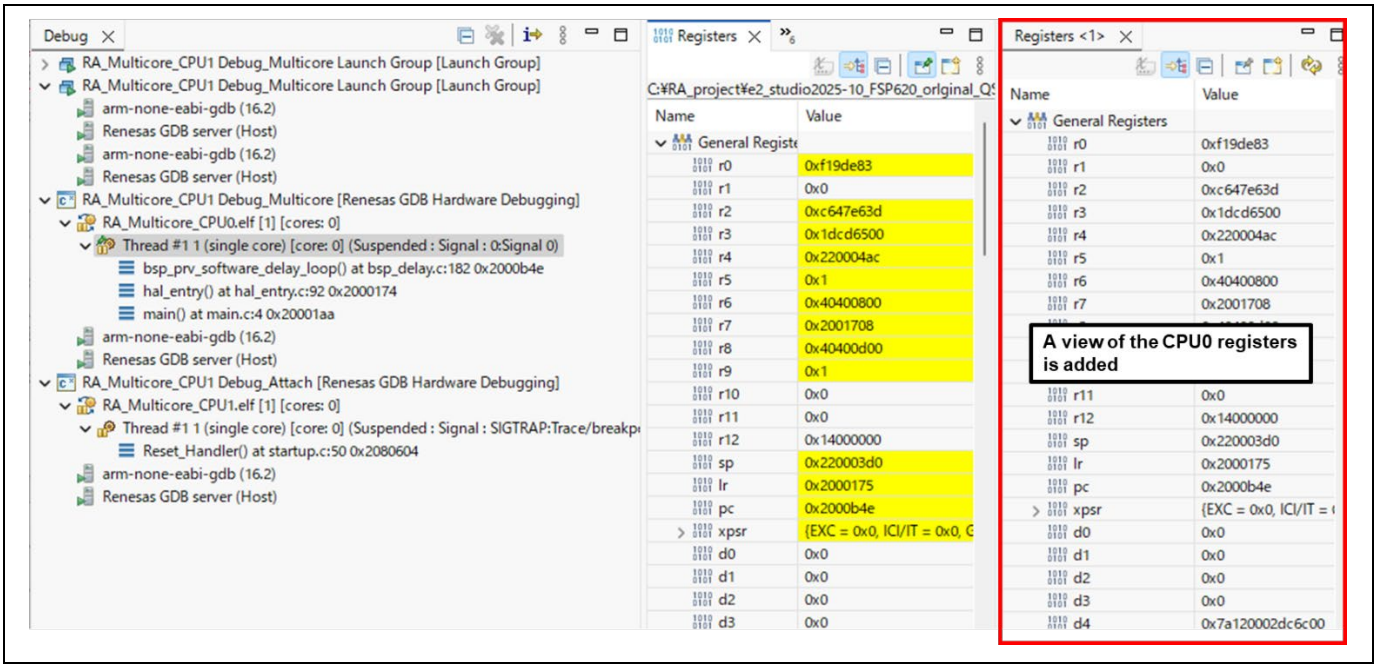

Figure 4-11 Debug – New [Registers] View

Click on [Pin to Debug Context] in the [Registers] view while CPU0 is selected in the [Debug] view.

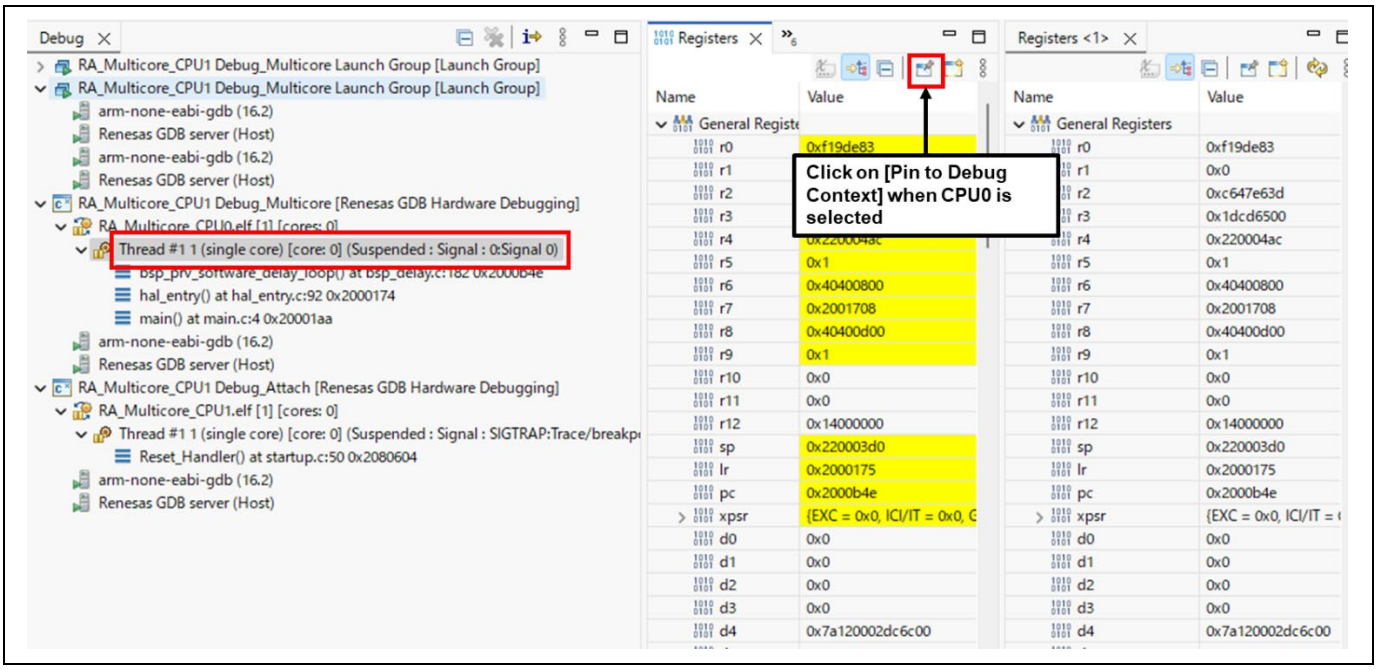

Figure 4-12 Debug – Pinning to CPU0

The pin on the [Pin to Debug Context] icon in the [Registers] view will become the same color (green in the example) as the pin on the CPU0 icon in the [Debug] view.

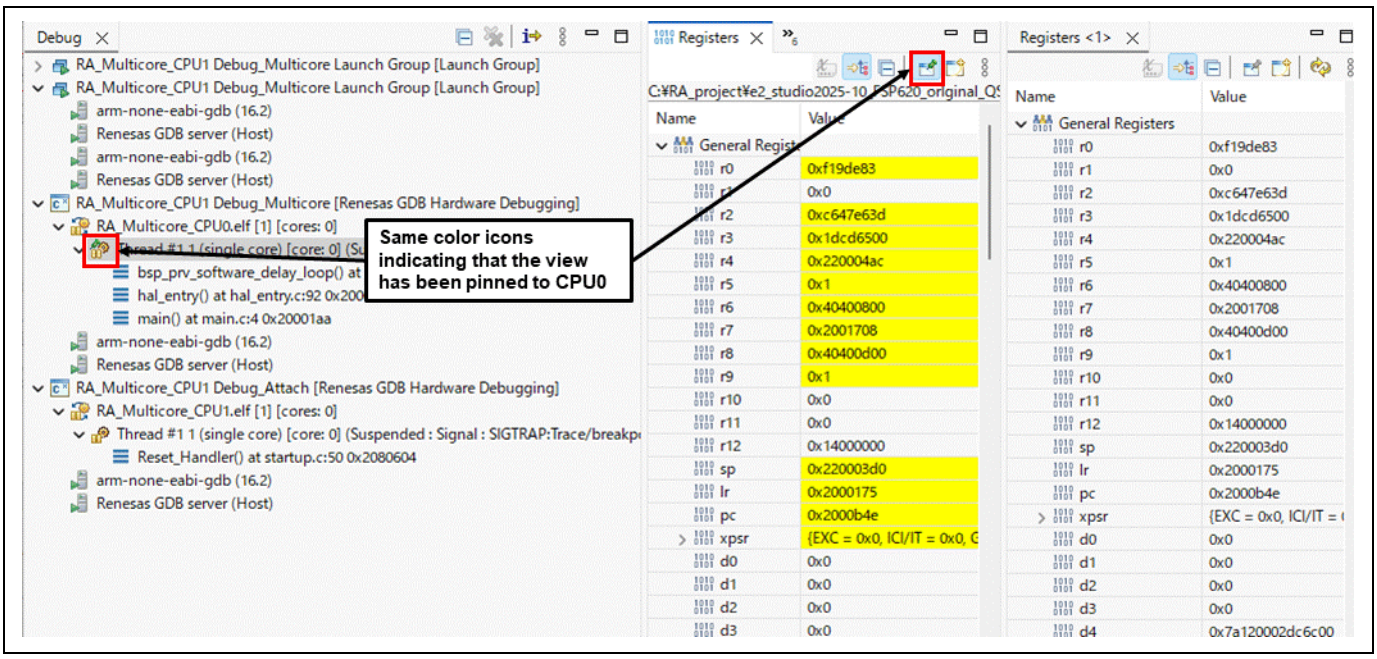

Figure 4-13 Debug – Icons Indicating That the View Has Been Pinned to CPU0

In the newly opened [Registers] view, click on [Pin to Debug Context] while CPU1 is selected in the [Debug] view.

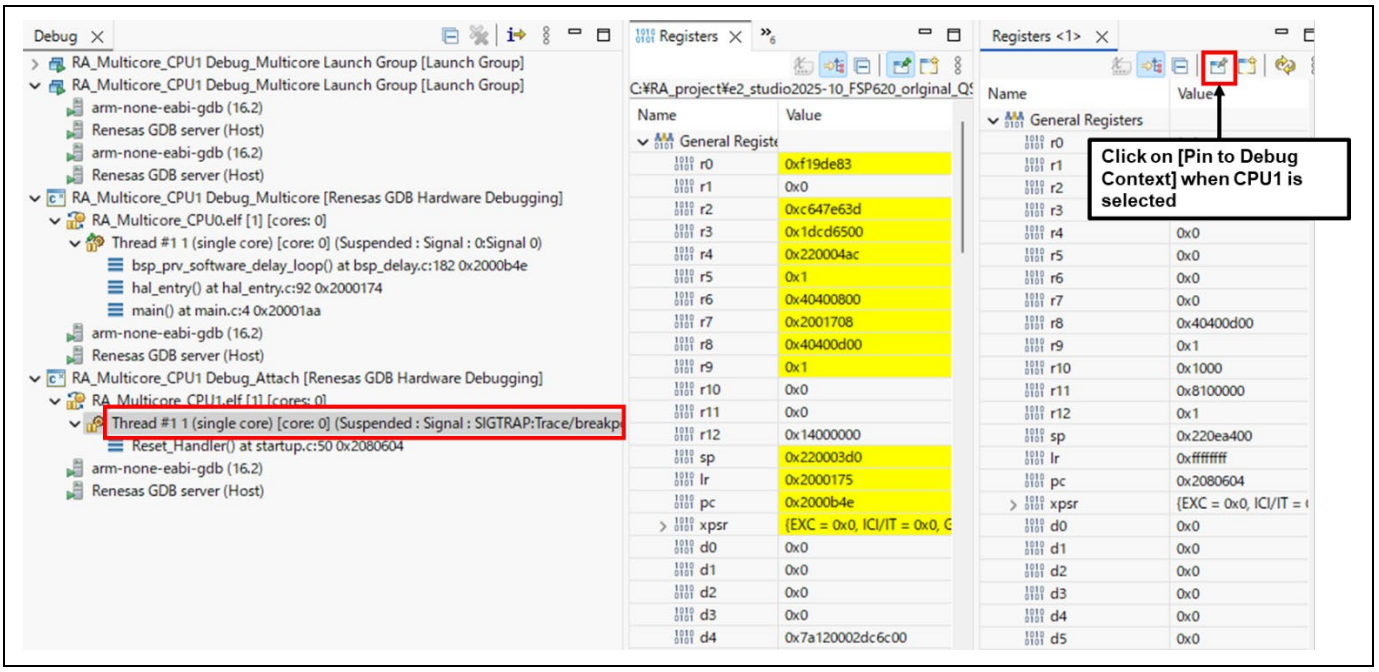

Figure 4-14 Debug – Pinning to CPU1

The pin on the [Pin to Debug Context] icon in the [Registers] view will become the same color (red in the example) as the pin on the CPU1 icon in the [Debug] view.

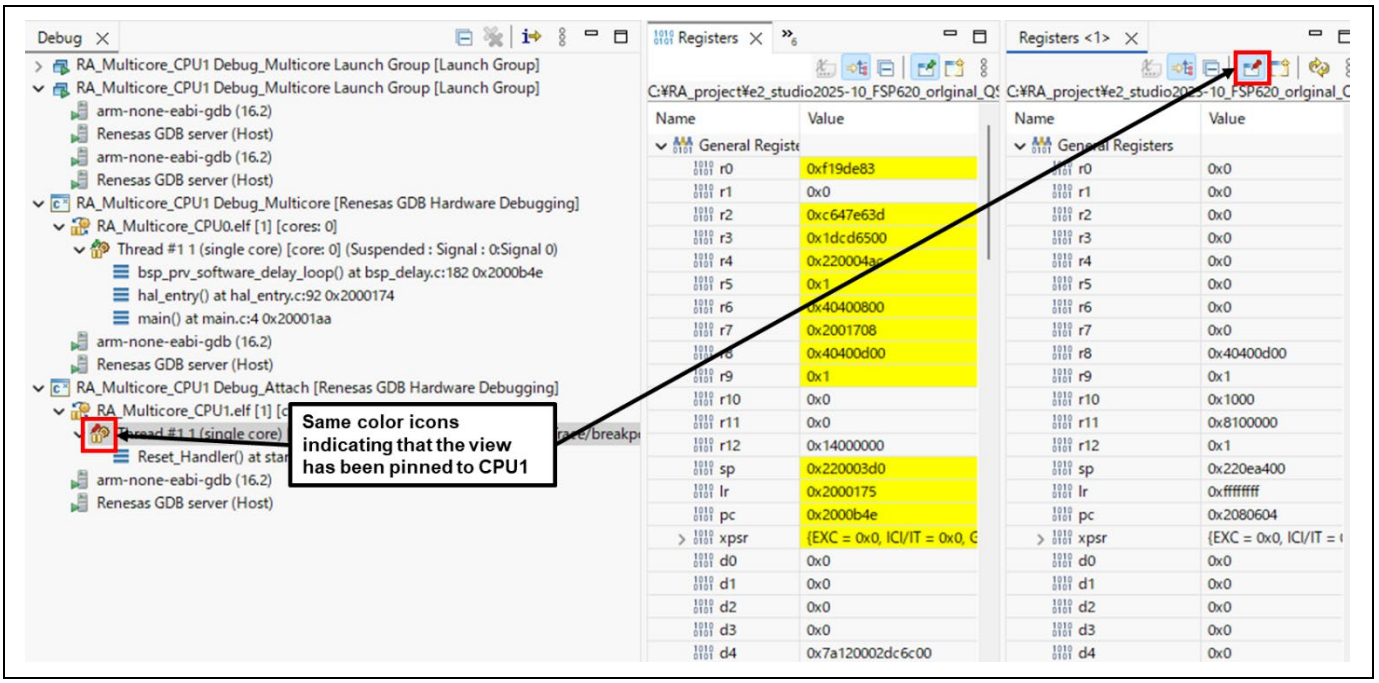

Figure 4-15 Debug – Icons Indicating That the View Has Been Pinned to CPU1

4.3 Procedure for Using Existing Debugging Functions in Multicore Debugging

This section describes the procedure for using existing debugging functions in multicore debugging.

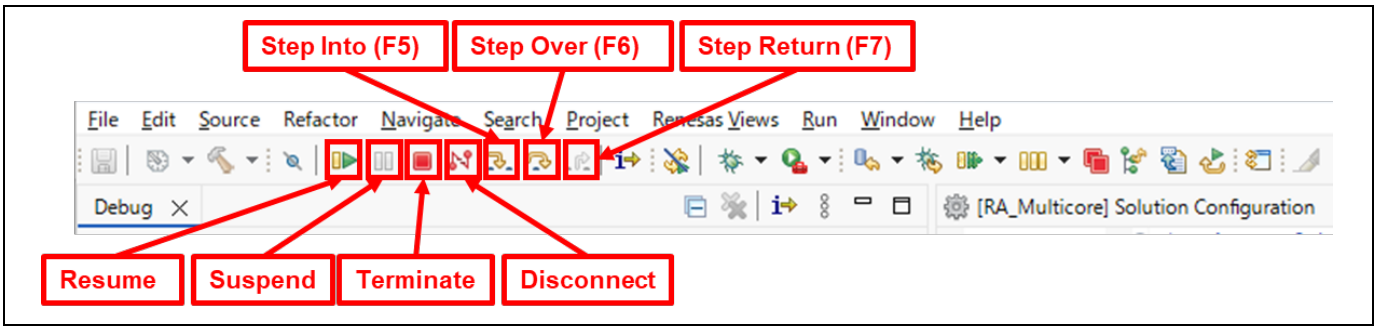

Figure 4-16 Debug – Existing Debugging Functions

In multicore debugging, threads of the selected CPU or CPUs are the targets for handling. To select threads of both CPUs, select threads while pressing the Ctrl key.

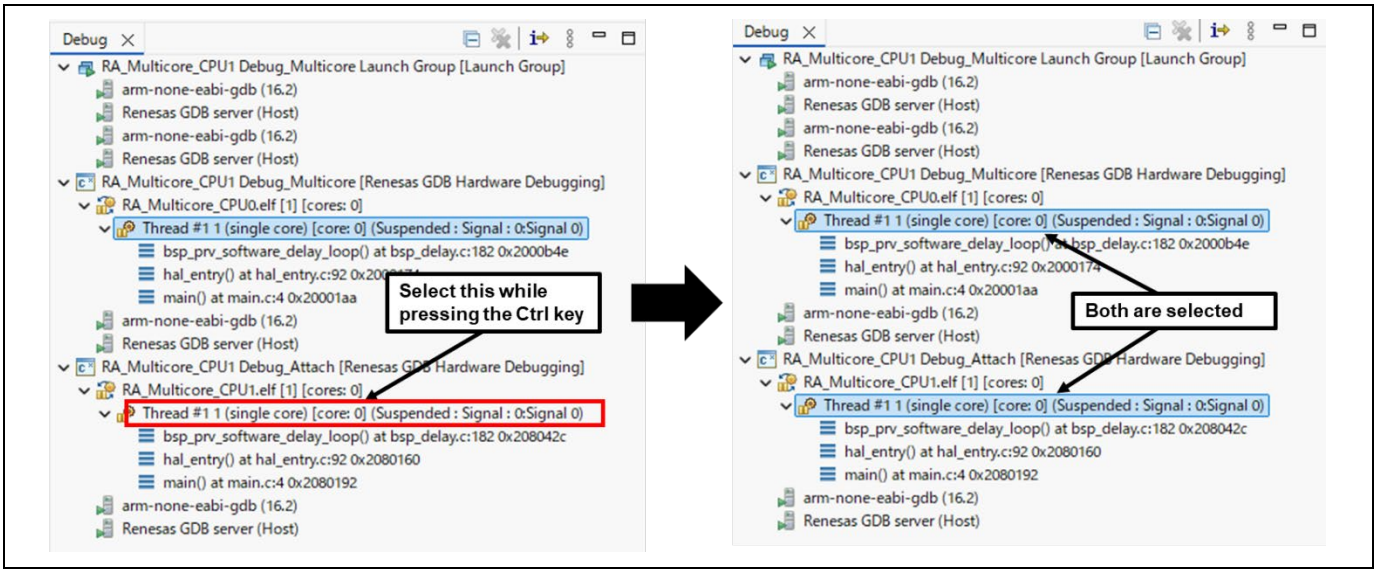

Figure 4-17 Debug – Selecting Multiple Threads in the [Debug] View

4.3.1 Operation of Existing Debugging Functions in the J-Link (When Multiple Threads are Selected)

- The  button is used to start execution by CPU0 and then start execution by CPU1.
- The  button is used to suspend execution by CPU0 and then suspend execution by CPU1.
- The  button is used to end the debug session of CPU0 and then end the debug session of CPU1.
- The  button is used to disconnect the debug session of CPU0 and then disconnect the debug session of CPU1.
- The  button or F5 key is used to perform step-in execution by CPU0 and then perform step-in execution by CPU1.
- The  button or F6 key is used to perform step-over execution by CPU0 and then perform step-over execution by CPU1.
- The  button is grayed out and cannot be selected.

4.3.2 Operation of Existing Debugging Functions in the E2/E2 Lite (When Multiple Threads are Selected)

- The  button is grayed out and cannot be selected. Use the  button to simultaneously start execution by both CPUs on the hardware level by the E2/E2 Lite.
- The  button is grayed out and cannot be selected. Use the  button to simultaneously suspend execution by both CPUs on the hardware level on the E2/E2 Lite.
- The  button is used to end the debug session of CPU0 and then end the debug session of CPU1.
- The  button is used to disconnect the debug session of CPU0 and then disconnect the debug session of CPU1.

  Note: If this button is used when both debug sessions are selected in FSP v6.1.0, an error occurs. Therefore, this button should be used only when one of the debug sessions is selected.
- The  button is grayed out and cannot be selected.
- The  button is grayed out and cannot be selected.
- The  button is grayed out and cannot be selected.

4.3.3 [Debug] View in Multicore Debugging

The functions being executed by each CPU are displayed for each thread in the [Debug] view. The state of the thread is also displayed: “Suspended” indicates that the program is stopped, and “Running” indicates that the program is being executed.

When the J-Link is in use, CPU0 and CPU1 are controlled by independent [Renesas GDB Hardware Debugging] debug configurations in the [Debug] view because CPU0 and CPU1 are separately controlled in a single debug session.

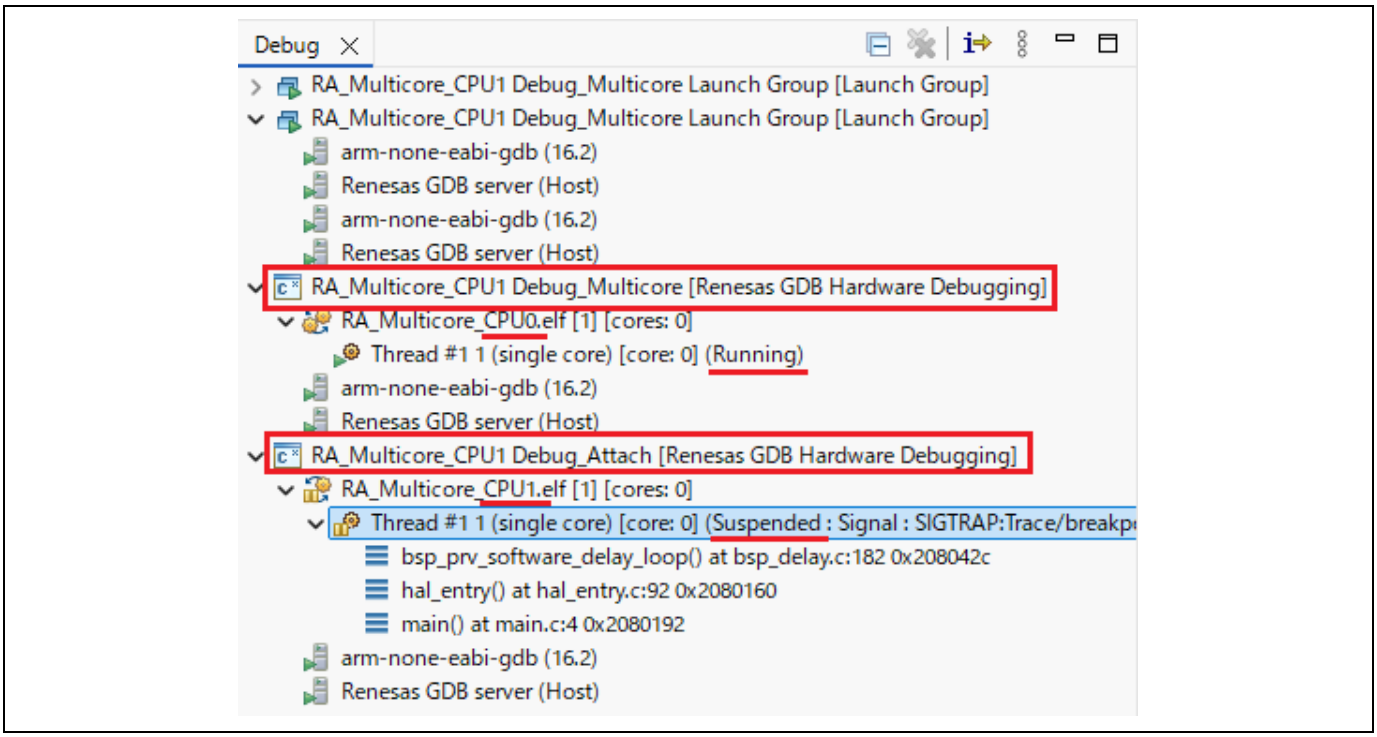

Figure 4-18 Debug – [Debug] View When Using the J-Link

When the E2/E2 Lite is in use, CPU0 and CPU1 are controlled by a single [Renesas GDB Hardware Debugging] debug configuration in the [Debug] view because CPU0 and CPU1 are simultaneously controlled in a single debug session.

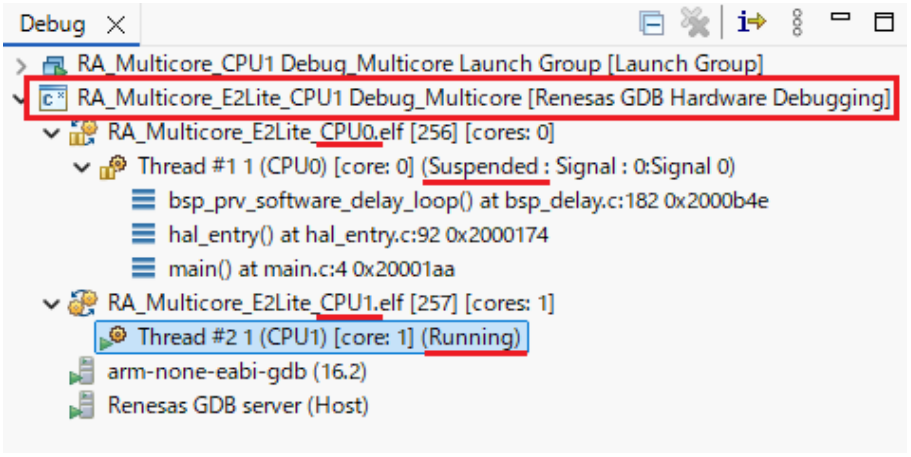

Figure 4-19 Debug – [Debug] View When Using the E2/E2 Lite

Revision History

e² studio
Additional Document for Quick Start Guide
Multicore Project

Rev.	Date	Description	
		Page	Summary
1.00	Feb.17.26	—	First Edition issued

e² studio
Additional Document for Quick Start Guide
Multicore Project

Publication Date: Rev.1.00 Feb.17.26

Published by: Renesas Electronics Corporation

e² studio
Integrated Development Environment
Additional Document for Quick Start Guide
Multicore Project

Renesas Electronics Corporation

R20UT5776EJ0100