Renesas RZ Family

Migrating Projects to RZ FSP v4.x.x

Introduction

This application note describes how to migrate projects that use the legacy RZ/A, RZ/G, RZ/N, RZ/T, and RZ/V Flexible Software Packages (FSPs) to the new unified RZ FSP v4.x.x.

The new unified RZ FSP consolidates all previous family-specific FSPs into a single package, providing a consistent development experience across all RZ product series. This integration simplifies project maintenance and improves development efficiency by eliminating differences between the individual FSPs of each RZ series. While the legacy RZ/A, RZ/G, RZ/N, RZ/T, and RZ/V FSPs will continue to be supported until the end of 2026, future feature enhancements and bug fixes will be provided exclusively through the unified RZ FSP. Therefore, it is strongly recommended to migrate existing projects to RZ FSP v4.x.x.

Prerequisites and Intended Audience

Prerequisites

- Basic understanding of Renesas Flexible Software Package (FSP).
- Prior experience working with Example Project (EP) using FSP (e.g., configuring stacks and drivers).
- Prior experience working with Integrated Development Environment such as e^2 studio, IAR.

Intended Audience

- Embedded Software Engineers or Firmware Developers working with Renesas MCU/MPU.
- System Integrators migrating projects to the RZ FSP.
- Individuals with intermediate or higher experience in FSP and EP configuration.

Required Resources

The following resources are referenced throughout this application note.

Development Tools and Software

- e^2 studio: version 2025-12 or later.
- IAR Embedded Workbench for Arm: 9.60.3.
- Renesas FSP Smart Configurator: version 2025-12 or later.
- Flexible Software Package (FSP) for Renesas RZ MPU Family v4.0.0 or later.

Target Devices

Below are the Renesas MPU products to which the information within this document is applicable:

- RZ/A series: RZ/A3UL, RZ/A3M.
- RZ/G series: RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/G3E, RZ/G3S.
- RZ/N series: RZ/N2L, RZ/N2H.
- RZ/T series: RZ/T2L, RZ/T2M, RZ/T2ME, RZ/T2H.
- RZ/V series: RZ/V2L, RZ/V2N, RZ/V2H.

Supported Kit

RZ/A series

- RZ/A3UL Evaluation Board Kit QSPI Edition.
- RZ/A3UL Evaluation Board Kit OCTAL-SPI Edition.
- RZ/A3M Evaluation Board Kit Edition.

RZ/G series

- RZ/G2L Evaluation Board Kit.
- RZ/G2LC Evaluation Board Kit.
- RZ/G2UL Evaluation Board Kit.
- RZ/G3S Evaluation Board Kit.
- RZ/G3E Evaluation Board Kit.

RZ/N series

- Renesas Starter Kit+ RZ/N2L.
- Evaluation Board Kit for RZ/N2H.

RZ/T series

- Renesas Starter Kit+ RZ/T2M.
- Renesas Starter Kit+ RZ/T2L.
- Renesas Starter Kit+ RZ/T2ME.
- Evaluation Board Kit for RZ/T2H.

RZ/V series

- RZ/V2L Evaluation Board Kit.
- RZ/V2H Evaluation Board Kit.
- RZ/V2N Evaluation Board Kit.

Contents

1. Why Are RZ FSPs Integrated into a Single Package?

RZ FSP v4.x.x is a unified Flexible Software Package that supports multiple Renesas RZ device families (RZA, RZG, RZV, RZT, RZN) under one common framework. Instead of maintaining separate FSP versions for each RZ series, this single package provides:

- Common architecture for drivers, middleware, and BSP.
- Consistent API across different RZ devices.
- Simplified migration between RZ families.
- Reduced maintenance and easier updates.

2. Installing RZ FSP Pack

Refer to the Getting Started Flexible Software Package documents of each RZ series:

- RZ/A: RZA Getting Started Flexible Software Package
- RZ/G: RZG Getting Started Flexible Software Package
- RZ/T, RZ/N: RZT2 RZN2 Getting Started Flexible Software Package
- RZ/V: RZ/V Getting Started with Flexible Software Package

3. Migrating the Existing Project to e² studio

A sample project is used to demonstrate the migration procedure. The same approach can be applied to any project that currently uses an existing FSP.

The migration process consists of the following essential steps. These steps are summarized below and explained in detail in the subsequent sections.

3.1 Creating a New Project for the Example Project

The creation of a new Renesas RZ C/C++ FSP Project is the first step in the migration of an existing project. All information provided below uses the GTM project of RZ/A3M as a reference.

Launch e² studio, click **File > New > Renesas C/C++ Project > Renesas RZ**, and select **Renesas RZ C/C++ FSP Project > Next**.

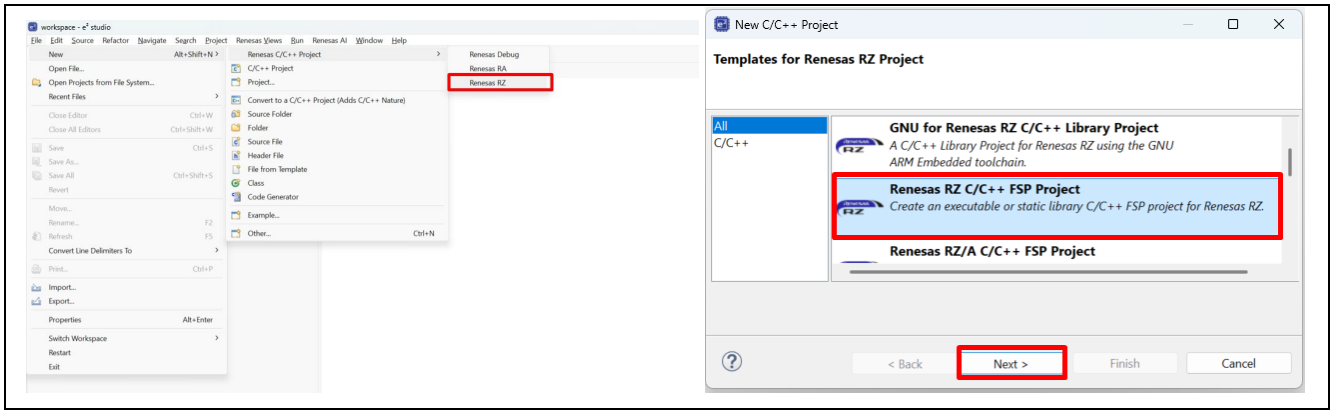


Figure 1. New Renesas RZ C/C++ FSP project

Assign a name for this new project, such as **ek_rza3m_gtm,** then > **Next**.

In the Renesas RZ C/C++ Project window, select the **Board** type as EK-RZ/A3M NOR Boot (Exec with DDR SDRAM), choose **GCC ARM A-Profile (AArch64 bare-metal)** for **Toolchains**, and J-Link ARM for **Debugger,** then > **Next**.
If no Toolchains are available in the list, go to “Manage Toolchains…” to download/add the **GCC ARM A-Profile (AArch64 bare-metal)** toolchain.

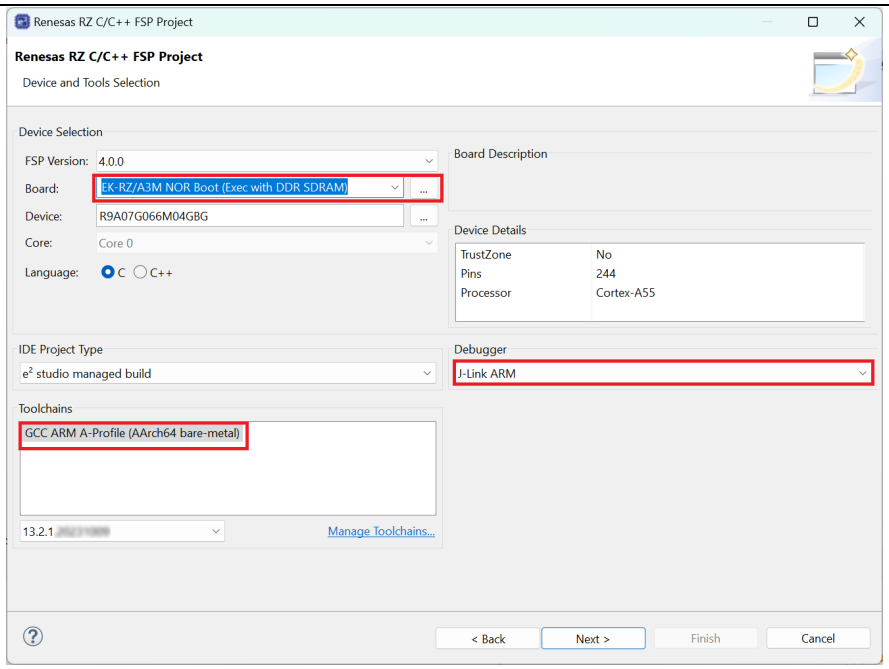


Figure 2. Device and Tools selection

If an MPU with multiple cores is used, select the **Preceding Project** option. In this example, the EK-RZ/A3M board has a single core, so simply click **Next**.

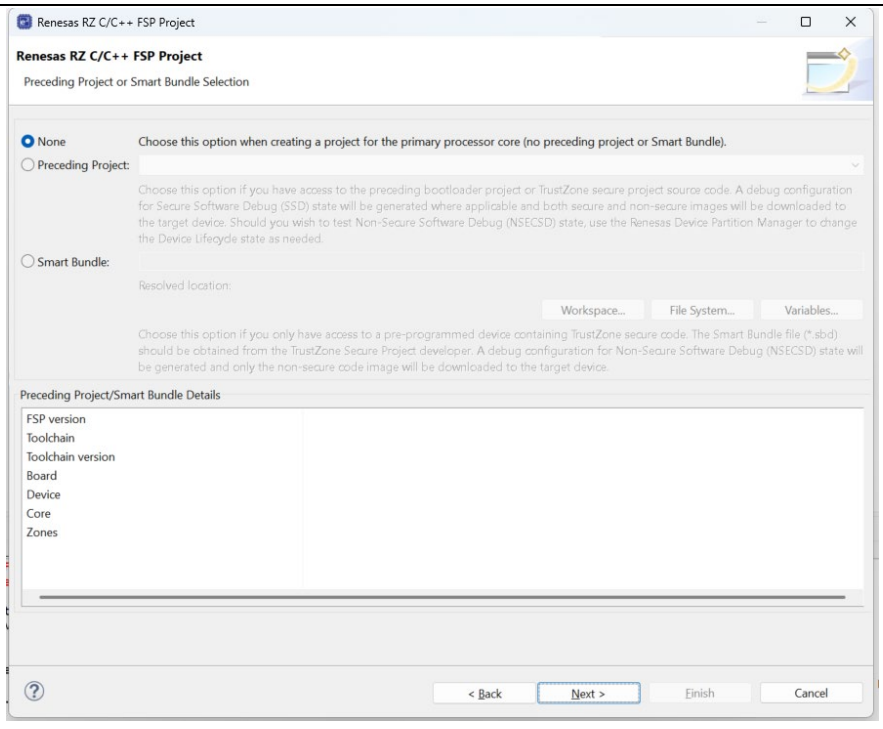


Figure 3. Select project type

This project does not use RTOS, so select **No RTOS** and click **Next**.

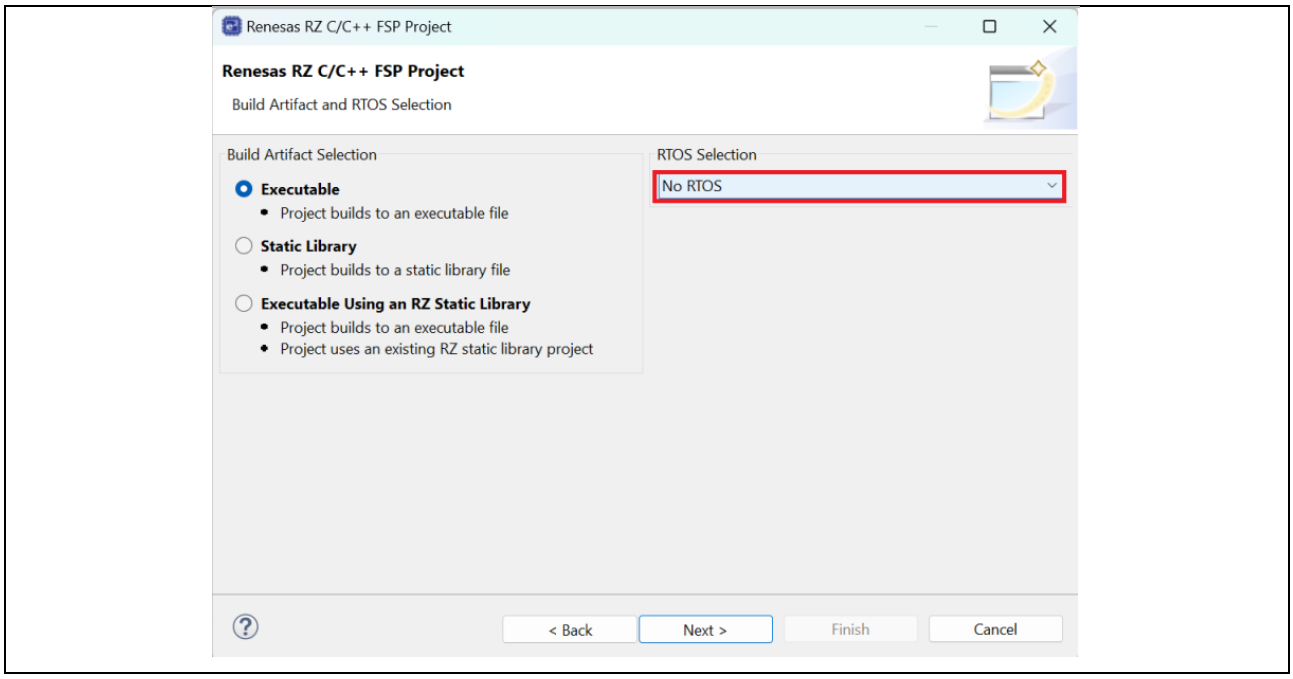


Figure 4. Build Artifact and RTOS selection

Select **Bare Metal – Minimal** template for this example and click **Finish**.

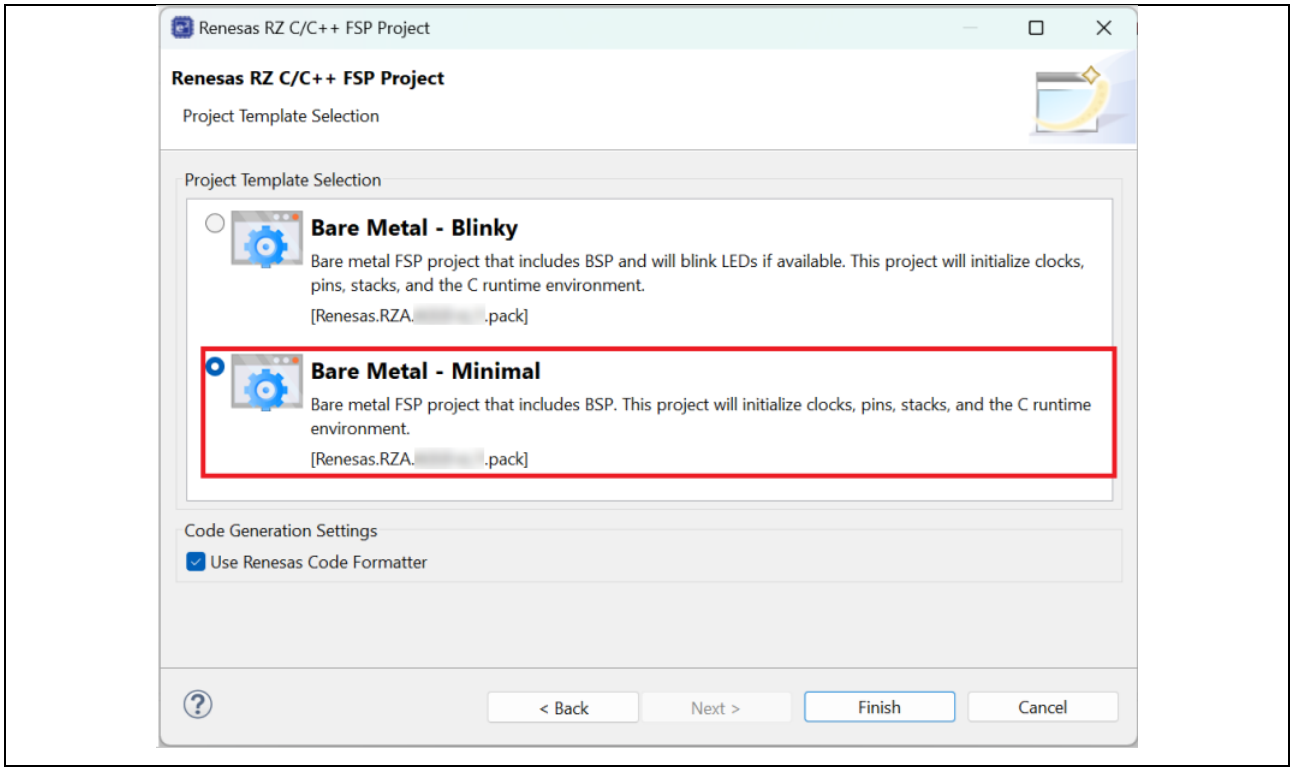


Figure 5. Project Template selection

After completing these steps, the project will be created successfully, as shown in Figure 6.

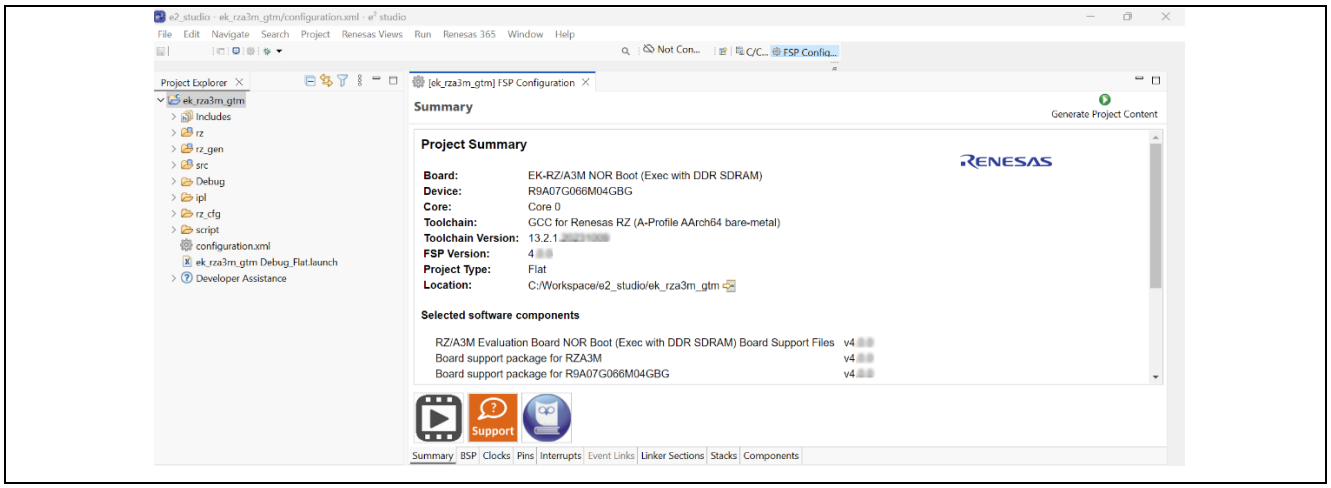


Figure 6. Successful project creation

3.2 Migrate Source Codes and FSP Configurations

3.2.1 Migrate source codes

- If the above steps are completed successfully, the src folder will be available in the workspace, as shown in Figure 7.

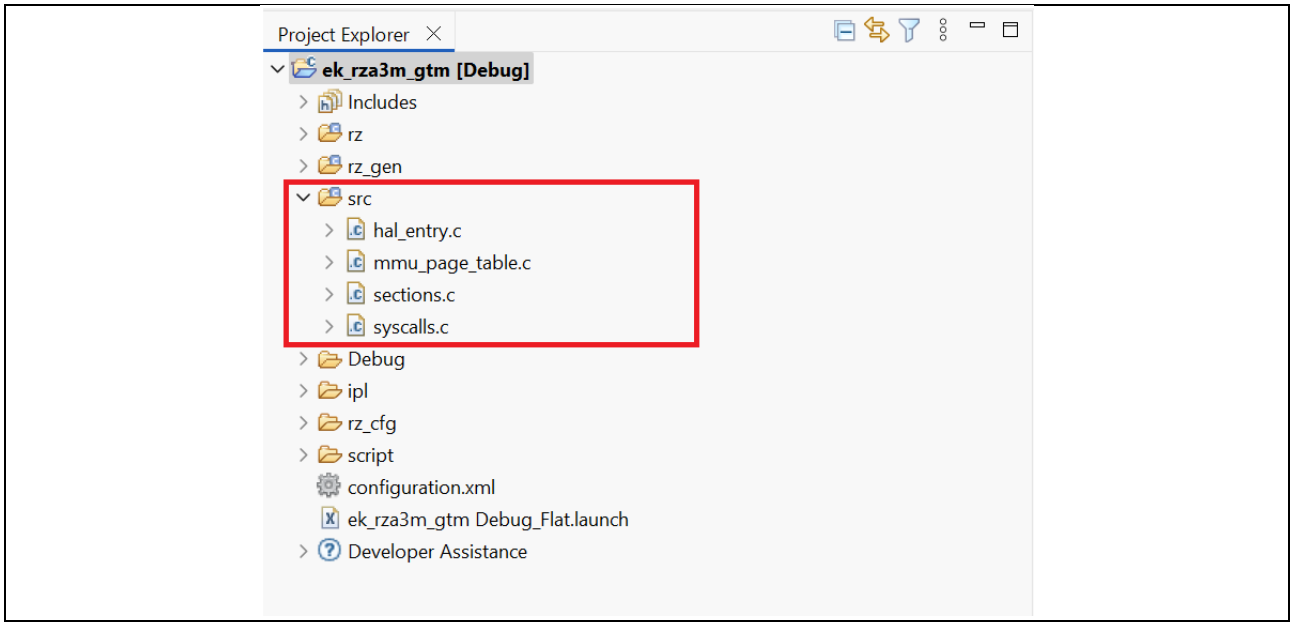


Figure 7. e² studio Project folder structure

- Update the src folder by copying example code from the existing project's src folder, excluding hal_entry.c and all FSP-generated files, as shown in Figure 8. Do not replace FSP-generated files, as doing so may cause unexpected errors.

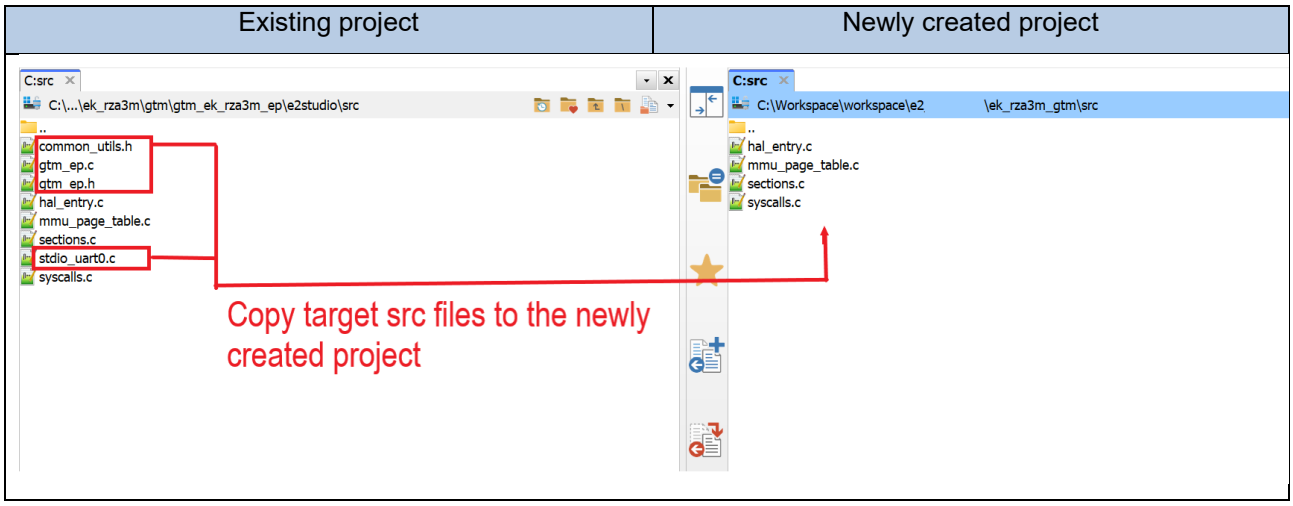


Figure 8. Copy existing source files to the new project

- In hal_entry.c, update only the source code related to the example project while keeping the generated template. As shown in Figure 9, update only other functions in hal_entry.c, excluding R_BSP_WarmStart().

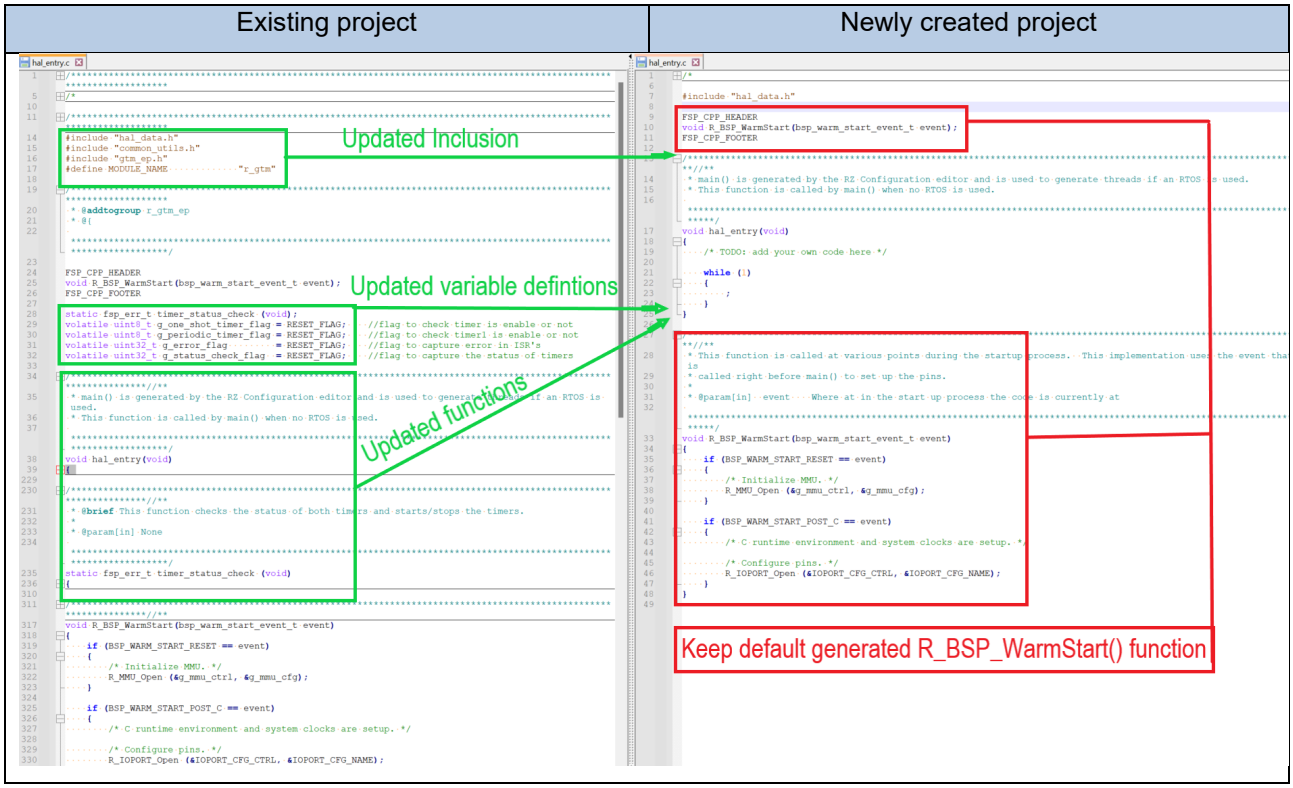


Figure 9. Keep generated R_BSP_WarmStart() function

Note: In case of migrating source codes of projects for RZ/T, RZ/N devices, the project template will also generate R_BSP_WarmStart_StackLess() function in hal_entry.c file. You need to keep that generated function.

3.2.2 FSP Configuration

After adding the source code, the configuration needs to be migrated.

Click on the configuration.xml, then a screen will appear as shown below with multiple tabs.

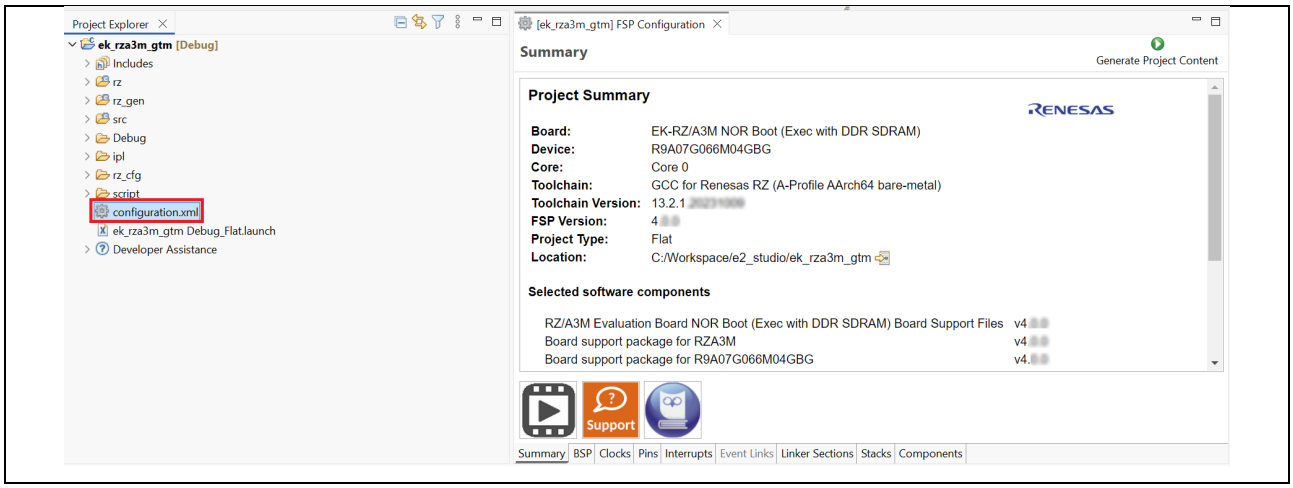


Figure 10. e² studio Project Summary Tab

Required tabs can be selected in the tabs bar, and the properties of individual components can be configured as per the project requirement.

Generally, the following tabs are configured for Project setup before generating the project:

3.2.2.1 Configuring the example project BSP tab

The BSP screen presents the BSP Properties configuration in e² studio. It is used to define board parameters and system initialization settings, ensuring proper startup and compatibility with the target hardware.
To configure BSP properties, please refer to the BSP settings of an existing project or related project documentation. Then configure correct values for each property, especially below:

- Secure stack size (bytes)
- Main stack size (bytes)
- Heap size (bytes)
- Parameter checking

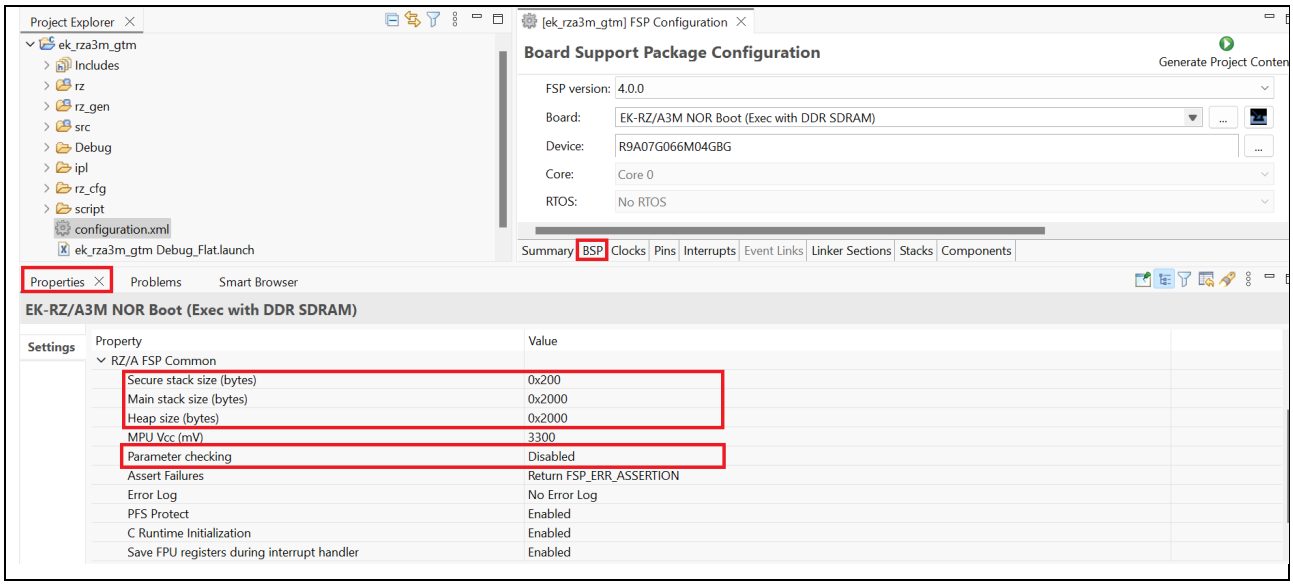


Figure 11. e² studio Project BSP Tab

3.2.2.2 Configuring the Example Project Clock Tab

This screen is used to configure clock sources and division ratios for the RZ/A3M device. To configure the Clock tab, please refer to the Clock settings of an existing project or related project documentation. Then configure correct values for each property, especially below:

- System clock
- Clock division ratio

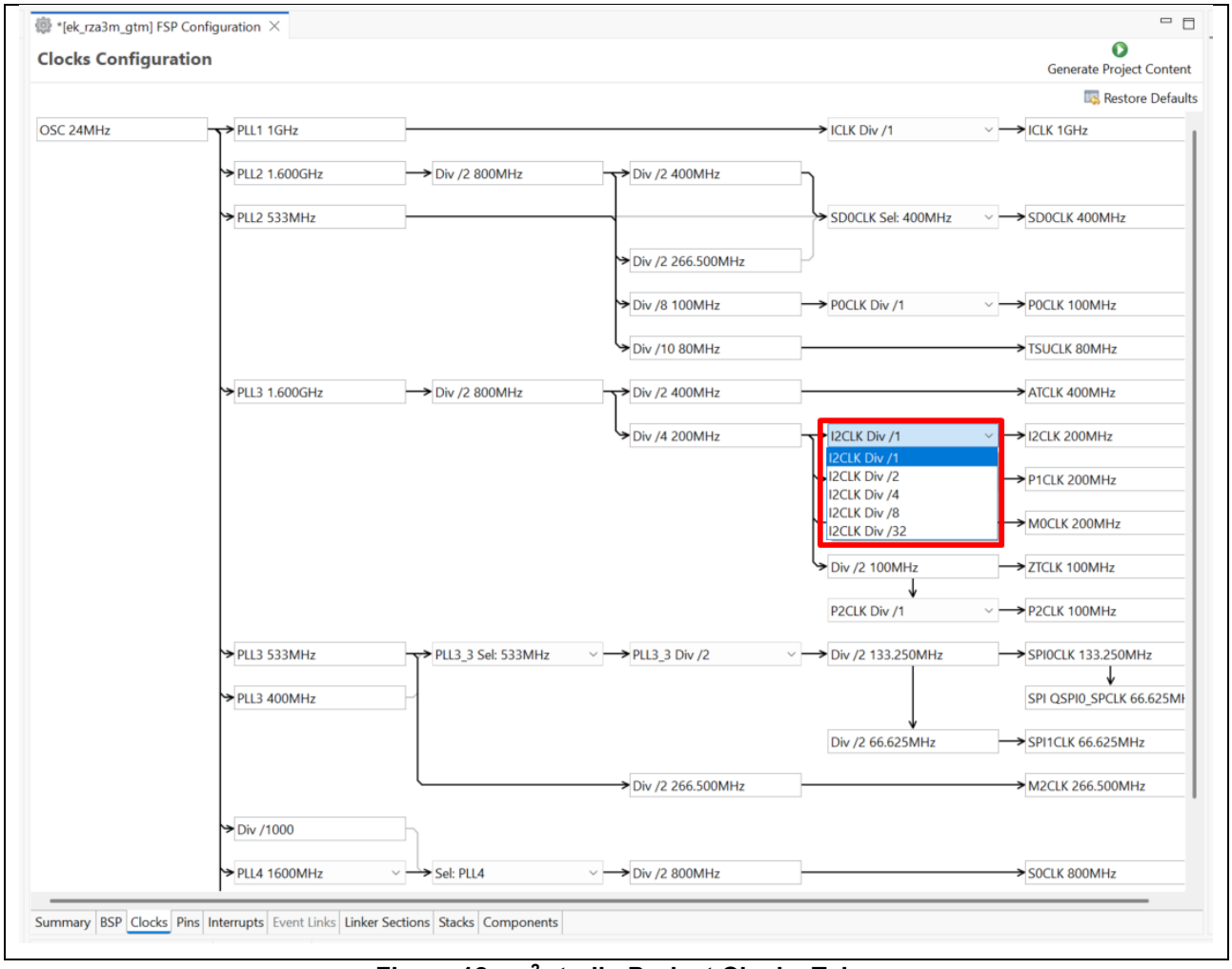


Figure 12. e² studio Project Clocks Tab

3.2.2.3 Configuring the Example Project Pin Tab

Pin Configuration screen for assigning pin functions and electrical characteristics. To configure the Pins tab, please refer to the Pins setting of the existing project or the related project documentation. Then configure the correct value of the required Port pins and Peripherals.

The example project uses the SCIF0 peripheral for UART functionality. Therefore, set the **Operation Mode** to "**Custom**", and then configure pins for **SCIF0_RXD** and **SCIF0_TXD**.

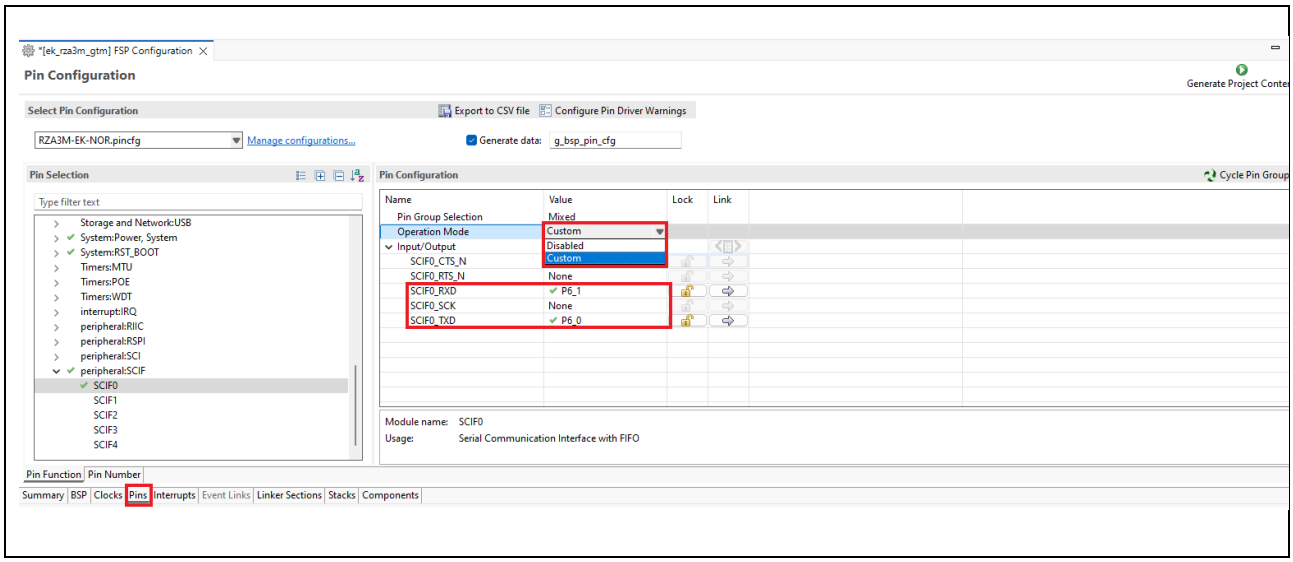


Figure 13. e² studio Project Pins Tab

3.2.2.4 Configuring the Example Project Stack Tab

To configure the Stacks tab, please refer to the Stacks setting of an existing project or related project documentation. Please be aware that the Stacks tab includes:

- Threads
- Objects
- HAL/Common Stacks

Add all required **Threads**, **Objects**, and **HAL/Common Stacks** according to the requirements of the existing project.

The example project uses the following stacks:

- **Timer (r_gtm)**: One-shot timer and Periodic timer
- **UART Driver on r_scif_uart**

The respective module stack can be chosen from the **New Stack** option.

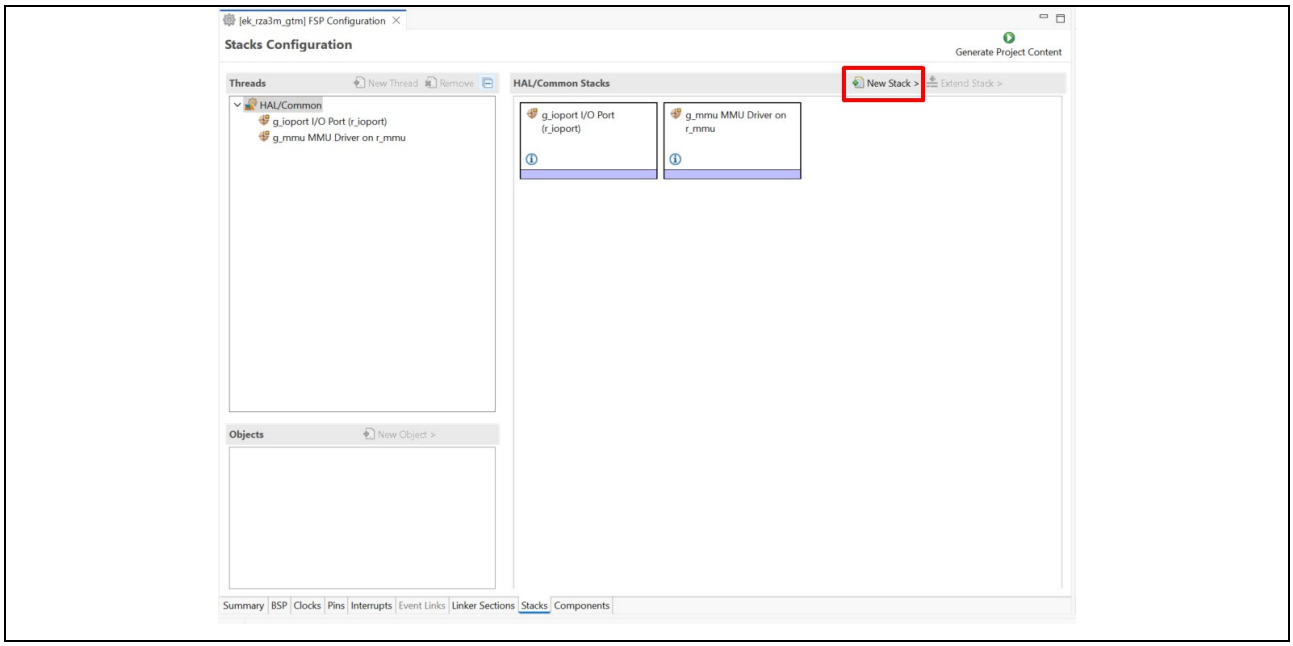


Figure 14. Stack Configuration

Select **Timers > Timer (r_gtm)** stack.

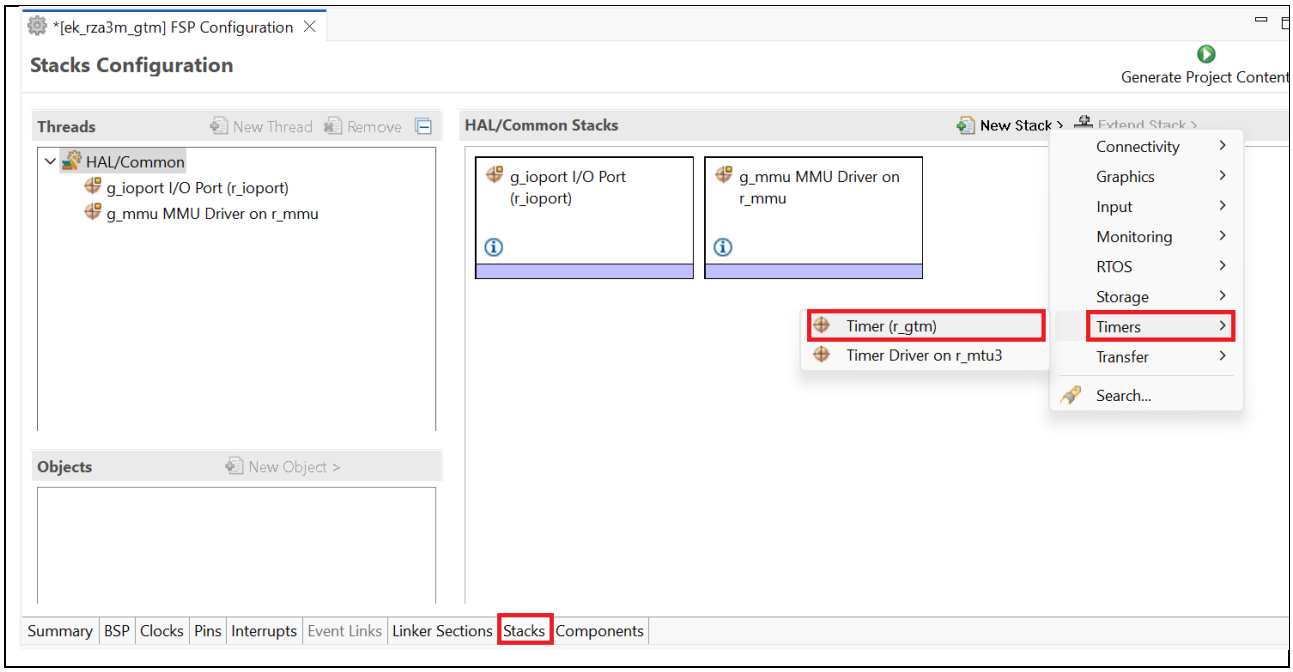


Figure 15. Timer (r_gtm) stack

Continue adding one more **Timer (r_gtm)** and **UART Driver on r_scif_uart** stacks.

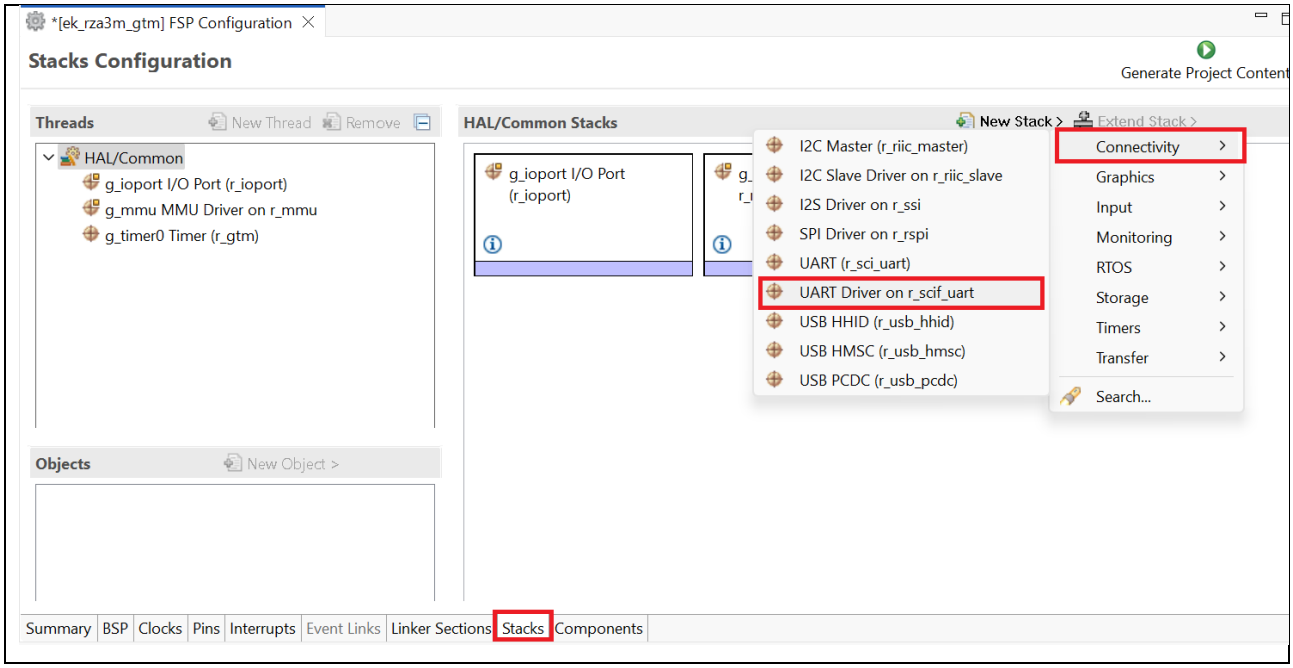


Figure 16. UART Driver on r_scif_uart stack

For each newly added stack, open **Properties** and configure each property based on the configuration of the existing project.

As the example Timer (r_gtm) stack, you need to configure important parameters below according to the existing project configuration:

- Name
- Channel
- Mode
- Period

- Period Unit
- Callback
- Underflow Interrupt Priority

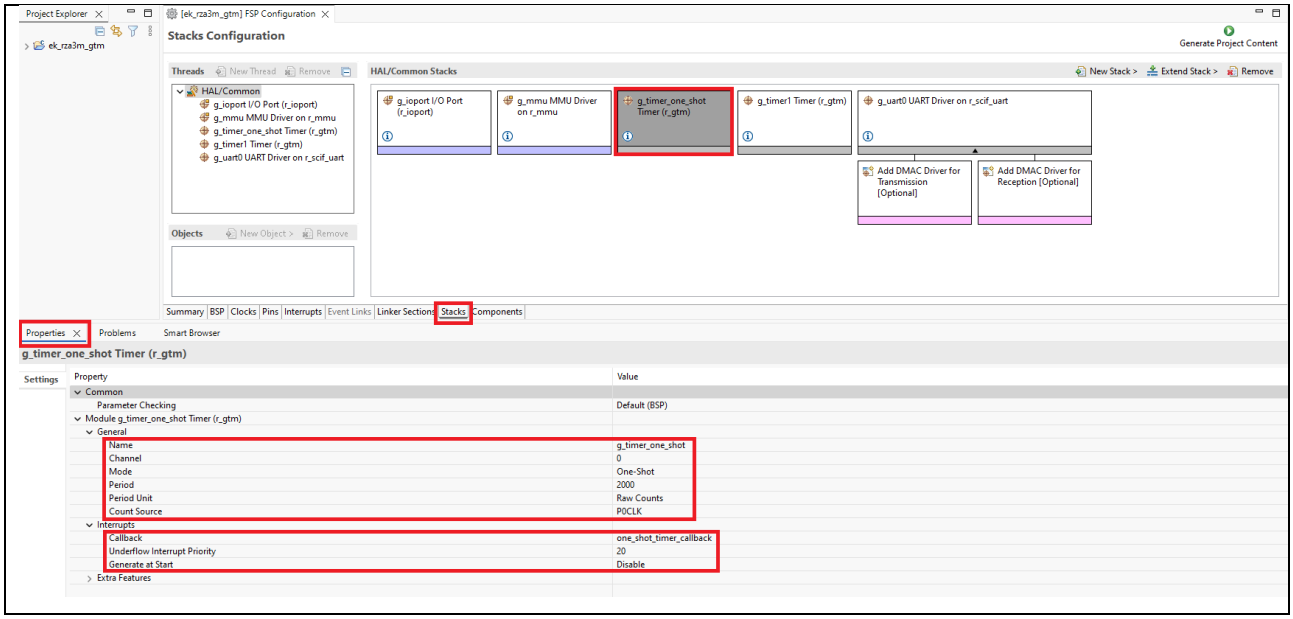


Figure 17. e² studio Project Stack Properties

After the completion of setting one stack, please continue to configure all remaining stacks to ensure that the configuration is the same as the existing project configuration.

3.2.2.5 Configuring the Example Project Components Tab

This screen shows the Components Configuration interface in e² studio. To configure the Components tab, please refer to the Components setting of an existing project or related project documentation. Then add the necessary components by ticking the boxes.

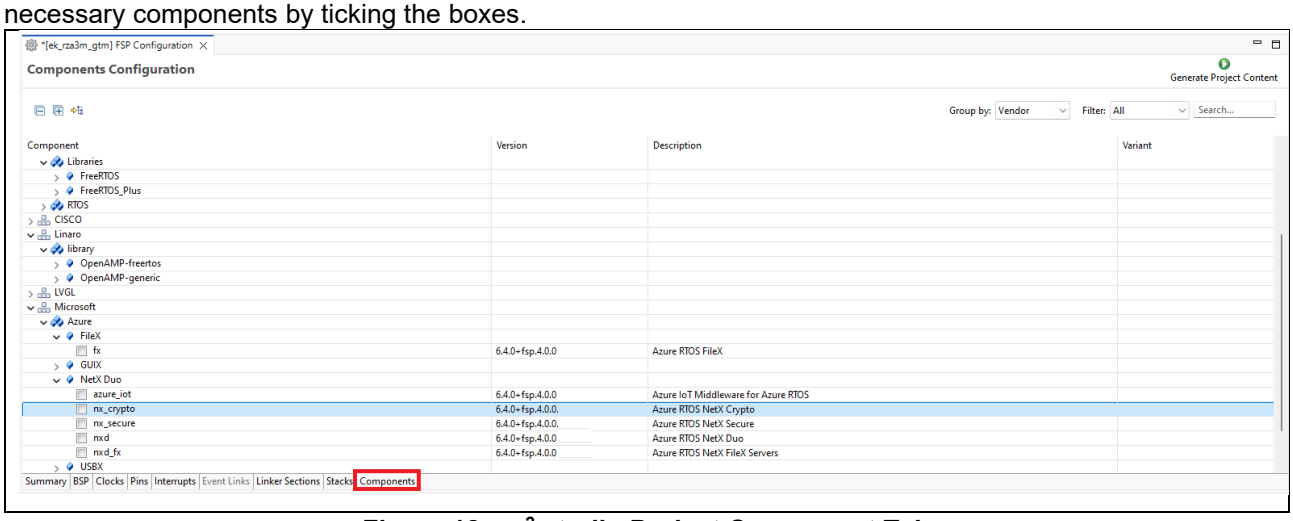


Figure 18. e² studio Project Component Tab

Then please untick the default template project, such as **Projects > rza > baremetal_minimal**.

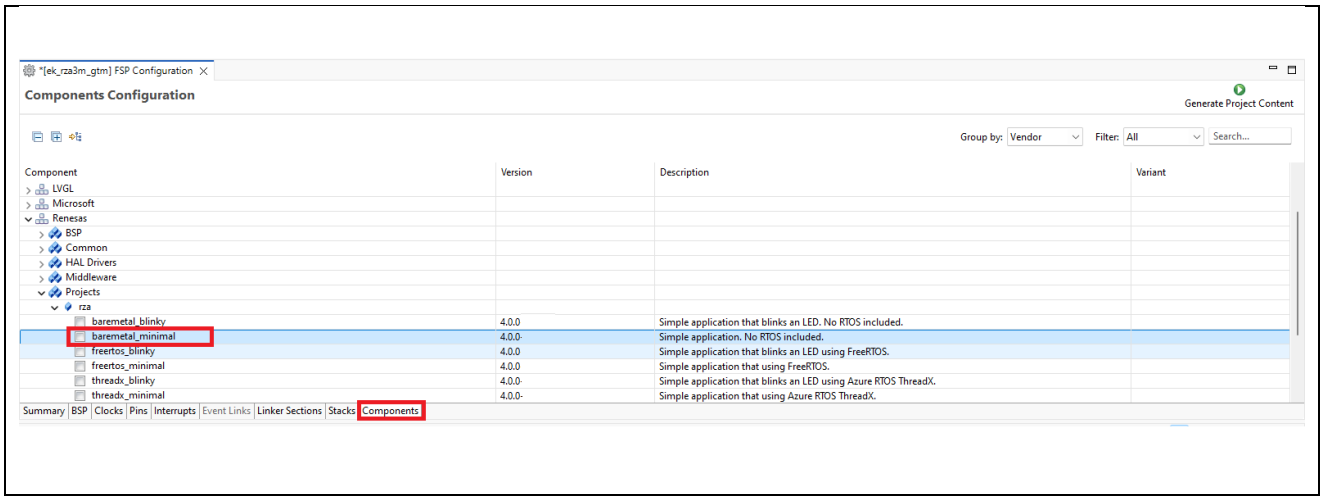


Figure 19. Component baremetal_minimal

3.2.2.6 Generating Project Content

Click the **Generate Project Content** button to apply all configured stacks and update the project files accordingly.

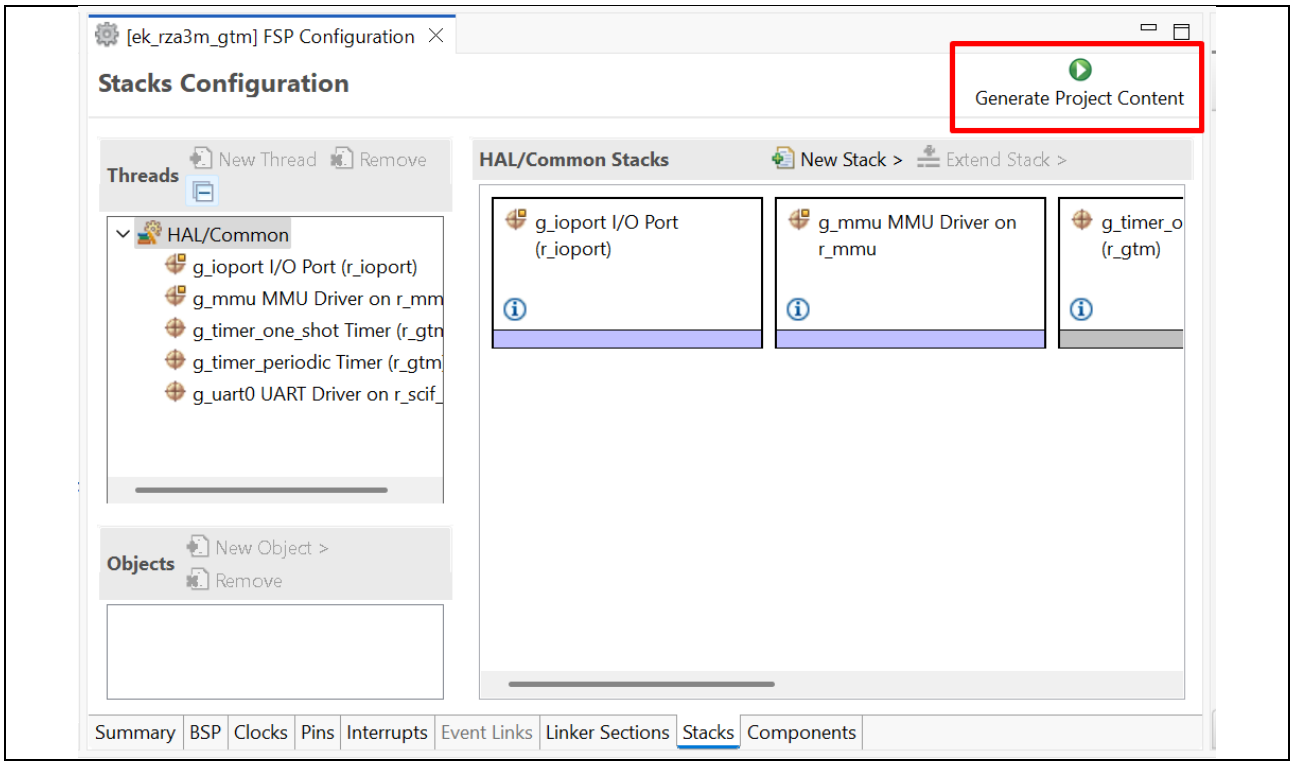


Figure 20. e² studio Project Configuration Tab

3.3 Configuring and Compiling the Project

3.3.1 Building the Project

Before building the project, make sure the following pre-build operations are completed:

- Module-specific src folder is present inside the project folder directory
- Project files are generated with required configurations (BSP, Clocks, Pins, etc.)

Right-click on the project, and click on the **Build Project** option.

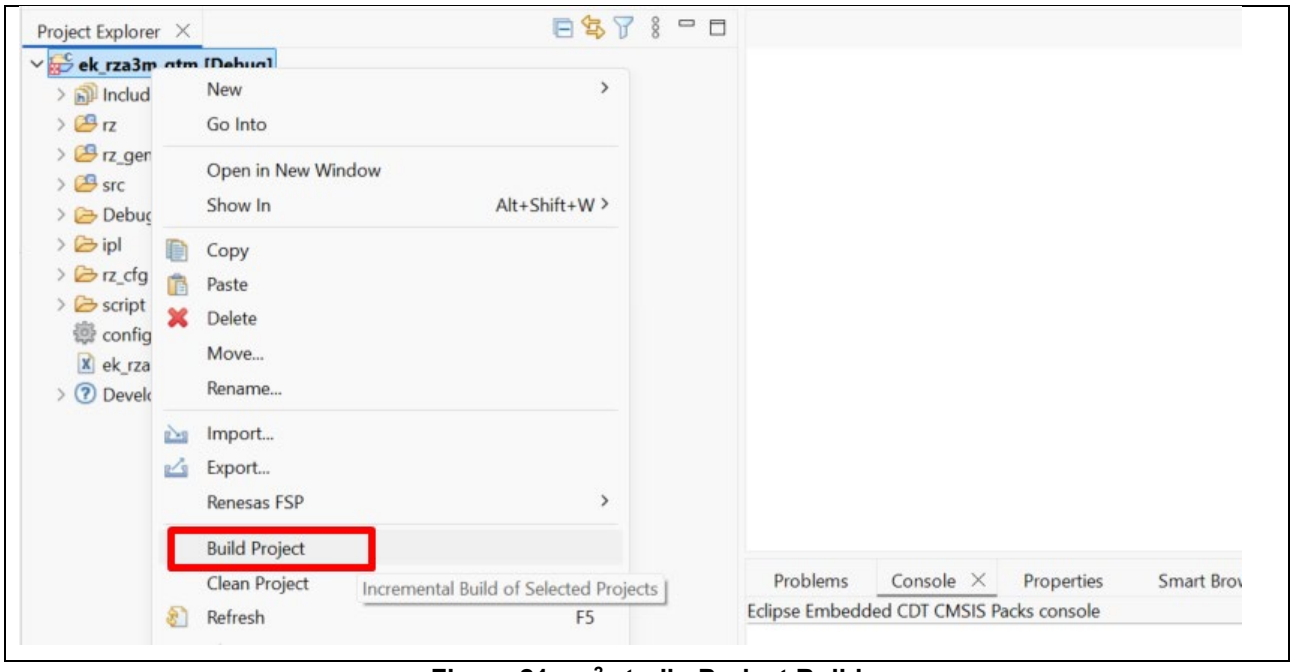


Figure 21. e² studio Project Build

3.3.2 Addressing Warnings and Errors in Build Operation

On completion of the Generating and Building Procedure of the Project. The successful or failed operation can be observed in the Build logs in the Console window.

Build Window Error and Warning Messages can be observed in detail, with the total number of Errors and Warnings found. Sample as shown in Figure 22:

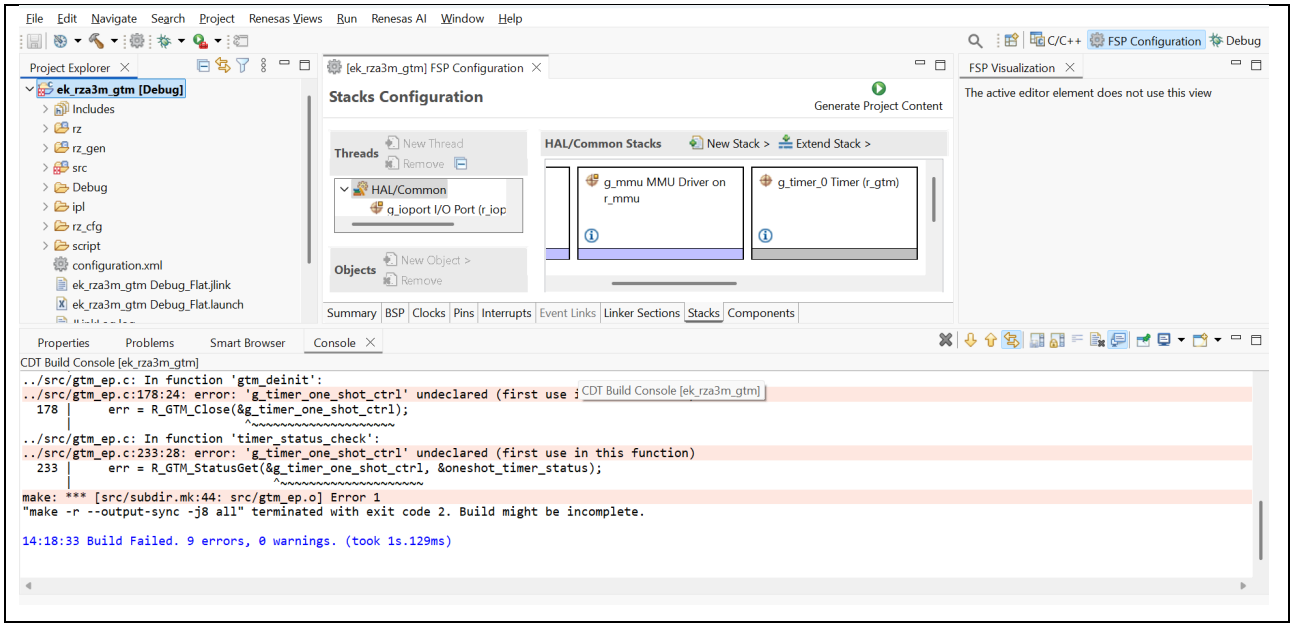


Figure 22. e² studio Project Build Console

The build fails due to undeclared variables in the source code, as shown in Figure 22. During project migration, such errors may occur. To resolve this, review and adjust the **stack configuration** properties accordingly.

The error “**g_timer_one_shot_ctrl’ undeclared (first use in this function)**” occurs because the source code references an object named **g_timer_one_shot_ctrl**, which has not been defined.

In the current example, the r_gtm stack already exists, but during migration, the step to configure the value of the Name property to **g_timer_one_shot** was missing, which caused the build error.

When a stack named **g_timer_one_shot** is correctly configured, please click the **Generate Project Content** button again to apply all configurations. FSP will automatically generate the object **g_timer_one_shot_ctrl** in the generated source code.

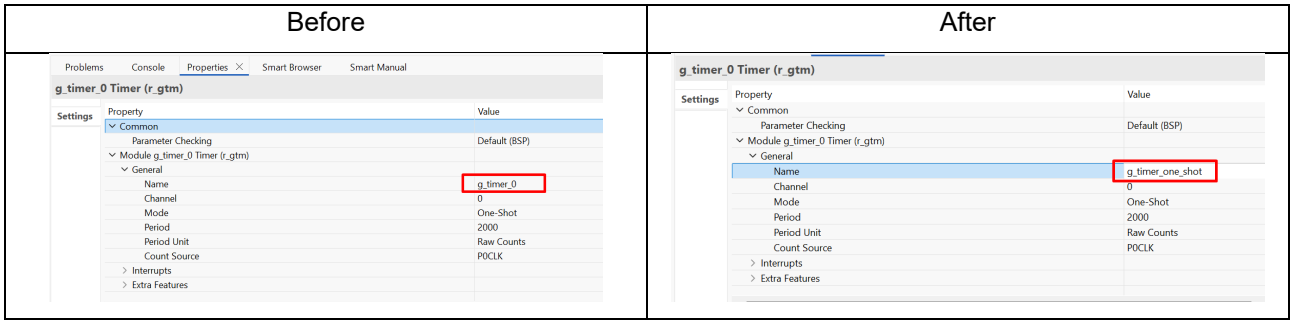


Figure 23. GTM timer property Settings

After making these changes, the building should be completed successfully.

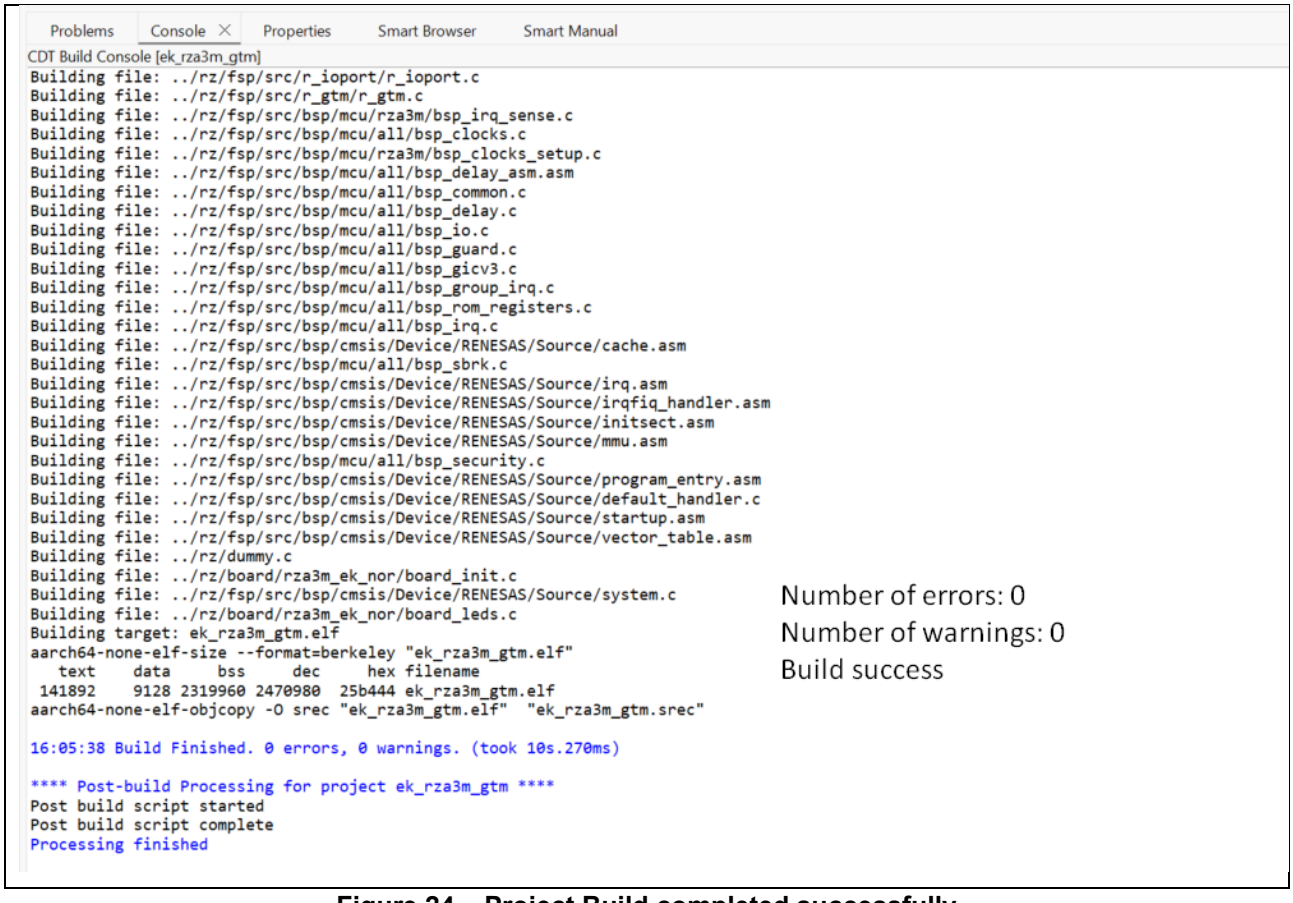


Figure 24. Project Build completed successfully

3.4 Downloading and Debugging the Project

3.4.1 Debug Prerequisites

To debug the project on a board, the following are required:

- The board must be connected to a **PC**.
- The debugger must be configured to communicate with the board.
- The application must be programmed onto the microprocessor.

Applications run from the internal RAM of the microprocessor. To run or debug the application, it must first be programmed into RAM using a JTAG debugger. The evaluation board includes a JTAG header and requires an external JTAG debugger connected to this header.

3.4.2 Debug Configuration Settings

Before debugging, make sure there are no errors, and all the build warnings have been reviewed. Confirm that the target board is connected to the host system.

Right-click on the project, select **Debug As** → **Debug Configurations**. Select your debugger configuration in the window. If it is not visible, then it must be created by clicking the **New** icon in the top left corner of the window. Once selected, the **Debug Configuration** window displays the **Debug configuration** for your project. Verify that the correct target MPU and debug interface are configured.

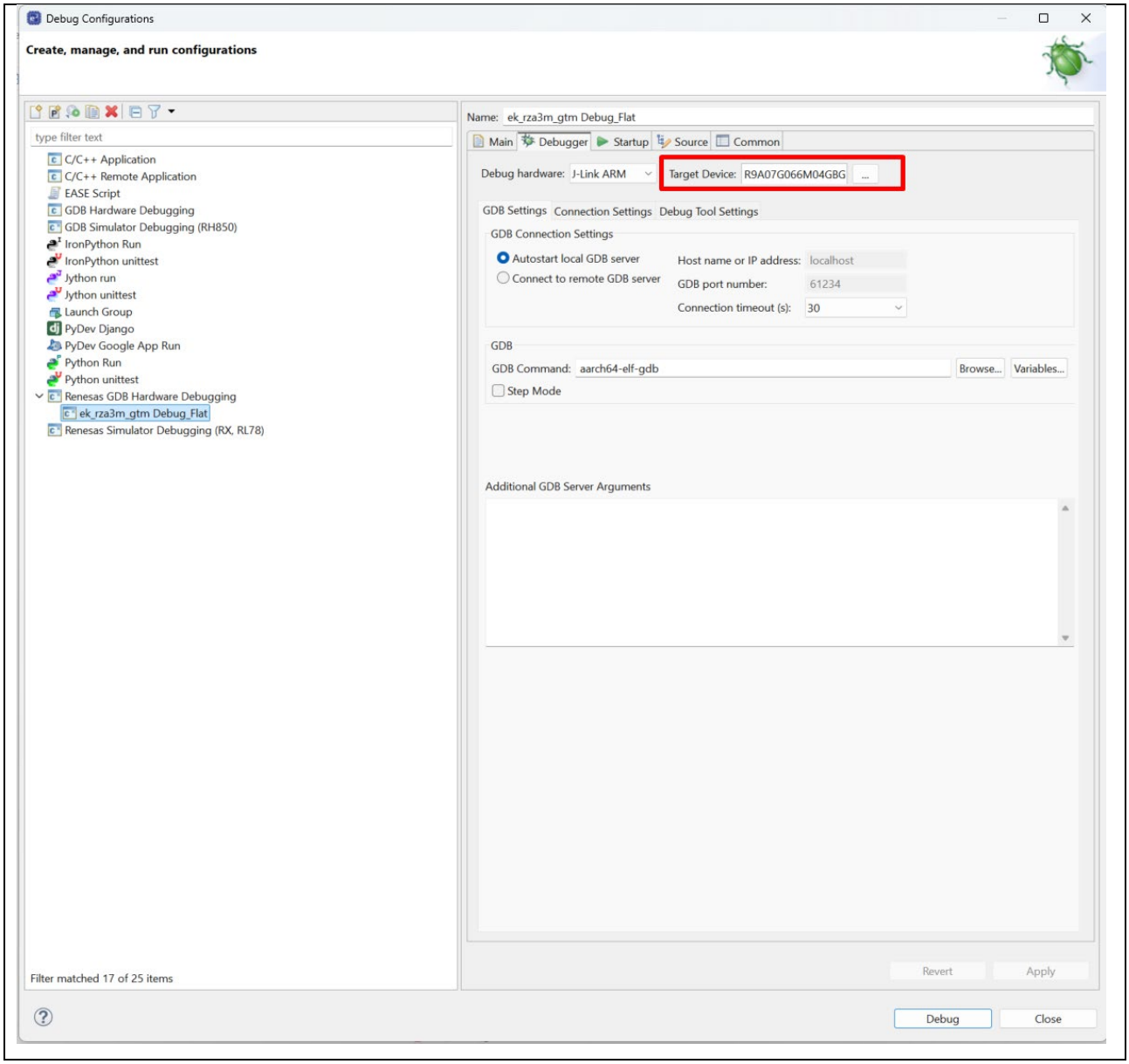


Figure 25. Select the correct target MPU

Note: In case of setting Debug Configuration of projects for specific RZ devices, please refer to the “Debug Steps” part at section “4.5 Debug the Blinky Project” in the “Getting Started Flexible Software Package” documents.

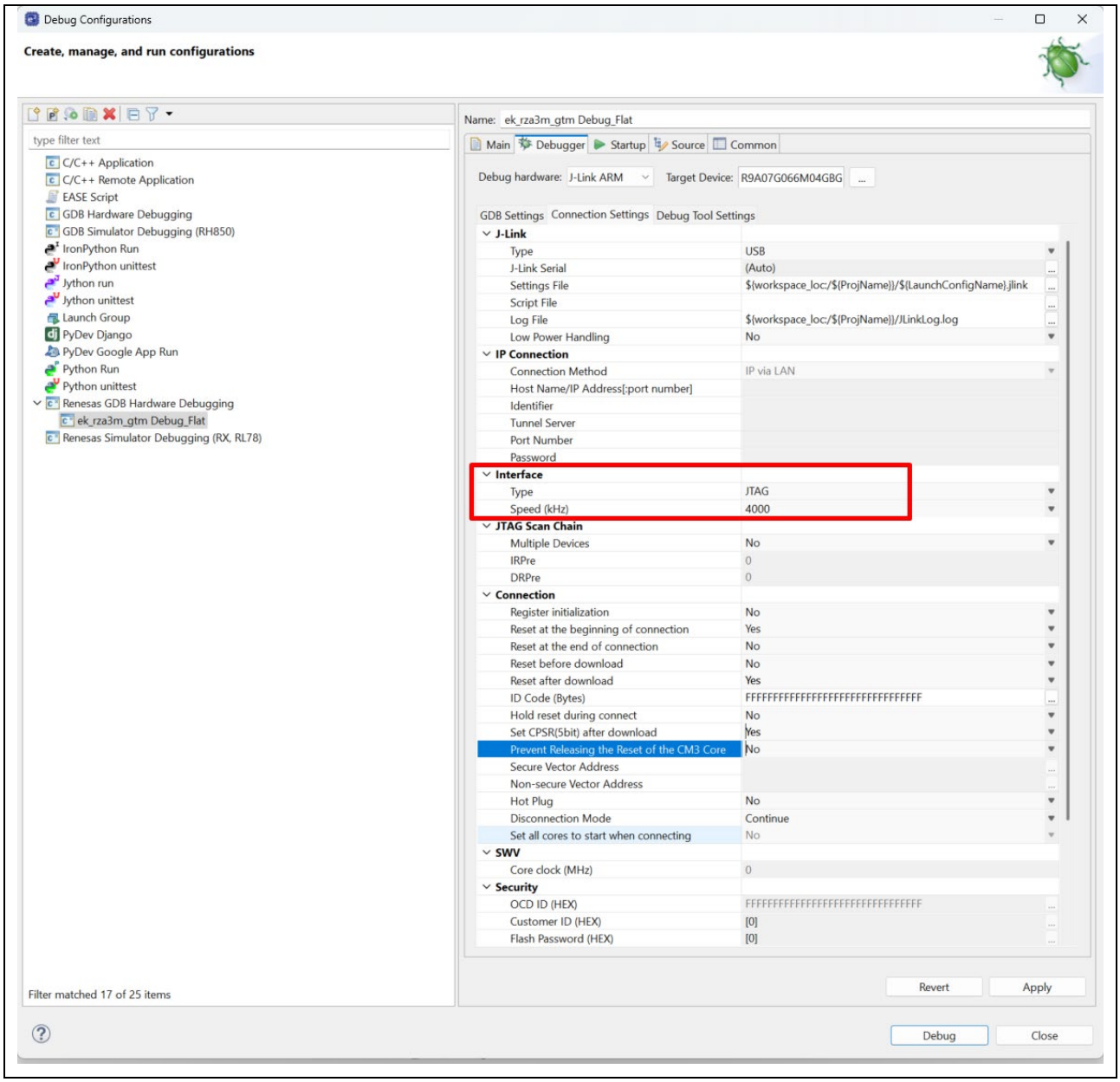


Figure 26. Select the correct Debug Interface

- Select **Apply,** then **Debug**.
- Once downloaded and debugged successfully, click on the **Go** option to run the code.

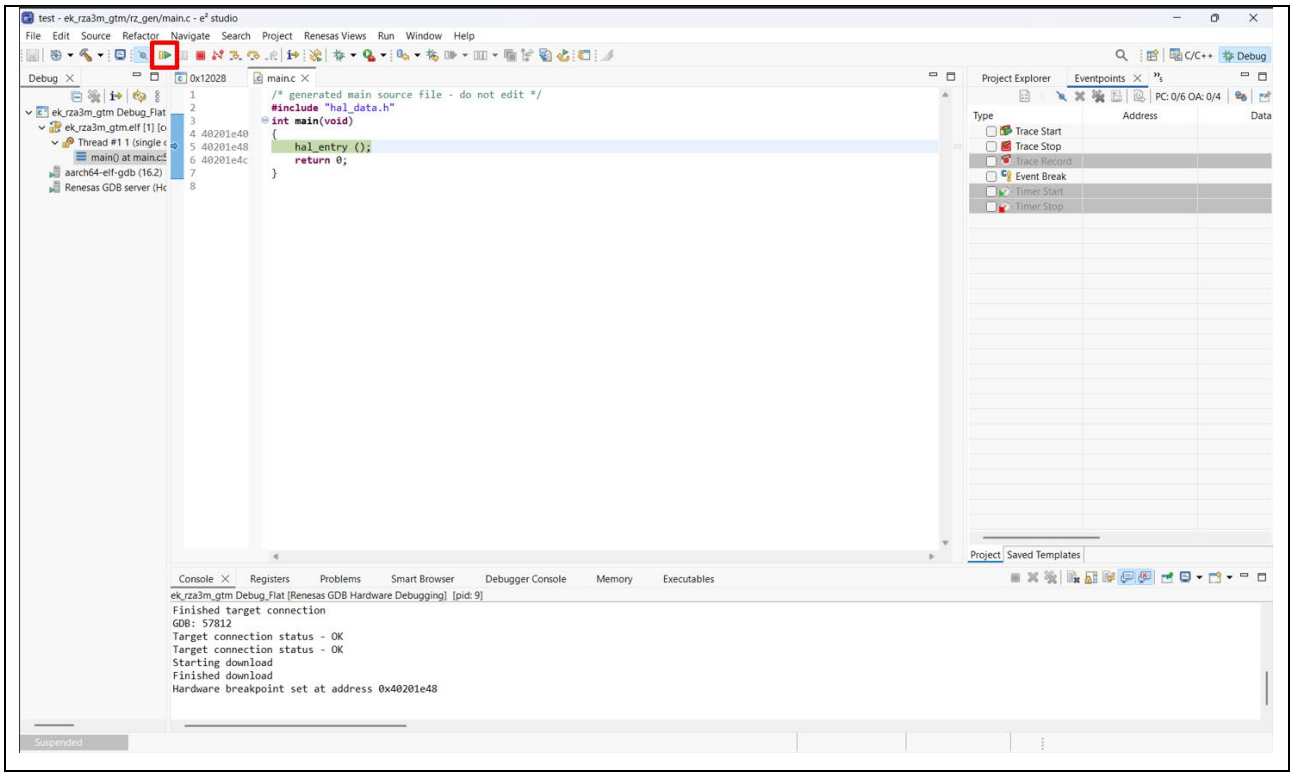


Figure 27. e² studio Debugger Configurations window

3.4.3 Executing Example Project via Binary File and Check Logs

SEGGER J-Link Tools, such as J-Flash, J-Flash Lite, and J-Link Commander, can be used to program the executable binary into the target MPU. Refer to User Manuals UM08001 and UM08003.

- Using J-flash lite to flash the executable binary into the target MPU.

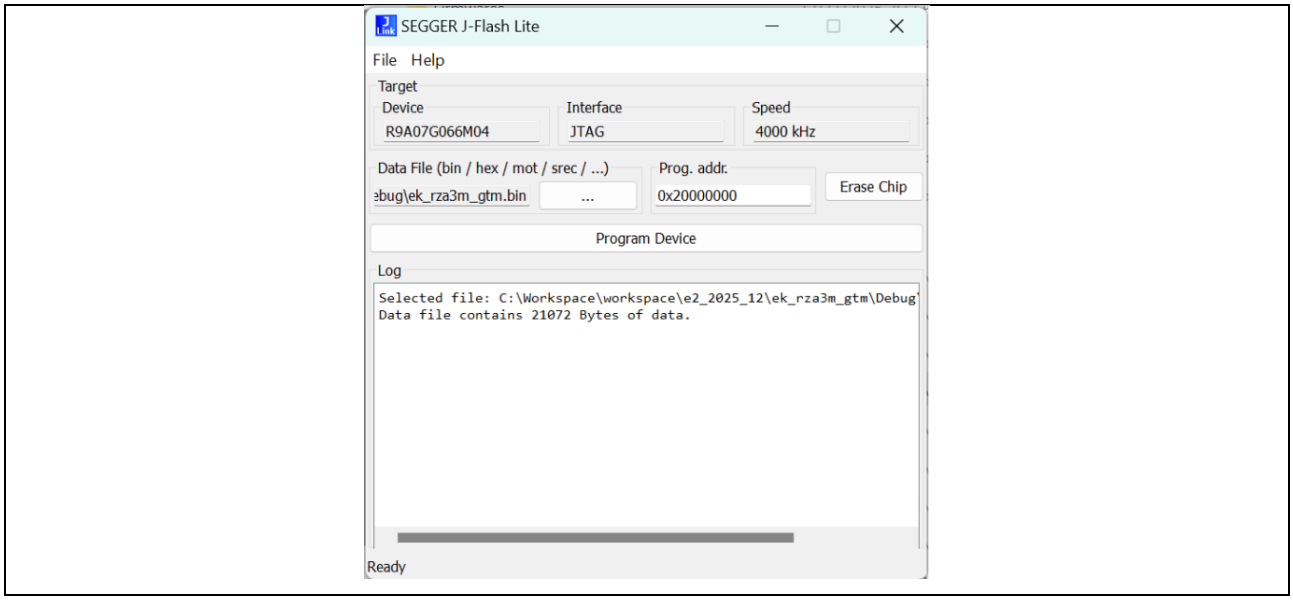


Figure 28. J-Flash Lite Window

Note: Use Tera Term for existing projects that support Terminal Emulator. For existing projects that support J-link RTT, use SEGGER J-Link RTT Viewer. Refer to the following sections for instructions to check the log.

3.4.3.1 Check logs using Tera Term

- Open **Tera Term** by double-clicking **ttermpro.exe** in the downloaded folder.
- Select the serial port.

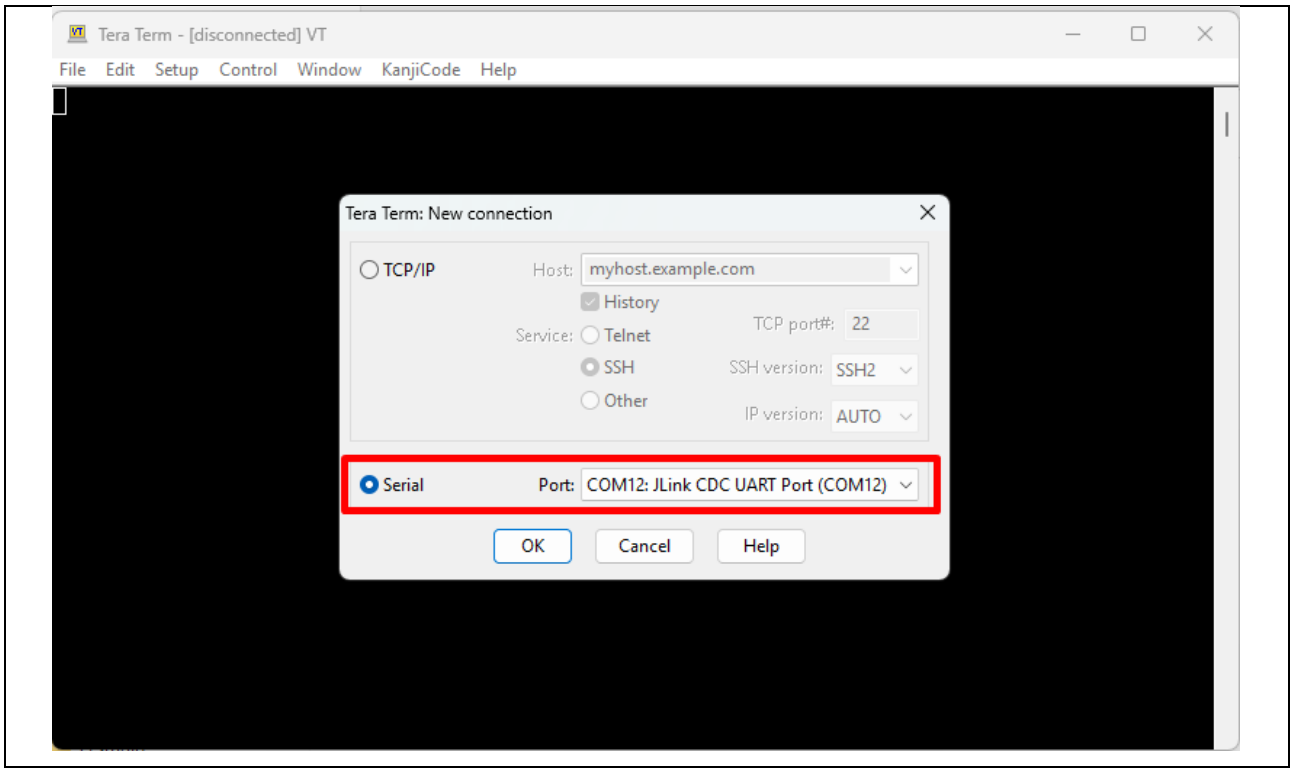


Figure 29. Tera Term New Configuration Window

- Set up Speed.

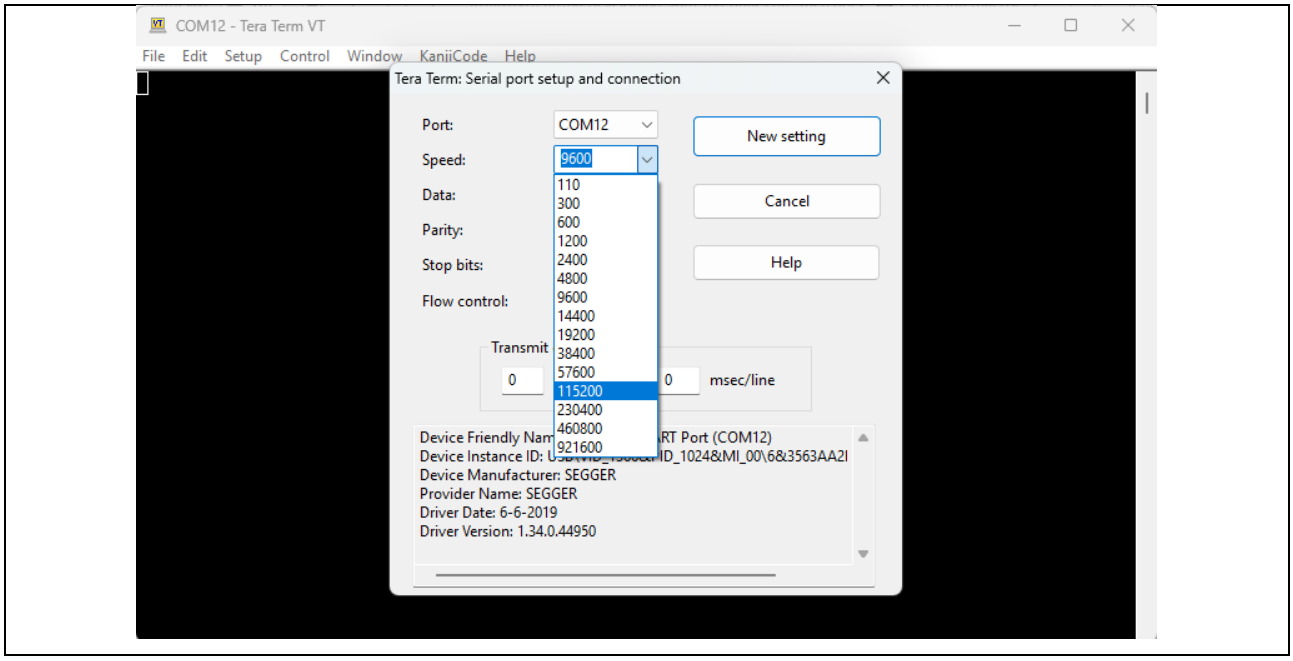


Figure 30. Tera Term Serial Port Setup and Configuration

- Check the log in Tera Term.

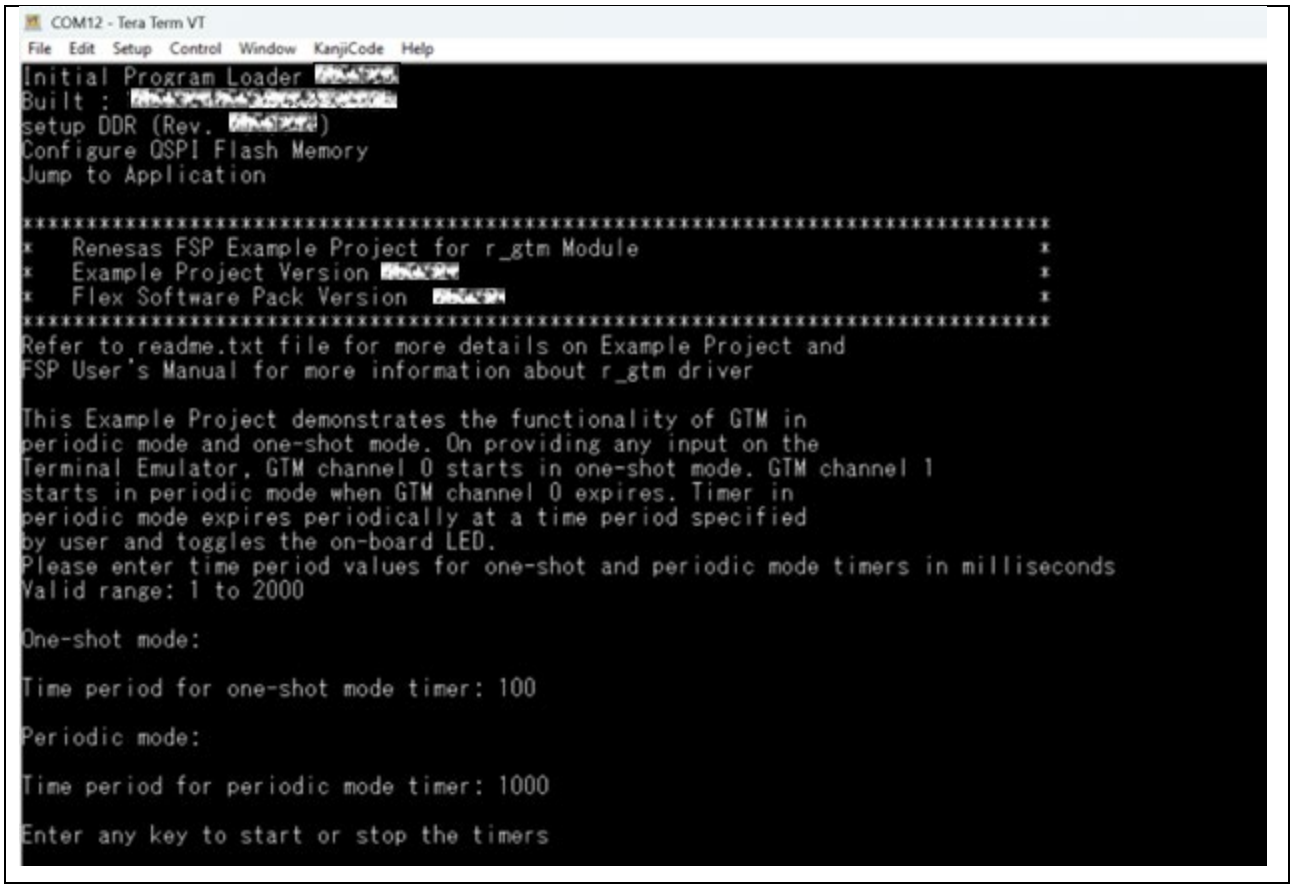


Figure 31. Tera Term Console

3.4.3.2 Check Logs Using J-Link RTT Viewer

To use J-link RTT Viewer to check logs:

- Open **J-LINK RTT Viewer** by double-clicking **JlinkRTTViewer.exe** in the downloaded folder SEGGER\JLink

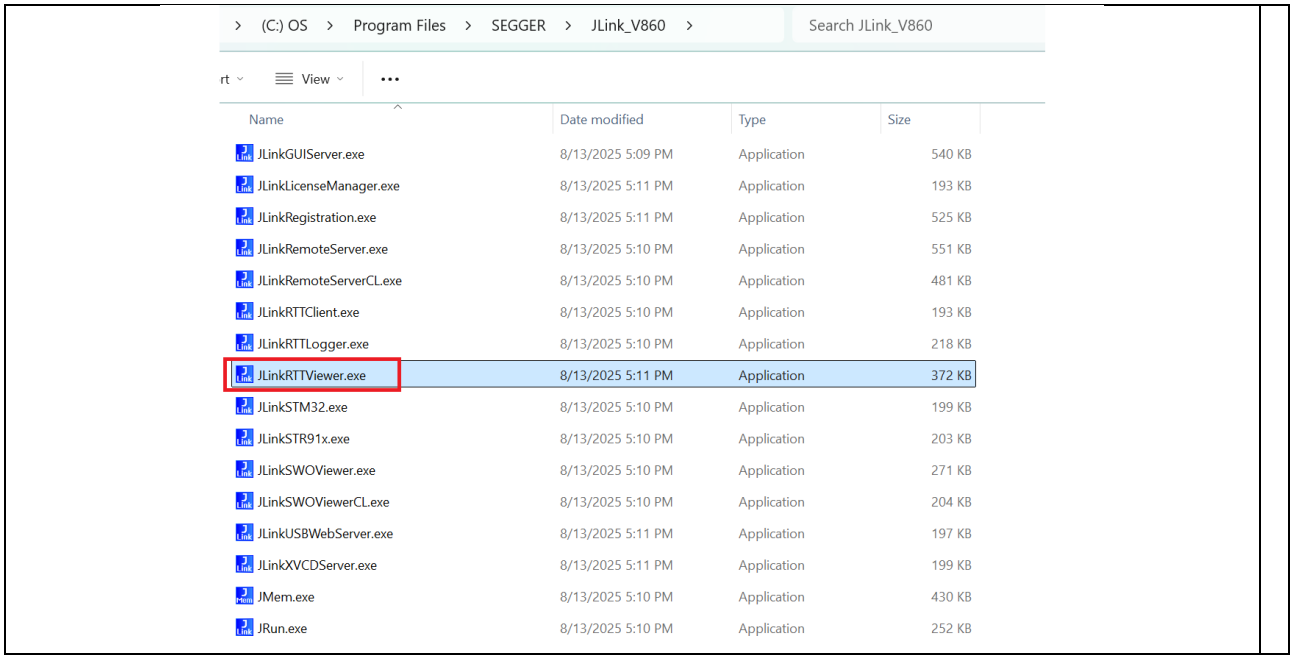


Figure 32. J-Link RTT Viewer tool

- Specify Target Device, click on the "…" tab to select the Renesas RZ device. Then specify Target Interface.

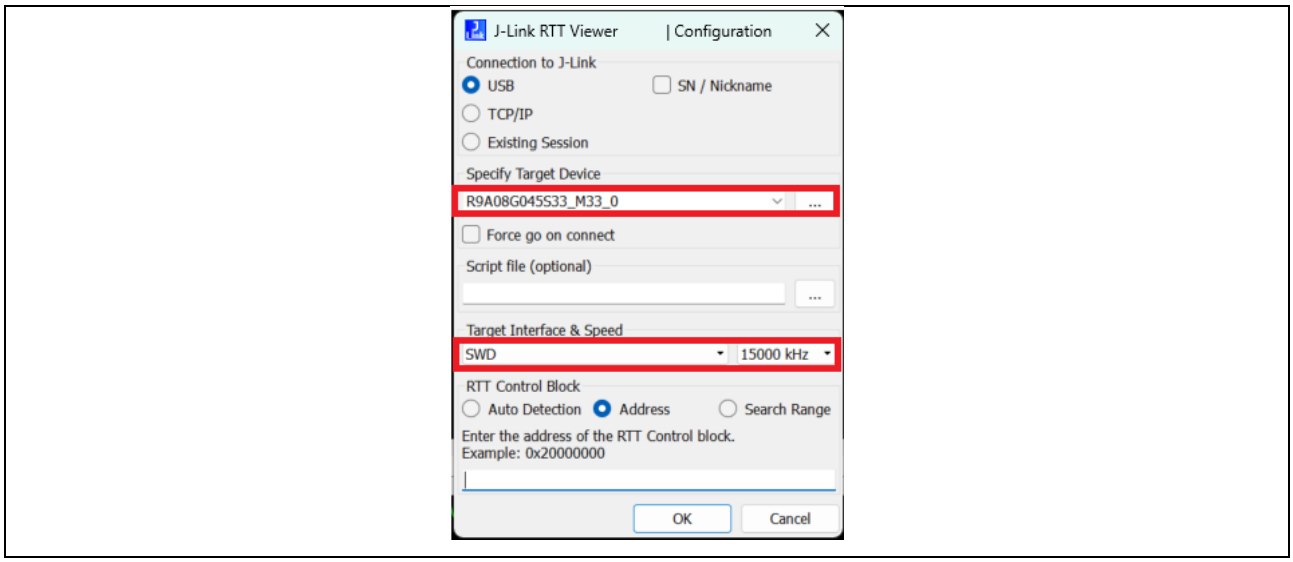
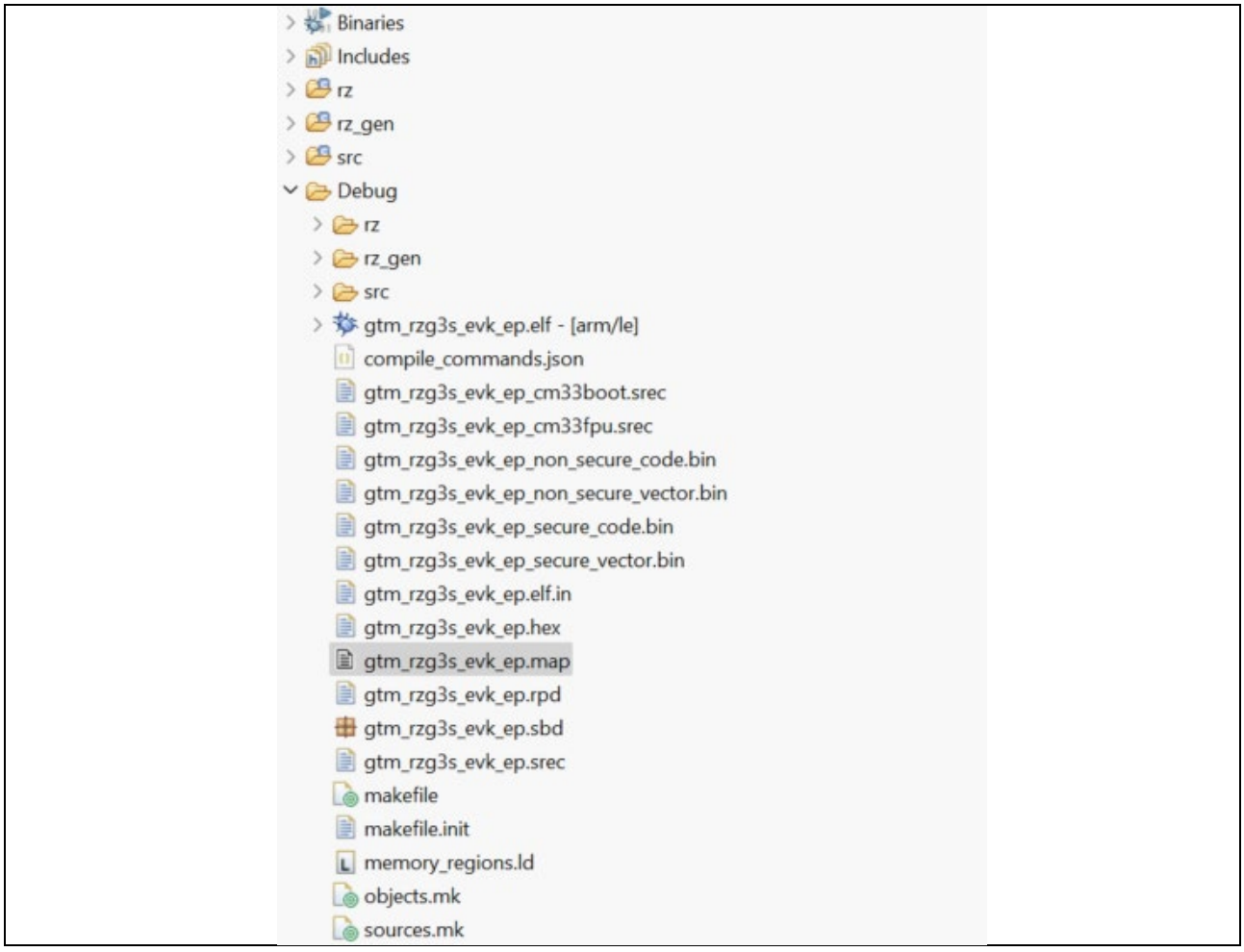


Figure 33. J-Link RTT Viewer Configuration

- To specify the address of the RTT control block, search the _SEGGER_RTT variable in the map file, generated upon successfully building a configuration of the project, which is by default located in the address space for On-chip SRAM.



```
*(.bss*)
.bss._acDownBuffer
                0x00026ca0       0x10 ./src/SEGGER_RTT/SEGGER_RTT.o
.bss._acUpBuffer
                0x00026cb0      0x400 ./src/SEGGER_RTT/SEGGER_RTT.o
.bss._SEGGER_RTT
                0x000270b0       0xa8 ./src/SEGGER_RTT/SEGGER_RTT.o
                0x000270b0                _SEGGER_RTT
```

Figure 34. J-Link RTT Viewer Configuration

- Enter the exact address of the variable into the Address of RTT Control Block.

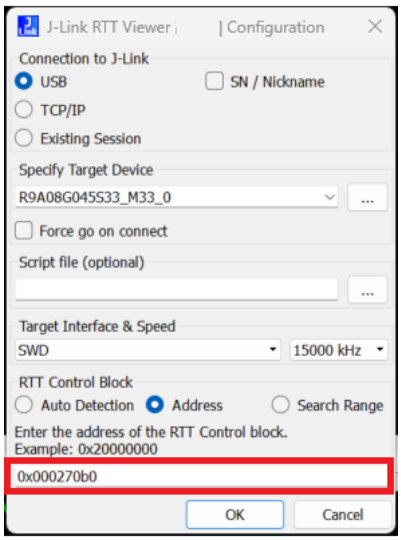


Figure 35. J-Link RTT Viewer Address of RTT Control Block

After connecting successfully, check the log in RTT Viewer.

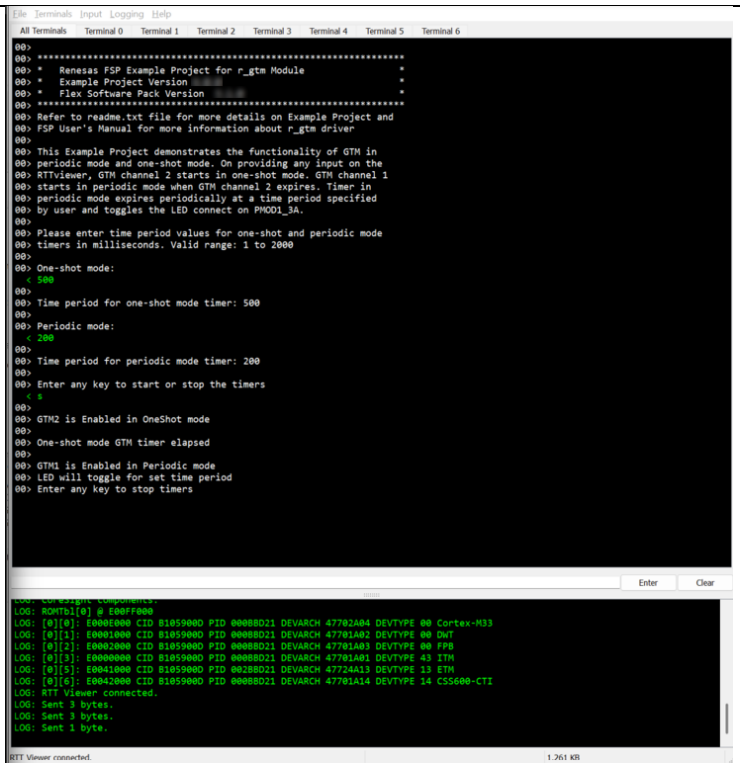


Figure 36. J-Link RTT Viewer Terminals Console

4. Migrating the Existing Project to IAR

A sample project is used to demonstrate the migration procedure. The same approach can be applied to any project that currently uses an existing FSP.

The migration process consists of the following essential steps. These steps are summarized below and explained in detail in the subsequent sections.

Note: IAR Embedded Workbench for ARM (IAR EWARM) includes support for Renesas RZ/T2, RZ/N2 devices.

4.1 Using FSP Smart Configurator with IAR

When using a 3rd-party IDE and toolchain like IAR EWARM, the Renesas FSP Smart Configurator (FSP SC) is a suitable application to configure device hardware, such as clock setup and pin assignment, as well as initialization of FSP software components. FSP SC generates a “Project Connection” file that can be loaded directly into IAR EWARM to update project files.

Note: In case of setting up FSP SC for specific RZ devices like RZ/T2ME, RZ/T2H, and RZ/N2H, please refer to section “5.3.1 Prerequisites” in “RZ/T2, RZ/N2 Getting Started with Flexible Software Package” to apply specific patch files.

4.2 Creating a New Project for the Example Project

The creation of a new Renesas FSP Project is the first step in the migration of an existing project. All information provided below uses the CMT project of RZ/N2L as a reference.

Launch FSP SC, click **File > New > FSP Project**.

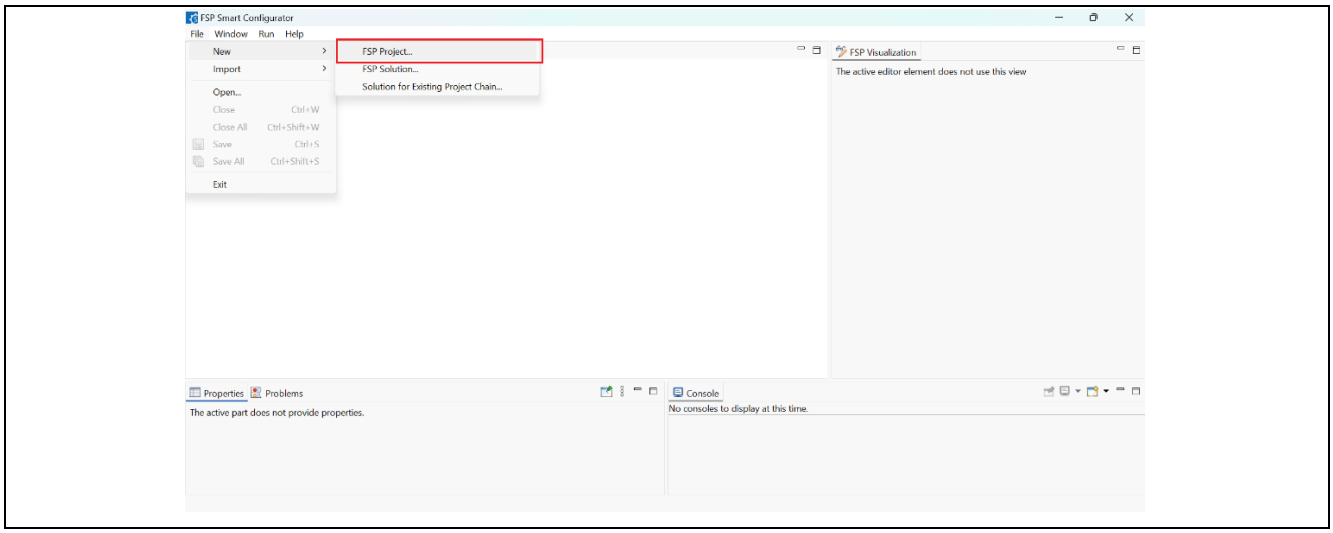


Figure 37. New Renesas FSP project

Assign a name for this new project, such as **cmt_rzn2l_rsk_ep,** then > **Next**.
In the New Renesas RZ Project window, select the **Board** type as RSK+RZN2L (RAM execution without flash memory), choose IAR EWARM [v9.60+] for **IDE project type**, and choose IAR Toolchain for ARM for **Toolchains,** then > **Next**.

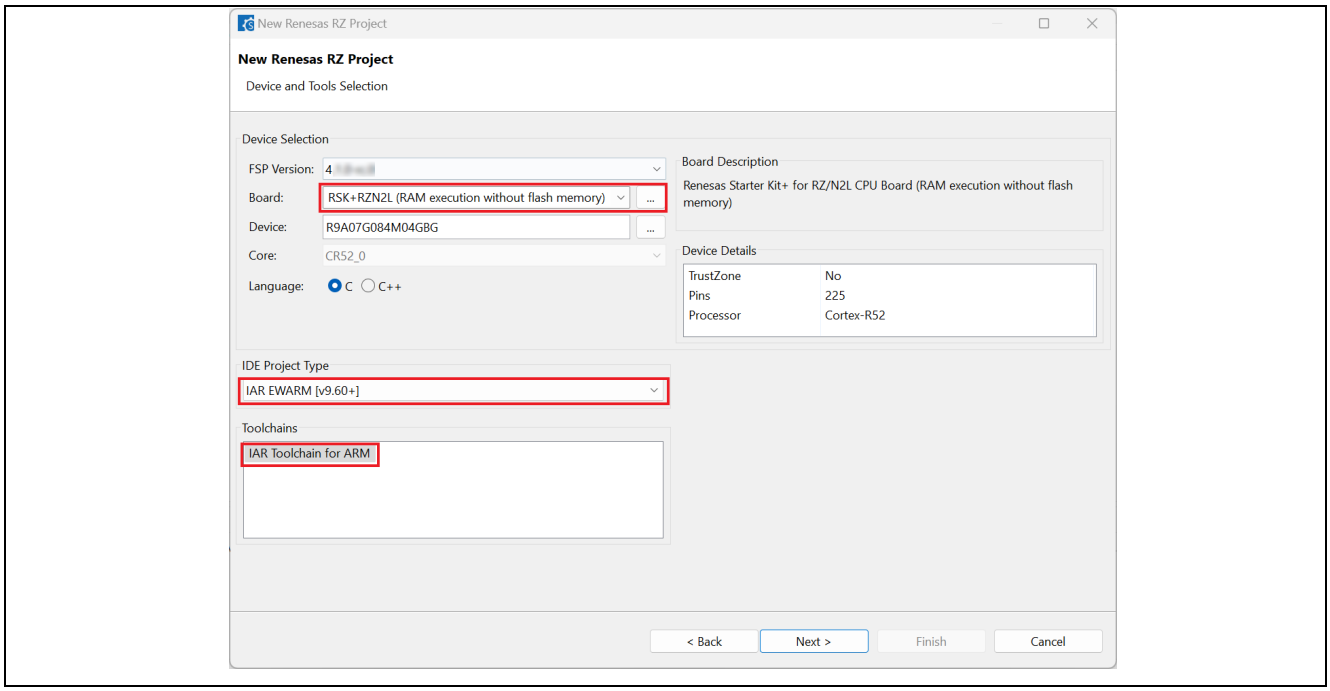


Figure 38. Device and IDE Project Type selection

If an MPU with multiple cores is used, select the **Use Smart bundle** option. In this example, the RZN2L-RSK board has a single core, so simply click **Next**.

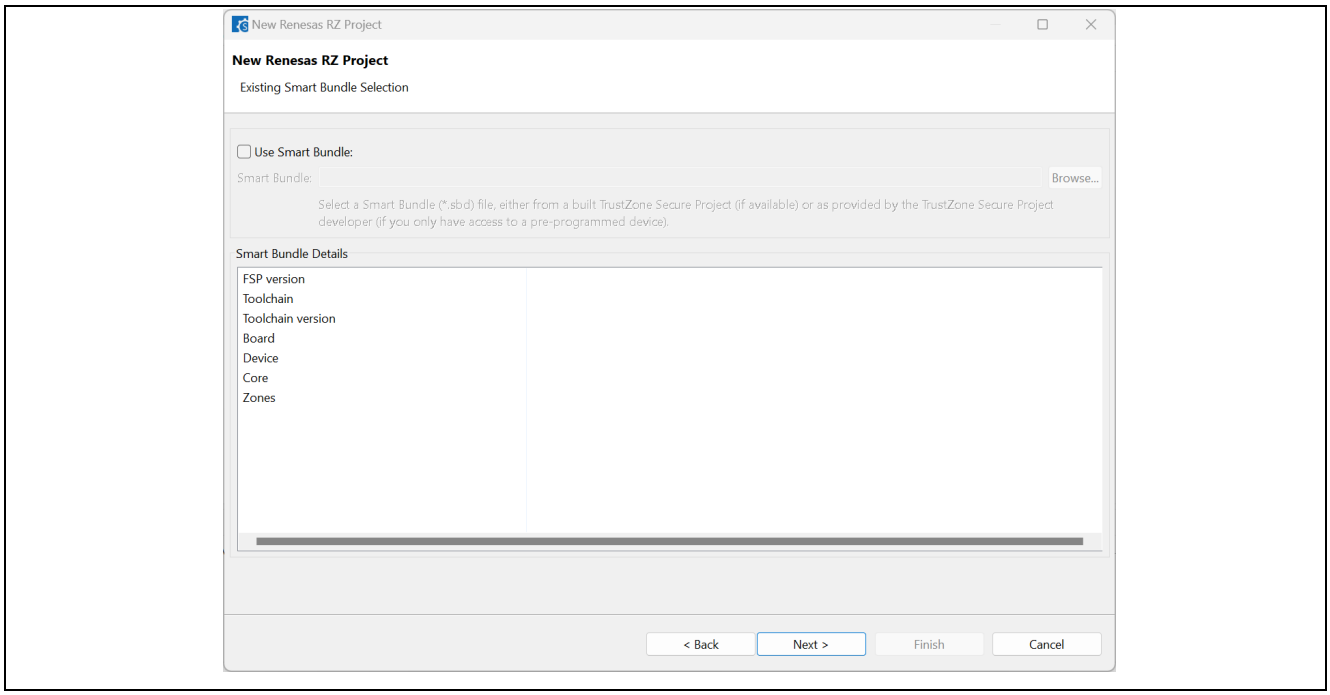


Figure 39. Select project type

This project does not use RTOS, so select **No RTOS** and click **Next**.

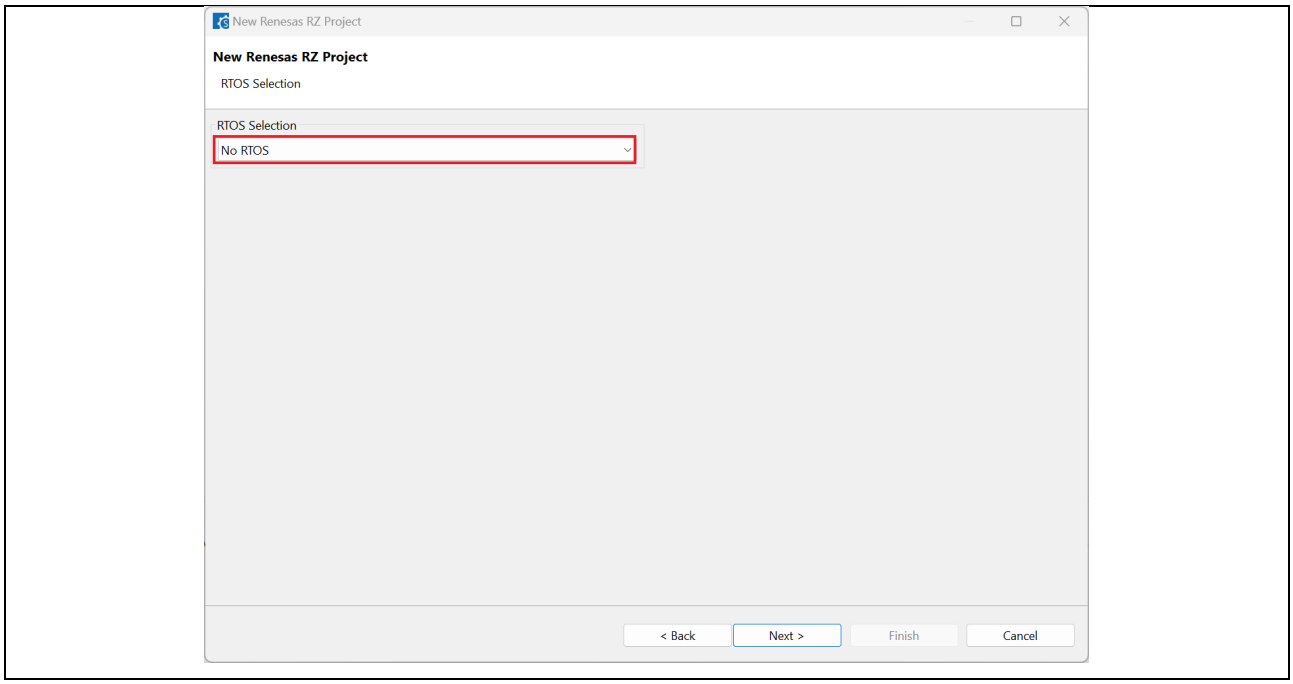


Figure 40. RTOS selection

Select **Bare Metal – Minimal** template for this example and click **Finish**.

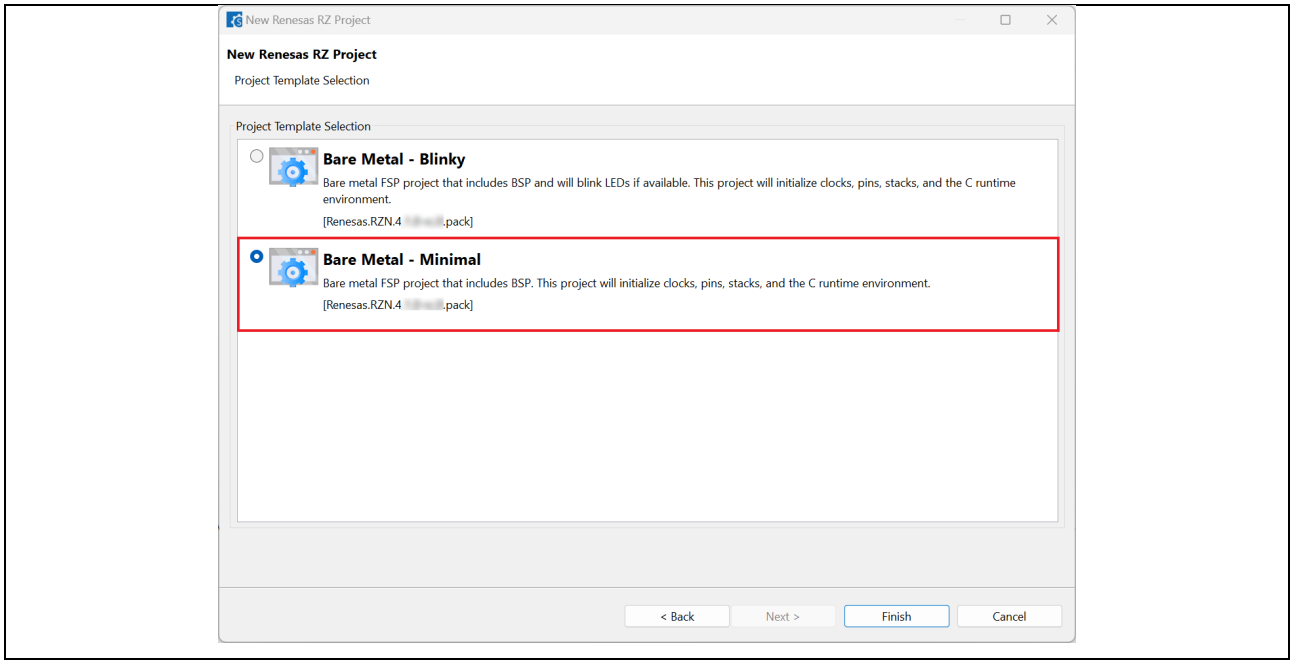


Figure 41. Project Template selection

After completing these steps, the project will be created successfully, as shown in Figure 42.

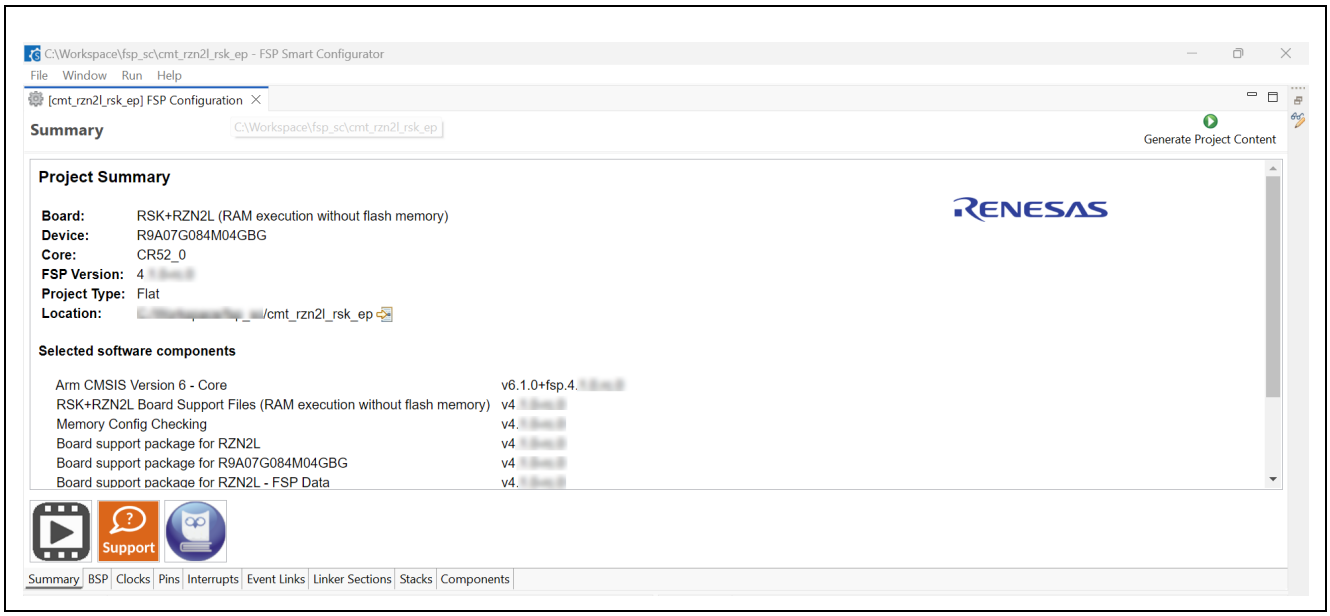


Figure 42. Successful project creation

4.3 Migrate Source Codes and FSP Configurations

4.3.1 Migrate source codes

- If the above steps are completed successfully, the src folder will be available in the workspace, as shown in Figure 43.

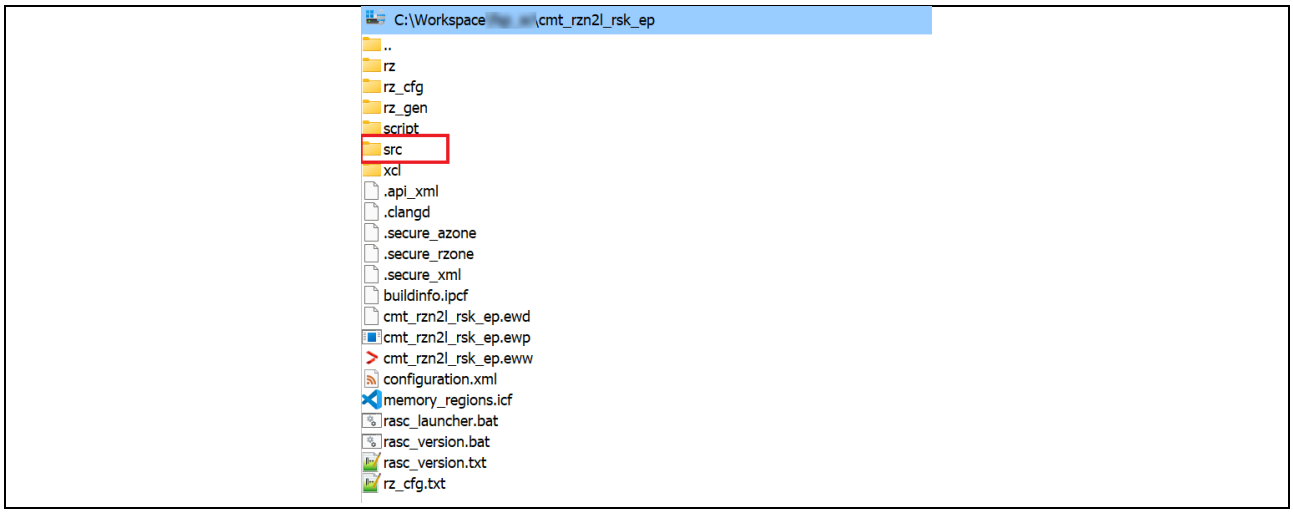


Figure 43. FSP SC Project folder structure

- Update the src folder by copying example code from the existing project's src folder, excluding hal_entry.c and all FSP-generated files.
- In hal_entry.c, update only the source code related to the example project while keeping the generated template. As shown in Figure 9, update only other functions in hal_entry.c, excluding R_BSP_WarmStart() and R_BSP_WarmStart_Stackless() functions.

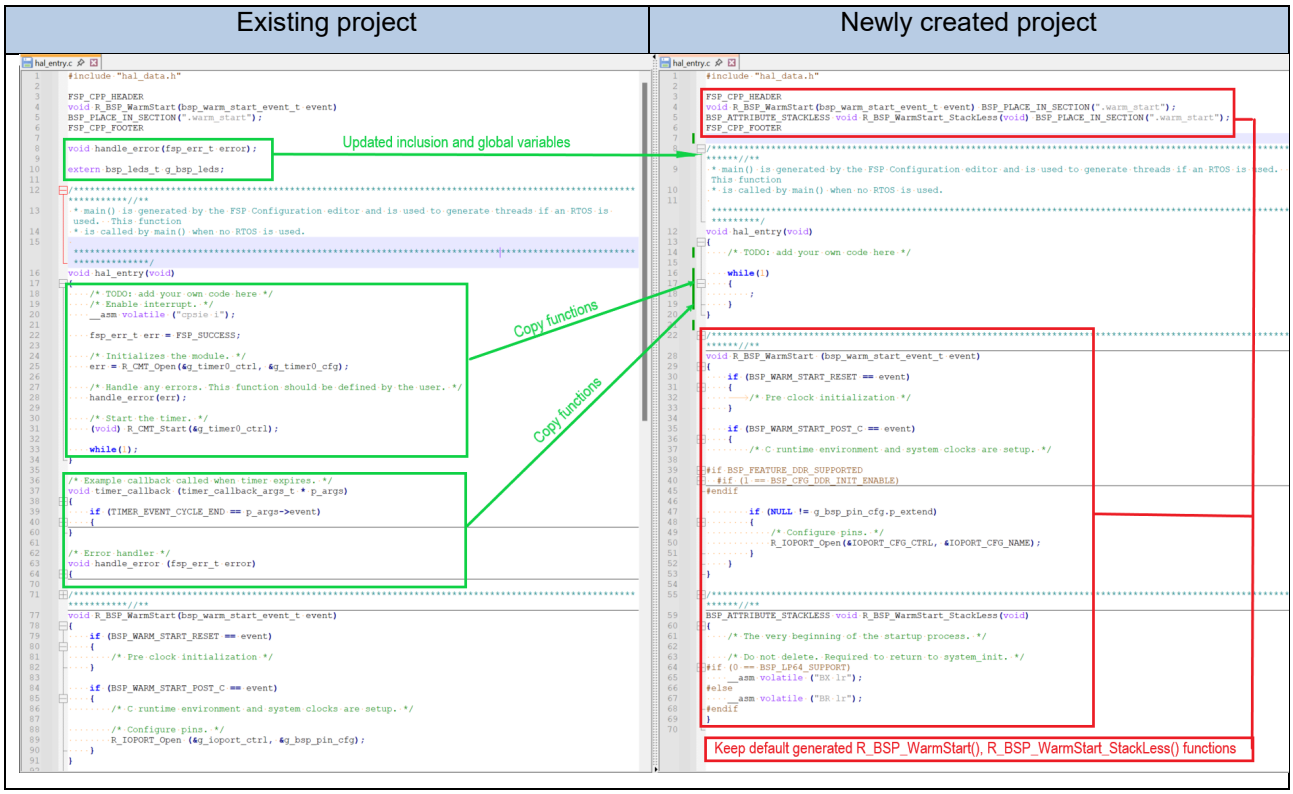


Figure 44. Keep generated R_BSP_WarmStart(), R_BSP_WarmStart_StackLess() functions

4.3.2 FSP Configuration

After adding the source code, the configuration needs to be migrated. The project can be configured in the BSP tab, Clock tab, Pin tab, and Stack tab in the same way as in e^2 studio. Please refer to the **3.2.2 FSP Configuration** learn how to configure the FSP.

The CMT example project of RZ/N2L uses the default configuration of BSP, Clock, and Pin tabs. To configure the Stack tab, please add the following stack:

- **Timer, Compare Match (r_cmt):** Periodic timer
  - **Channel:** 0
  - **Mode:** Periodic
  - **Period:** 1
  - **Period Unit:** Seconds
  - **Callback:** timer_callback

4.3.3 Generating Project Content

Click the **Generate Project Content** button to apply all configured stacks and update the project files accordingly.

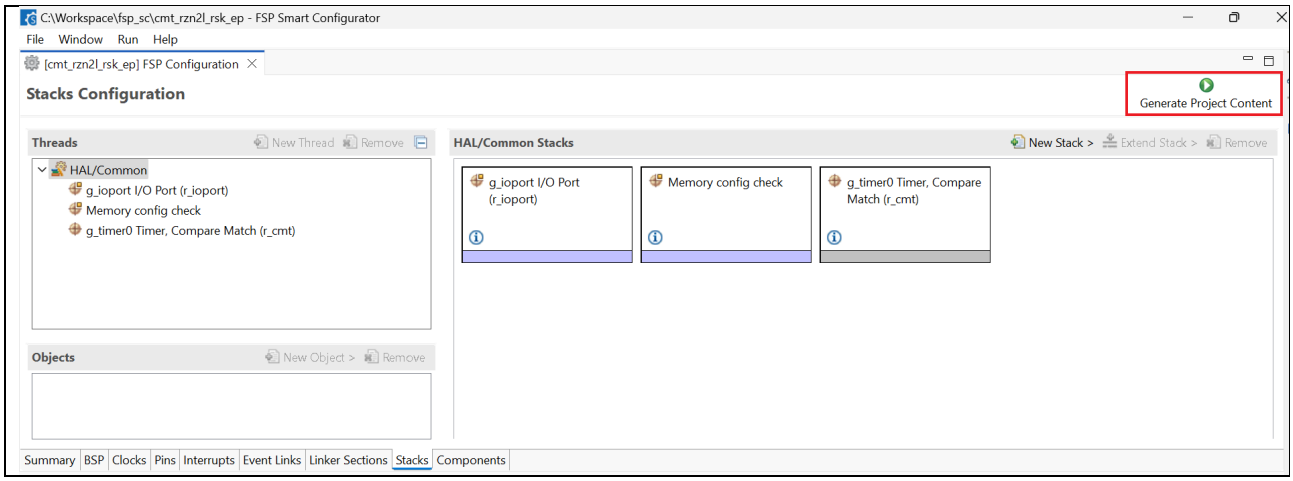


Figure 45. FSP SC Generate Project Content

4.4 Configuring and Compiling the Project

4.4.1 Configuring the Project

Before building the project, make sure the following pre-build operations are completed:

- Module-specific src folder is present inside the project folder directory
- Project files are generated with required configurations (BSP, Clocks, Pins, etc.)

Double-click the IAR EWARM Workspace file (.eww) to open IAR EWARM with the workspace.

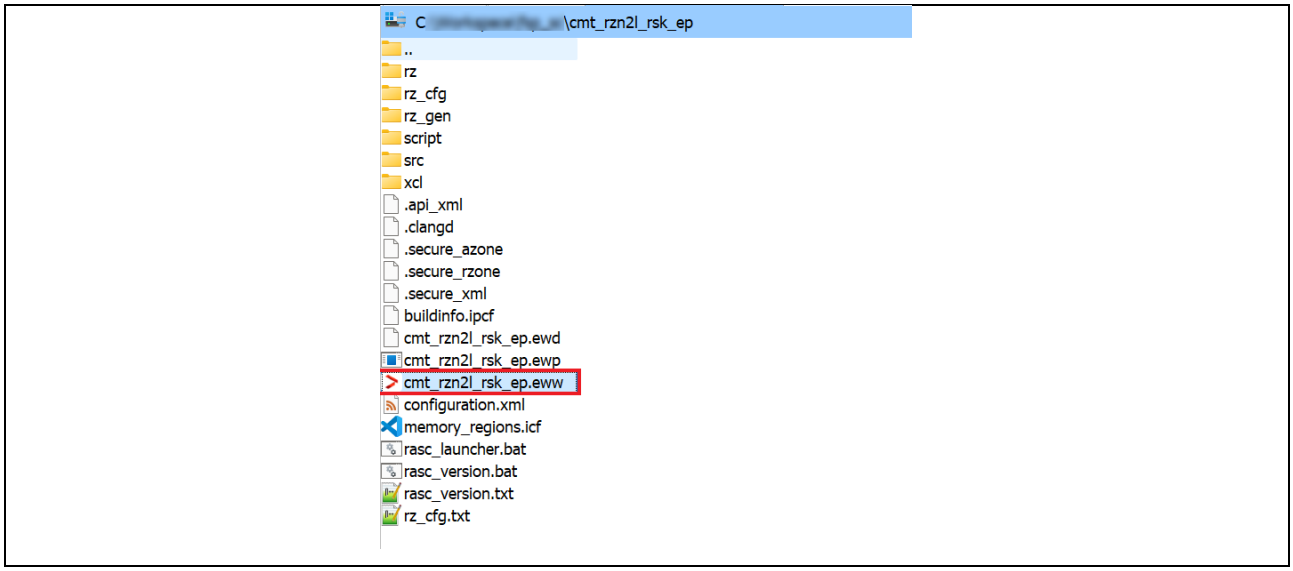


Figure 46. FSP SC Project Workspace

1. Click on **Project** and then click on **Option**…to open the project option window.
2. Select the **General Options** category and select the **Target** tab. Confirm the device name at **Device** match with the selected device in project creation.

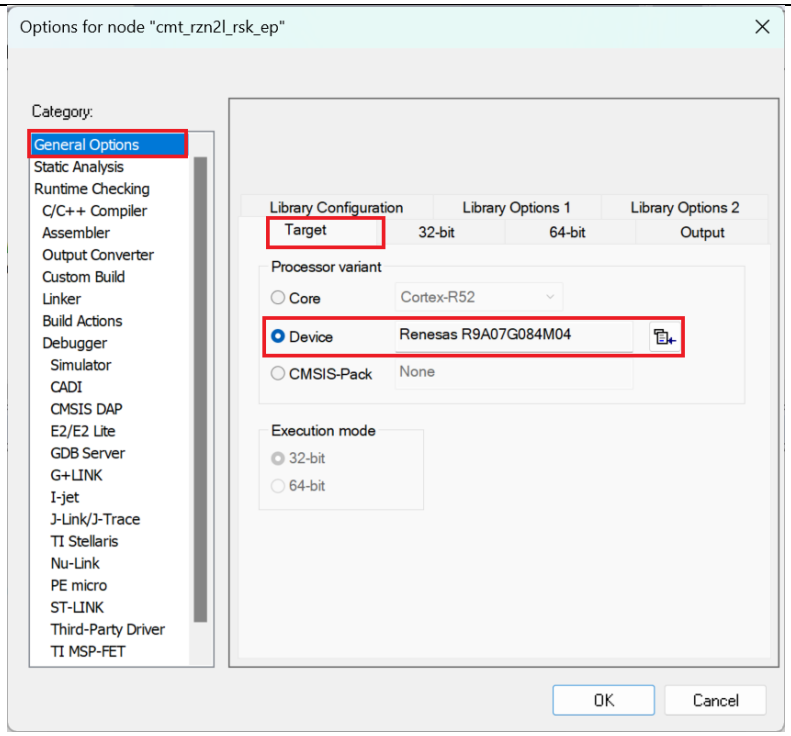


Figure 47. IAR General Options

3. Click **Debugger** > **Setup** and select I-jet of **Driver**.

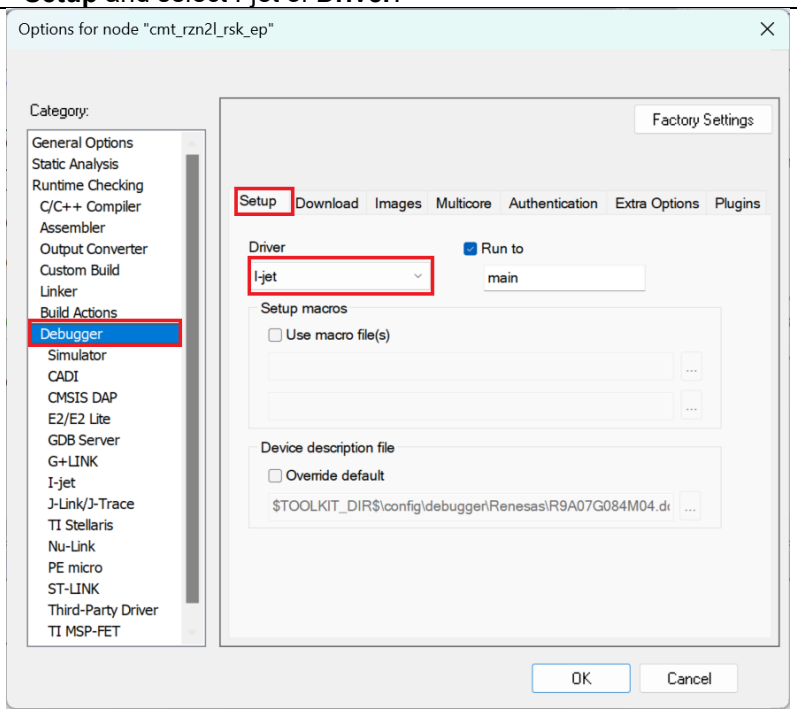


Figure 48. IAR Debugger Options

4. Click **Linker** > **Extra Options** and add "--redirect Reset_Handler=system_init" to **Command line options (one per line)**:

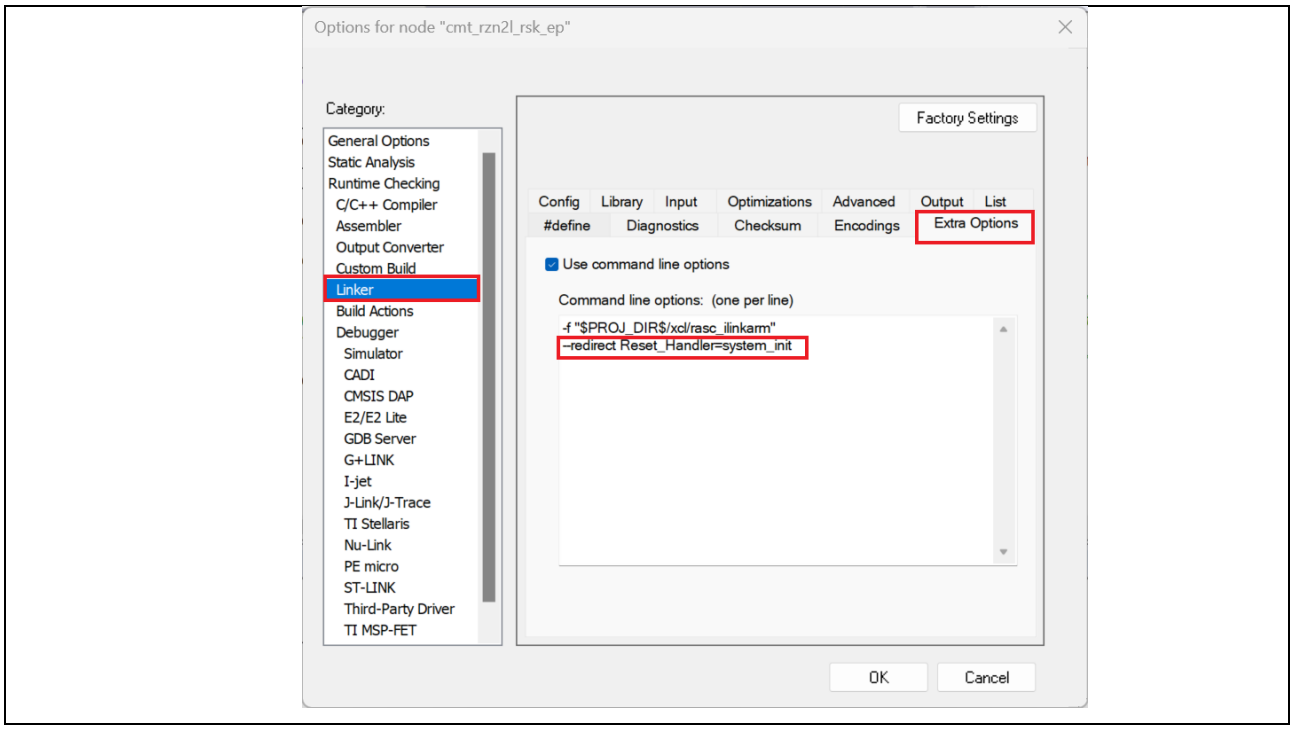


Figure 49. IAR Linker Extra Options

5. Click on **Project -> Make** from the menu bar or the **Make** button on the toolbar to build.

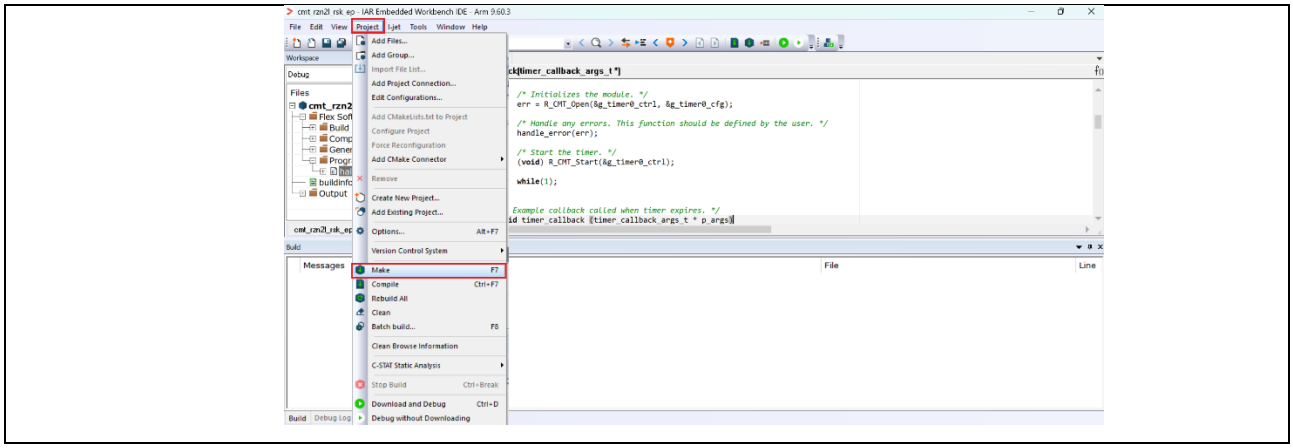


Figure 50. IAR Build project

Once the build is completed, the build message is displayed in the Build Console window, which displays compilation target files and the number of errors/warnings.

4.4.2 Addressing Warnings and Errors in Build Operation

On completion of the Generating and Building Procedure of the Project. The successful or failed operation can be observed in the Build logs in the Console window.
Warnings or errors like those encountered during migration in e² studio may also occur during this process. For guidance on identifying and resolving these issues, please refer to section 3.3.2 Addressing Warnings and Errors in Build Operation.

4.5 Downloading and Debugging the Project

4.5.1 Debug Prerequisites

To debug the project on a board, the following are required:

- The board must be connected to a **PC**.

- The debugger I-Jet must be configured to communicate with the board.
- The application must be programmed onto the microprocessor.

Applications run from the internal RAM of the microprocessor. To run or debug the application, it must first be programmed into RAM using a JTAG debugger. The evaluation board includes a JTAG header and requires an external JTAG debugger (I-Jet) connected to this header.

4.5.2 Download and Debug the Project

Before debugging, make sure there are no errors, and all the build warnings have been reviewed. Confirm that the target board is connected to the host system.

- Click on **Project -> Download and Debug** from the menu bar or the **Download and Debug** button on the toolbar to download and debug.

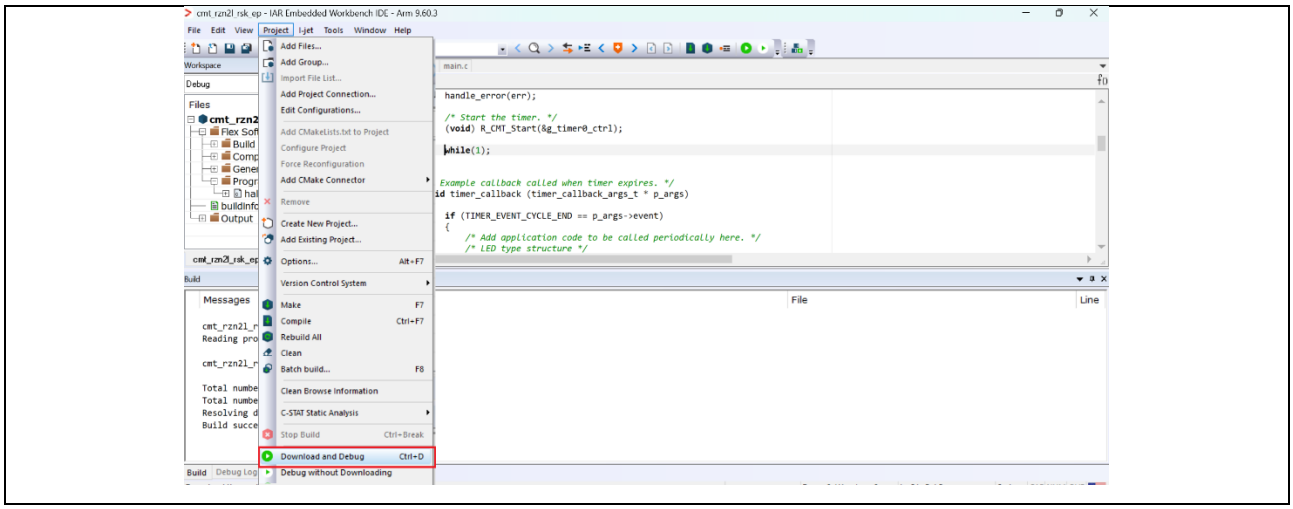


Figure 51. IAR Download and Debug

- Once downloaded and debugged successfully, the program breaks at the beginning of **main** in **main.c**. Click on **Debug->Go** from the menu bar or the **Go** button on the toolbar to run the project.

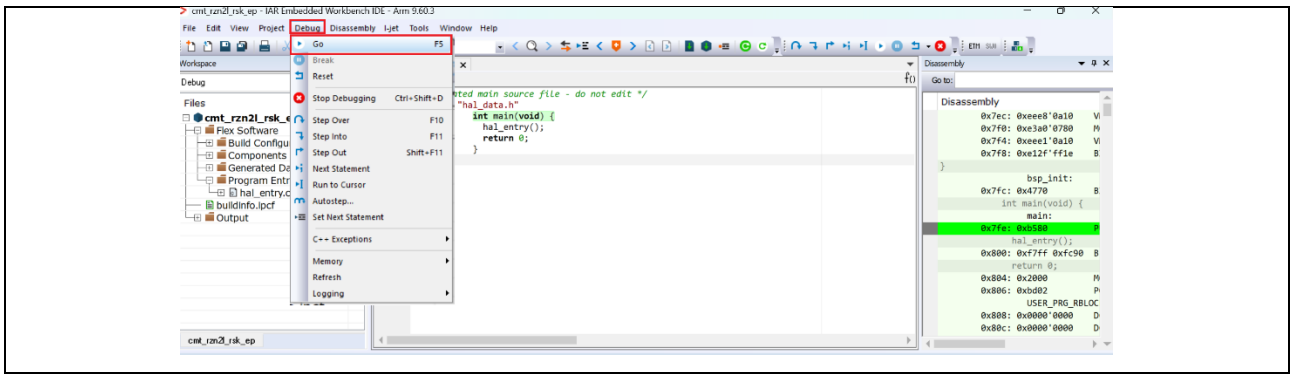


Figure 52. IAR Go button

5. Reference Documents

Getting Started Flexible Software Package document of each RZ series:

- RZ/A: RZA Getting Started Flexible Software Package
- RZ/G: RZG Getting Started Flexible Software Package
- RZ/T2, RZ/N2: RZT2 RZN2 Getting Started Flexible Software Package
- RZ/V: RZ/V Getting Started with Flexible Software Package

Example Projects of Renesas RZ MPU devices:

- RZ-FSP-Examples: renesas/rz-fsp-examples

Revision history

Rev.	Date	Description	
		Page	Summary
1.00	Feb.27.26	-	Initial release.
1.10	May.20.26	p.1	Added IAR and FSP SC at “Development tools and software”.
		p.25	Added new chapter “4 Migrating the Existing Project to IAR”.

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.

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

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/.