Renesas RA Family

Converting Applications from e² studio to IAR or Keil

Introduction

Customers developing products with Renesas RA MCUs have the option of working with a variety of Integrated Development Environments. This application note provides guidelines to create projects in IDEs such as IAR Embedded Workbench for ARM and Keil Microcontroller Development Kit - ARM while using projects in Renesas e^2 studio IDE as reference. This process is here after referred to as "IDE migration". These guidelines apply to most example projects and application projects available on the Renesas website. Upon completion of the recommended steps in this document, the reader is advised to perform thorough testing to compare operation between the resulting project and the original e^2 studio project running on the RA MCU due to differences in underlying compilers and corresponding options selected.

Required Resources

Development Tools and Software

- e^2 studio IDE v2021-10 or later (link)
- RA Standalone Configurator (RASC) v2021-10 or later (link)
- IAR Embedded Workbench for ARM v8.50.9 or later (link)
- Keil MDK 5.34 or higher (link)
- Flexible Software Package v3.1.0 or later (link)

Prerequisites and Intended Audience

This application note assumes that you have some experience with the Renesas e^2 studio IDE and RA Flexible Software Package (FSP). Before you perform the procedures in this application note, follow the procedure in the FSP User Manual to build and run the Blinky project. Doing so enables you to become familiar with the e^2 studio and the FSP, and also validates that the debug connection to your board functions properly.

The intended audience for this application note are users who want to extend example project and application project support typically released with e^2 studio and GCC to other IDE and compilers.

Note: If you are a first-time user of e^2 studio and FSP, we highly recommend you install e^2 studio and FSP on your system in order to run the Blinky Project and to get familiar with the e^2 studio and FSP development environment before proceeding to the next sections.

Prerequisites

- Access to latest documentation for identified Renesas Flexible Software Package
- Prior knowledge of operating e^2 studio and built-in (or standalone) RA Configurator
- Prior knowledge of operating IAR Embedded Workbench and IAR Compiler
- Prior knowledge of operating Keil MDK and ARM Compiler

Contents

1. Overview

The process of IDE migration is performed by following common needed steps for both IAR EW and Keil MDK. These steps, mentioned below, are described in detail in the upcoming sections.

1. IDE configuration and component installation
    1. RASC Installation and registration
    2. FSP Installation
    3. Patch/DFP installation
2. Creating a Blank Template Project
    1. Create blank project
3. Add Source code
    1. Add Source files
    2. Generate FSP auto-generated files
4. Compiling the Binary
    1. Adjust compiler options (include paths, compiler flags, linker flags, etc.)
    2. Build Project
    3. Addressing warnings and errors
5. Download and debug binary
    1. Debugger setup
    2. Verifying download

1.1 RASC Installation

The available RASC executable can be run directly. Follow the steps below to install.

1. Once you install RASC and begin setup, you must select **All Users** if you would like to install to a different location. If you select **Current user**, you might not be able to install to a different location.

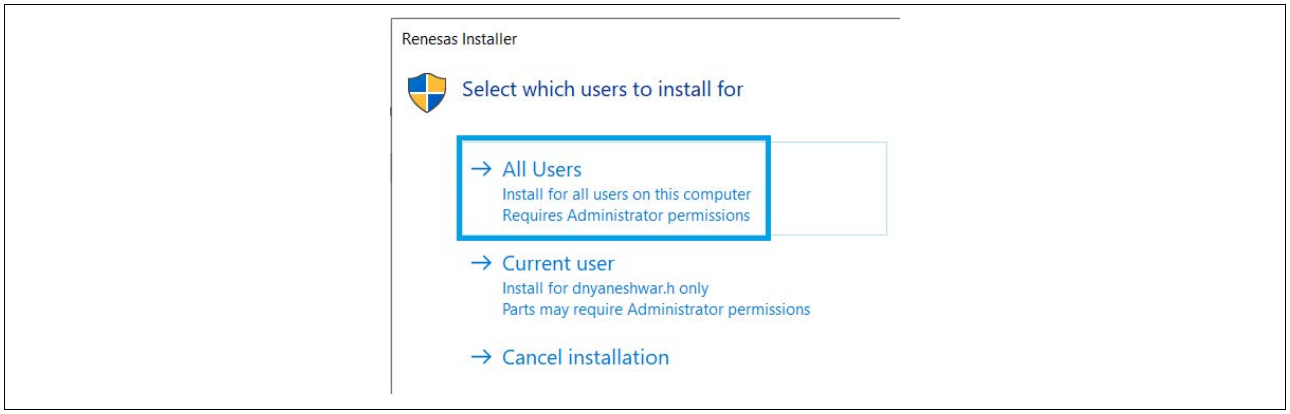

Figure 1. Install for All Users

2. Select **Install** (Install to a different location) to create a separate instance for new RASC. Then click on **Next** to begin the nstallation.

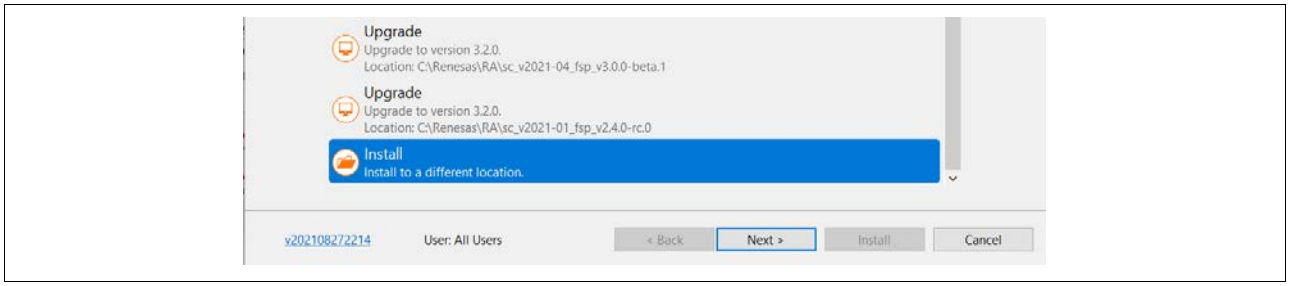

Figure 2. Select Install

The process will check the necessary prerequisites for the installation to complete. By default, a separate installed folder is created in the `C:\Renesas\RA\` directory but this can be changed to any desired location.

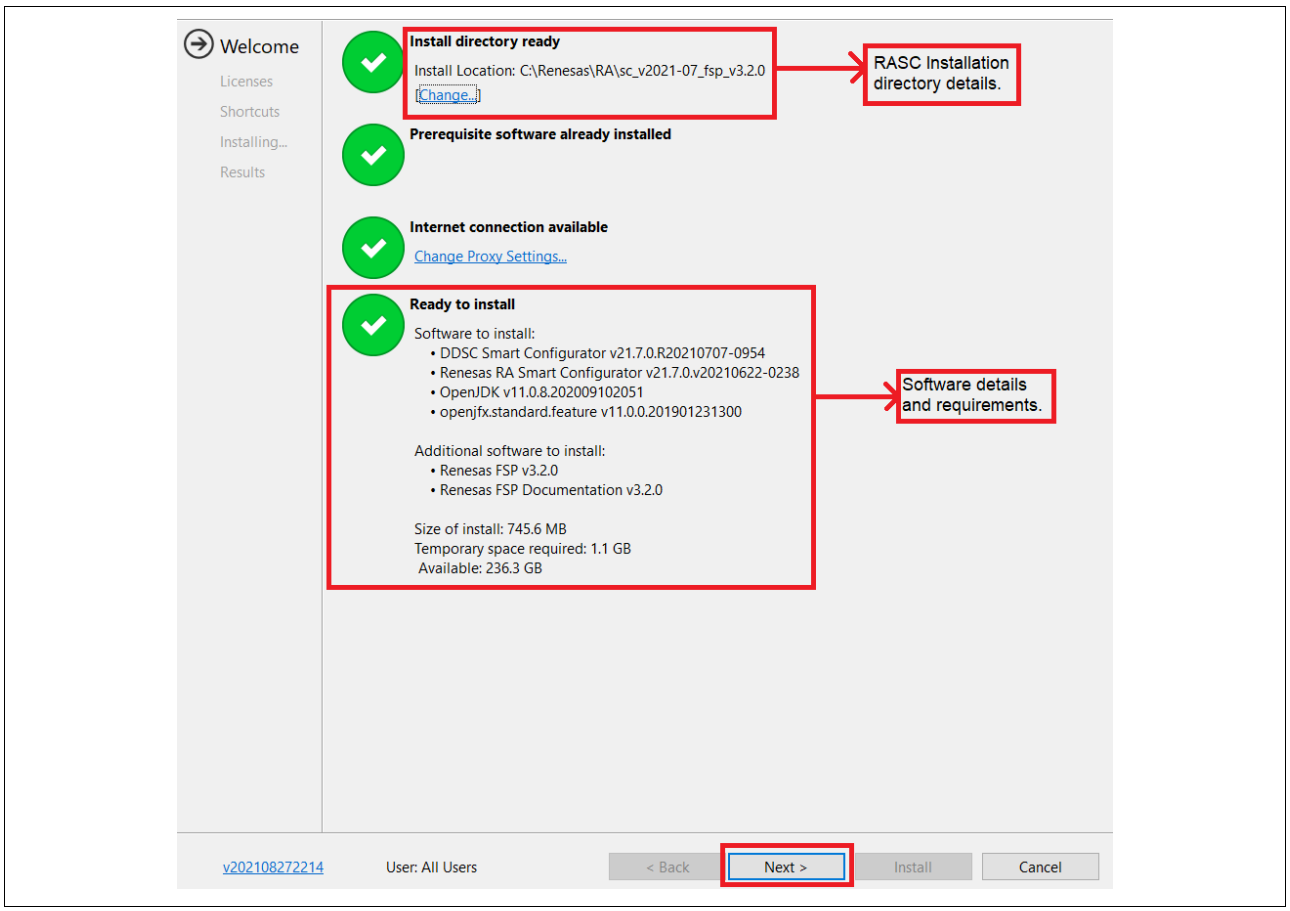

Figure 3. Select Install Directory

3. Click on **Next** to move to further steps.

4. Click on the checkbox to agree with the terms and conditions then click the **Install** button to start the installation. Below is the appearance after clicking the final **Install.**

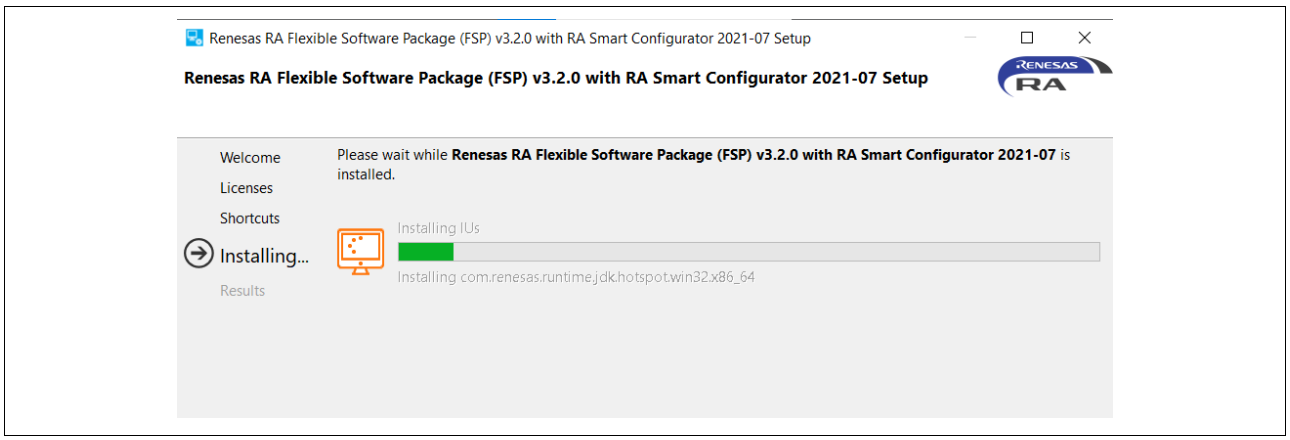

Figure 4. Installing

5. Locate the installation folder and check for the newly installed RASC folder for the desired FSP version. It will have folders as shown in Figure 5 (installed as a new instance in default directory):

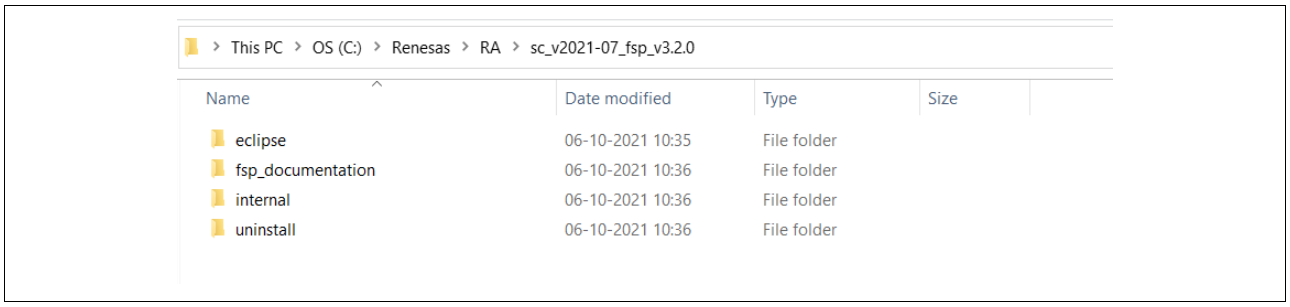

Figure 5. Installation Folders

Note: Newly installed packs should be present inside the internal directory in the RASC folder as shown in Figure 6:

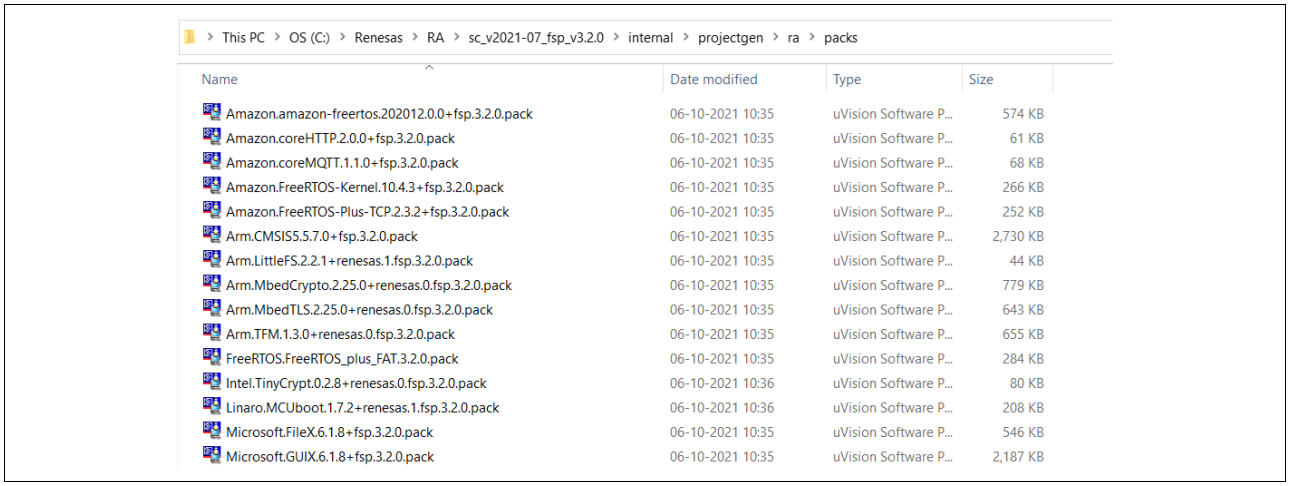

Figure 6. Internal Directory

1.2 FSP Installation

This step is common for migration of projects in both Keil and IAR IDEs.

This can be achieved in following possible ways:

- The new RASC can be downloaded directly with pre-installed packs of the specific version.
- The packs are present in `<rasc installation directory>\sc_v2021-04_fsp_v3.2.0\internal\projectgen\ra\packs`
- The FSP packs(pack files) can be directly placed inside currently installed RASC pack folder which is: `<installation directory>\sc_v2021-04_fsp_v3.2.0\internal\projectgen\ra\packs`

2. Migrating e² studio projects to IAREW/IARCC

2.1 IAR IDE configuration and component installation

User may use the reference document available in the latest FSP User Manual → Starting Development → Using RA Smart Configurator with IAR EWARM

https://renesas.github.io/fsp/_s_t_a_r_t__d_e_v.html#using-ra-smart-configurator-with-ewarm

2.1.1 IAR IDE and RASC Configuration

Requirements:

- IAR EW for ARM (current version used: 8.50.9)
- Flex Software Package (FSP) (current version used: 3.1.0)
- RA Smart Configurator (RASC) (current version used: 3.1.0)

Steps to configure RASC for installed IAR EW software:

Open the installed IAR Embedded Workbench. Go to **Tools** → **Configure Tools...**

1. When you click the **Configure tools** → **New** option, the following window appears. Configure the related fields as shown in Figure 7.

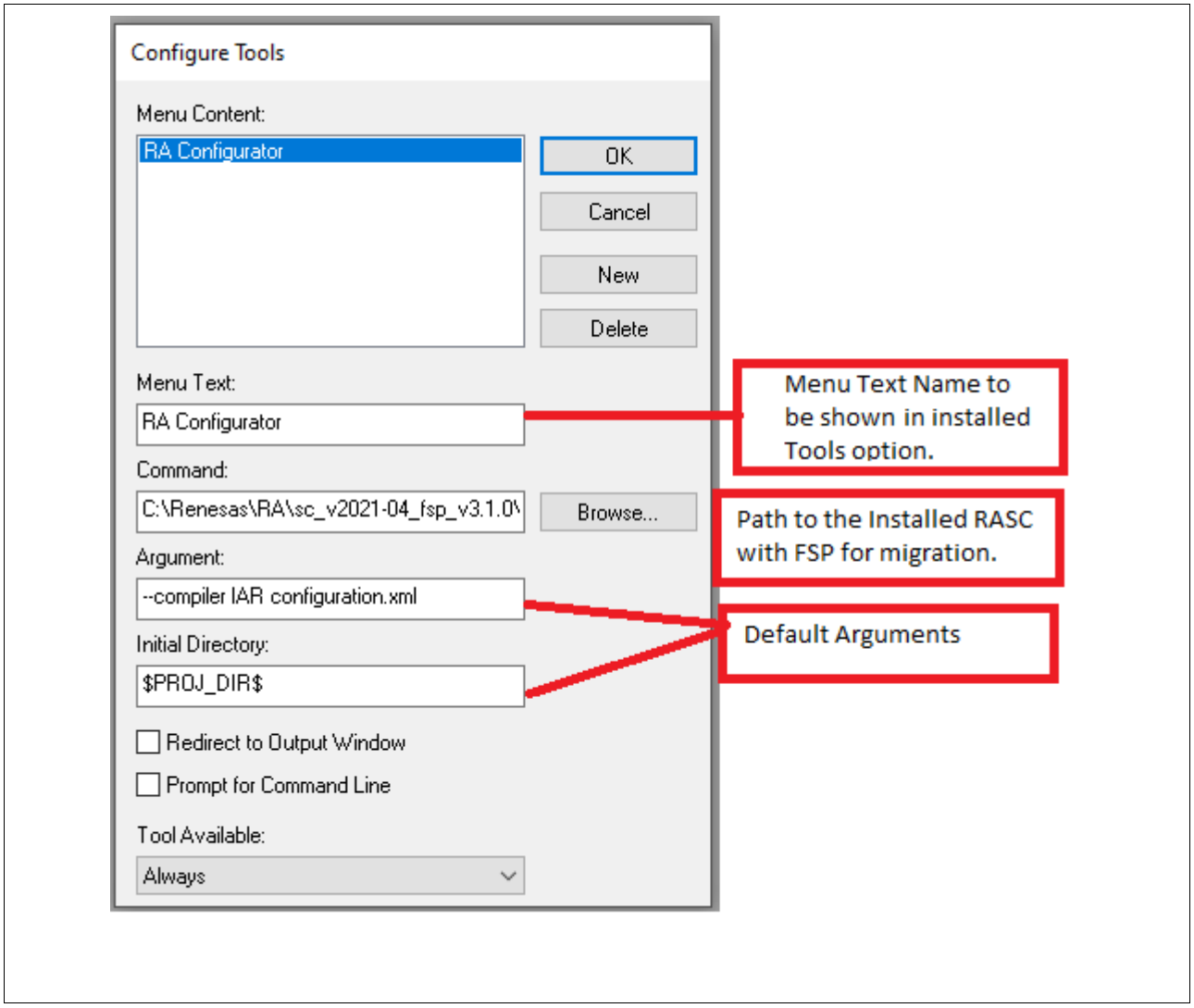

Figure 7. Configure Tools Options

2. Set the **Configure Tools** options as shown in Figure 7, as described below:
   - **Menu Text:** Name of the RA Smart configurator Tool to be shown in configured tools. The name of the tool should be given correctly for identification
   - **Command:** This is the path to the installed RASC to be used for migration. Browse and navigate to the installed `rasc.exe` (for example: `C:\Renesas\RA\sc_v2021-04_fsp_v3.1.0\eclipse`) in the new RASC installed and select it.
   - **Argument:** Set to **--compiler IAR configuration.xml.** This flag configures RASC to generate code compatible with IAR compiler
   - **Initial Directory:** Use **$PROJ_DIR$**

Note: When changing the FSP version, adjust the path to the rasc.exe in the **Command** field accordingly.

For example: Here FSP v3.1.0 is used, so the configured path is `C:\Renesas\RA\sc_v2021-04_fsp_v3.1.0\eclipse\rasc.exe`'

3. Click on **OK** to complete the RASC configuration with IDE.

Now the configured RASC can be run from the IDE with the name configured in the **Menu Text** field of **Configure Tools**. Open IAR IDE and Click on **Tools** → **RA configurator**. RA configurator can be run from here as shown:

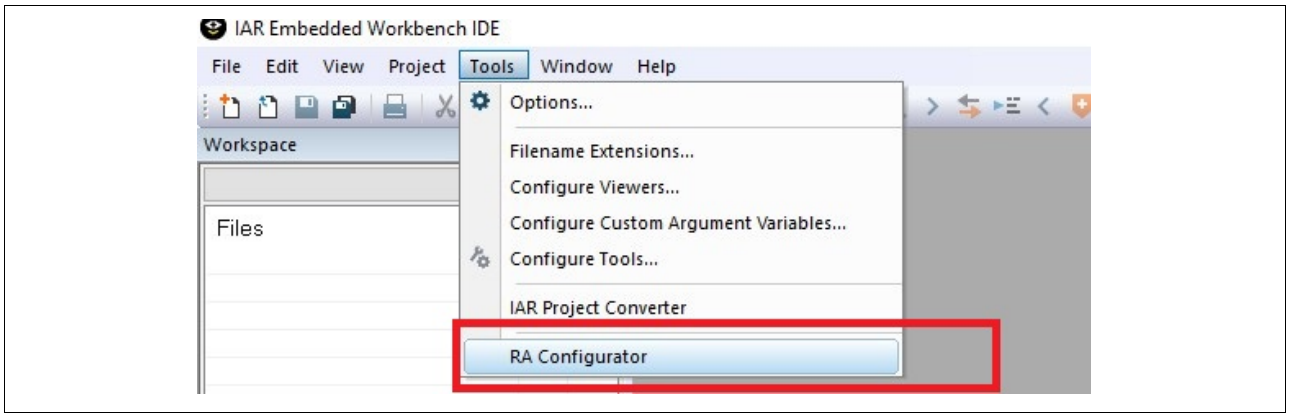

Figure 8. Run RA Configurator

2.1.2 Patch Installation for IAR

Patching IAR may be required if the IAR Embedded Workbench does not include support for the desired RA MCU. Without this patch you may not be able to generate or operate projects for the microcontroller of your choice.

Communication with IAR is recommended for accessing the patch. Once available, the patch for supported boards should be copied Inside the **arm** directory of Installed IAR version.

Default Location: `C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.4\arm`

1. Locate the `arm` directory in the installed IAR directory.
2. Unzip the patch files to get newer directories, that is, `config`, `inc`, `src` as shown:

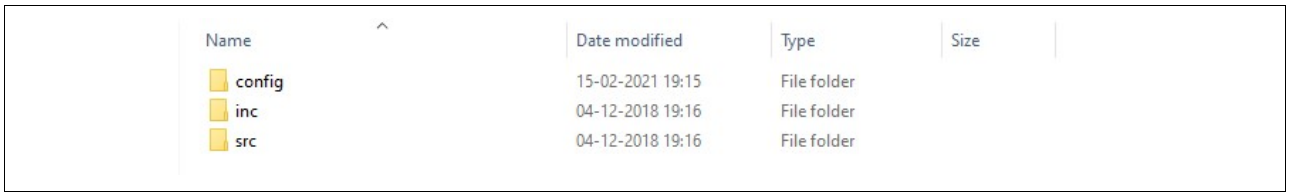

Figure 9. Newer Directories

3. Copy and paste the above directories to the default location or to the `arm` directory location where IAR is installed.

2.2 Creating a Blank Template Project for IAR

Below are the requirements for creating a new project in IAR:

- IAR EW for ARM (current version used: 8.50.9)
  - Note: Latest available IDE version associated with a FSP release can be checked in the FSP release note available at github.
- Flex Software Package (FSP) (current version used: 3.1.0)
- RA Smart Configurator (RASC) (current version used: 3.1.0)

2.2.1 Steps for creating a Blank Template Project in IAR

1. Run the **RASC (**`rasc.exe`**)** after installing the packs as mentioned in the previous section.
2. Provide an appropriate **Project Name** for the project and **Location** for the workspace.

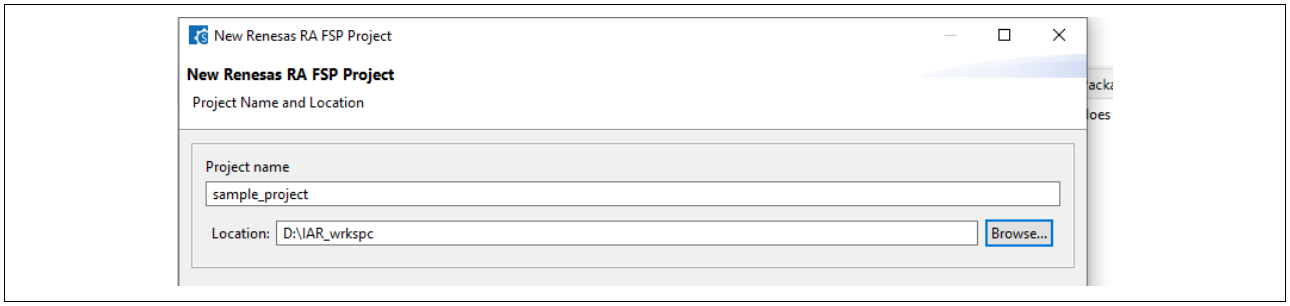

Figure 10. Set Project Name and Location

3. Click on **Next**. The following screen will appear. Select the **FSP version**, **Board**, **Device** and **IDE** in which project is to be created.

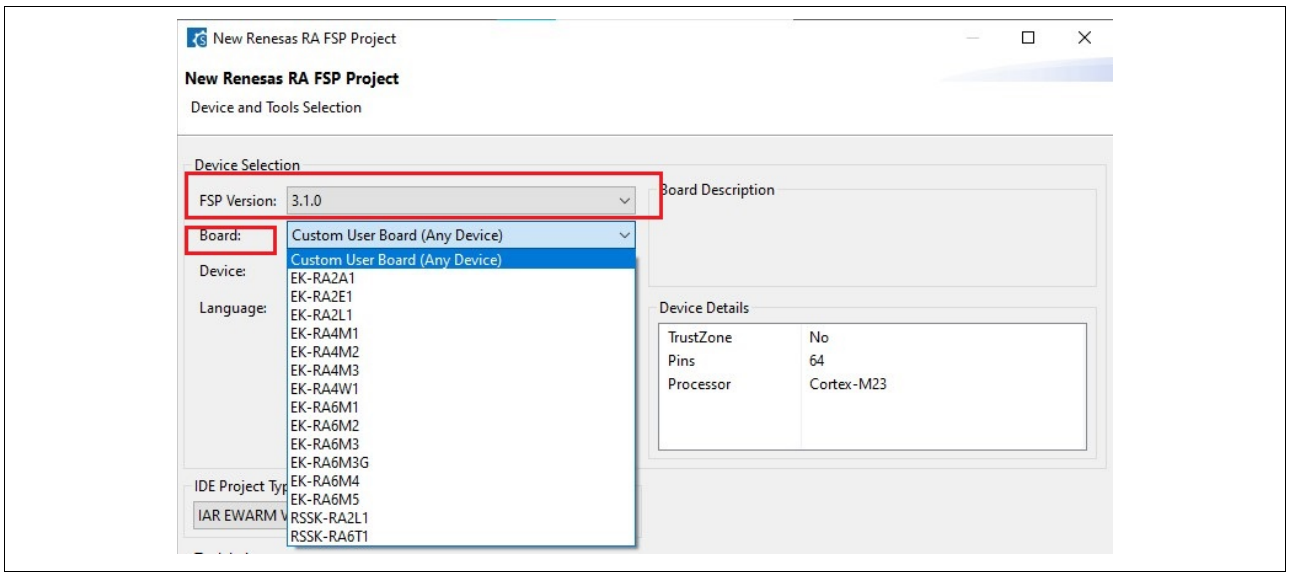

Figure 11. Set FSP Version, Board, Device and IDE

4. The **FSP Version** dropdown will show available packages. The packages must be present inside the **internal** folder in the installed directory of RASC.
5. **Device** and **Language** are automatically selected to be **Default**.
6. Select **IDE Project type** to be **IAR EWARM Version 8** for creating the sample project in IAR.

Figure 12. Set IDE Project Type

7. Click on **Next** and follow the steps beow for **Flat** project creation. If you select an MCU with TrustZone support (RA4M2, RA4M3, RA6M4, RA6M5), you will be prompted to select a **Flat**, **Secure** or **Non secure** project. Otherwise only Flat project can be selected.

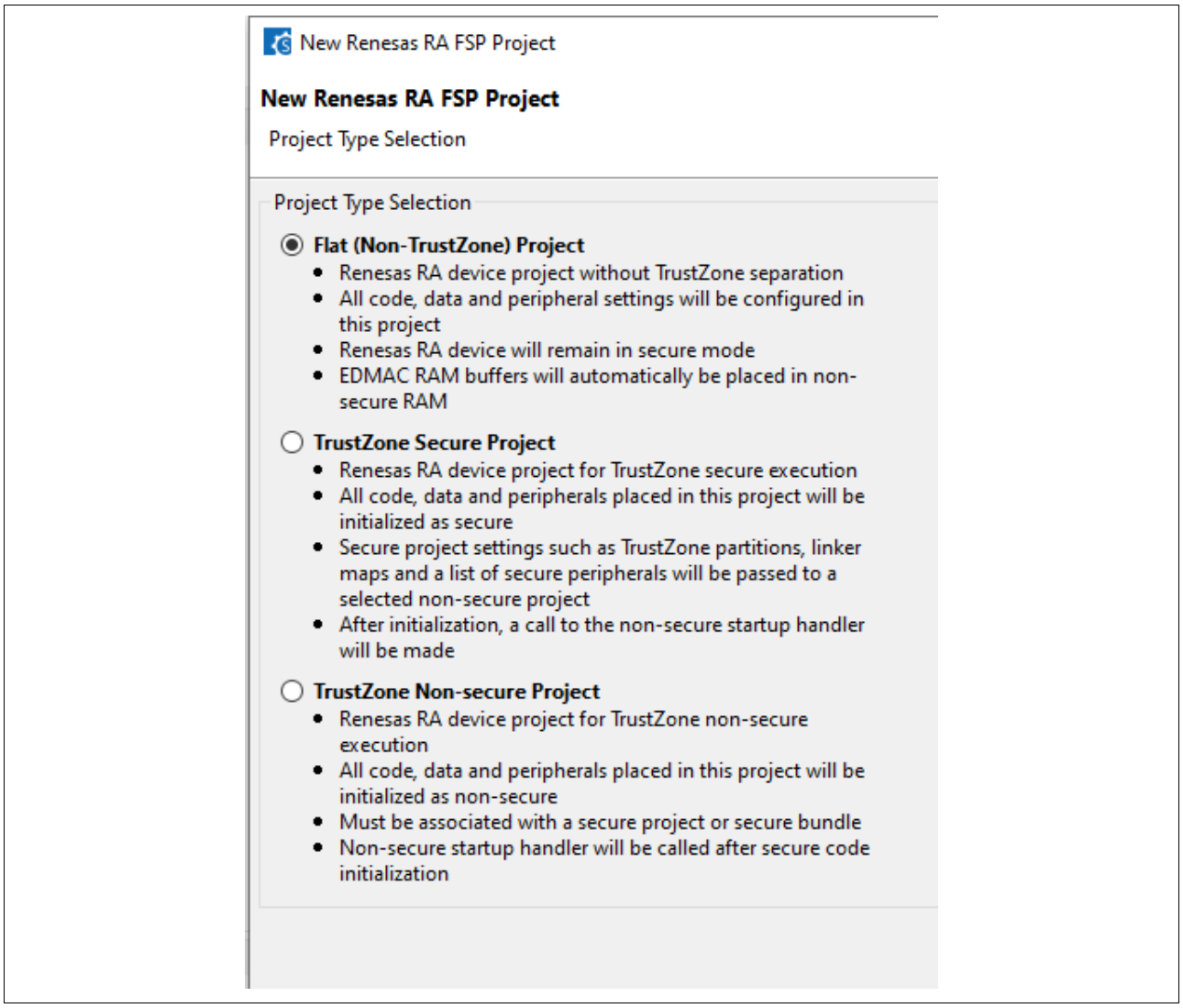

Figure 13. Flat, Secure, Non-Secure Options

8. After selecting the project type (in step 6), select **RTOS** as per project requirements. By default, it is set to **No RTOS**. Click on **Next** to proceed.

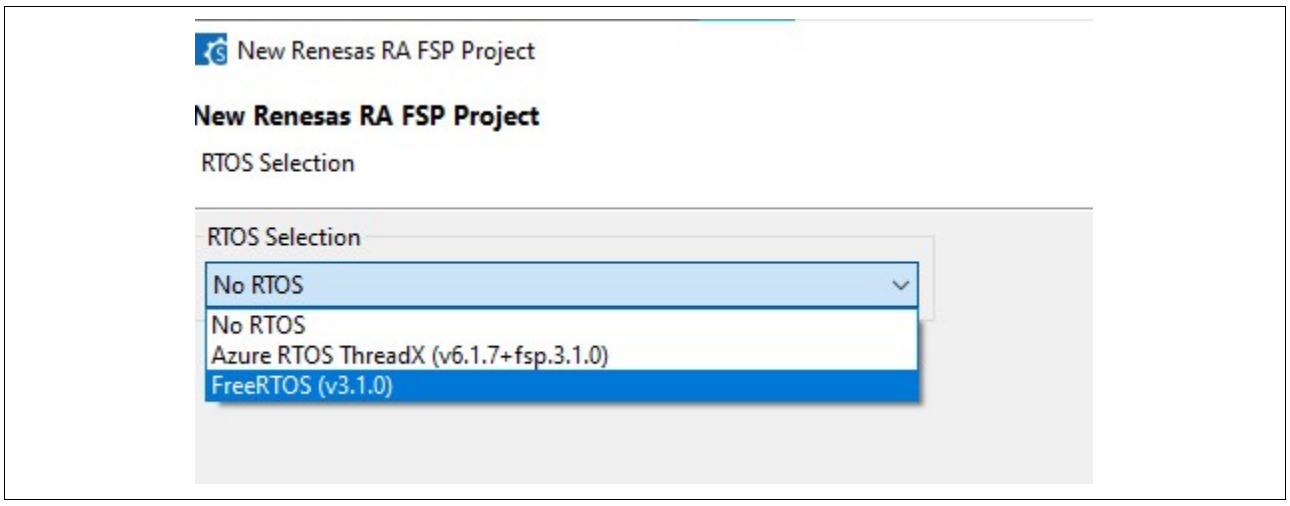

Figure 14. RTOS Selection

9. Select **Bare Metal - Minimal Project** to create the project sample template. Click on **Finish** to complete the Project Template creation.

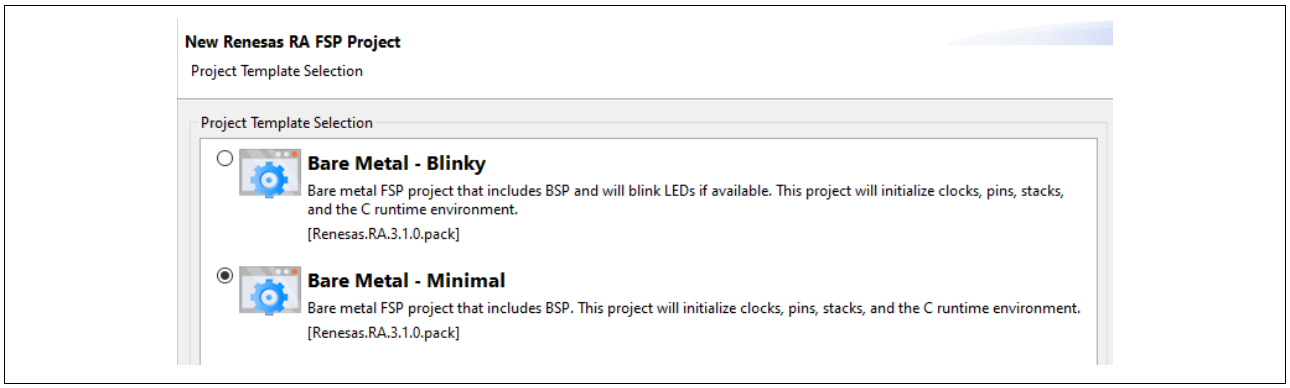

Figure 15. Select Bare Metal - Minimal

2.3 Adding the source code

2.3.1 Adding src files

- If the above steps are completed successfully, the `src` folder will already be present in the same location where the IAR Project template is created.

This `src` folder in the project template can be replaced with the `src` folder from the e² studio project folder.

2.3.2 Configuration and Generating project after src files addition.

After project template creation, the project is created in the selected project location as an `.eww` file with all the required initialization and other configuration files.

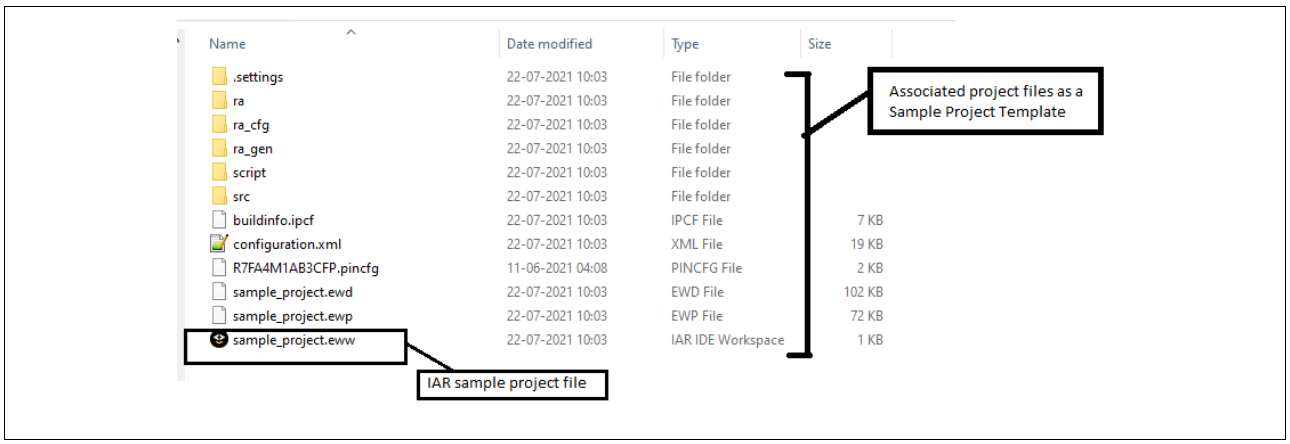

Figure 16. Project Files

Steps Involved for Project Generation:

1. As stated above, a `<project name>.eww` file is created in project template creation. Run the `.eww` file to open the project in IAR EW. .
2. Click on **Tools** option to see **RA Configurator** as RASC is configured with IAR IDE (Refer to Steps to configure RASC for installed IAR EW software above)
3. After RASC opens, a screen appears as in Figure 17 with multiple tabs and options:

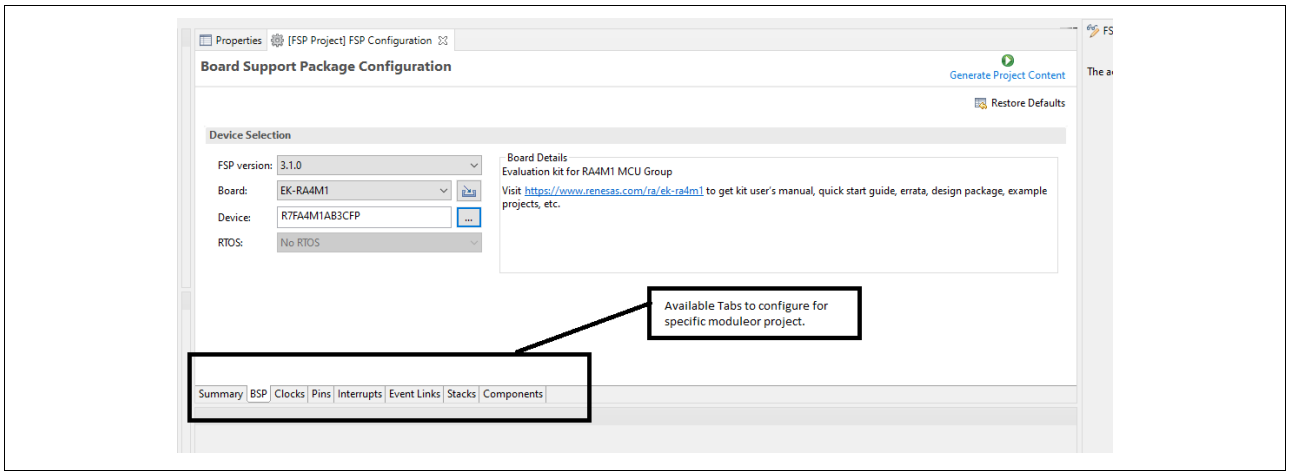

Figure 17. Board Support Package Configuration

4. The required **Tabs** can be selected in the Tabs Bar of RASC and **Properties** of individual components can be configured as per the project requirement.
5. Generally, the following tabs are configured for roject setup before generating the project:
   a. Select the **BSP** tab and set up BSP parameters in **Properties** Window.
   b. Add sacks in the RASC configuration according to the requirement. The respective module stack can be chosen from the **New Stack** option.

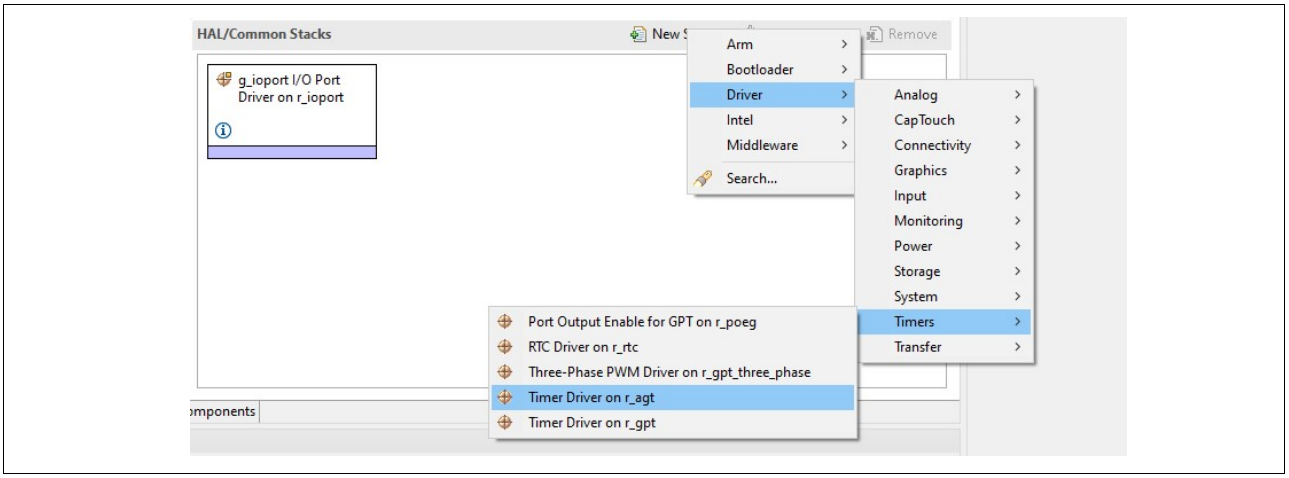

Figure 18. Adding New Stacks

   c. Set up the **Pins** tab according to the module to be configured.

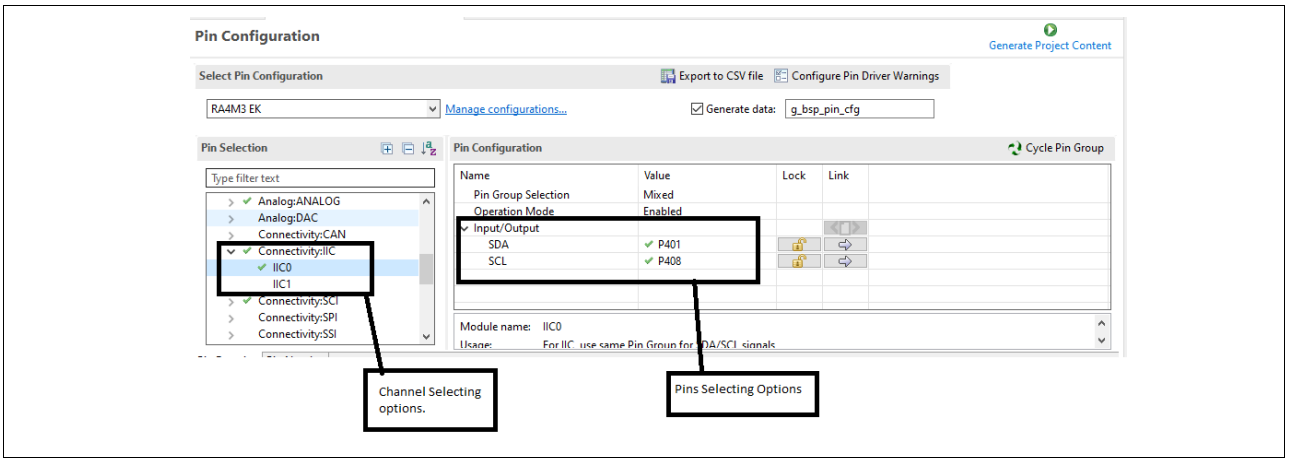

Figure 19. Setting Pins Tab

6. After setting up all required tabs (**BSP**, **Stack**, **Clocks**, and so on) for the desired module, save (**Ctrl+S**) and click on **Generate Project Content** as shown:

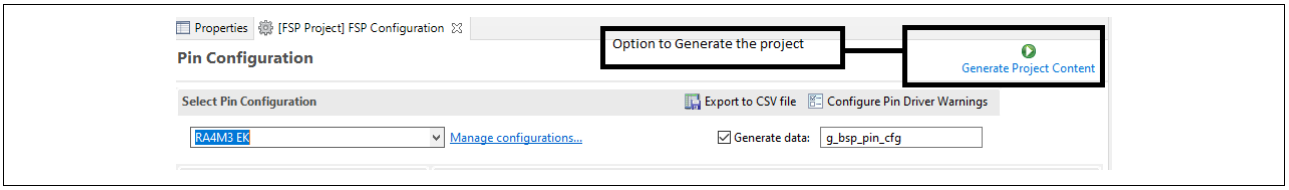

Figure 20. Generate Project Content

2.4 Configuring and Compiling the Project

2.4.1 Setting up compiler options

The compiler options for project can be changed with following steps:

1. Open the `<project_name>.eww` file. In the workspace click on **Project** → **Options** or use the **Alt+F7** shortcut.
2. In the **Options** window, select **C/C++ compiler** in the **Category** section. Select **Optimizations** (on the right side) as shown in Figure 21:

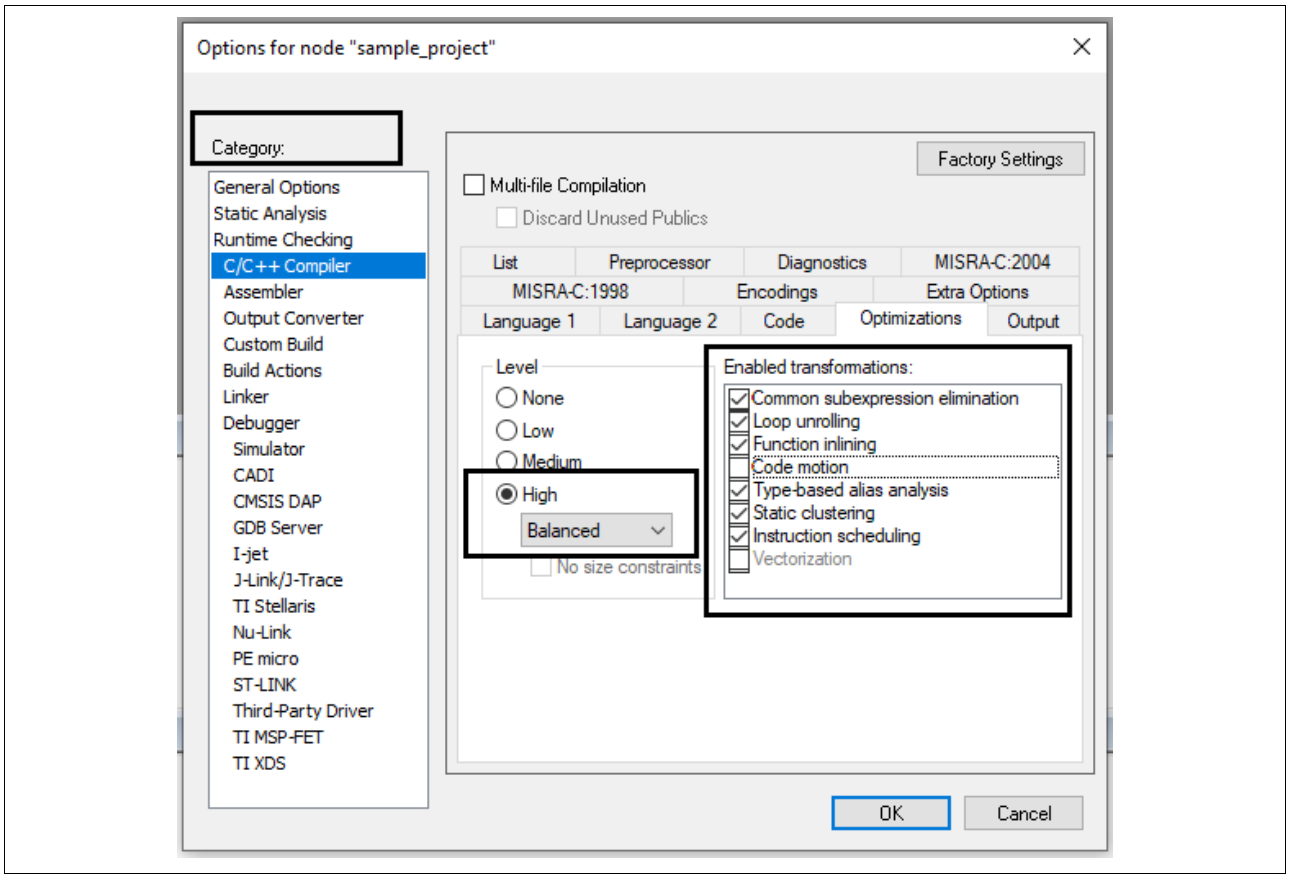

Figure 21. Optimizations

3. As shown in Figure 21, select optimization **Level** as **High**, select **Balanced** in the menu below it and uncheck **Code Motion.**

4. In the same **Options** menu**,** select **Output Converter** in **Category.** Check **Generate additional output** and optionally **Output format** as **Intel Extended Hex** as shown in Figure 22:

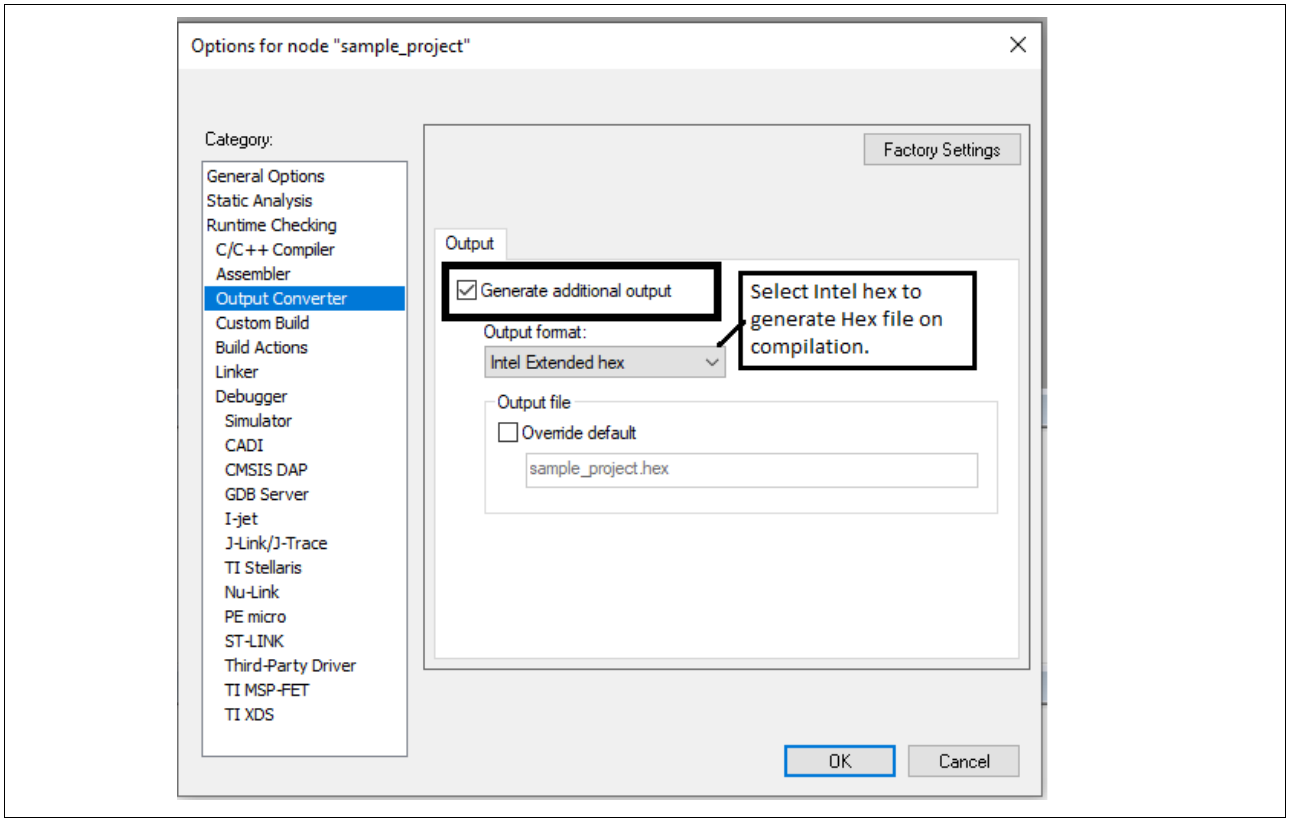

Figure 22. Output Converter

5. Click on **OK** that the bottom of the **Options** window to apply the settings.

2.4.2 Building the Project

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

- The module-specific `src` folder is present inside the project folder directory.
- Project files are generated with required configurations (**BSP**, **Clocks**, **Pins**, and so on)
- Compiler adjustments are completed.
- The IAR EW License is active.

1. After generating the project, open the project (.eww file)
2. Right-click on the project and click on the **Rebuild All** option.

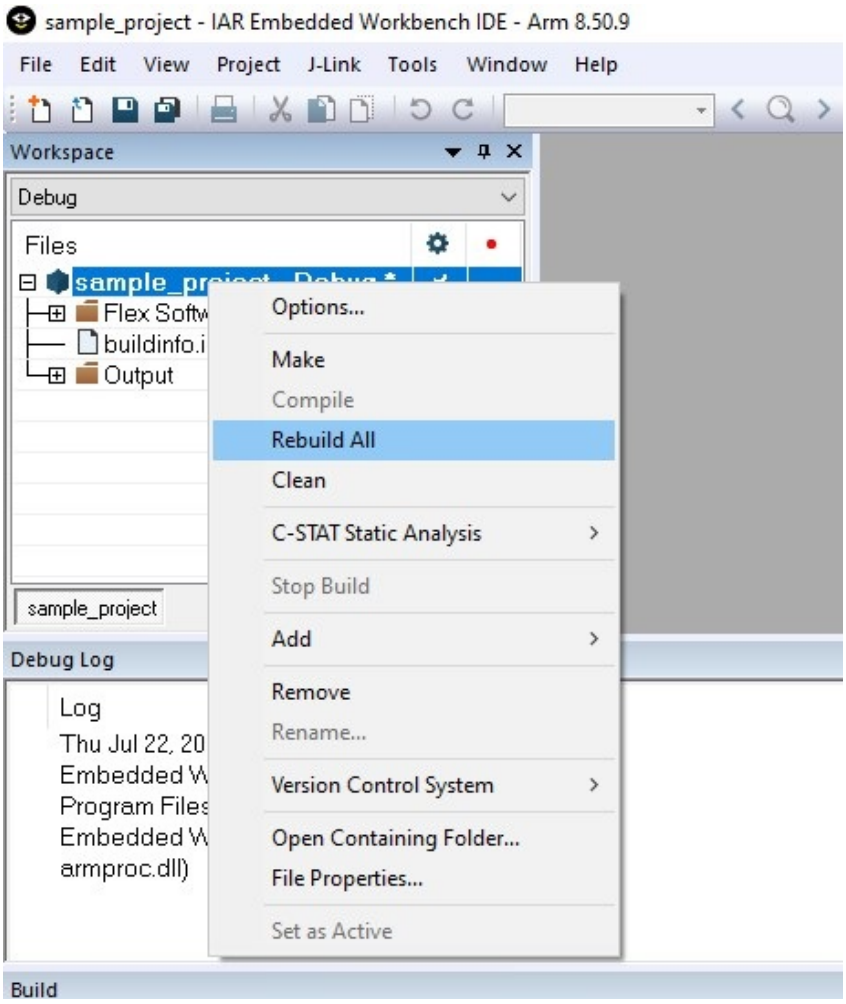

Figure 23. Rebuild All

3. Logs can be captured by enabling the Live Logging option
   - Right-click on the **Build** window below the workspace
   - Select **Live Log to File** and choose the file location for the log to be created.

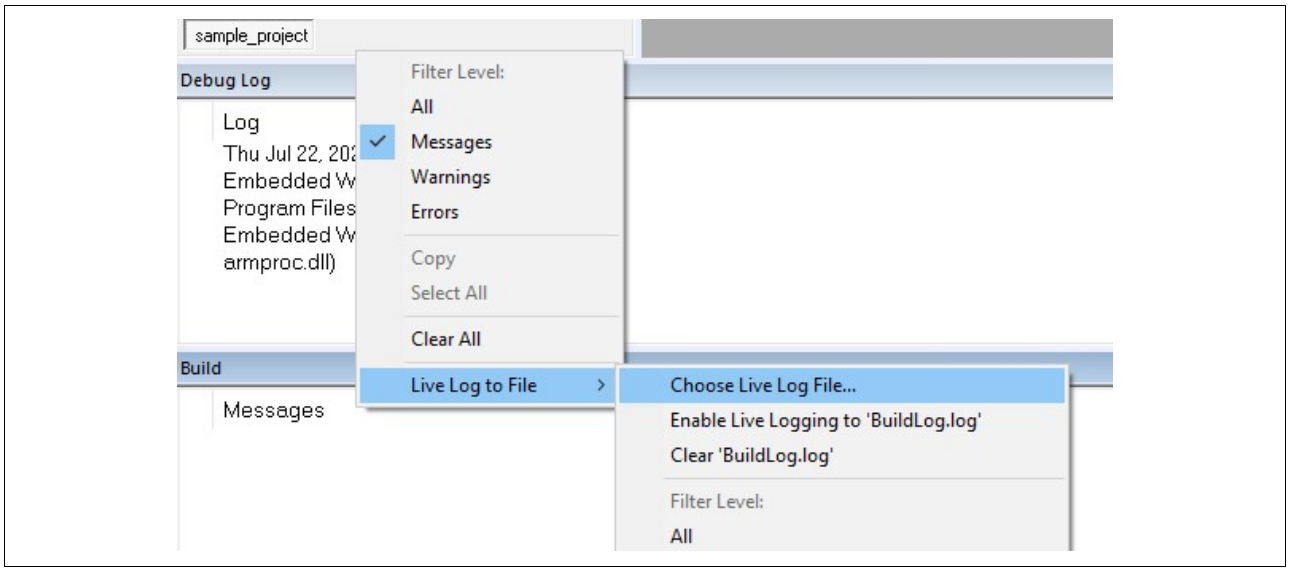

Figure 24. Live Log to File

2.4.3 Addressing Warnings and Errors in Build Operation

On completion of generating and build procedure of the project, the successful or failed operation can be observed in Build logs in the **Build** window.

If the **Build** window is not visible, select **View** → **Messages** → **Build**.

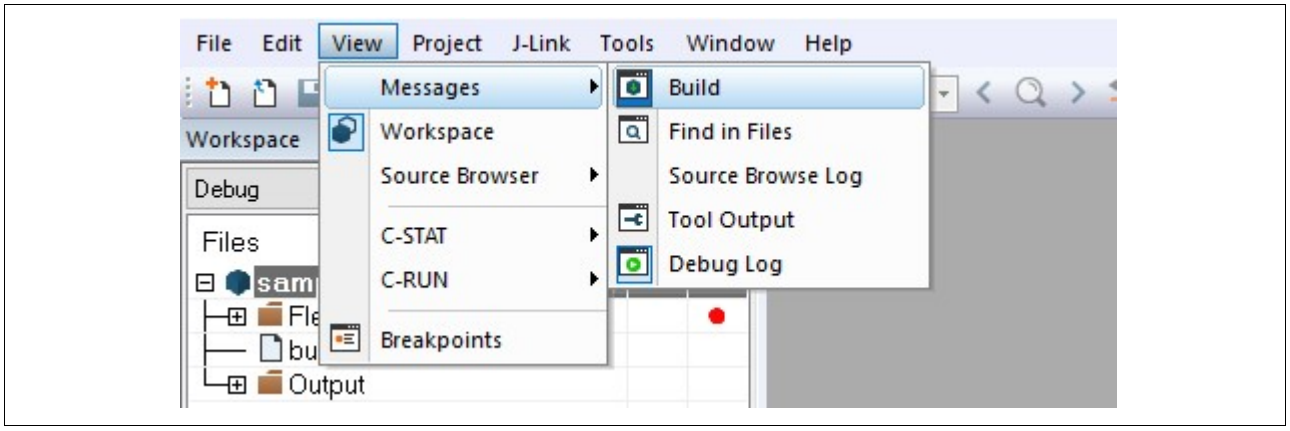

Figure 25. Show Build Window

In the **Build** window, error and warning messages can be observed in detail with total number of errors and warnings found. See Figure 26.

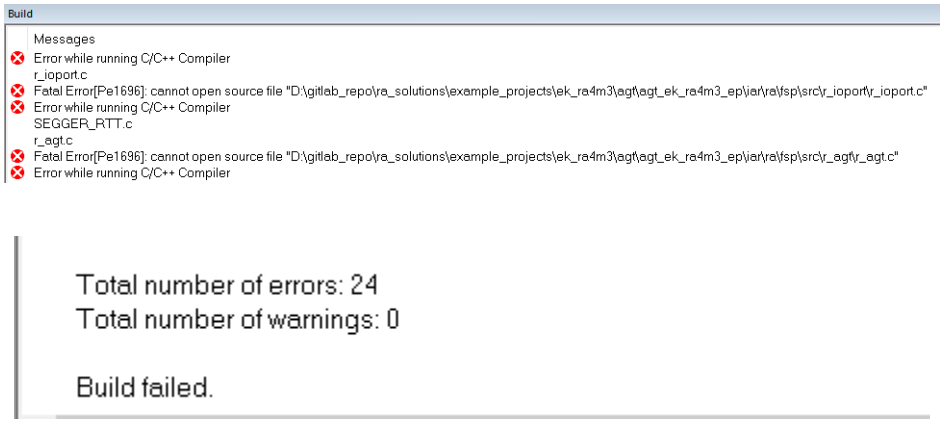

Figure 26. Errors and Warnings

2.5 Downloading and Debugging the Application

Before debugging make sure there are 0 errors and review the compiler generated warnings. Confirm that the target board is connected to the host system.

- To debug the binary, click the **Download and Debug option** on the options panel as shown

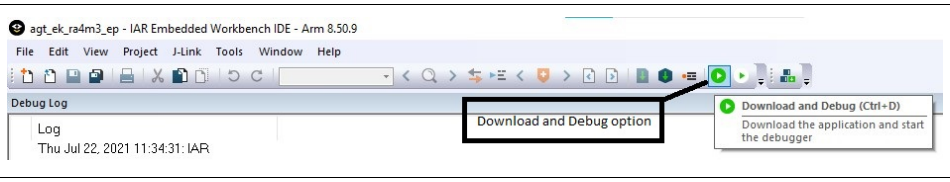

Figure 27. Download and Debug Option

- Once download and debug successfully completes, click on the **Go** option to run the code.

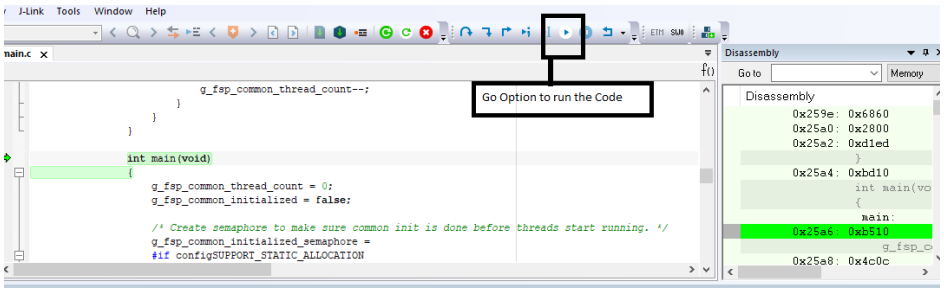

Figure 28. Run Code

- RTT will be visible if it is compiled into the binary and invoked by the application.

Migrating e² studio projects to Keil MDK and ARM C Compiler

Refer to the following document for more on how to use Keil IDE with the RA boards:
https://www.keil.com/appnotes/files/apnt_330_v0.35.pdf

Refer to the following document for Using RA Smart Configurator with Keil MDK -
https://renesas.github.io/fsp/_s_t_a_r_t__d_e_v.html#using-ra-smart-configurator-with-keil-mdk

Please refer to the FSP Installation section for RASC/FSP installation process.

Note: When changing the FSP version, adjust the path to the `rasc.exe` in the **Options for Target** in the **User Command** field accordingly.

2.5.1 DFP installation

1. Download latest available DFP pack and extract it.
   All the latest Device Family Packs are available at the following link - https://www.keil.com/dd2/pack/
2. Open the Keil IDE (Note: Administrative access is required to install the packs).
3. Open the pack installer.

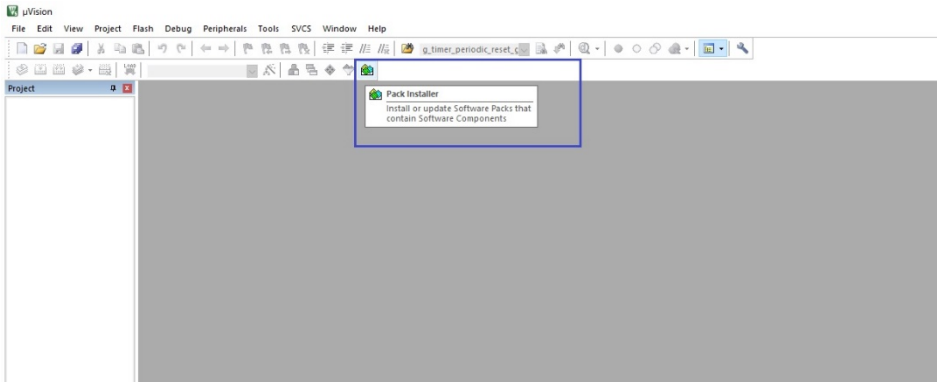

Figure 29. Pack Installer

4. Go to **File** → **Import from Folder…** as shown in Figure 30.

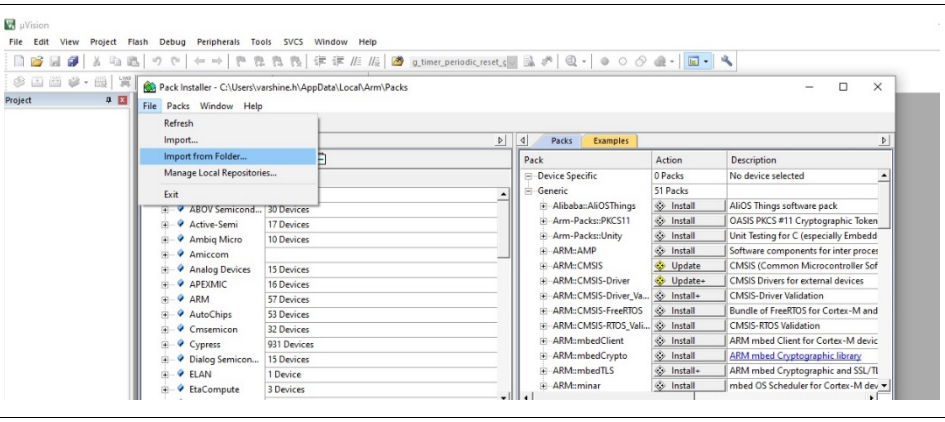

Figure 30. Import from Folder

5. Select the extracted folder (from in step 1) from the window that opens.
6. The installed pack can be checked for a particular device selected under packs tab as shown below:

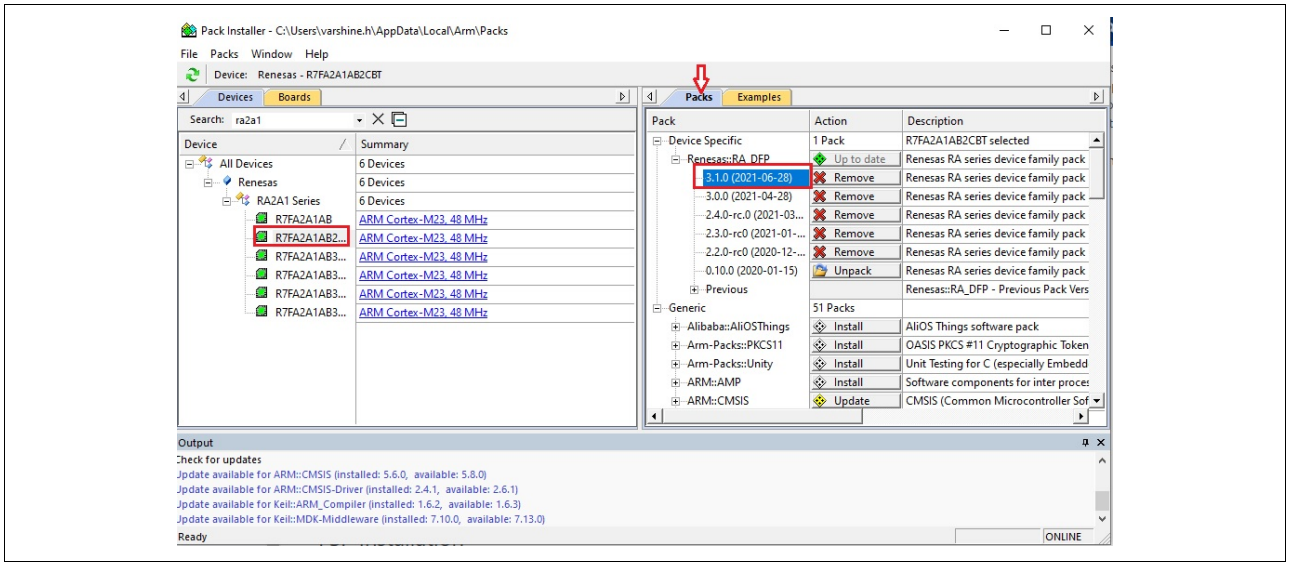

Figure 31. Installed Packs

2.5.2 Creating a Blank Template Project for Keil

1. Navigate to folder `<rasc installation directory>\sc_v2021-04_fsp_v3.1.0\eclipse` and open `rasc.exe`.
2. Enter the project name and project location in the opened smart configurator and click **Next.**

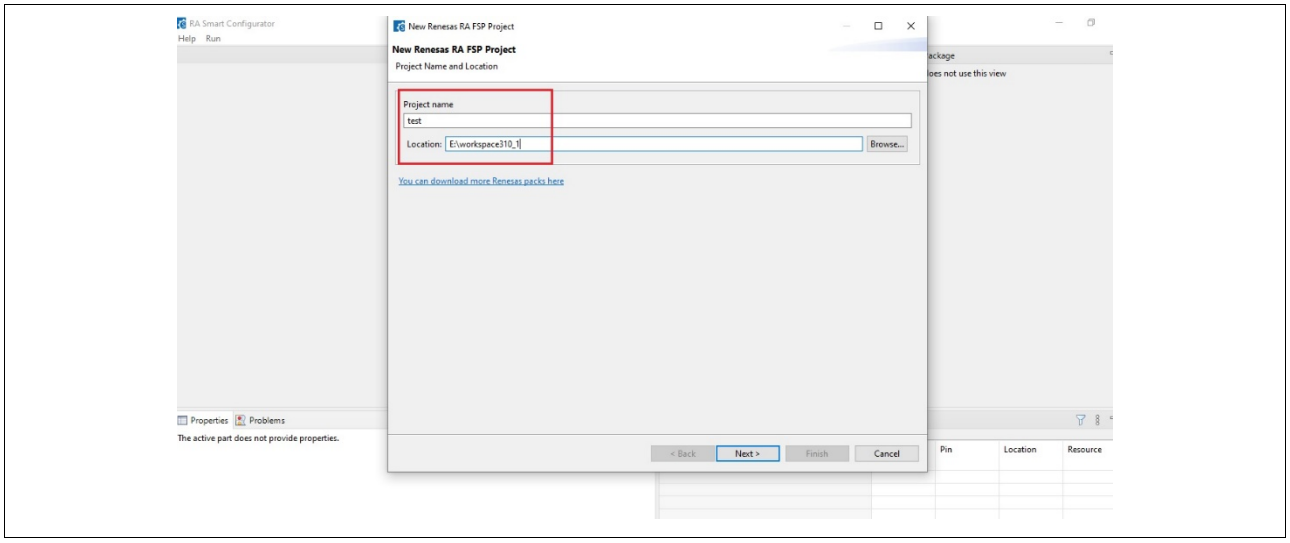

Figure 32. Smart Configurator

3. The next step is to select the required FSP version and board. The project type of the particular IDE can be selected in this step as shown in Figure 33. Select **Keil MDK version 5**.

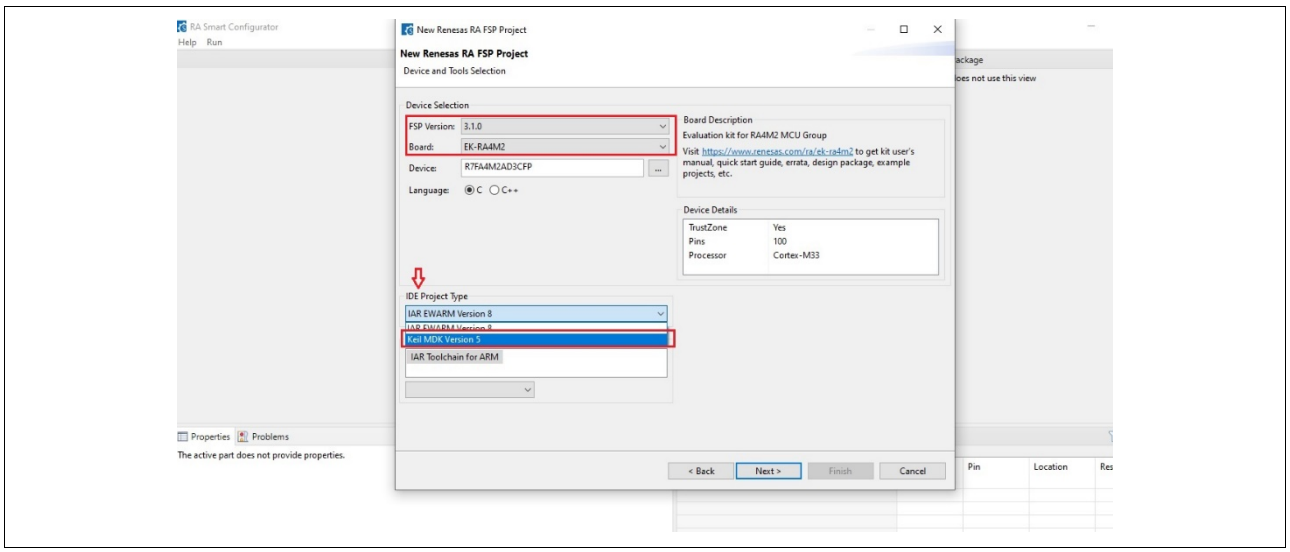

Figure 33. Select Project Type

4. In the next step, select the project type as **Flat**, **TrustZone Secured**, or **TrustZone Non-secured** as per requirement. This selection option is available only for MCUs with TrustZone support.

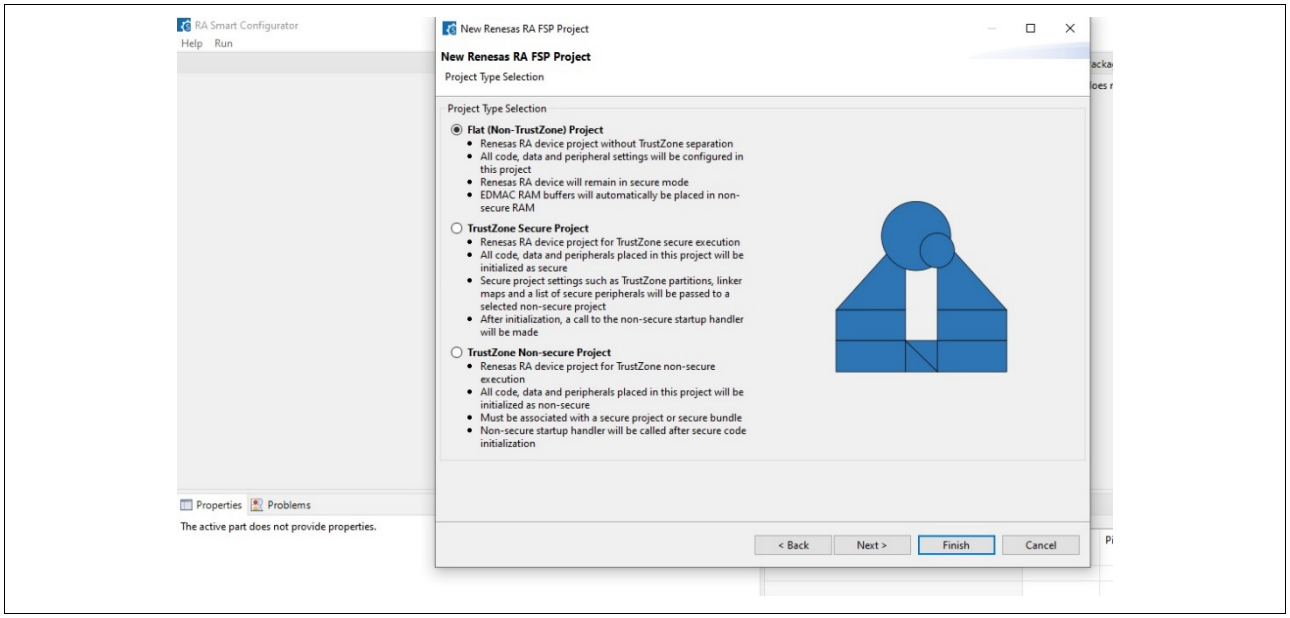

Figure 34. Select Flat, TrustZone Secured, or TrustZone Non-secured

5. Next, RTOS selection is available with options as shown below. Select the desired option for the project.

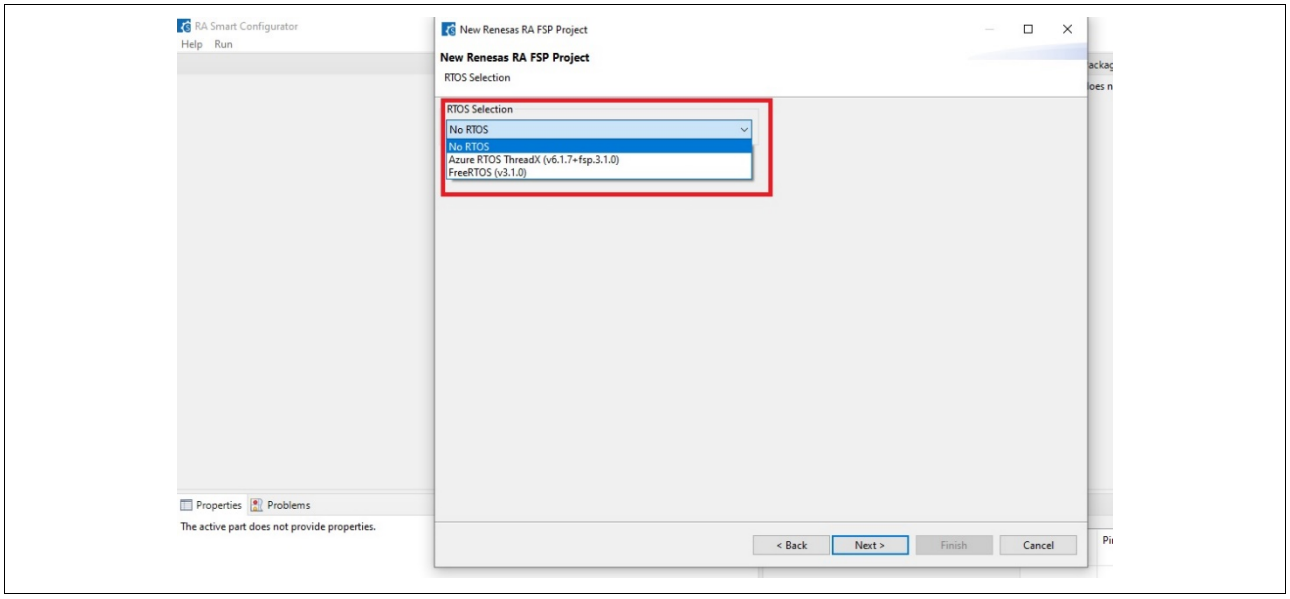

Figure 35. Select RTOS

6. Select **Bare Metal - Minimal** project to create the sample template of the project. Click on **Finish** to complete the project template creation.

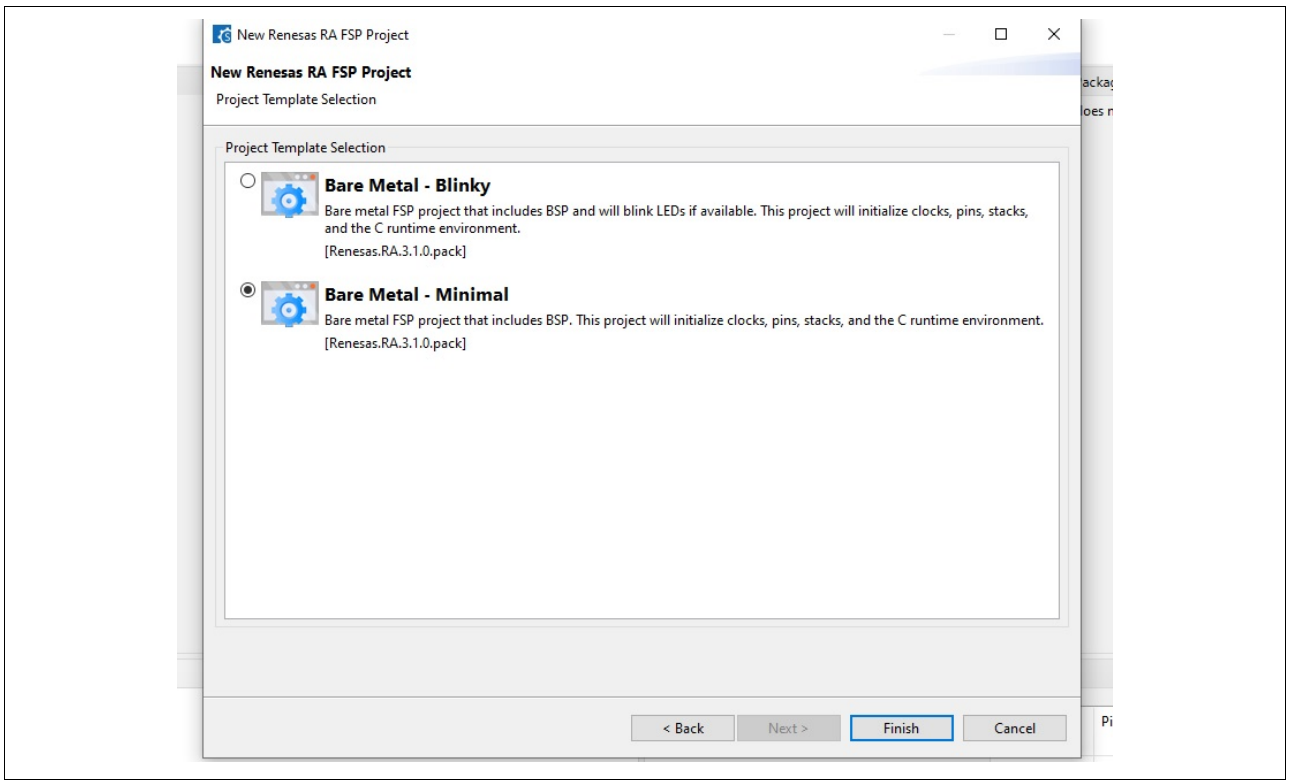

Figure 36. Project Template

2.5.3 Add Source code

1. After completing project template creation process, navigate to the folder containing the created Keil project.
   In the project folder, the generated `src` folder should be replaced with the `src` folder of the e² studio project.

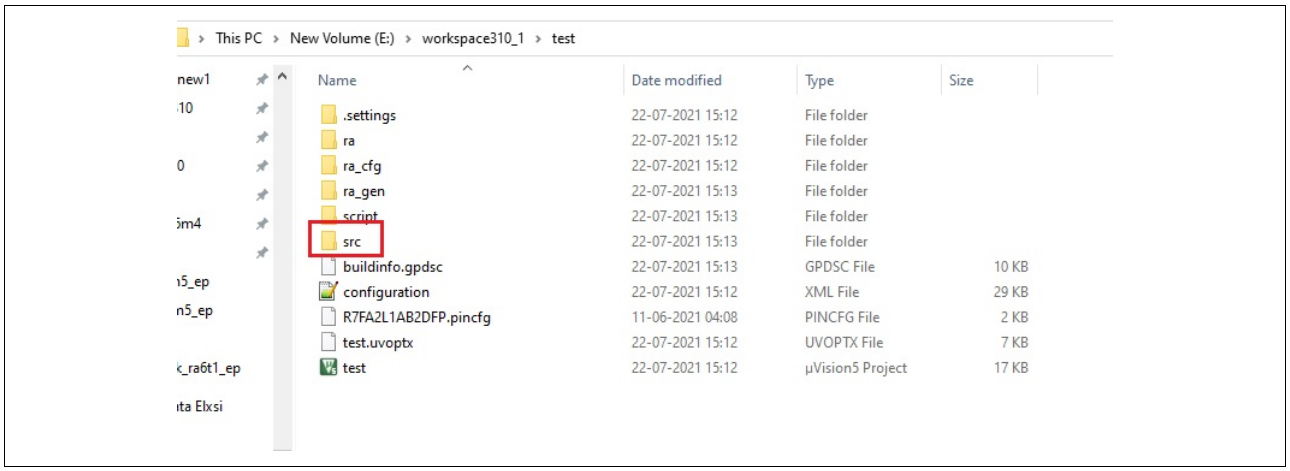

Figure 37. e² studio src Folder

2. Open the uVision5 project file in the folder to open the project in Keil IDE.

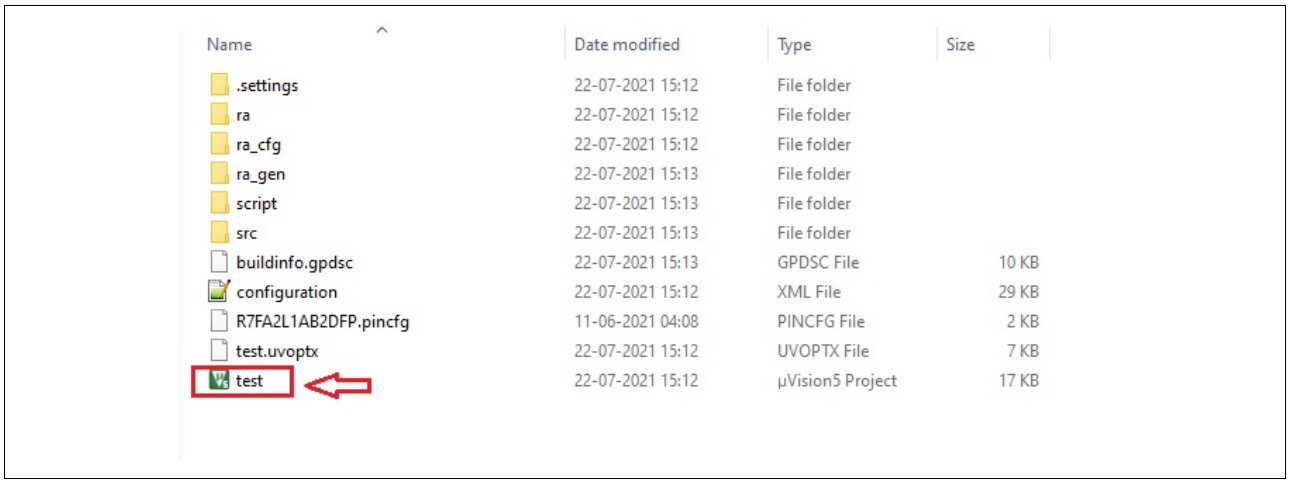

Figure 38. uVision5 Project File

3. The Keil project will be opened as shown in Figure 39.

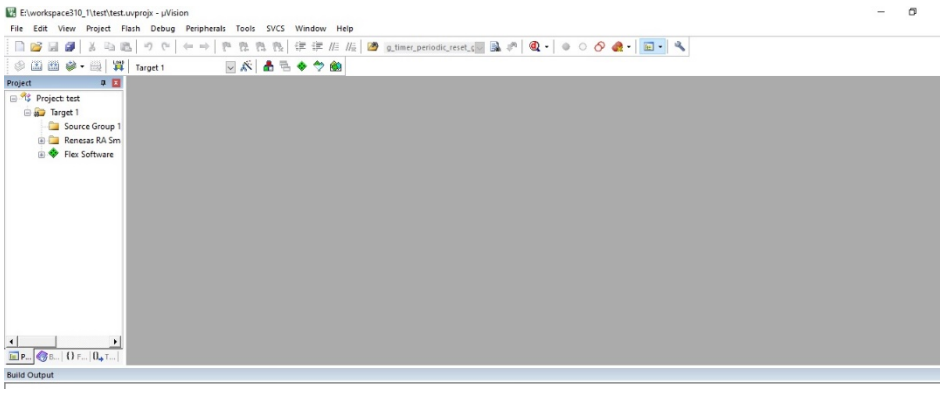

Figure 39. Keil Project

4. To open RASC, select the **Manage Run-Time Environment** icon.

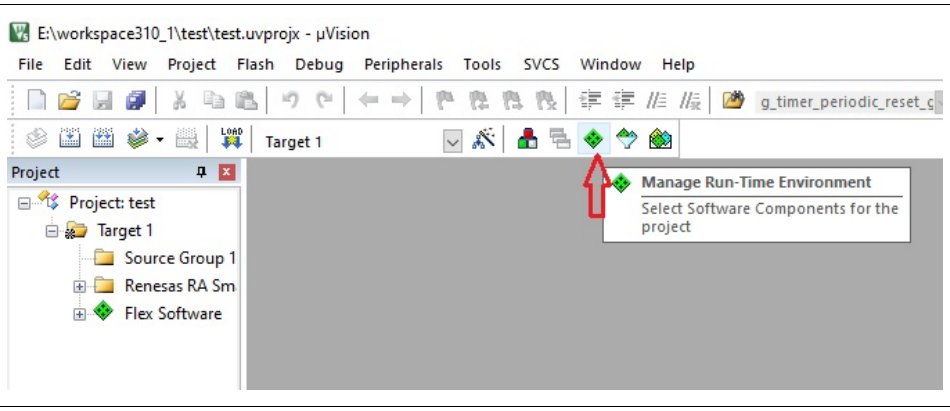

Figure 40. Manage Run-Time Environment

5. The following dialog box opens:

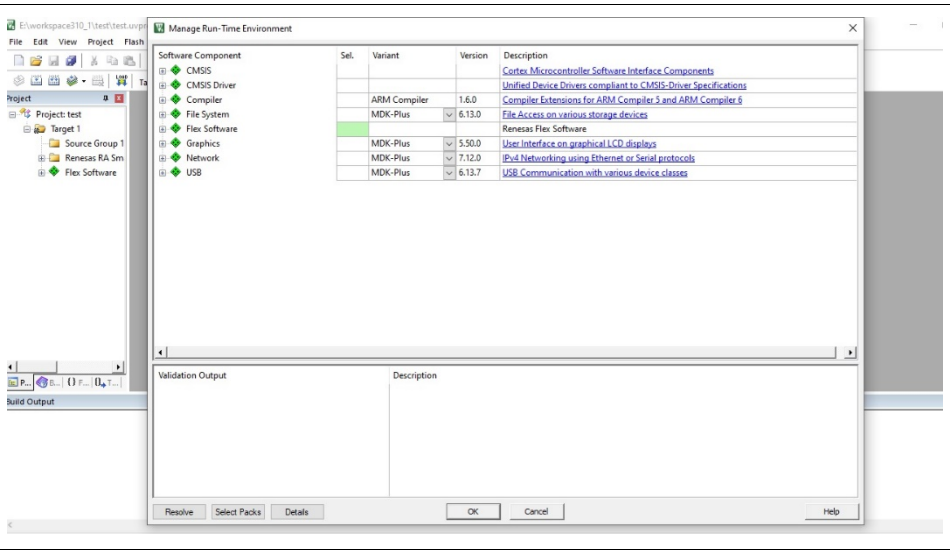

Figure 41. Manage Run-Time Environment Dialog

6. To run RASC, expand **Flex Software** and run **RA Configuration** as shown

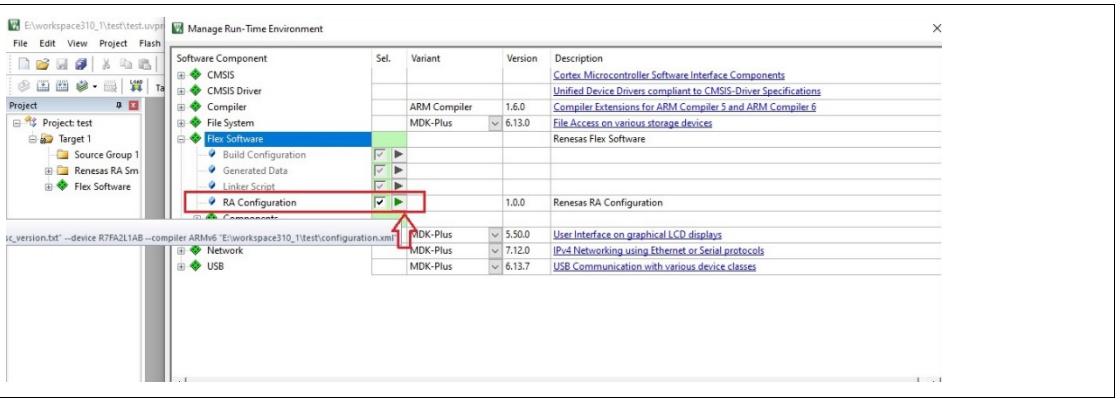

Figure 42. RA Configuration

7. A window with installed RASC versions will be displayed. Select the required version by entering the relevant number.

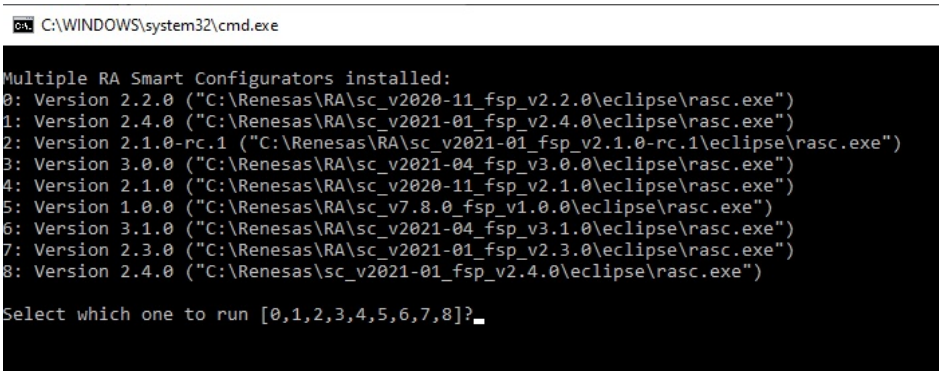

Figure 43. Select RASC Version

8. The smart configurator opens to allow configuration of various properties like **BSP**, **Pins**, and **Stacks** through the available tabs as highlighted.

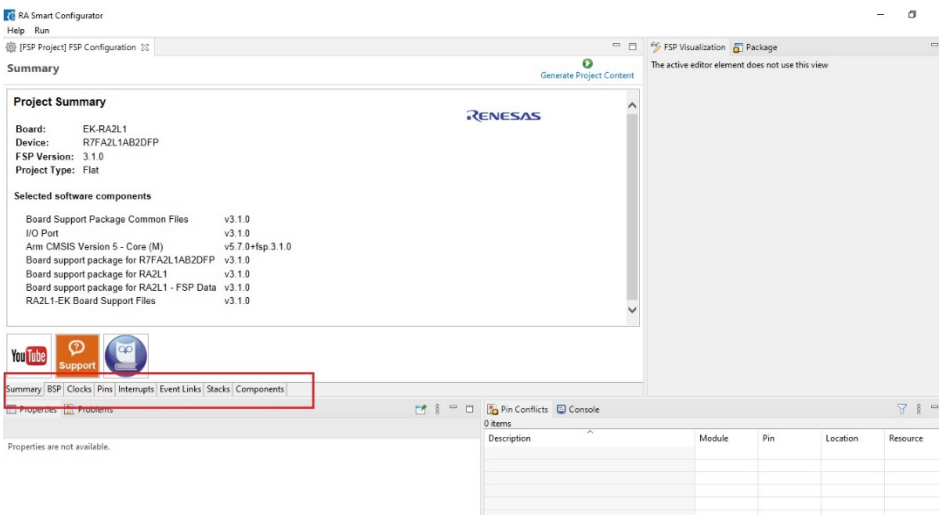

Figure 44. Smart Configurator

The parameters to be configured under the **BSP**, **Pins**, **Stacks**, and **Clocks** tabs should be adjusted to match the e² studio project configuration.

9. Once the desired configurations are done, perform **Ctrl+S** (Save) and select **Generate Project Content**.

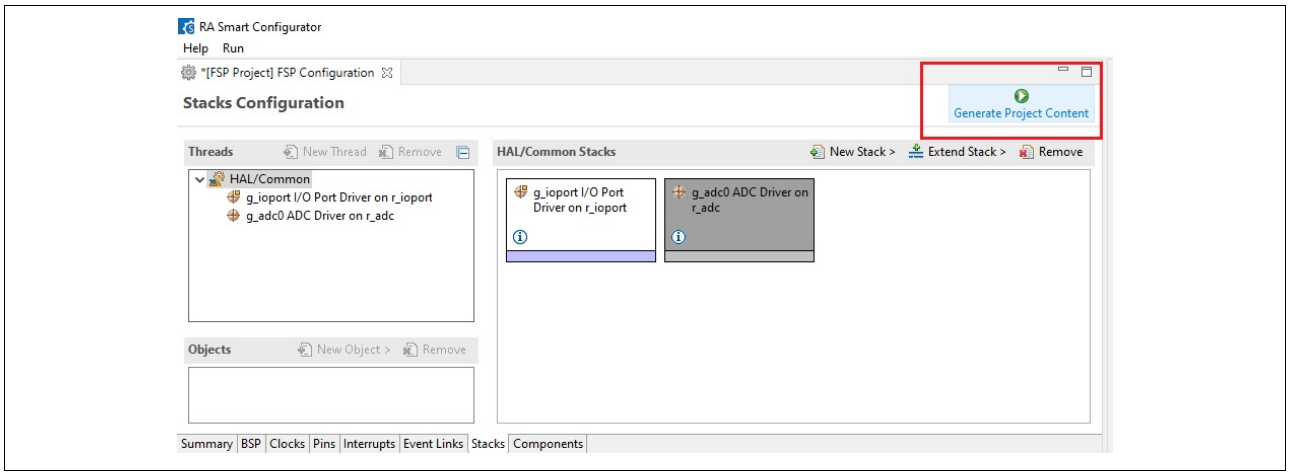

Figure 45. Generate Project Content

10. Close the RASC, then select **OK** in the **Manage Run-Time Environment** window.
11. Once done, the following dialog box appears. Select **Yes** to import the changes made in configurator.

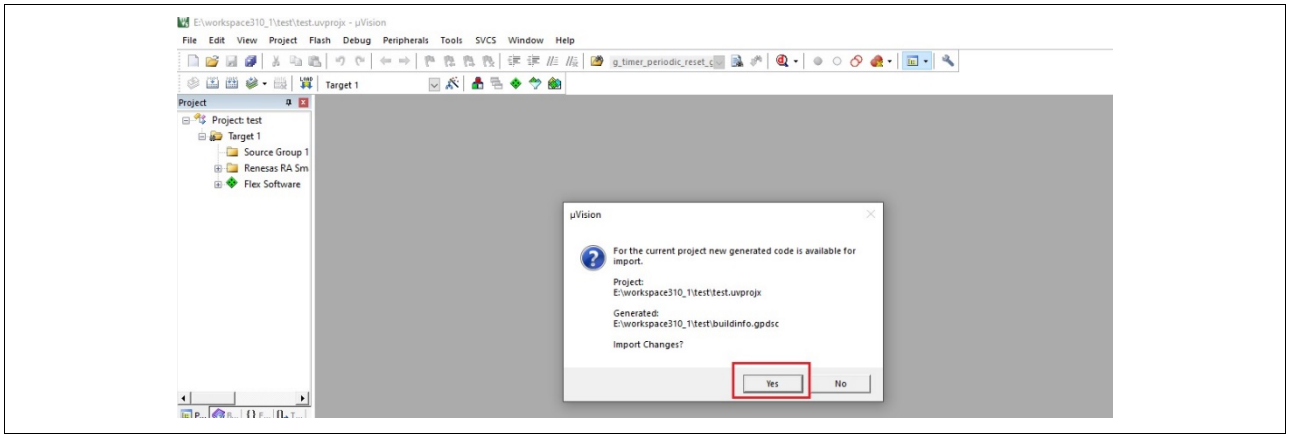

Figure 46. Save Changes

2.5.4 Compiling the Project

1. Set the following target options in the window that appears. This window can be selected from the **Target** icon as highlighted.

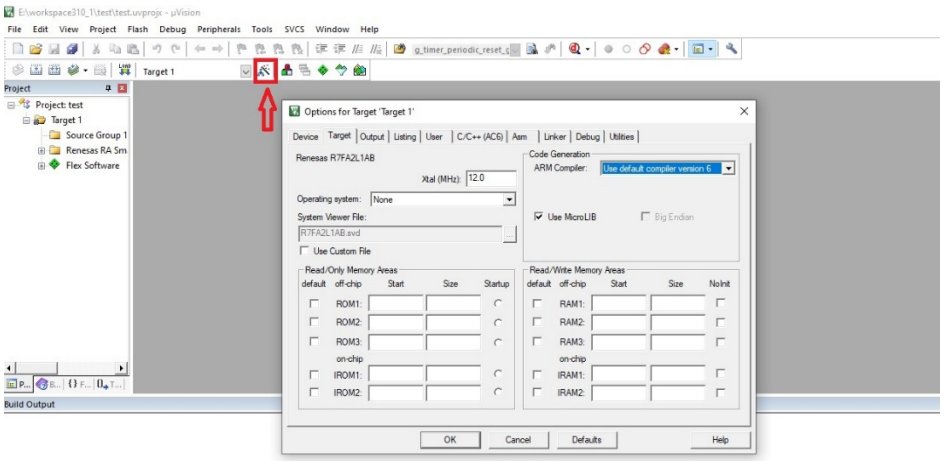

Figure 47. Target Options

2. Set the **ARM Compiler** version as shown in Figure 48.

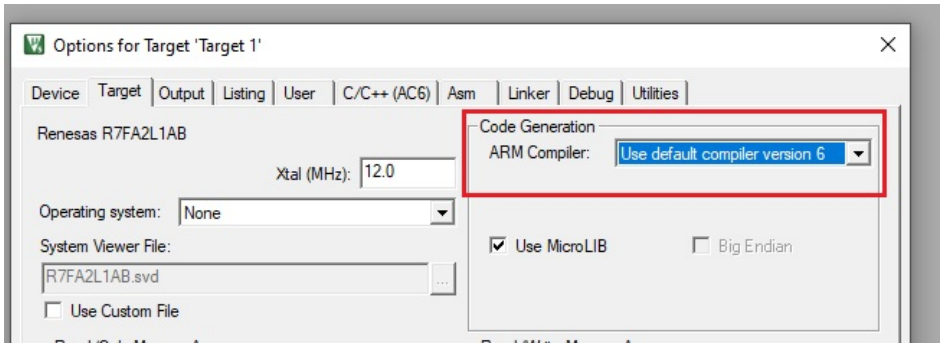

Figure 48. Arm Compiler Version

3. Under the **Output t**ab, ensure that the **Create HEX File** option is selected to generate the hex file during build.

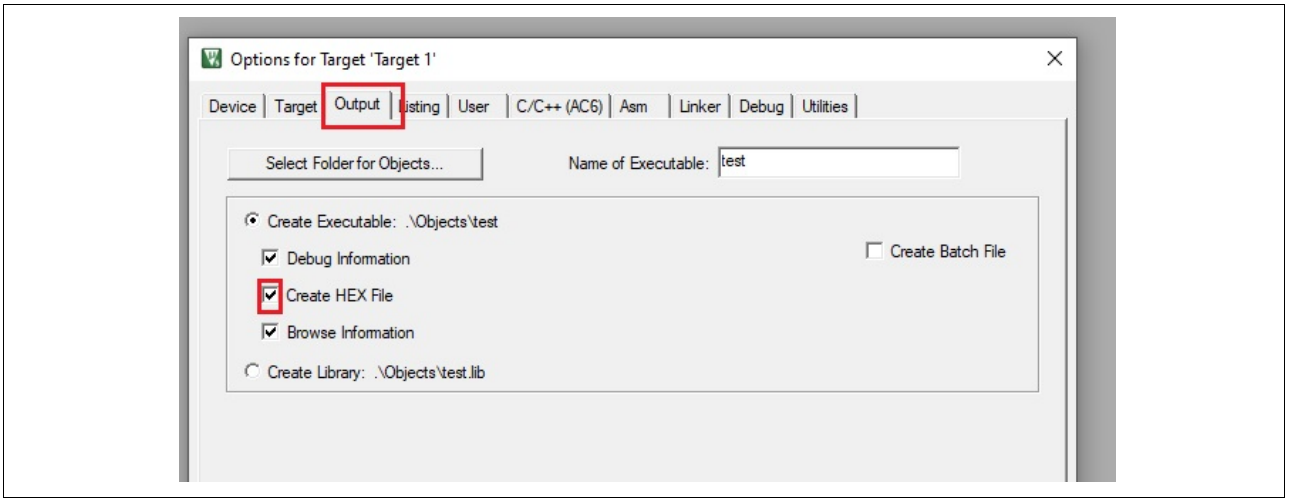

Figure 49. Create HEX File

4. Set optimization for the target under the **C/C++ (AC6)** tab. Select optimization as -**O2**. Under **Misc Controls**, add parameter -**Os** as shown

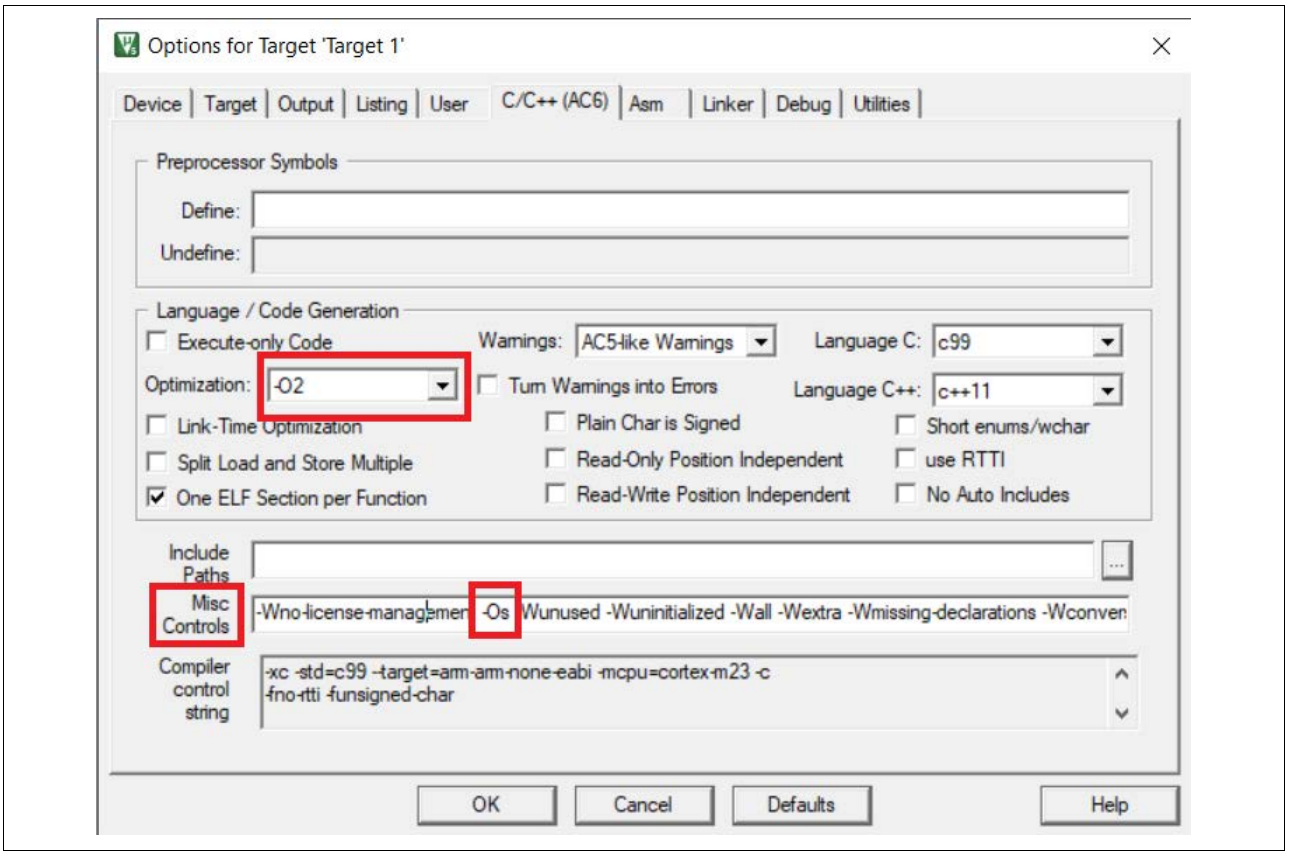

Figure 50. Optimization and Misc. Controls

5. Under **Debug** options, select **J-Link/J-TRACE Cortex** option. Also select the **Settings** option and ensure J-Link Settings are set and generated.

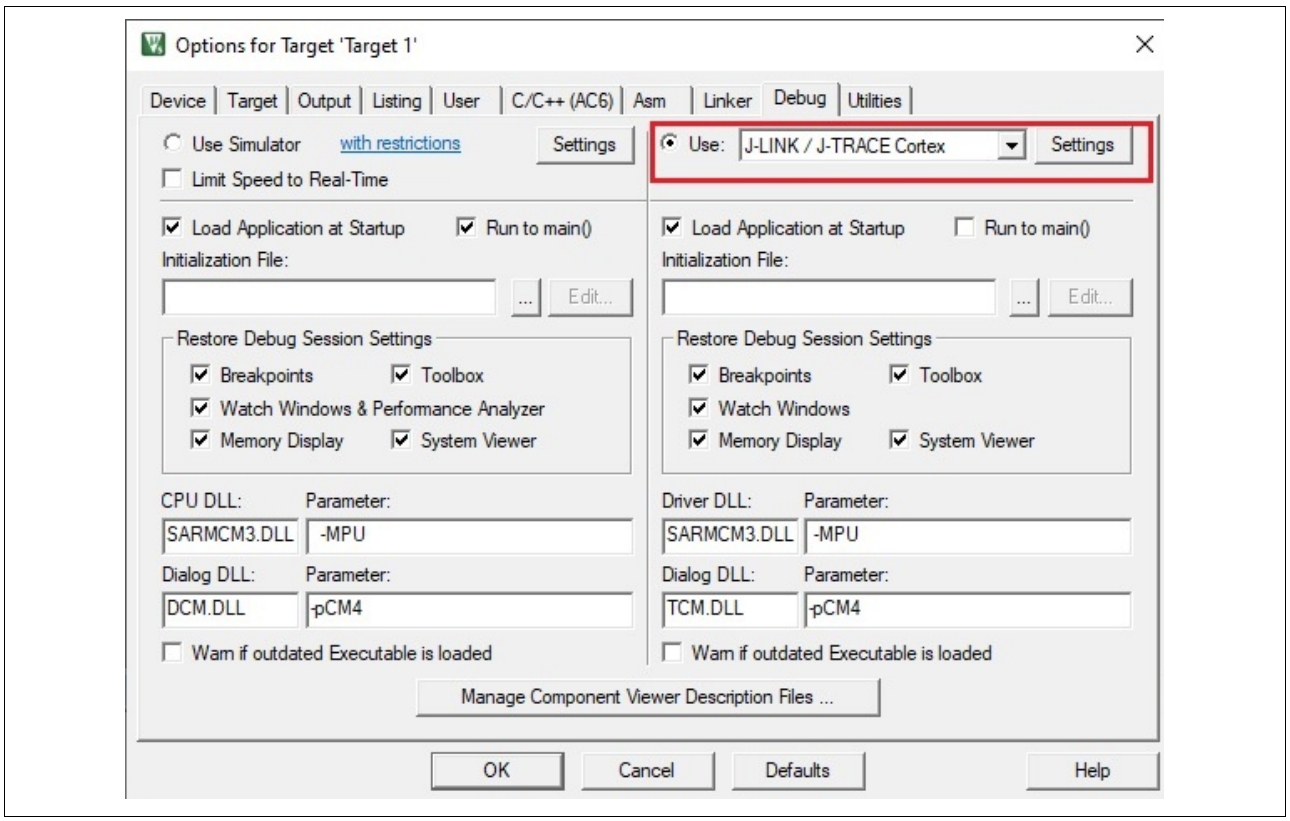

Figure 51. Debug Options

6. The following window appears after **Settings** is selected under debug window as shown above. Ensure the appropriate J-Link ID is selected.

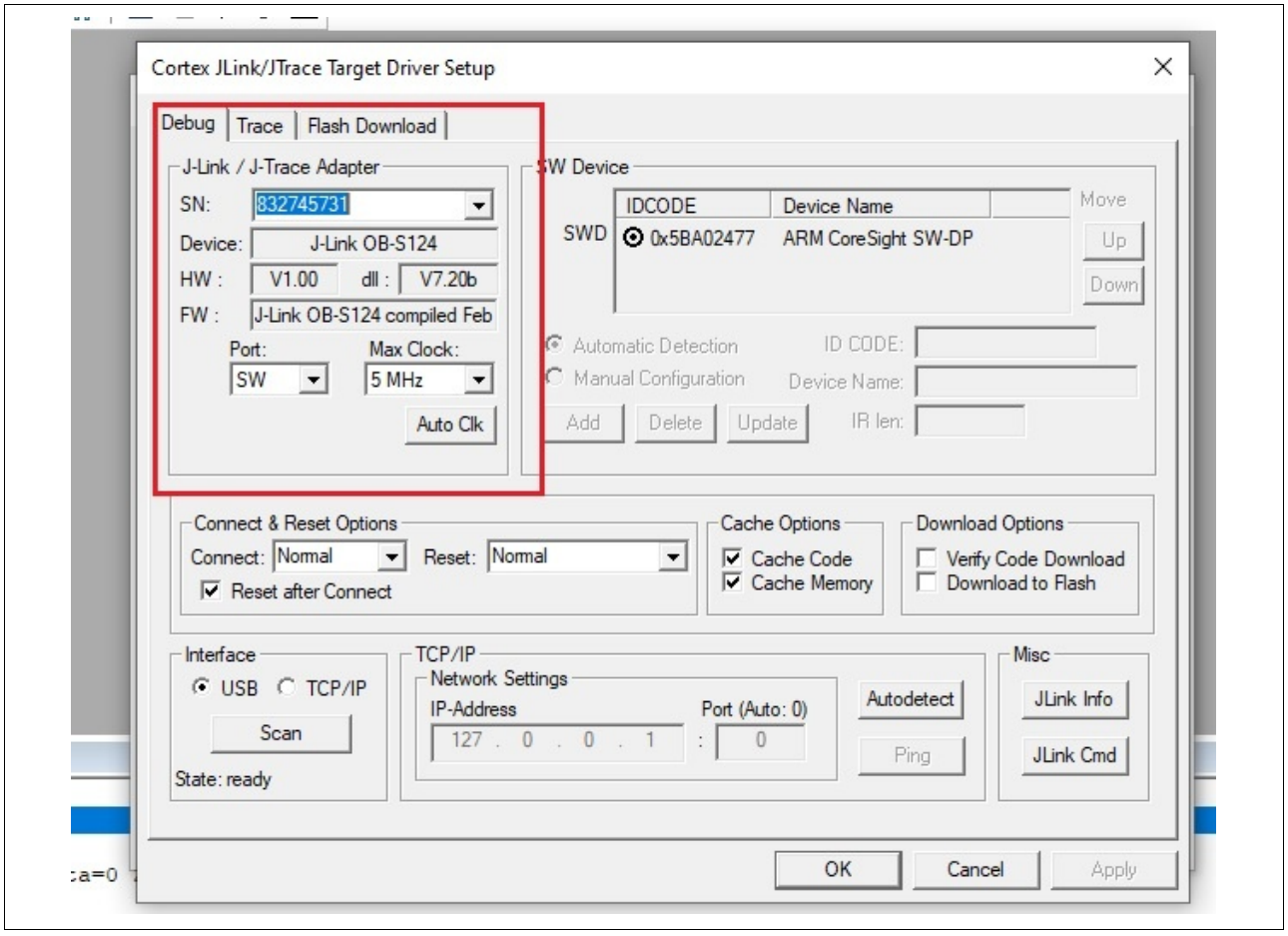

Figure 52. Select J-Link ID

7. Select **OK.**

8. Select the project. Select **Project** → **Build Target**. After the project builds, observe any warnings and errors.

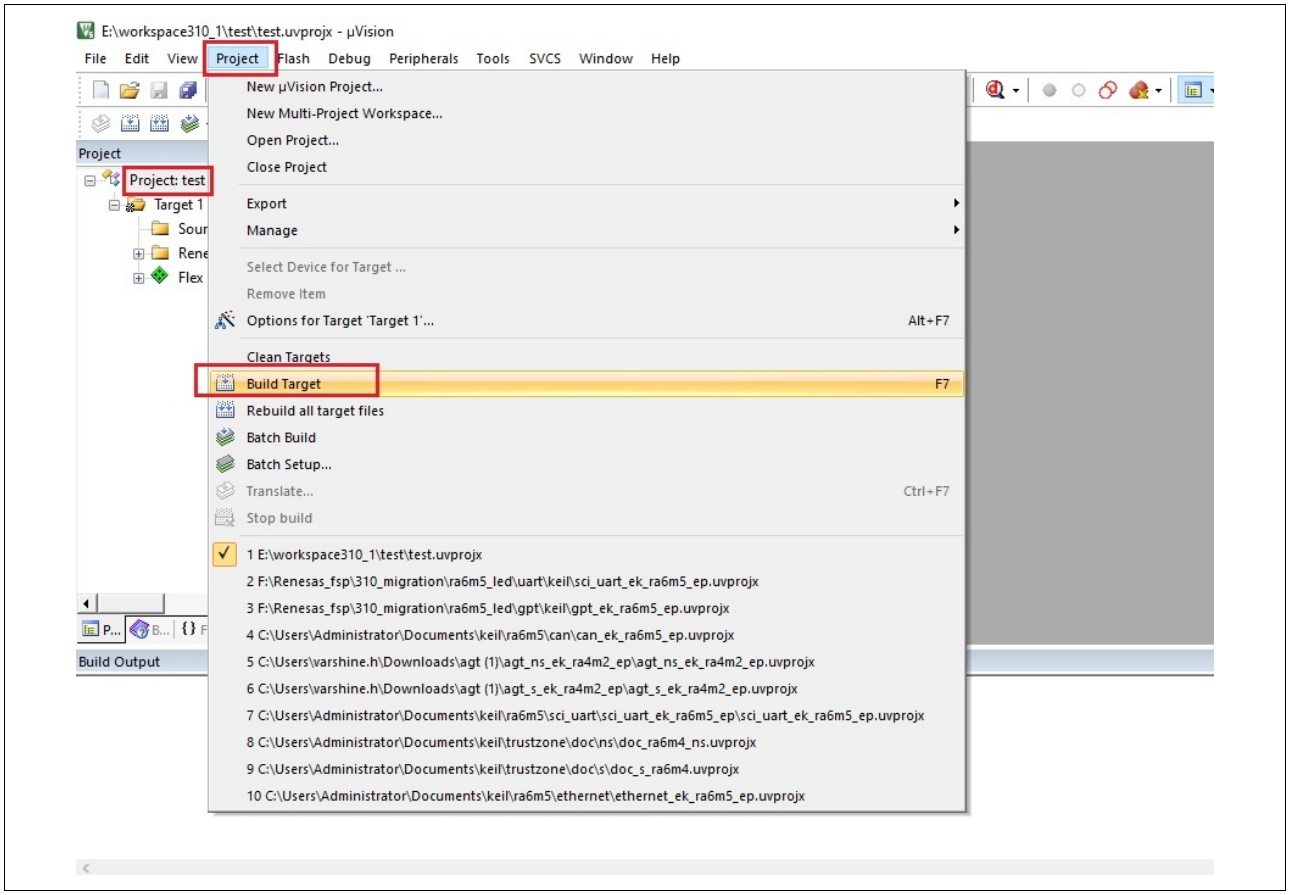

Figure 53. Build Target

2.5.5 Downloading and Debugging the Project

1. Upon successful project build with no errors and no warnings, select the **Start/Stop Debug Session** icon. Make sure the J-Link settings are set while configuring the **Target** options.

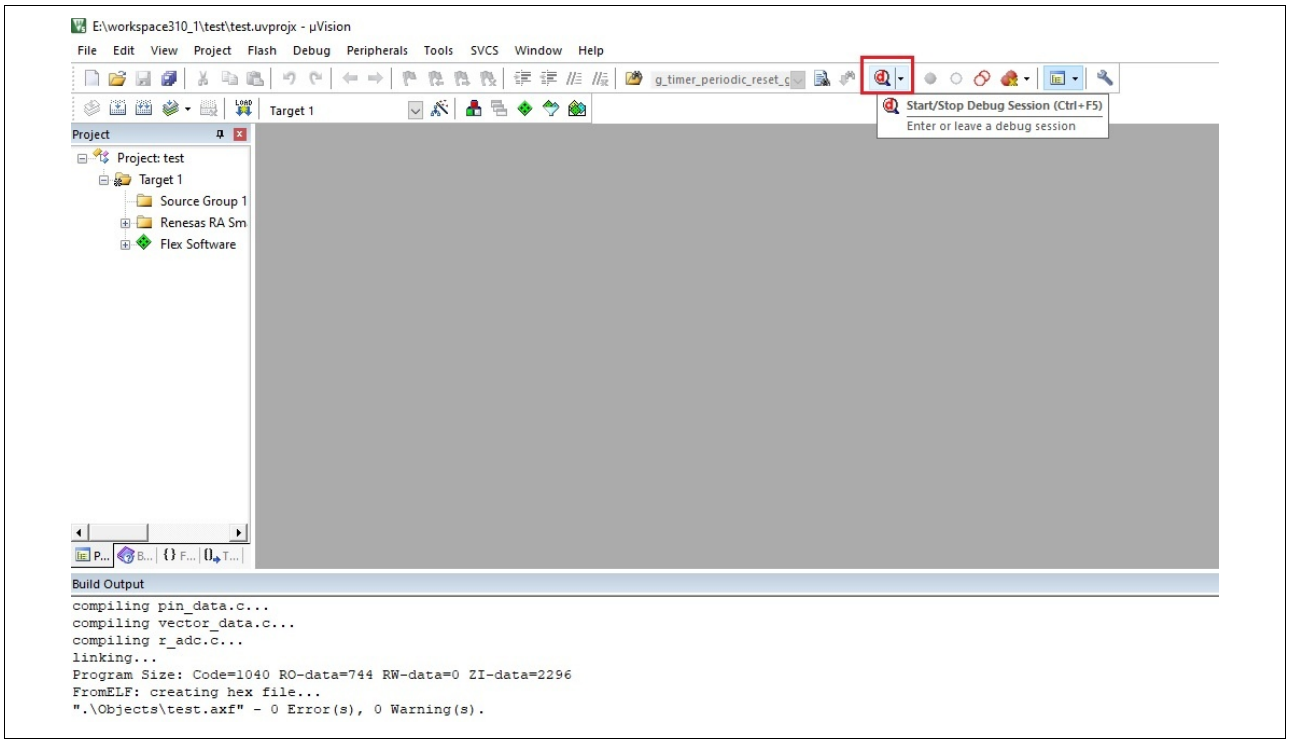

Figure 54. Start Debugging Session

2. The **Debug** window appears. Select the **Reset** and **Run** icons to run the project.

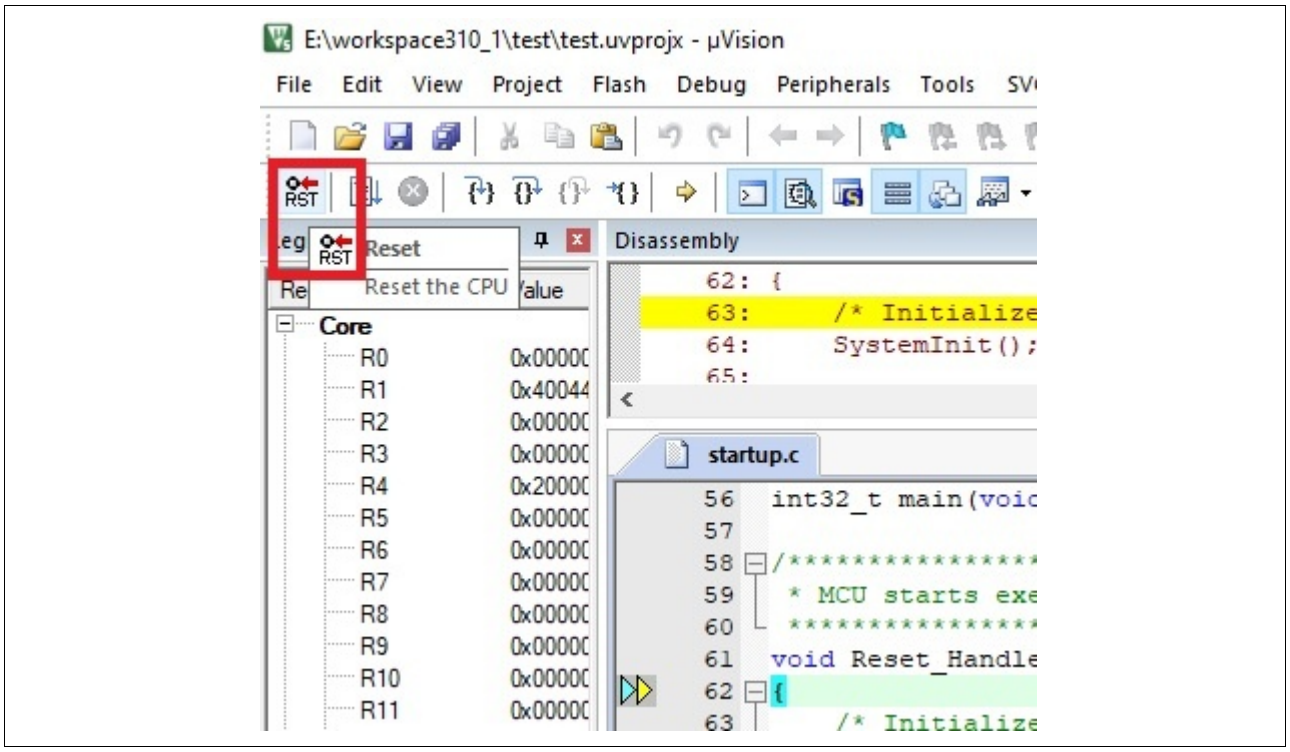

Figure 55. Reset Icon

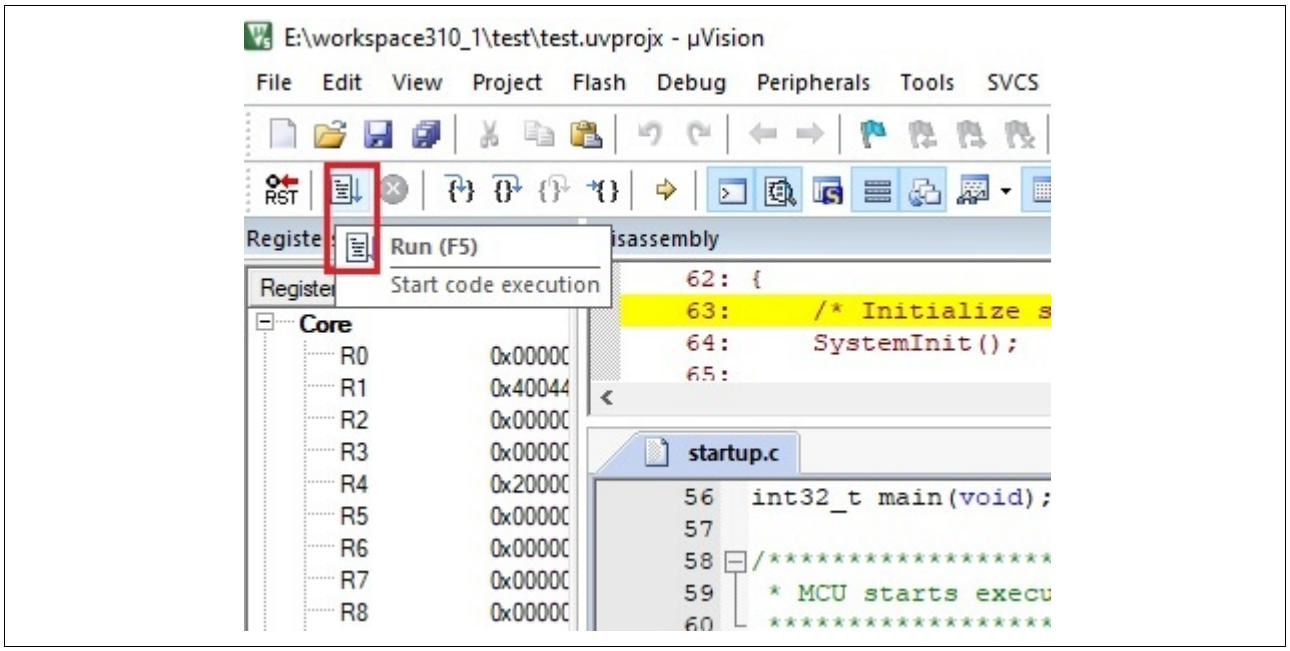

Figure 56. Run Icon

3. The debug session can be stopped by selecting the **Stop** icon.

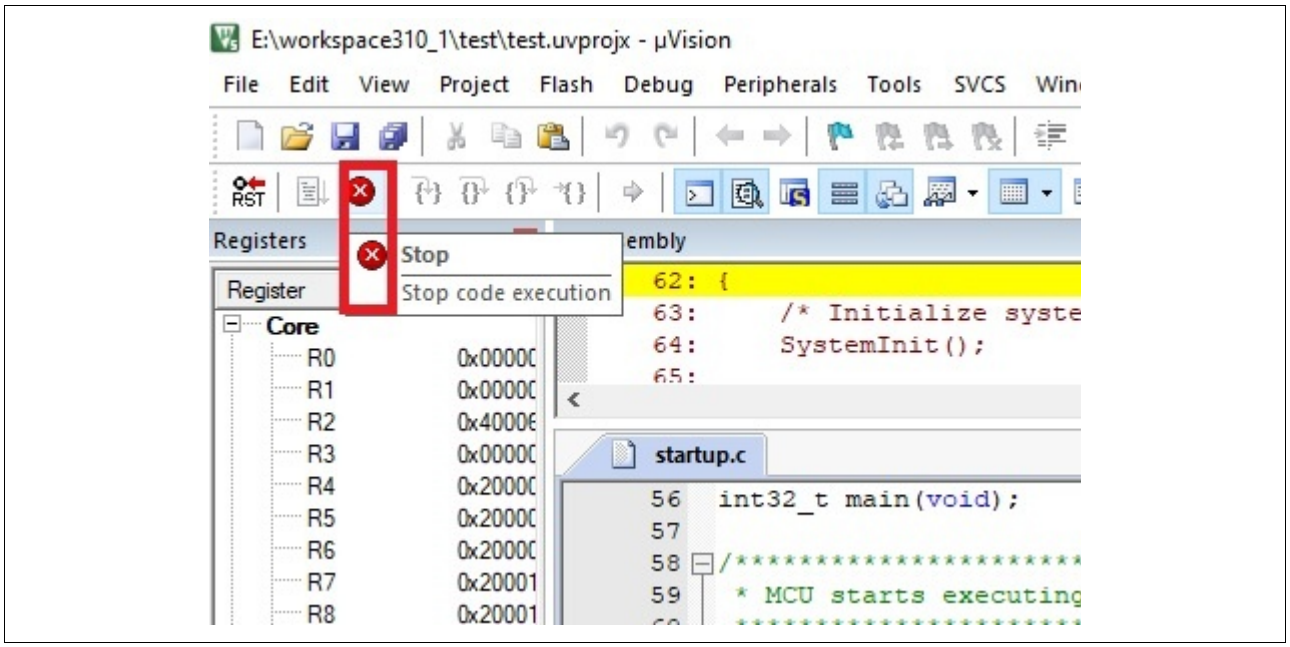

Figure 57. Stop Icon

After running, you can observe the output on SEGGER RTT Viewer (available only if it is compiled with and used by the application binary).

Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.15.21	—	First release document

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/.