RZ/A3UL

Getting Started with RZ/A Flexible Software Package

Introduction

This manual describes how to use the RZ/A Flexible Software Package (FSP) for developing applications for the RZ microprocessor series.

Target Device

RZ/A3UL

Contents

1. Introduction

1.1 Overview

This application note describes how to use the Renesas RZ/A Flexible Software Package (FSP) running on the Cortex®-A55 (hereinafter referred to as CA55) incorporated on RZ/A3UL.

1.2 Introduction to FSP

1.2.1 Purpose

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

1.2.2 e2 studio IDE

FSP provides a host of efficiency enhancing tools for developing projects targeting the Renesas RZ series of MPU devices. The e2 studio IDE provides a familiar development cockpit from which the key steps of project creation, module selection and configuration, code development, code generation, and debugging are all managed.

1.3 Limitations

1.3.1 Hardware Initial Setup

RZ/A FSP expects the initial setup of hardware should be carried out beforehand by RZ/A Initial Program Loader (hereinafter referred to as IPL). For detail on IPL, please refer to the Application Note.

2. Starting Development Introduction

2.1 e2 studio setup

2.1.1 What is e2 studio?

Renesas e2 studio is a development tool encompassing code development, build, and debug. e2 studio is based on the open-source Eclipse IDE and the associated C/C++ Development Tooling (CDT).

When developing for RZ MPUs, e2 studio hosts the RZ/A FSP. The FSP provides a wide range of time saving tools to simplify the selection, configuration, and management of modules and threads, to easily implement complex applications.

2.1.2 e2 studio Prerequisites

2.1.2.1 Obtaining an RZ MPU Kit

To develop applications with RZ/A FSP, start with RZ/A3UL Evaluation Board Kit.

RZ/A3UL Evaluation Board Kit related information is available at RZ/A3UL Evaluation Board Kit.

2.1.2.2 PC Requirements

The following are the minimum PC requirements to use e2 studio:

- Windows 10 or Ubuntu 20.04 LTS Desktop(64-bit) with Intel i5 or i7, or AMD A10-7850K or FX
- Memory: 8-GB DDR3 or DDR4 DRAM (16-GB DDR4/2400-MHz RAM is preferred)
- Minimum 250-GB hard disk

2.1.2.3 Licensing

FSP licensing includes full source code, limited to Renesas hardware only.

2.1.3 e2 studio installation for Windows PC

This chapter describes how to install the e2 studio IDE on Windows PC.

2.1.3.1 Download

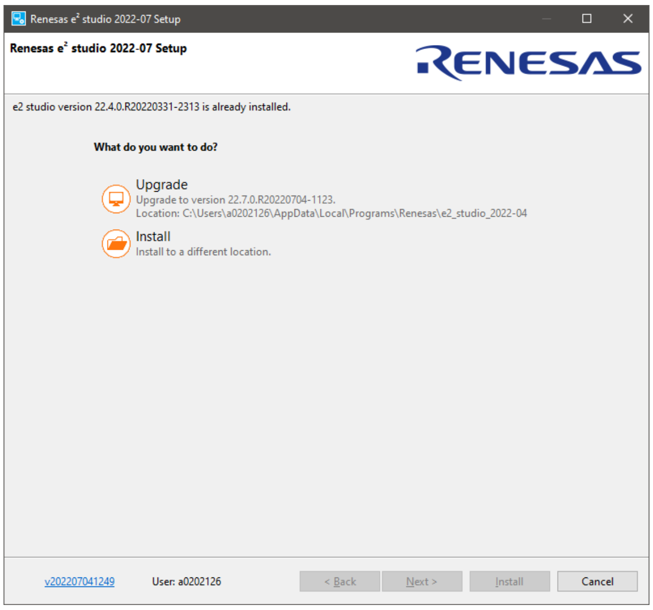
The latest e2 studio IDE installer package can be downloaded from Renesas website for free. Please check detailed information from: https://www.renesas.com/e2studio. Note that user has to login to the Renesas account (in MyRenesas page) for the software download.

2.1.3.2 Installation of e2 studio IDE

1. Double-click on e2 studio installer to invoke the e2 studio installation wizard page. Click [Install].

Note:

If e2 studio was installed in your PC, the option to upgrade the existing version or install e2 studio to a different location will be displayed



Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup

e2 studio version 22.4.0.R20220331-2313 is already installed.

What do you want to do?

Upgrade
Upgrade to version 22.7.0.R20220704-1123.
Location: C:\Users\a0202126\AppData\Local\Programs\Renesas\e2_studio_2022-04

Install
Install to a different location.

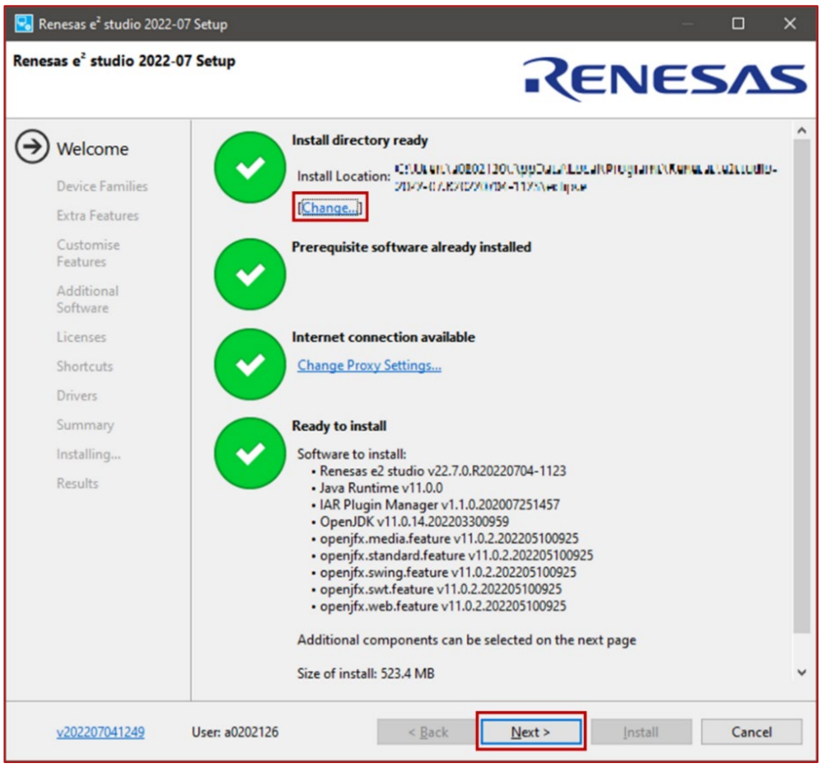
v202207041249 User: a0202126

Figure 1: e2 studio installation wizard

2. Welcome page
   User can change the install folder by clicking [Change…]. Click [Next] to continue.

Notes:

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



Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup

Install directory ready

Install Location: [illegible]

[Change...]

Prerequisite software already installed

Internet connection available

Change Proxy Settings...

Ready to install

Software to install:
- Renesas e2 studio v22.7.0.R20220704-1123
- Java Runtime v11.0.0
- IAR Plugin Manager v1.1.0.202007251457
- OpenJDK v11.0.14.202203300959
- openjfx.media.feature v11.0.2.202205100925
- openjfx.standard.feature v11.0.2.202205100925
- openjfx.swing.feature v11.0.2.202205100925
- openjfx.swt.feature v11.0.2.202205100925
- openjfx.web.feature v11.0.2.202205100925

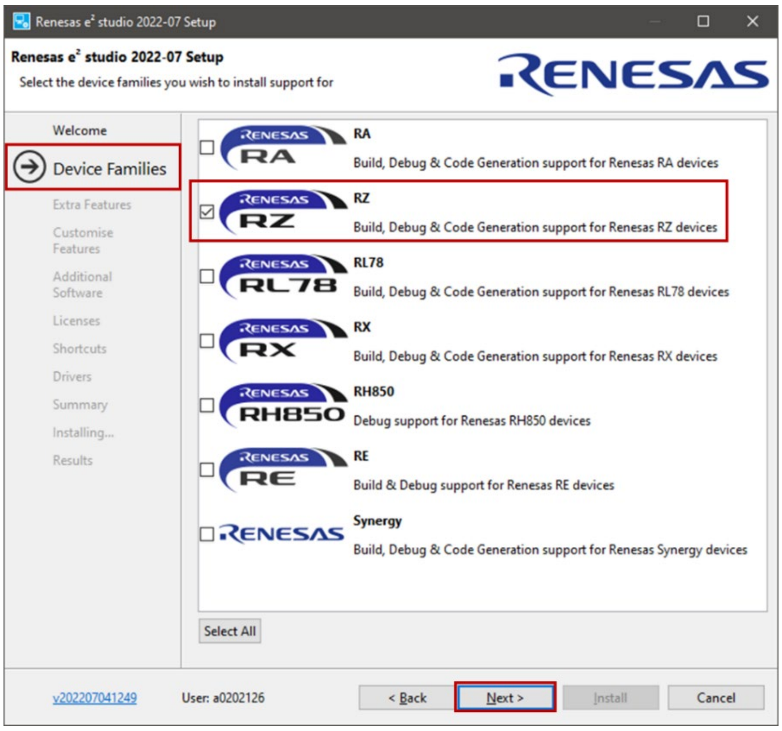
Additional components can be selected on the next page

Size of install: 523.4 MB

v202207041249 User: a0202126

Figure 2: Installation of e2 studio – Welcome page

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


Renesas e² studio 2022-07 Setup

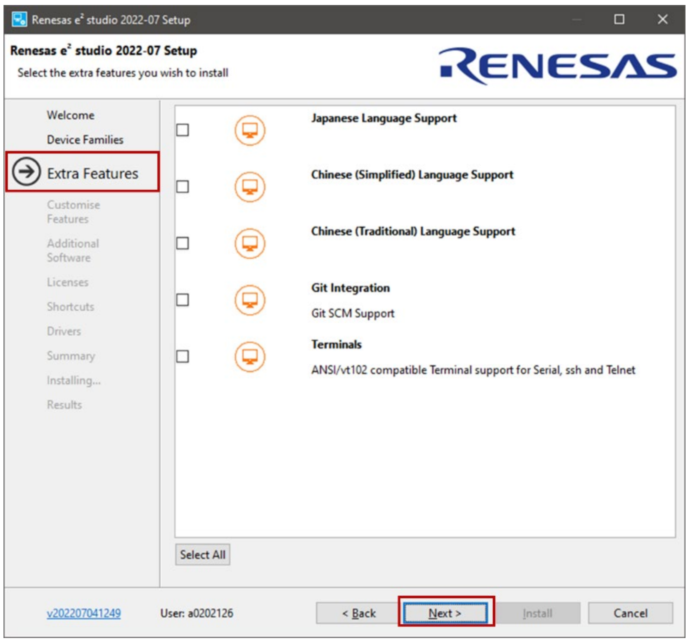
Renesas e² studio 2022-07 Setup
Select the device families you wish to install support for

- [ ] RA — Build, Debug & Code Generation support for Renesas RA devices
- [✓] RZ — Build, Debug & Code Generation support for Renesas RZ devices
- [ ] RL78 — Build, Debug & Code Generation support for Renesas RL78 devices
- [ ] RX — Build, Debug & Code Generation support for Renesas RX devices
- [ ] RH850 — Debug support for Renesas RH850 devices
- [ ] RE — Build & Debug support for Renesas RE devices
- [ ] Synergy — Build, Debug & Code Generation support for Renesas Synergy devices

v202207041249 User: a0202126

Figure 3: Installation of e2 studio – Device Families

4. Extra Features
   Select Extra Features (i.e., Language packs, SVN & Git support, RTOS support…) to be installed. For non-English language users, please select Language packs at this step if needed.
   Click the [Next] button to continue.


Renesas e² studio 2022-07 Setup

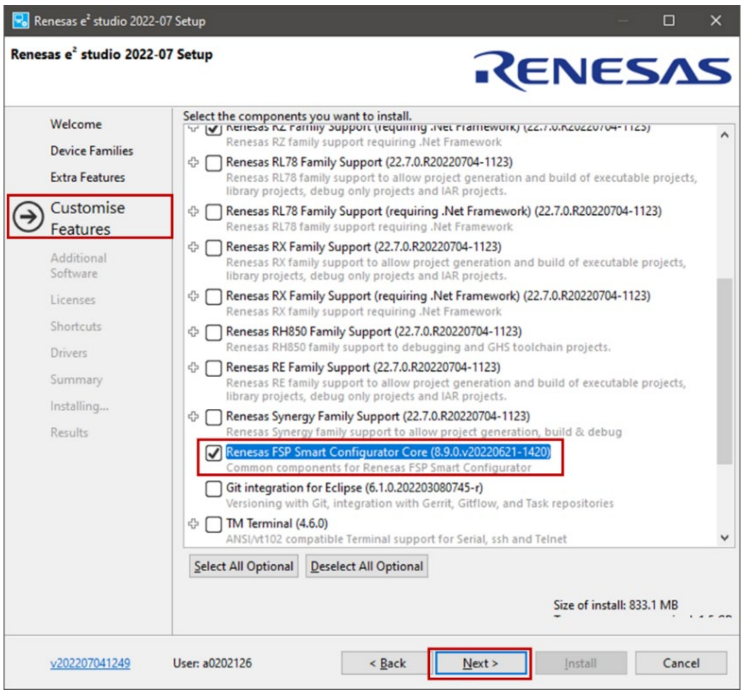
Renesas e² studio 2022-07 Setup
Select the extra features you wish to install

- [ ] Japanese Language Support
- [ ] Chinese (Simplified) Language Support
- [ ] Chinese (Traditional) Language Support
- [ ] Git Integration — Git SCM Support
- [ ] Terminals — ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet

v202207041249 User: a0202126

Figure 4: Installation of e2 studio – Extra Features

5. Customise Features
   Select the components to install and click the [Next] button to continue. Be sure to choose “Renesas FSP Smart Configurator Core”. Otherwise, FSP won’t be built on e2 studio successfully.



Select the components you want to install.

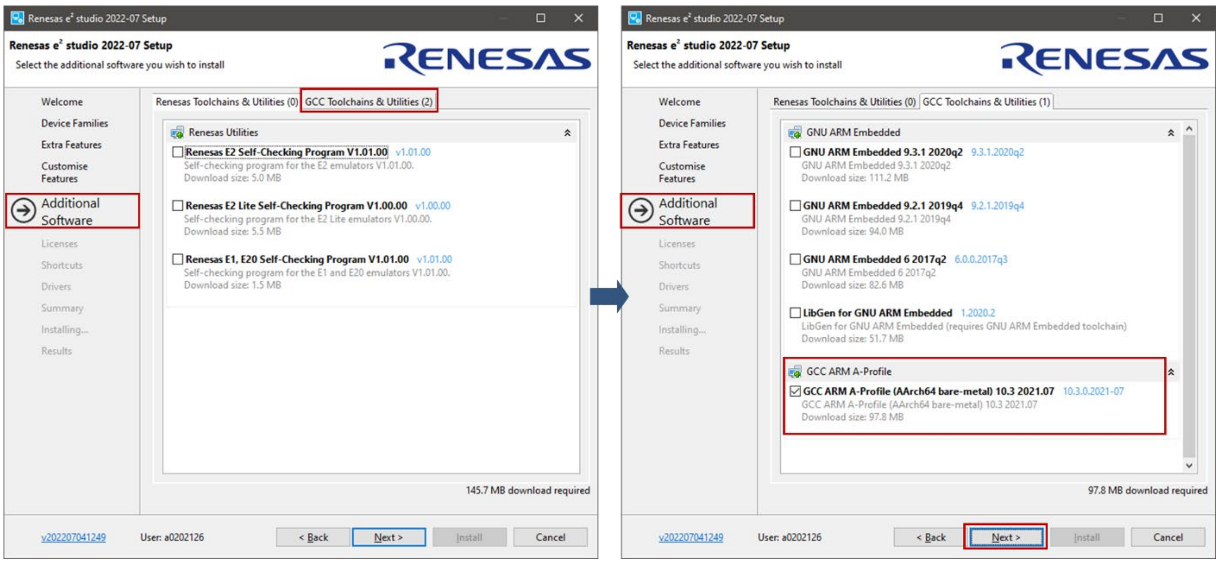
- [✓] Renesas RZ Family Support (requiring .Net Framework) (22.7.0.R20220704-1123)
  Renesas RZ family support requiring .Net Framework
- [ ] Renesas RL78 Family Support (22.7.0.R20220704-1123)
  Renesas RL78 family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
- [ ] Renesas RL78 Family Support (requiring .Net Framework) (22.7.0.R20220704-1123)
  Renesas RL78 family support requiring .Net Framework
- [ ] Renesas RX Family Support (22.7.0.R20220704-1123)
  Renesas RX family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
- [ ] Renesas RX Family Support (requiring .Net Framework) (22.7.0.R20220704-1123)
  Renesas RX family support requiring .Net Framework
- [ ] Renesas RH850 Family Support (22.7.0.R20220704-1123)
  Renesas RH850 family support to debugging and GHS toolchain projects.
- [ ] Renesas RE Family Support (22.7.0.R20220704-1123)
  Renesas RE family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
- [ ] Renesas Synergy Family Support (22.7.0.R20220704-1123)
  Renesas Synergy family support to allow project generation, build & debug
- [✓] Renesas FSP Smart Configurator Core (8.9.0.v20220621-1420)
  Common components for Renesas FSP Smart Configurator
- [ ] Git integration for Eclipse (6.1.0.202203080745-r)
  Versioning with Git, integration with Gerrit, Gitflow, and Task repositories
- [ ] TM Terminal (4.6.0)
  ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet

Size of install: 833.1 MB

Figure 5: Installation of e2 studio – Features

6. Additional Software
   Select additional software (i.e., compilers, utilities, QE…) to be installed. Be sure that you select the "GCC ToolChains & Utilities" tab, choose the following items, and click [Next] to continue:

   - GCC ARM A-Profile (AArch64 bare-metal) 10.3.2021.07


Renesas e² studio 2022-07 Setup
Renesas e² studio 2022-07 Setup
Select the additional software you wish to install
Welcome
Device Families
Extra Features
Customise Features
Additional Software
Licenses
Shortcuts
Drivers
Summary
Installing...
Results
Renesas Toolchains & Utilities (0)
GCC Toolchains & Utilities (2)
v202207041249
User: a0202126
< Back
Next >
Install
Cancel
Renesas Toolchains & Utilities (0)
GCC Toolchains & Utilities (1)

Renesas Utilities

- [ ] Renesas E2 Self-Checking Program V1.01.00 v1.01.00
  Self-checking program for the E2 emulators V1.01.00.
  Download size: 5.0 MB
- [ ] Renesas E2 Lite Self-Checking Program V1.00.00 v1.00.00
  Self-checking program for the E2 Lite emulators V1.00.00.
  Download size: 5.5 MB
- [ ] Renesas E1, E20 Self-Checking Program V1.01.00 v1.01.00
  Self-checking program for the E1 and E20 emulators V1.01.00.
  Download size: 1.5 MB

145.7 MB download required

GNU ARM Embedded

- [ ] GNU ARM Embedded 9.3.1 2020q2 9.3.1.2020q2
  GNU ARM Embedded 9.3.1 2020q2
  Download size: 111.2 MB
- [ ] GNU ARM Embedded 9.2.1 2019q4 9.2.1.2019q4
  GNU ARM Embedded 9.2.1 2019q4
  Download size: 94.0 MB
- [ ] GNU ARM Embedded 6 2017q2 6.0.0.2017q3
  GNU ARM Embedded 6 2017q2
  Download size: 82.6 MB
- [ ] LibGen for GNU ARM Embedded 1.2020.2
  LibGen for GNU ARM Embedded (requires GNU ARM Embedded toolchain)
  Download size: 51.7 MB

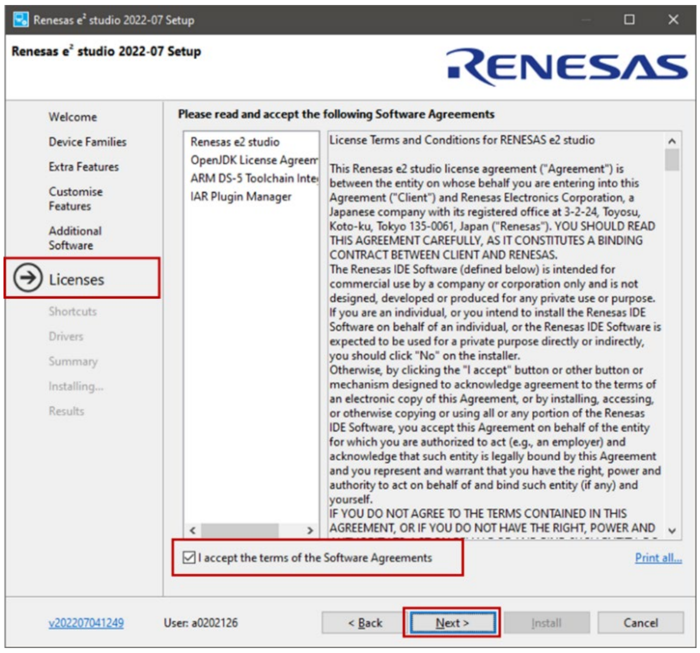
GCC ARM A-Profile

- [✓] GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 10.3.0.2021-07
  GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07
  Download size: 97.8 MB

97.8 MB download required

Figure 6: Installation of e2 studio – Additional Software

7. Licenses Agreement
   Read and accept the software license agreement. Click the [Next] button. Please note that user must accept the license agreement, otherwise installation cannot be continued.



Please read and accept the following Software Agreements

License Terms and Conditions for RENESAS e2 studio

This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS.
The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer.
Otherwise, by clicking the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of this Agreement, or by installing, accessing, or otherwise copying or using all or any portion of the Renesas IDE Software, you accept this Agreement on behalf of the entity for which you are authorized to act (e.g., an employer) and acknowledge that such entity is legally bound by this Agreement and you represent and warrant that you have the right, power and authority to act on behalf of and bind such entity (if any) and yourself.
IF YOU DO NOT AGREE TO THE TERMS CONTAINED IN THIS AGREEMENT, OR IF YOU DO NOT HAVE THE RIGHT, POWER AND

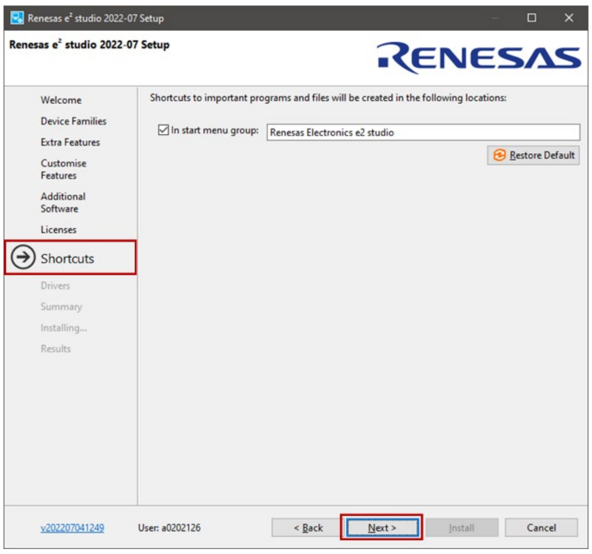
☑ I accept the terms of the Software Agreements

Figure 7: Installation of e2 studio - Licenses

8. Shortcuts
   Select shortcut name for start menu and click [Next] button to continue.

Note:

If e2 studio was installed in another location, it is recommended to rename to distinguish from the other e2 studio(s).

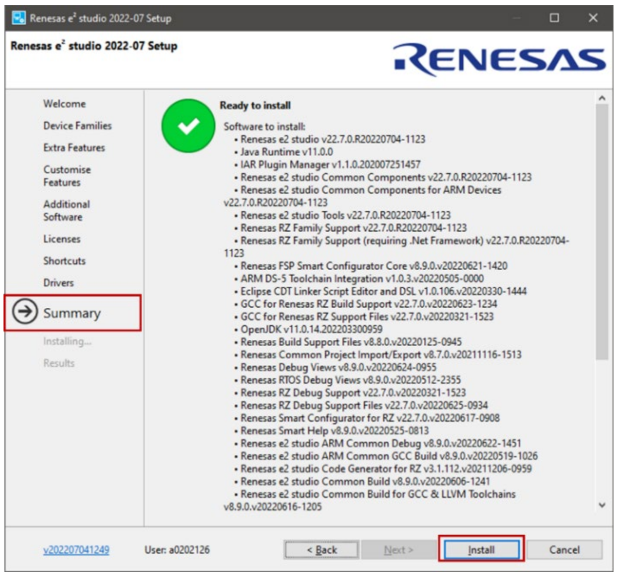


Shortcuts to important programs and files will be created in the following locations:

☑ In start menu group: Renesas Electronics e2 studio

Figure 8: Installation of e2 studio – Shortcuts

9. Summary
   Components list to be installed is shown. Please confirm the contents and click the [Install] button to install the Renesas e2 studio IDE.



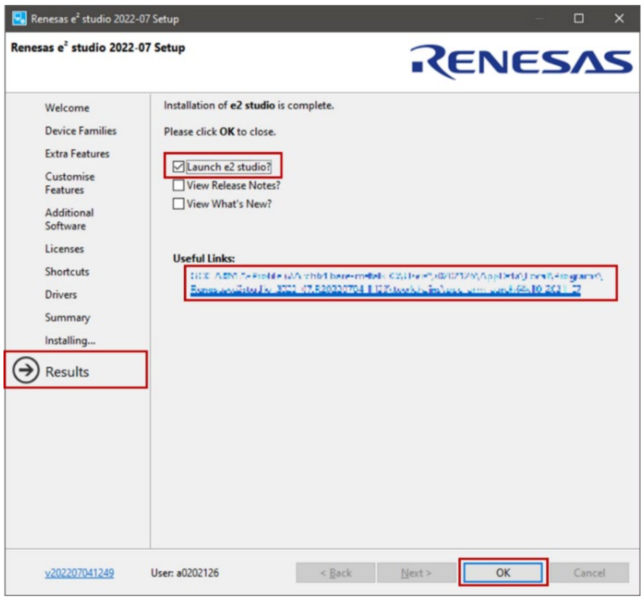
Ready to install

Software to install:
- Renesas e2 studio v22.7.0.R20220704-1123
- Java Runtime v11.0.0
- IAR Plugin Manager v1.1.0.202007251457
- Renesas e2 studio Common Components v22.7.0.R20220704-1123
- Renesas e2 studio Common Components for ARM Devices v22.7.0.R20220704-1123
- Renesas e2 studio Tools v22.7.0.R20220704-1123
- Renesas RZ Family Support v22.7.0.R20220704-1123
- Renesas RZ Family Support (requiring .Net Framework) v22.7.0.R20220704-1123
- Renesas FSP Smart Configurator Core v8.9.0.v20220621-1420
- ARM DS-5 Toolchain Integration v1.0.3.v20220505-0000
- Eclipse CDT Linker Script Editor and DSL v1.0.106.v20220330-1444
- GCC for Renesas RZ Build Support v22.7.0.v20220623-1234
- GCC for Renesas RZ Support Files v22.7.0.v20220321-1523
- OpenJDK v11.0.14.202203300959
- Renesas Build Support Files v8.8.0.v20220125-0945
- Renesas Common Project Import/Export v8.7.0.v20211116-1513
- Renesas Debug Views v8.9.0.v20220624-0955
- Renesas RTOS Debug Views v8.9.0.v20220512-2355
- Renesas RZ Debug Support v22.7.0.v20220321-1523
- Renesas RZ Debug Support Files v22.7.0.v20220625-0934
- Renesas Smart Configurator for RZ v22.7.0.v20220617-0908
- Renesas Smart Help v8.9.0.v20220525-0813
- Renesas e2 studio ARM Common Debug v8.9.0.v20220622-1451
- Renesas e2 studio ARM Common GCC Build v8.9.0.v20220519-1026
- Renesas e2 studio Code Generator for RZ v3.1.112.v20211206-0959
- Renesas e2 studio Common Build v8.9.0.v20220606-1241
- Renesas e2 studio Common Build for GCC & LLVM Toolchains v8.9.0.v20220616-1205

Figure 9: Installation of e2 studio - Summary

10. Installing...
    The installation is performed. Depending on selected items of additional software, new dialog prompts may appear during the installation process. At that time, please follow the instruction the installer indicates.

11. Results
    If the installation is successfully done, you should see the following information. Please note that the link to the installation place of GCC ARM A-Profile (AArch64 bare-metal) 10.3.2021.07 you selected as Additional Software is shown at "**Useful Links:**".



Installation of **e2 studio** is complete.

Please click **OK** to close.

[✓] Launch e2 studio?
[ ] View Release Notes?
[ ] View What's New?

Useful Links:
[illegible]

Figure 10: Results Page

2.1.4 e2 studio installation for Linux PC

This chapter describes how to install the e2 studio IDE on Linux PC.

2.1.4.1 Prerequisite

Please download the development tool related stuff:

- **SEGGER J-Link driver**
  Please download the driver V7.68 or after from:
  https://www.segger.com/downloads/jlink/JLink_Linux_x86_64.deb

- **e2 studio IDE installer**
  The latest e2 studio IDE installer package can be downloaded from Renesas website for free. Please check detailed information from: https://www.renesas.com/e2studio.

2.1.4.2 Installation

This section describes the procedure of each software installation. Filename, version number and the file path are just examples. Please replace those in accordance with your environment.

- **SEGGER J-Link driver**
  Open a terminal window and enter below commands.
  sudo dpkg -i JLink_Linux_V768_x86_64.deb
  (If the previous install fails with unmet dependencies, retry it as follows)
  sudo apt-get -f install
  sudo dpkg -i JLink_Linux_V768_x86_64.deb

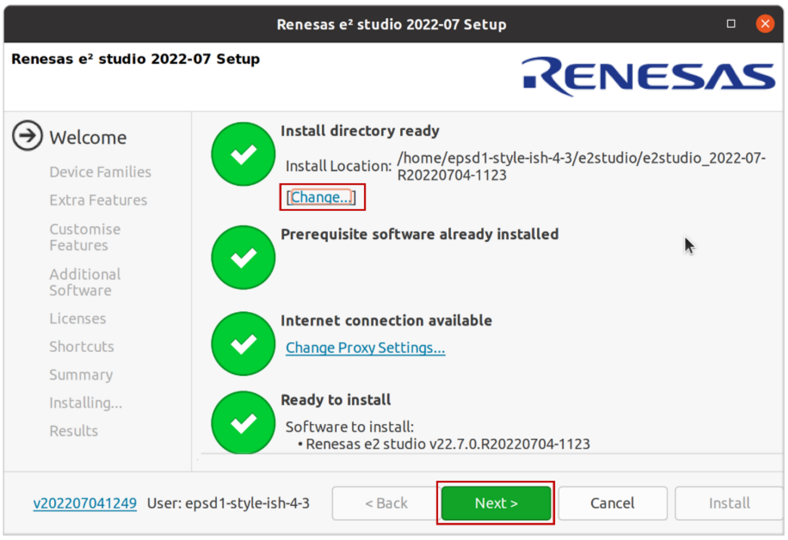
- **e2 studio IDE**

1. Run the e2 studio IDE Installer "./e2studio_installer-yyyy-mm_linux_host.run". (Before running the installer, check the execution permission of the installer.)

2. Welcome page
   User can change the install folder by clicking [Change…]. Click [Next] to continue.

Notes:

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


Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup

Install directory ready

Install Location: /home/epsd1-style-ish-4-3/e2studio/e2studio_2022-07-R20220704-1123

[Change...]

Prerequisite software already installed

Internet connection available

Change Proxy Settings...

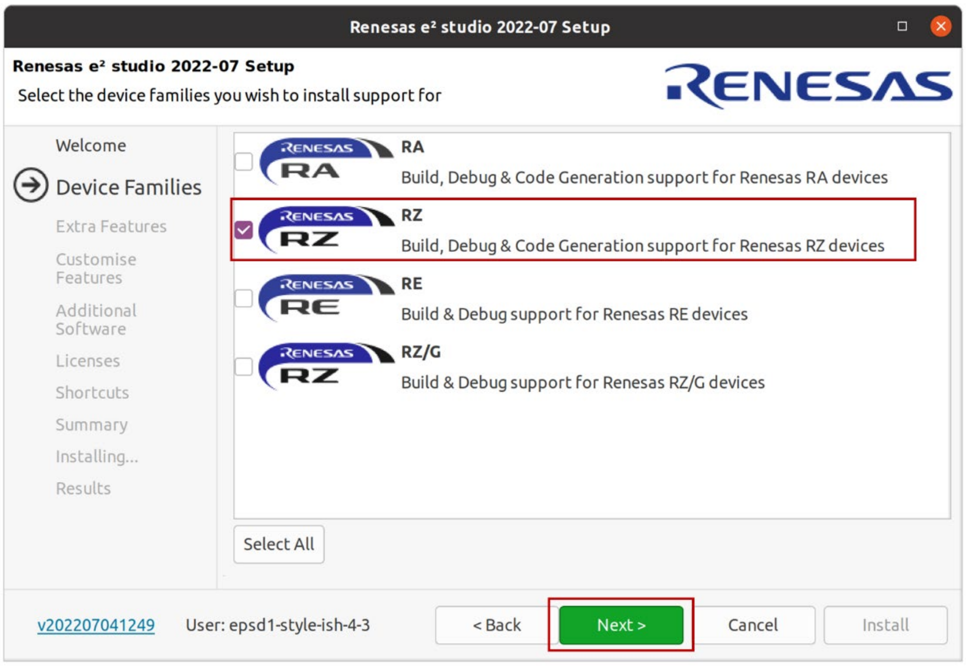
Ready to install

Software to install:
• Renesas e2 studio v22.7.0.R20220704-1123

v202207041249 User: epsd1-style-ish-4-3

Figure 11: Installation of e2 studio – Welcome page

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


Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup
Select the device families you wish to install support for

☐ RA
Build, Debug & Code Generation support for Renesas RA devices

☑ RZ
Build, Debug & Code Generation support for Renesas RZ devices

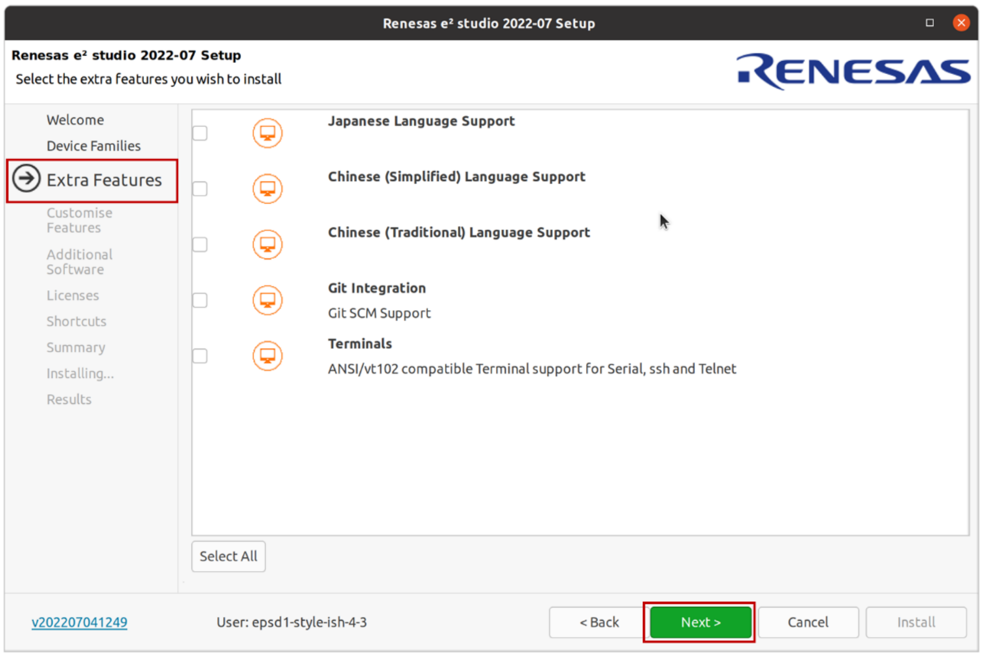
☐ RE
Build & Debug support for Renesas RE devices

☐ RZ/G
Build & Debug support for Renesas RZ/G devices

v202207041249 User: epsd1-style-ish-4-3

Figure 12: Installation of e2 studio – Device Families

4. Extra Features
   Select Extra Features (i.e., Language packs, SVN & Git support, RTOS support…) to be installed. For non-English language users, please select Language packs at this step if needed. Then, please click the [Next] button to continue.



Renesas e² studio 2022-07 Setup
Select the extra features you wish to install

Japanese Language Support

Chinese (Simplified) Language Support

Chinese (Traditional) Language Support

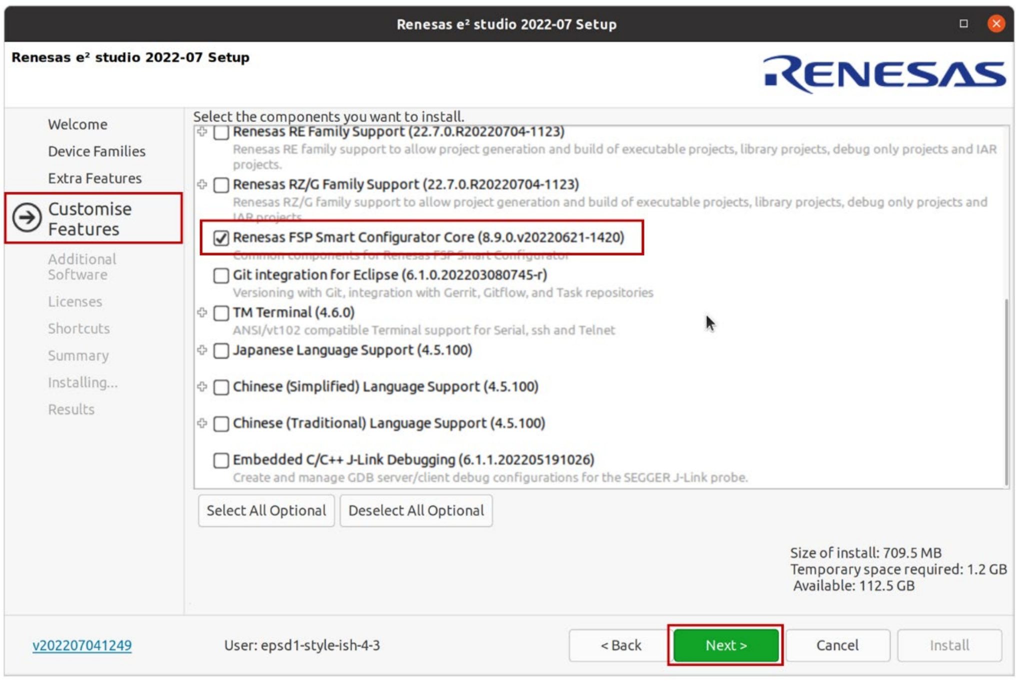
Git Integration
Git SCM Support

Terminals
ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet

v202207041249 User: epsd1-style-ish-4-3

Figure 13: Installation of e2 studio – Extra Features

5. Customize Features
   Select the components to install and click the [Next] button to continue. Be sure that "Renesas FSP Smart Configurator Core" is certainly selected.



Renesas e² studio 2022-07 Setup

Select the components you want to install.

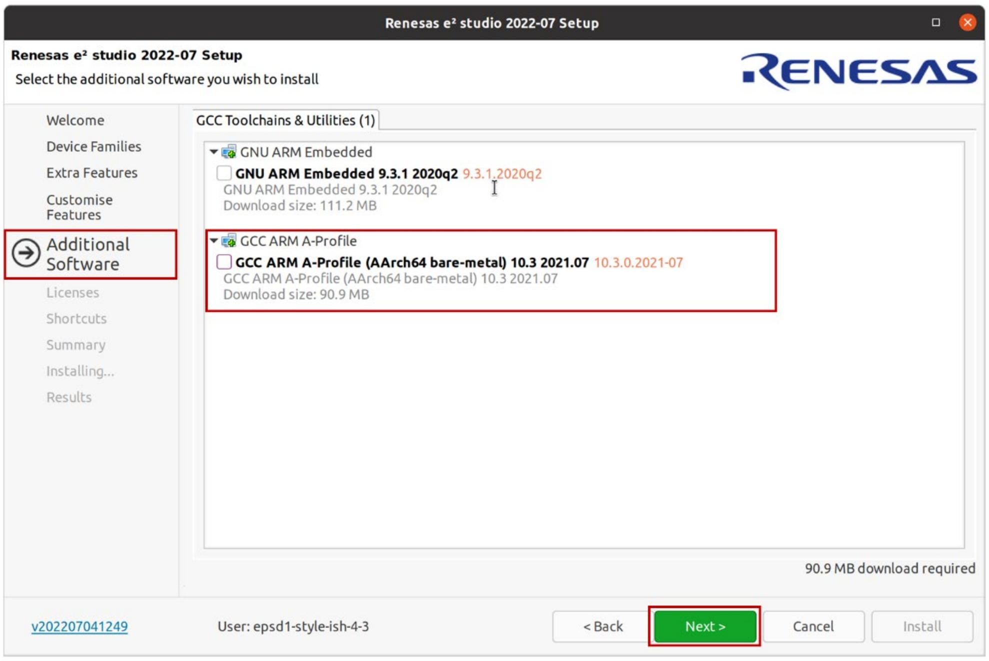
- [ ] Renesas RE Family Support (22.7.0.R20220704-1123)
  Renesas RE family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
- [ ] Renesas RZ/G Family Support (22.7.0.R20220704-1123)
  Renesas RZ/G family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
- [x] Renesas FSP Smart Configurator Core (8.9.0.v20220621-1420)
  Common components for Renesas FSP Smart Configurator
- [ ] Git integration for Eclipse (6.1.0.202203080745-r)
  Versioning with Git, integration with Gerrit, Gitflow, and Task repositories
- [ ] TM Terminal (4.6.0)
  ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
- [ ] Japanese Language Support (4.5.100)
- [ ] Chinese (Simplified) Language Support (4.5.100)
- [ ] Chinese (Traditional) Language Support (4.5.100)
- [ ] Embedded C/C++ J-Link Debugging (6.1.1.202205191026)
  Create and manage GDB server/client debug configurations for the SEGGER J-Link probe.

Size of install: 709.5 MB
Temporary space required: 1.2 GB
Available: 112.5 GB

v202207041249 User: epsd1-style-ish-4-3

Figure 14: Installation of e2 studio – Features

6. Additional Software
   Be sure to choose “GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07” as the additional software to be installed. Then click [Next].



Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup
Select the additional software you wish to install

GNU ARM Embedded
- [ ] **GNU ARM Embedded 9.3.1 2020q2** 9.3.1.2020q2
  GNU ARM Embedded 9.3.1 2020q2
  Download size: 111.2 MB

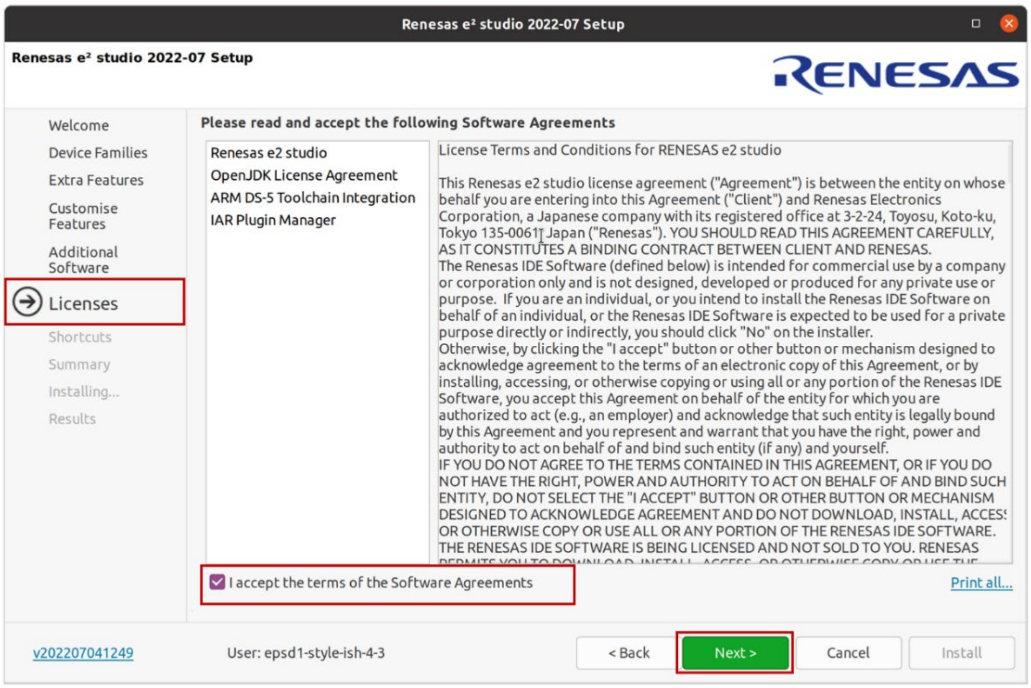
GCC ARM A-Profile
- [ ] **GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07** 10.3.0.2021-07
  GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07
  Download size: 90.9 MB

90.9 MB download required

v202207041249 User: epsd1-style-ish-4-3

Figure 15: Installation of e2 studio – Additional Software

7. License Agreement
   Read and accept the software license agreement. Click the [Next] button. Please note that user must accept the license agreement, otherwise installation cannot be continued.



Renesas e² studio 2022-07 Setup

Renesas e² studio 2022-07 Setup

Please read and accept the following Software Agreements

Renesas e2 studio
OpenJDK License Agreement
ARM DS-5 Toolchain Integration
IAR Plugin Manager

License Terms and Conditions for RENESAS e2 studio

This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061 Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS.
The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer.
Otherwise, by clicking the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of this Agreement, or by installing, accessing, or otherwise copying or using all or any portion of the Renesas IDE Software, you accept this Agreement on behalf of the entity for which you are authorized to act (e.g., an employer) and acknowledge that such entity is legally bound by this Agreement and you represent and warrant that you have the right, power and authority to act on behalf of and bind such entity (if any) and yourself.
IF YOU DO NOT AGREE TO THE TERMS CONTAINED IN THIS AGREEMENT, OR IF YOU DO NOT HAVE THE RIGHT, POWER AND AUTHORITY TO ACT ON BEHALF OF AND BIND SUCH ENTITY, DO NOT SELECT THE "I ACCEPT" BUTTON OR OTHER BUTTON OR MECHANISM DESIGNED TO ACKNOWLEDGE AGREEMENT AND DO NOT DOWNLOAD, INSTALL, ACCES OR OTHERWISE COPY OR USE ALL OR ANY PORTION OF THE RENESAS IDE SOFTWARE. THE RENESAS IDE SOFTWARE IS BEING LICENSED AND NOT SOLD TO YOU. RENESAS

- [x] I accept the terms of the Software Agreements

Print all...

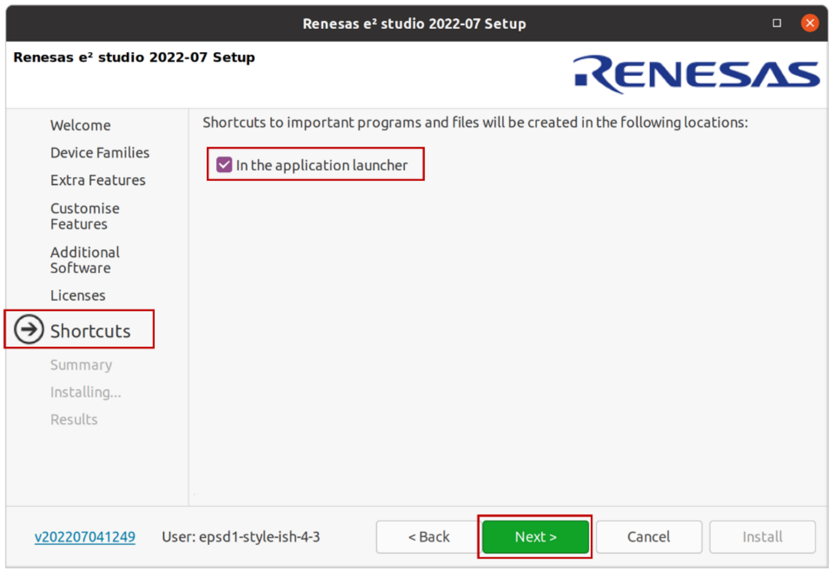
v202207041249 User: epsd1-style-ish-4-3

Figure 16: Installation of e2 studio – Licenses

8. Shortcuts
   Select shortcut name for start menu and click [Next] button to continue.

Note:

If e2 studio was installed in another location, it is recommended to rename to distinguish from the other e2 studio(s).

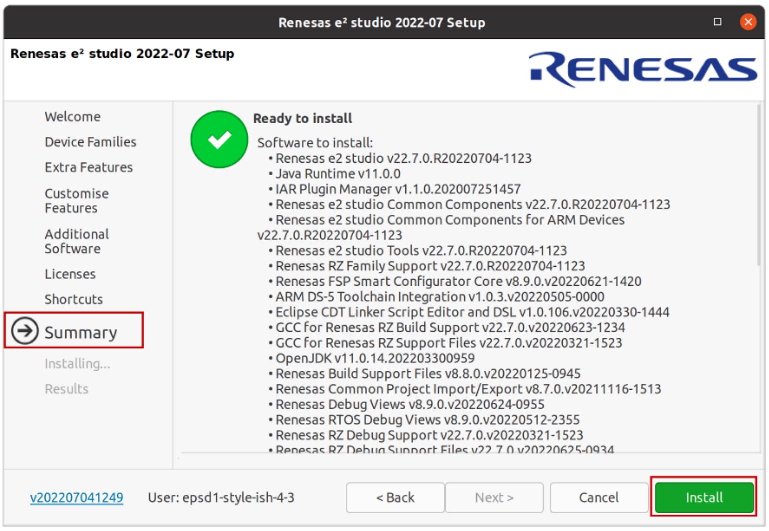


Shortcuts to important programs and files will be created in the following locations:

☑ In the application launcher

Figure 17: Installation of e2 studio – Shortcuts

9. Summary
   Components list to be installed is shown. Please confirm the contents and click the [Install] button to install the Renesas e2 studio IDE.



Ready to install

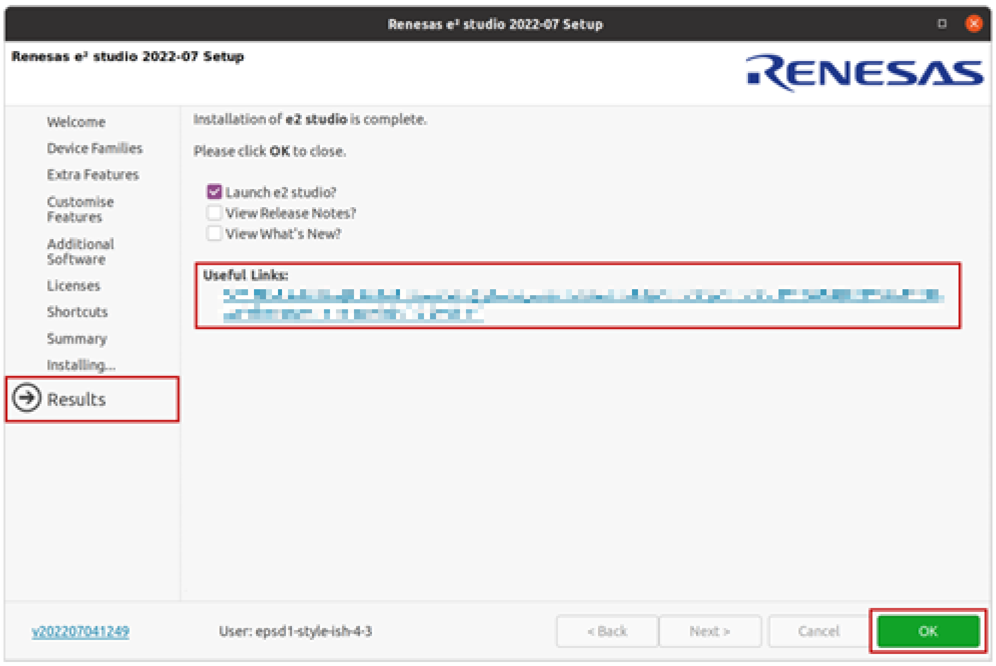
Software to install:
- Renesas e2 studio v22.7.0.R20220704-1123
- Java Runtime v11.0.0
- IAR Plugin Manager v1.1.0.202007251457
- Renesas e2 studio Common Components v22.7.0.R20220704-1123
- Renesas e2 studio Common Components for ARM Devices v22.7.0.R20220704-1123
- Renesas e2 studio Tools v22.7.0.R20220704-1123
- Renesas RZ Family Support v22.7.0.R20220704-1123
- Renesas FSP Smart Configurator Core v8.9.0.v20220621-1420
- ARM DS-5 Toolchain Integration v1.0.3.v20220505-0000
- Eclipse CDT Linker Script Editor and DSL v1.0.106.v20220330-1444
- GCC for Renesas RZ Build Support v22.7.0.v20220623-1234
- GCC for Renesas RZ Support Files v22.7.0.v20220321-1523
- OpenJDK v11.0.14.202203300959
- Renesas Build Support Files v8.8.0.v20220125-0945
- Renesas Common Project Import/Export v8.7.0.v20211116-1513
- Renesas Debug Views v8.9.0.v20220624-0955
- Renesas RTOS Debug Views v8.9.0.v20220512-2355
- Renesas RZ Debug Support v22.7.0.v20220321-1523
- Renesas RZ Debug Support Files v22.7.0.v20220625-0934

Figure 18: Installation of e2 studio – Summary

10. Installing...
    The installation is performed. Depending on selected items of additional software, new dialog prompts may appear during the installation process. Please see chapter 0 for more detailed information.

11. Results

If the installation is successfully done, you should see the following information. Please note that the link to the installation place of GCC ARM A-Profile (AArch64 bare-metal) 10.3.2021.07 you selected as Additional Software is shown at "**Useful Links:**".



Renesas e² studio 2022-07 Setup

Installation of **e2 studio** is complete.

Please click **OK** to close.

- [x] Launch e2 studio?
- [ ] View Release Notes?
- [ ] View What's New?

Useful Links:

v202207041249

User: epsd1-style-ish-4-3

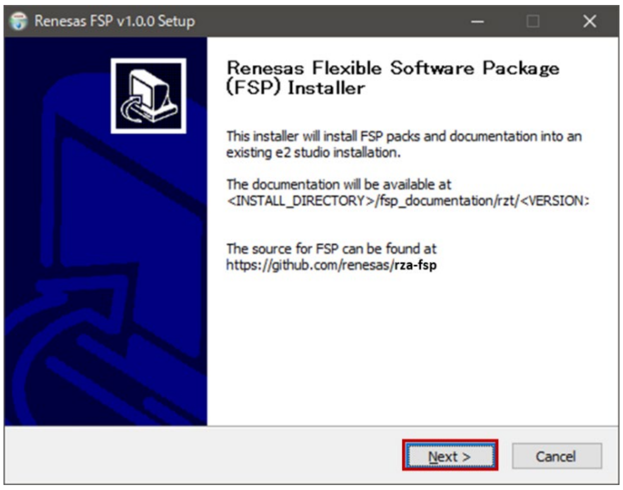
Figure 19: Summary Page

2.2 FSP setup

2.2.1 Installation of FSP Packs

FSP Packs are showcased at here. This section describes how to integrate FSP to e² studio using **RZA_FSP_Packs.exe**.

1. Exit e2 studio
2. Invoke **RZA_FSP_Packs.exe**
3. Click **Next >** to start the installation



Renesas FSP v1.0.0 Setup

Renesas Flexible Software Package (FSP) Installer

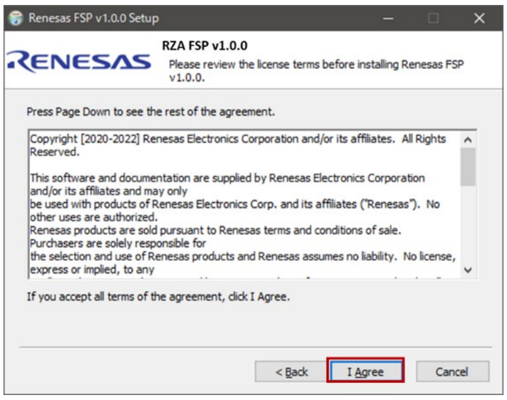
This installer will install FSP packs and documentation into an existing e2 studio installation.

The documentation will be available at <INSTALL_DIRECTORY>/fsp_documentation/rzt/<VERSION>

The source for FSP can be found at https://github.com/renesas/rza-fsp

Figure 20: FSP Package Installer

4. See the license term and click **I Agree** if it's acceptable



RZA FSP v1.0.0
Please review the license terms before installing Renesas FSP v1.0.0.

Press Page Down to see the rest of the agreement.

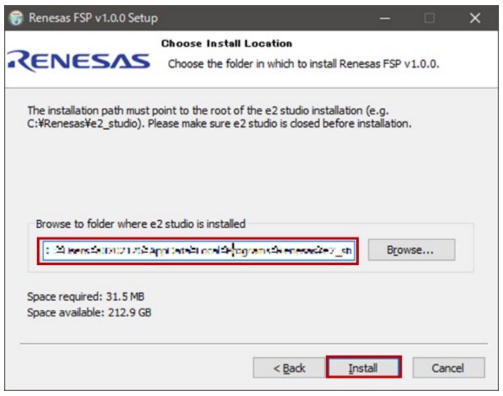
Copyright [2020-2022] Renesas Electronics Corporation and/or its affiliates. All Rights Reserved.

This software and documentation are supplied by Renesas Electronics Corporation and/or its affiliates and may only
be used with products of Renesas Electronics Corp. and its affiliates ("Renesas"). No other uses are authorized.
Renesas products are sold pursuant to Renesas terms and conditions of sale.
Purchasers are solely responsible for
the selection and use of Renesas products and Renesas assumes no liability. No license, express or implied, to any

If you accept all terms of the agreement, click I Agree.

Figure 21: FSP License Term

5. Specify e2 studio installation folder and click **Install**.



Choose Install Location
Choose the folder in which to install Renesas FSP v1.0.0.

The installation path must point to the root of the e2 studio installation (e.g. C:¥Renesas¥e2_studio). Please make sure e2 studio is closed before installation.

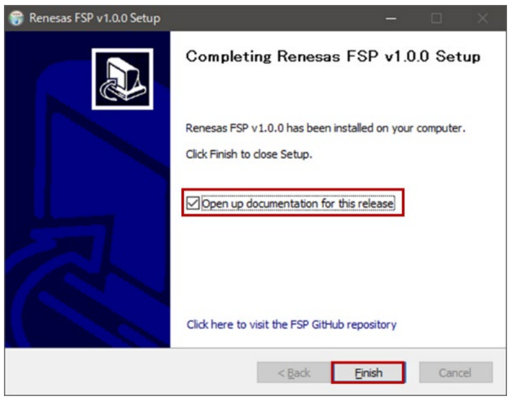
Browse to folder where e2 studio is installed

[illegible]

Space required: 31.5 MB
Space available: 212.9 GB

Figure 22: FSP Installation

6. Click **Finish** to complete the installation.



Completing Renesas FSP v1.0.0 Setup

Renesas FSP v1.0.0 has been installed on your computer.

Click Finish to close Setup.

☑ Open up documentation for this release

Figure 23: Completion of FSP Installation

If the box **Open up documentation for this release** is checked at that time, FSP documentation for the installed version of FSP should be opened.

3. Set up an SMARC EVK

Below is an example of a typical system configuration.

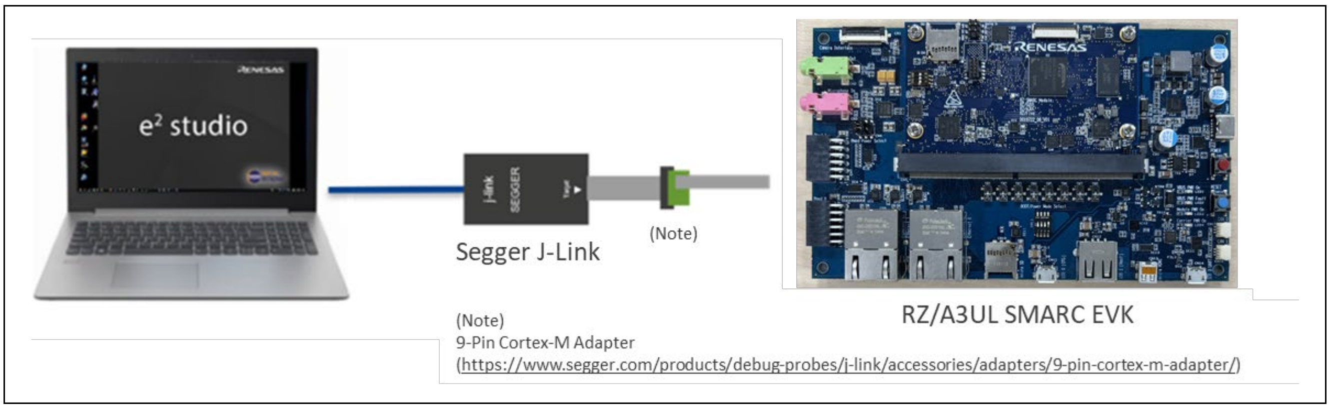

e² studio
Segger J-Link
(Note)
RZ/A3UL SMARC EVK
(Note)
9-Pin Cortex-M Adapter
(https://www.segger.com/products/debug-probes/j-link/accessories/adapters/9-pin-cortex-m-adapter/)

Figure 24: System Configuration Example – SMARC EVK

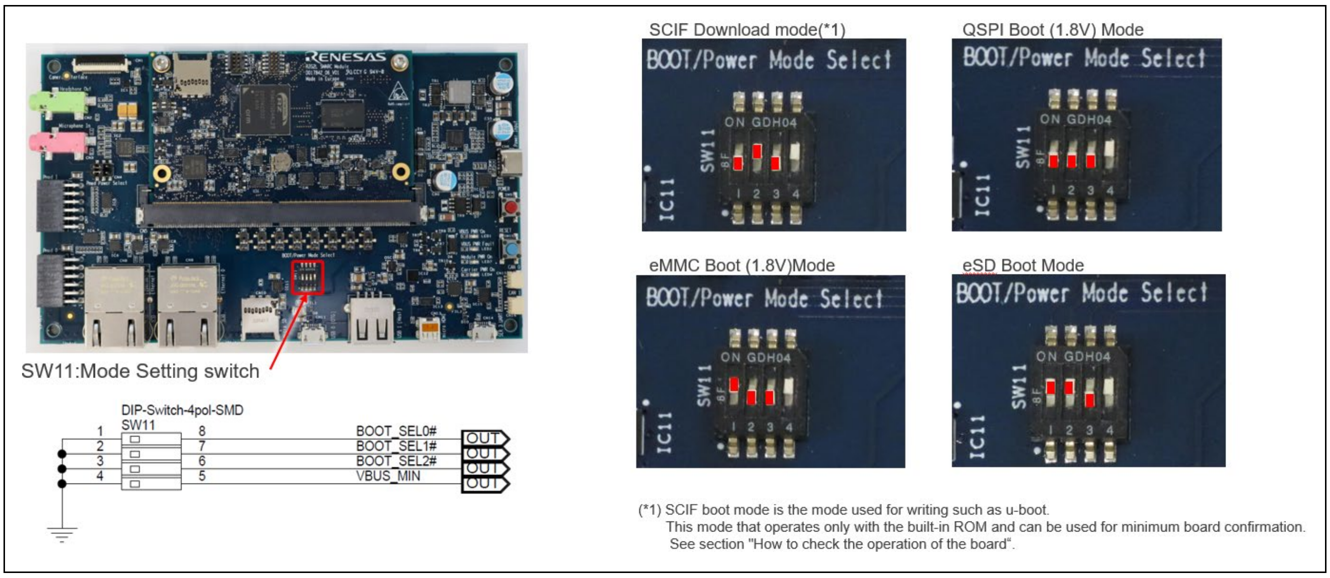
3.1 Supported Debugger

- SEGGER J-Link
  For details on SEGGER J-Link, please see J-Link Debug Probes by SEGGER – the Embedded Experts.

3.2 Board Setup

3.2.1 Boot MODE

To set the board to Boot mode 3(QSPI Boot (1.8V) Mode), set the SW11 as below.

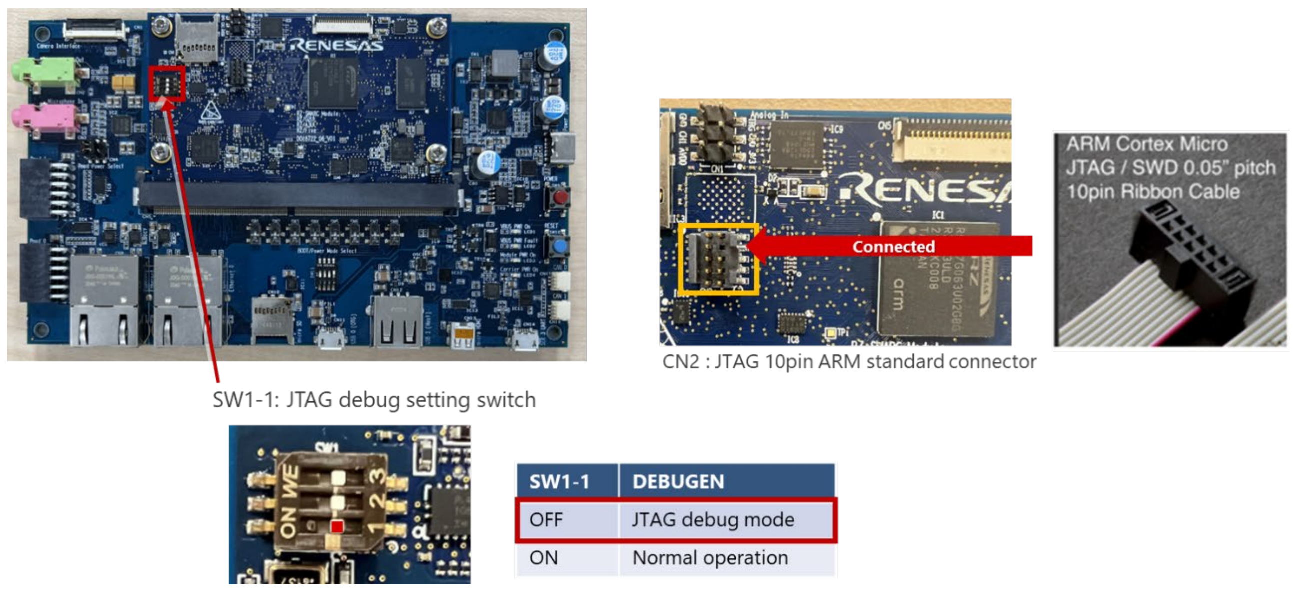


(*1) SCIF boot mode is the mode used for writing such as u-boot.
This mode that operates only with the built-in ROM and can be used for minimum board confirmation.
See section "How to check the operation of the board".

Figure 25: Boot MODE

3.2.2 JTAG connection

When connecting JTAG, you must set the DIP SW1 settings as follows:


SW1-1: JTAG debug setting switch
Connected
CN2 : JTAG 10pin ARM standard connector
ARM Cortex Micro JTAG / SWD 0.05" pitch 10pin Ribbon Cable

SW1-1	DEBUGEN
OFF	JTAG debug mode
ON	Normal operation

Figure 26: JTAG connection

Please note that RZ/A3UL SMARC EVK has CoreSight 10 connector and therefore, the following adapter must be needed to connect Segger J-Link.

https://www.segger.com/products/debug-probes/j-link/accessories/adapters/9-pin-cortex-m-adapter/

3.2.3 Debug Serial (console output)

Debug serial uses CN14. The baud rate is 115200bps.

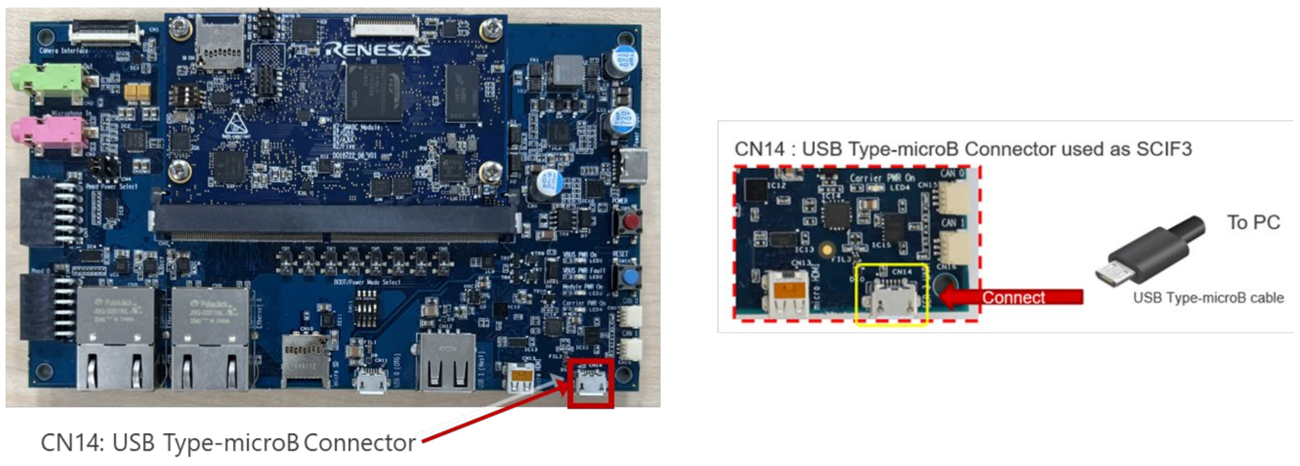

CN14 : USB Type-microB Connector used as SCIF3
To PC
Connect
USB Type-microB cable
CN14: USB Type-microB Connector

Figure 27: Debug Serial (console output)

3.2.4 Power Supply

Here are the proven power supply related goods to be used in Renesas' development. Please prepare for the equivalent ones for your development.

- USB Type-C cable CB-CD23BK (manufactured by Aukey)
- USB PD Charger Anker PowerPort III 65W Pod (manufactured by Anker)

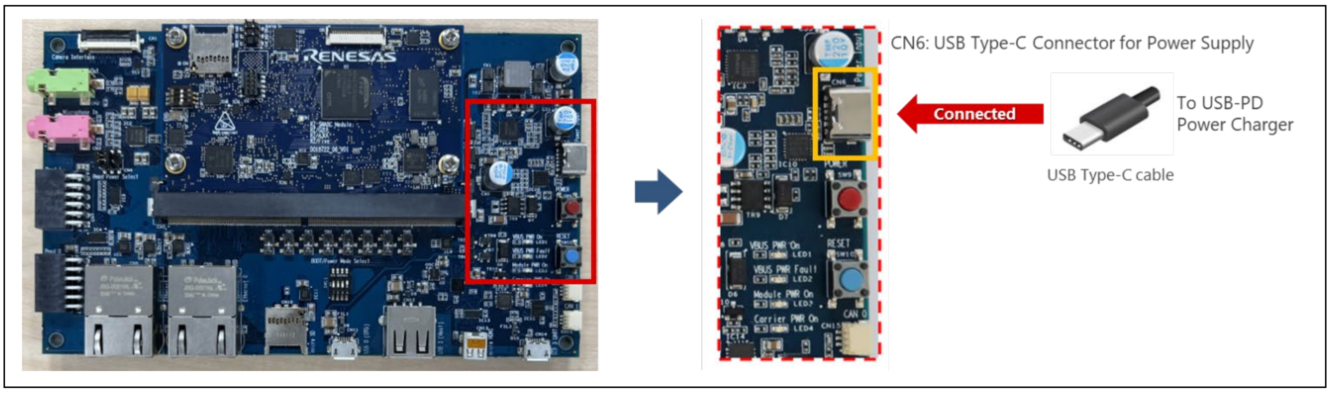


Figure 28: Power Supply

For power supply, please follow the following procedure:

1. Connect USB-PD Power Charger to USB Type-C Connector (CN6).
   Once USB-PD Power Charger is connected to the CN6, LED1 (VBUS PWR ON) and LED3 (Module PWR ON) should light up.
2. Press the power button (SW9) to turn on the power
   When turning on the power, you need to press and hold the power button for 1 second.
   When turning off the power, the power button should be pressed and held for 2 seconds.
3. If the power supply is successful, LED4 (Carrier PWR On) should light up.

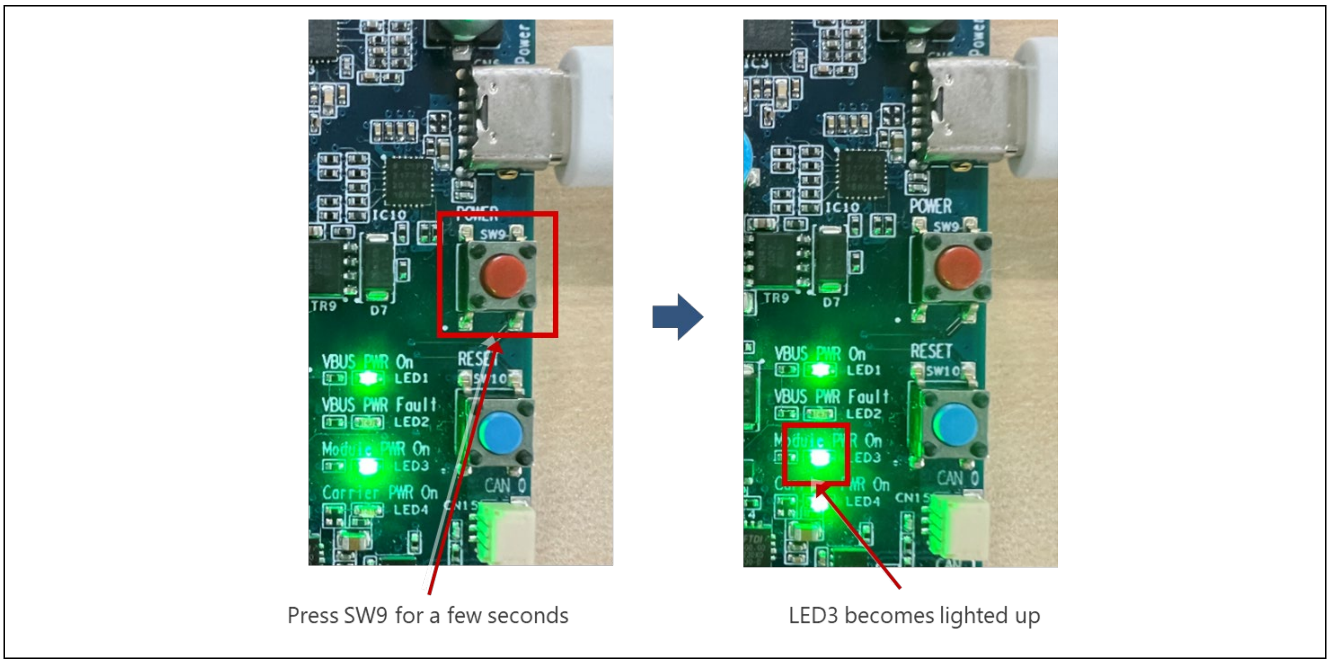
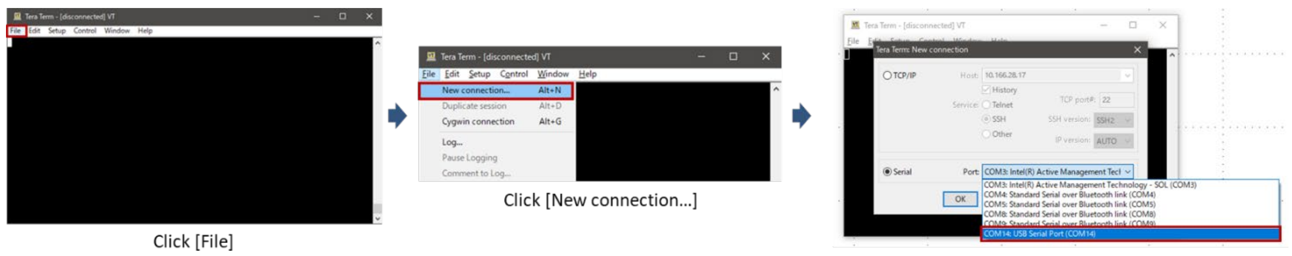


Figure 29: LED Status after Turning on EVK

3.2.5 How to check your board is operational

This section describes how to check if your board is operational.

1. Connect the board to your development PC as described in 3.2.3.
2. Turn on the board as described in 3.2.4.
3. Launch Terminal Software (e.g., Tera Term)
4. Establish the connection between the board and development PC as shown below:


Tera Term - [disconnected] VT
File Edit Setup Control Window Help
New connection... Alt+N
Duplicate session Alt+D
Cygwin connection Alt+G
Log...
Pause Logging
Comment to Log...
Tera Term: New connection
TCP/IP
Serial
OK
COM14: USB Serial Port (COM14)

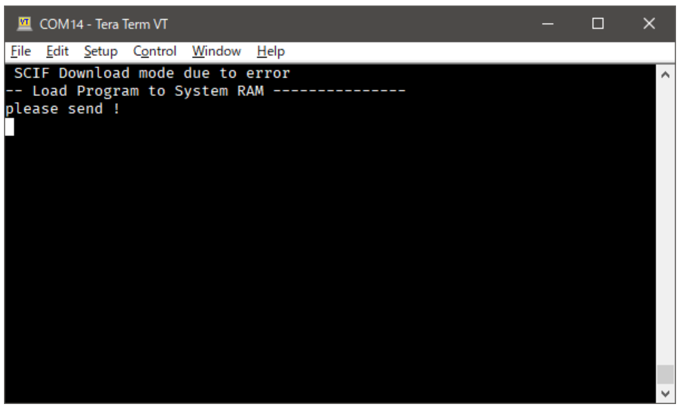
Click [File]

Click [New connection...]

Select "COM ##: USB Serial Port (COM ##) (note)

(note) The number of COM Port depends on your development PC.

5. You should see the following message on your Terminal Software. You can ignore the keyword "error" since the cause of error is that nothing is programmed to QSPI Flash or OctaFlash by default.



```
 SCIF Download mode due to error
-- Load Program to System RAM ---------------
please send !
```

4. Tutorial: Your First RZ MPU Project - Blinky

4.1 Tutorial Blinky

The goal of this tutorial is to quickly get acquainted with the Flexible Platform by moving through the steps of creating a simple application using e2 studio and running that application on an RZ MPU board.

4.2 What Does Blinky Do?

The application used in this tutorial is Blinky, traditionally the first program run in a new embedded development environment.

Blinky is the "Hello World" of microprocessors. If the LED blinks you know that:

- The toolchain is setup correctly and builds a working executable image for your chip.
- The debugger has installed with working drivers and is properly connected to the board.
- The board is powered up and its jumper and switch settings are probably correct.
- The microprocessor is alive, the clocks are running, and the memory is initialized.
- Timer (GTM) interrupt is intentionally fired and GPIO is properly controlled.

Note:

SRMAC EVK board does not have any LED. Thus, Blinky sample application used in this tutorial is designed to use the Pmod module described below alternatively:

- Pmod LED (Four High-brightness LEDs): https://reference.digilentinc.com/pmod/pmodled/start

This module is not included on the SRMAC EVK board and so, please prepare it beforehand.

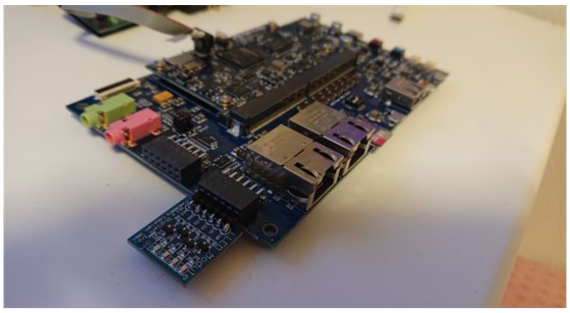
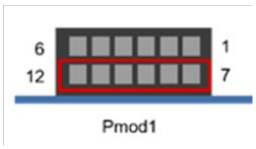



Figure 30: Connection Pmod LED module (410-076)

4.3 Create a New Project for Blinky

The creation and configuration of an RZ/A C/C++ FSP Project is the first step in the creation of an application. The base RZ/A pack includes a pre-written Blinky example application.

Follow these steps to create an RZ MPU project:

1. In e2 studio, click **File** > **New** > **C/C++ Project**.

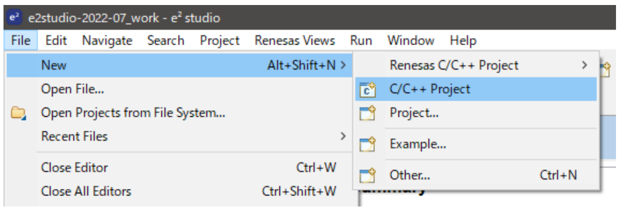

e2studio-2022-07_work - e² studio
File Edit Navigate Search Project Renesas Views Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Renesas C/C++ Project
C/C++ Project
Project...
Example...
Other... Ctrl+N

Figure 31: New C/C++ Project

2. Select **[Renesas RZ]** > **[Renesas RZ/A C/C++ FSP Project]** and Click Next.

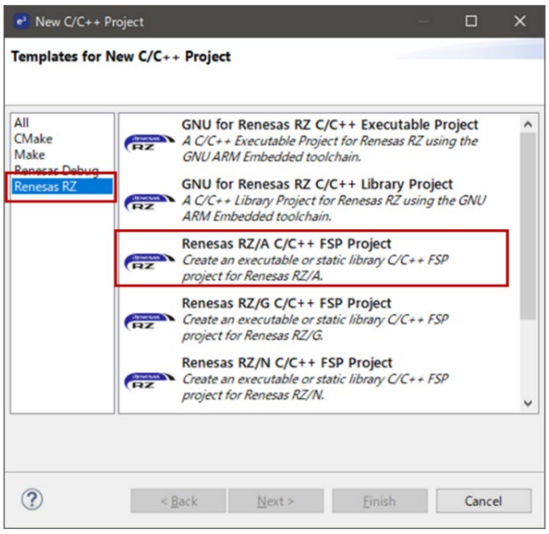
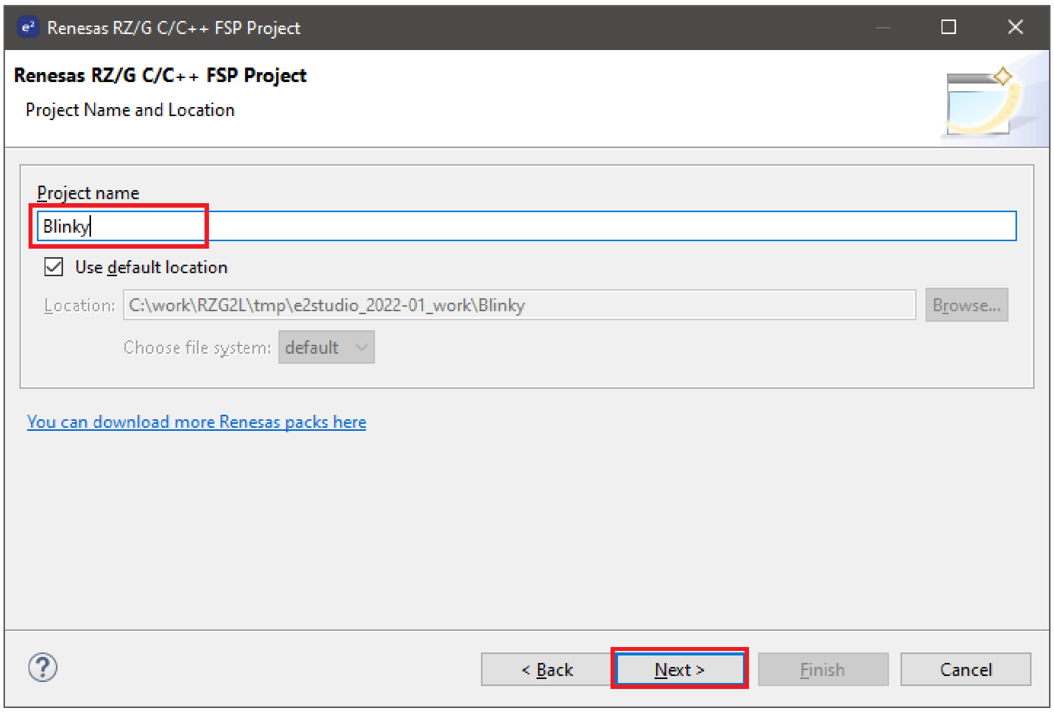

New C/C++ Project
Templates for New C/C++ Project
All
CMake
Make
Renesas Debug
Renesas RZ
GNU for Renesas RZ C/C++ Executable Project
A C/C++ Executable Project for Renesas RZ using the GNU ARM Embedded toolchain.
GNU for Renesas RZ C/C++ Library Project
A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain.
Renesas RZ/A C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/A.
Renesas RZ/G C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/G.
Renesas RZ/N C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/N.
< Back
Next >
Finish
Cancel

Figure 32: Renesas RZ/A C/C++ FSP Project

3. Assign a name to this new project. Blinky is a good name to use for this tutorial.
4. Click **Next**. The **Project Configuration** window shows your selection.


Renesas RZ/G C/C++ FSP Project

Renesas RZ/G C/C++ FSP Project

Project Name and Location

Project name: Blinky

[✓] Use default location

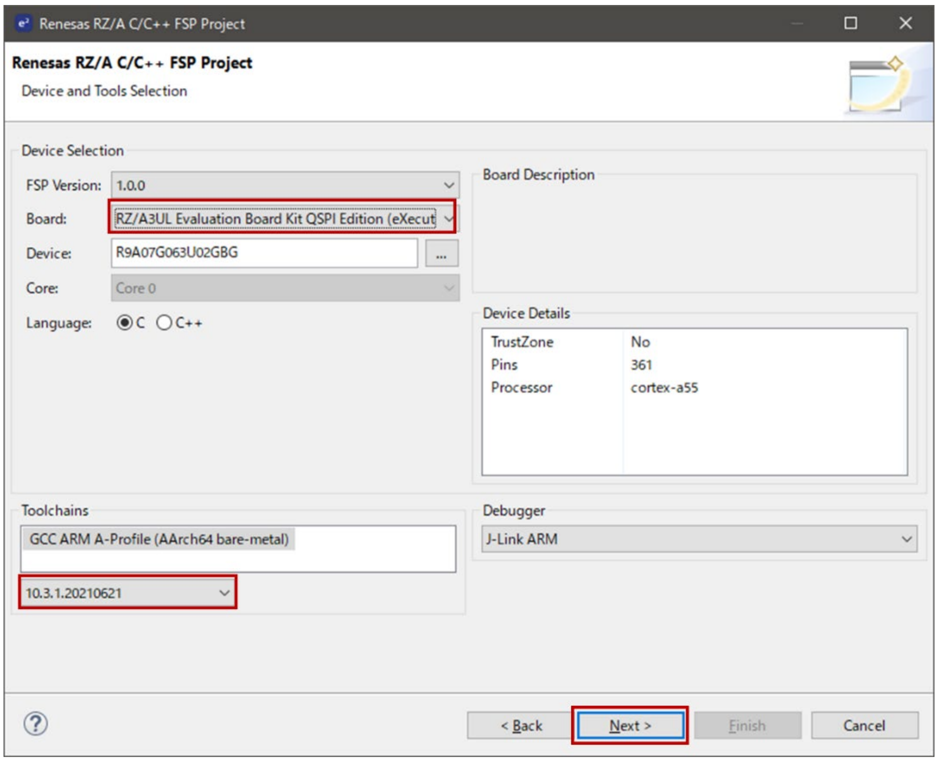
Location: C:\work\RZG2L\tmp\e2studio_2022-01_work\Blinky

Choose file system: default

You can download more Renesas packs here

Figure 33 : e2 studio Project Configuration window (part 1)

5. Select the board support package by selecting the package you would like to use from the Device Selection drop-down list. Select **GNU ARM A-Profile (AArch64 bare-metal)** in Toolchains and version is **10.3.1.20210621** and Click **Next**.


Renesas RZ/A C/C++ FSP Project

Renesas RZ/A C/C++ FSP Project

Device and Tools Selection

Device Selection

FSP Version: 1.0.0

Board: RZ/A3UL Evaluation Board Kit QSPI Edition (eXecut

Device: R9A07G063U02GBG

Core: Core 0

Language: [✓ C] C++

Board Description

Device Details

TrustZone	No
Pins	361
Processor	cortex-a55

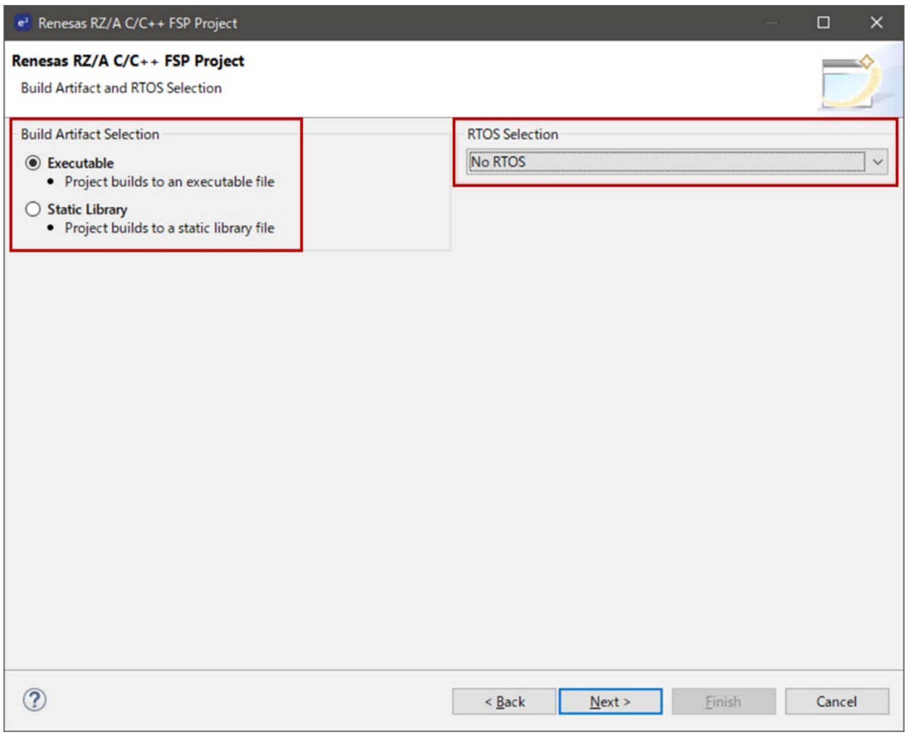
Toolchains: GCC ARM A-Profile (AArch64 bare-metal)

10.3.1.20210621

Debugger: J-Link ARM

Figure 34 : e2 studio Project Configuration window (part 2)

6. Select the **Build Artifact** and **RTOS**.



Renesas RZ/A C/C++ FSP Project

Renesas RZ/A C/C++ FSP Project

Build Artifact and RTOS Selection

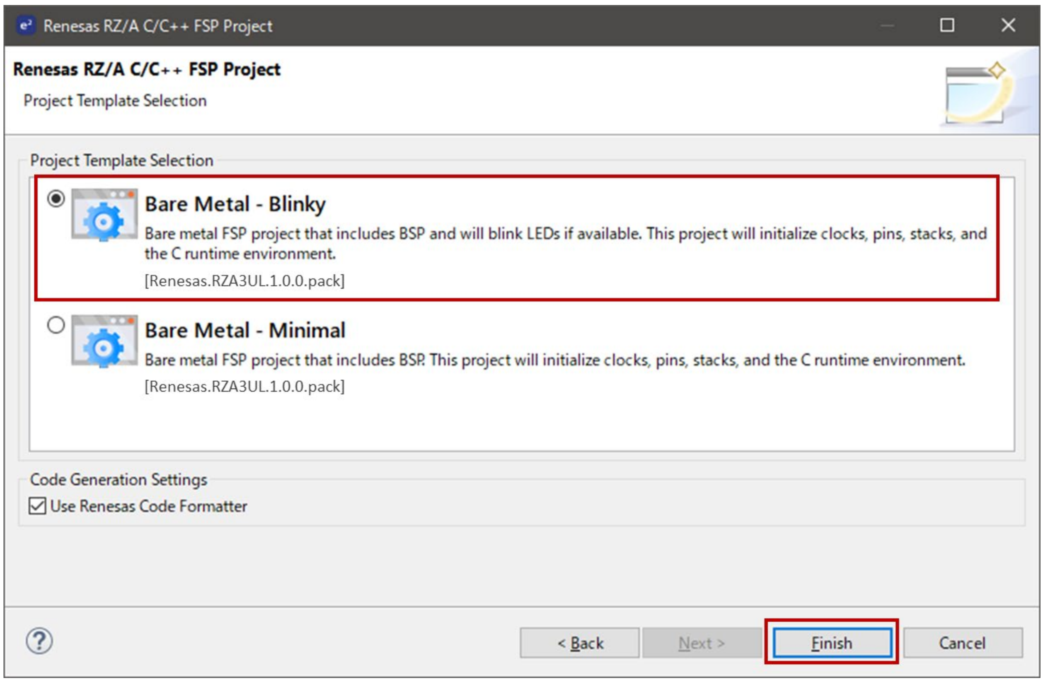
Build Artifact Selection

- [✓ Executable]
  - Project builds to an executable file
- Static Library
  - Project builds to a static library file

RTOS Selection: No RTOS

Figure 35 : e2 studio Project Configuration window (part 3)

7. Select the **Blinky** template for your board and click **Finish**.



Renesas RZ/A C/C++ FSP Project

Renesas RZ/A C/C++ FSP Project

Project Template Selection

Project Template Selection

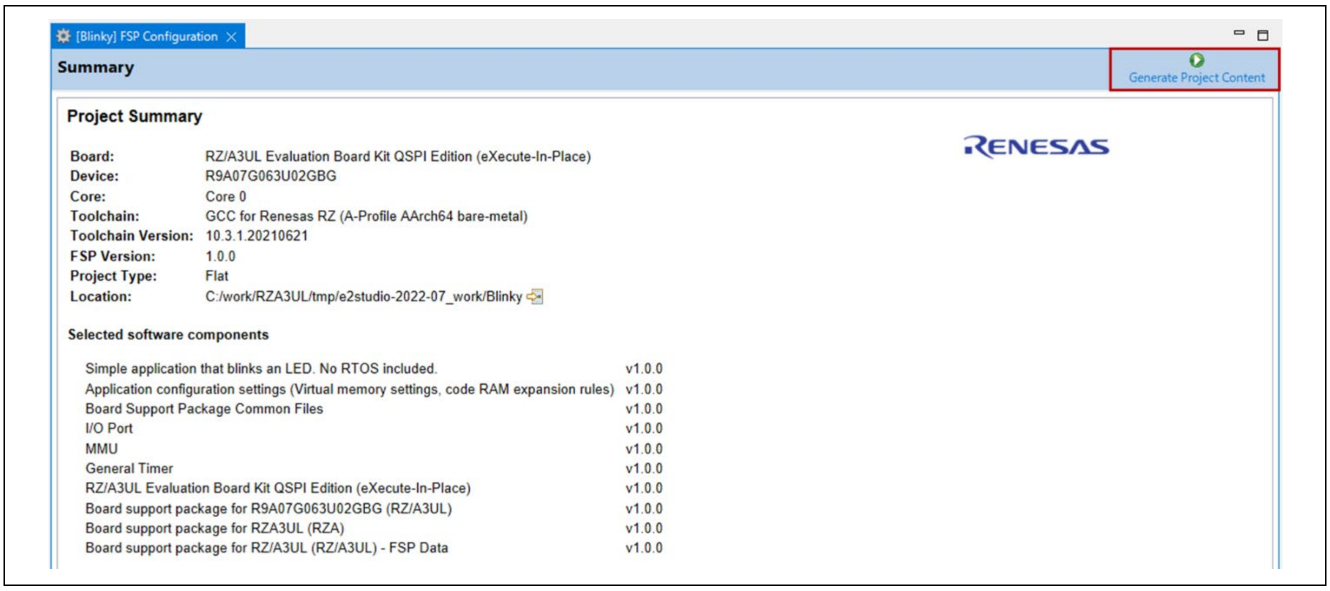
- [✓ Bare Metal - Blinky]
  Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment.
  [Renesas.RZA3UL.1.0.0.pack]
- Bare Metal - Minimal
  Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment.
  [Renesas.RZA3UL.1.0.0.pack]

Code Generation Settings

[✓] Use Renesas Code Formatter

Figure 36 : e2 studio Project Configuration window (part 4)

Once the project has been created, the name of the project will show up in the **Project Explorer** window of e2 studio. Now click the **Generate Project Content** button in the top right corner of the **Project Configuration** window to generate your board specific files.



Project Summary

Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)
Device:	R9A07G063U02GBG
Core:	Core 0
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)
Toolchain Version:	10.3.1.20210621
FSP Version:	1.0.0
Project Type:	Flat
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky

Selected software components

Simple application that blinks an LED. No RTOS included.	v1.0.0
Application configuration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0
Board Support Package Common Files	v1.0.0
I/O Port	v1.0.0
MMU	v1.0.0
General Timer	v1.0.0
RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0
Board support package for R9A07G063U02GBG (RZ/A3UL)	v1.0.0
Board support package for RZA3UL (RZA)	v1.0.0
Board support package for RZ/A3UL (RZ/A3UL) - FSP Data	v1.0.0

Figure 37 : e2 studio Project Configuration tab

Your new project is now created, configured, and ready to build.

4.3.1 Details about the Blinky Configuration

The Generate Project Content button creates configuration header files, copies source files from templates, and generally configures the project based on the state of the Project Configuration screen.

For example, if you check a box next to a module in the Components tab and click the Generate Project Content button, all the files necessary for the inclusion of that module into the project will be copied or created. If that same check box is then unchecked those files will be deleted.

4.3.2 Configuring the Blinky Clocks

By selecting the Blinky template, the clocks are configured by e2 studio for the Blinky application. The clock configuration tab (see 5.2.3. Configuring Clocks) shows the Blinky clock configuration. The Blinky clock configuration is stored in the BSP clock configuration file.

4.3.3 Configuring the Blinky Pins

By selecting the Blinky template, the GPIO pins used to toggle the LED1 are configured by e2 studio for the Blinky application. The pin configuration tab shows the pin configuration for the Blinky application (see 5.2.4.Configuring Pins). The Blinky pin configuration is stored in the BSP configuration file.

4.3.4 Configuring the Parameters for Blinky Components

The Blinky project automatically selects the following HAL components in the Components tab:

- r_gtm
- r_ioport
- r_mmu

To see the configuration parameters for any of the components, check the Properties tab in the HAL window for the respective driver (see 5.2.8.Adding and Configuring HAL Drivers).

4.3.5 Where is main()?

The main function is located in <project>/rza_gen/main.c. It is one of the files that are generated during the project creation stage and only contains a call to hal_entry(). For more information on generated files, see Adding and Configuring HAL Drivers.

4.3.6 Blinky Example Code

The blinky application is stored in the hal_entry.c file. This file is generated by e2 studio when you select the Blinky Project template and is located in the project's src/ folder.

The application performs the following steps:

1. Get the LED information for the selected board by bsp_leds_t structure.
2. Set the configuration of Timer (GTM) and the callback function that is called when interrupt is fired.
3. Define the output level HIGH for the GPIO pins controlling the LEDs for the selected board.
4. Toggle the LEDs by writing to the GPIO pin with "R_BSP_PinWrite((bsp_io_port_pin_t) pin, pin_level)" in callback function of GTM that is called with the specified interval.

4.4 Build the Blinky Project

Highlight the new project in the Project Explorer window by clicking on it and build it.

There are three ways to build a project:

1. Click on Project in the menu bar and select Build Project.
2. Click on the hammer icon.
3. Right-click on the project and select Build Project.

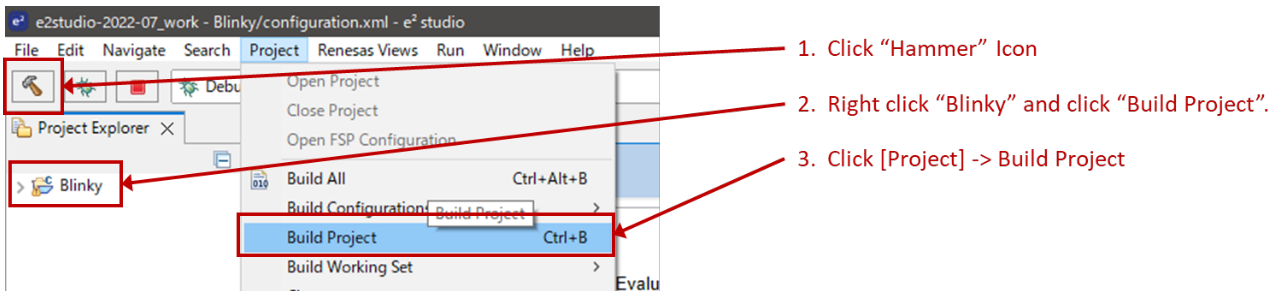
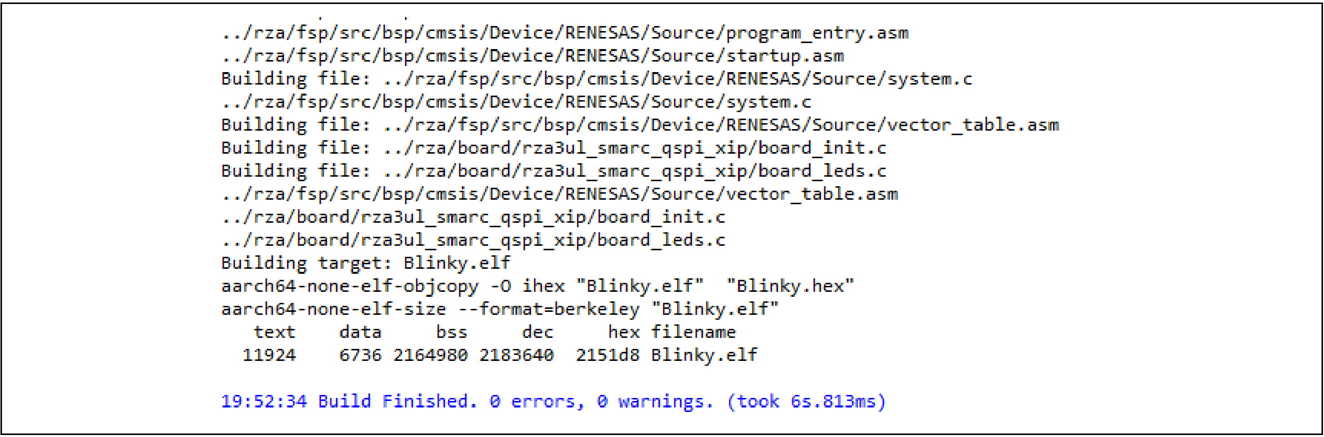

e2studio-2022-07_work - Blinky/configuration.xml - e² studio
File Edit Navigate Search Project Renesas Views Run Window Help
Debu
Project Explorer
Blinky
Open Project
Close Project
Open FSP Configuration
Build All Ctrl+Alt+B
Build Configurations
Build Project
Build Project Ctrl+B
Build Working Set
Evalu
1. Click "Hammer" Icon
2. Right click "Blinky" and click "Build Project".
3. Click [Project] -> Build Project

Figure 38 : e2 studio Project Explorer window

Once the build is complete, a message is displayed in the build Console window that displays the final image file name and section sizes in that image.



```
../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/program_entry.asm
../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/startup.asm
Building file: ../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/system.c
../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/system.c
Building file: ../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/vector_table.asm
Building file: ../rza/board/rza3ul_smarc_qspi_xip/board_init.c
Building file: ../rza/board/rza3ul_smarc_qspi_xip/board_leds.c
../rza/fsp/src/bsp/cmsis/Device/RENESAS/Source/vector_table.asm
../rza/board/rza3ul_smarc_qspi_xip/board_init.c
../rza/board/rza3ul_smarc_qspi_xip/board_leds.c
Building target: Blinky.elf
aarch64-none-elf-objcopy -O ihex "Blinky.elf"  "Blinky.hex"
aarch64-none-elf-size --format=berkeley "Blinky.elf"
   text    data     bss     dec     hex filename
  11924    6736 2164980 2183640  2151d8 Blinky.elf

19:52:34 Build Finished. 0 errors, 0 warnings. (took 6s.813ms)
```

Figure 39 : e2 studio Project Build console

4.5 Debug the Blinky Project

4.6 Debug prerequisites

To debug the project on a board, you need

- The board to be connected to e2 studio
- The debugger to be configured to talk to the board
- The application to be programmed to the microprocessor

Applications run from the internal ram or external ram of your microprocessor. To run or debug the application, the application must first be programmed to ram by JTAG debugger. SMARC EVK board has an JTAG header and requires an external JTAG debugger to the header.

4.7 Debug steps

To debug the Blinky application, follow these steps:

1. Configure the debugger for your project by clicking **Run** > **Debugger Configurations** ...

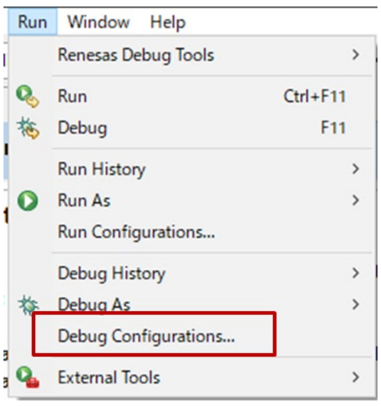


Figure 40 : e2 studio Debug icon

or by selecting the drop-down menu next to the bug icon and selecting **Debugger Configuration**s ...

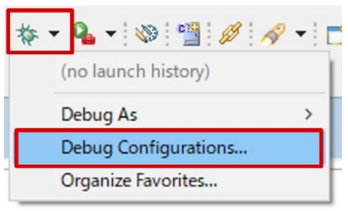
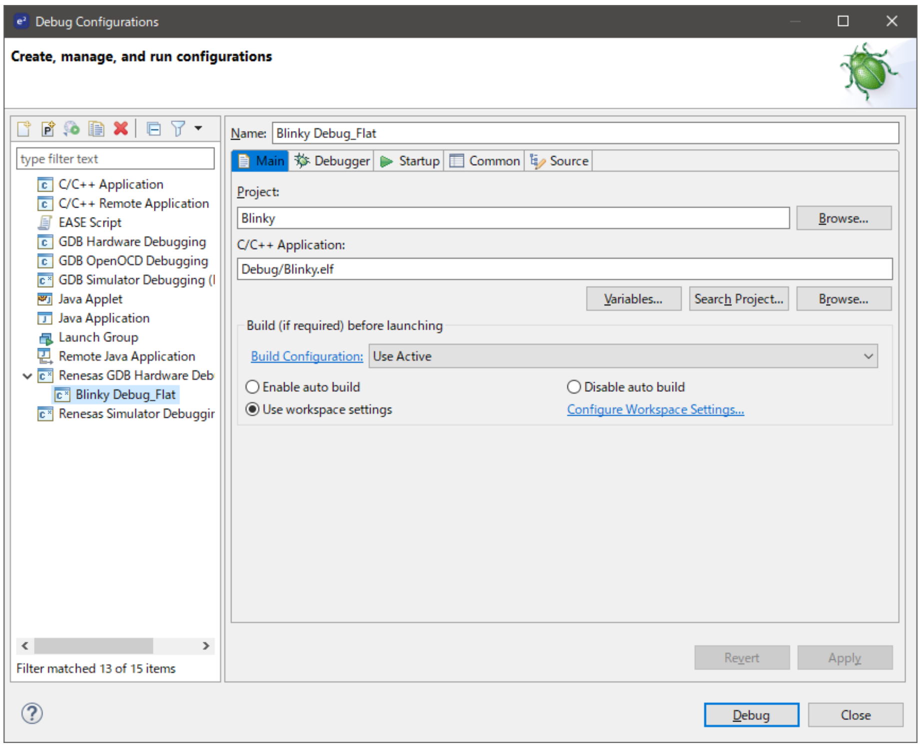


Figure 41 : e2 studio Debugger Configurations selection option

2. 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 **Blinky** project.


Debug Configurations

Create, manage, and run configurations

type filter text

C/C++ Application
C/C++ Remote Application
EASE Script
GDB Hardware Debugging
GDB OpenOCD Debugging
GDB Simulator Debugging (I
Java Applet
Java Application
Launch Group
Remote Java Application
Renesas GDB Hardware Deb
Blinky Debug_Flat
Renesas Simulator Debuggir

Filter matched 13 of 15 items

Name: Blinky Debug_Flat

Main | Debugger | Startup | Common | Source

Project: Blinky

C/C++ Application: Debug/Blinky.elf

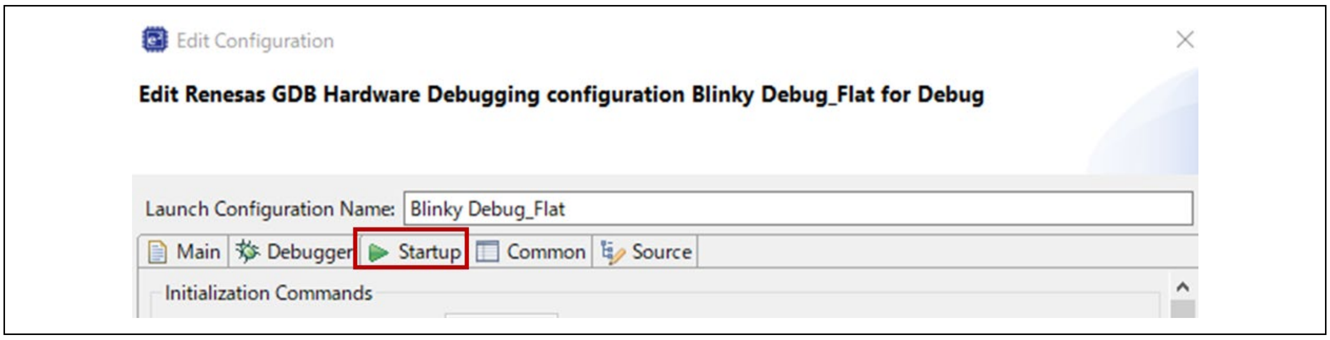
Build (if required) before launching

Build Configuration: Use Active

Enable auto build
Disable auto build
[✓ Use workspace settings]
Configure Workspace Settings...

Figure 42 : e2 studio Debugger Configurations window with Blinky project (1)

3. Select the debug configuration for the generated project and select the **Startup** tab.


Edit Configuration

Edit Renesas GDB Hardware Debugging configuration Blinky Debug_Flat for Debug

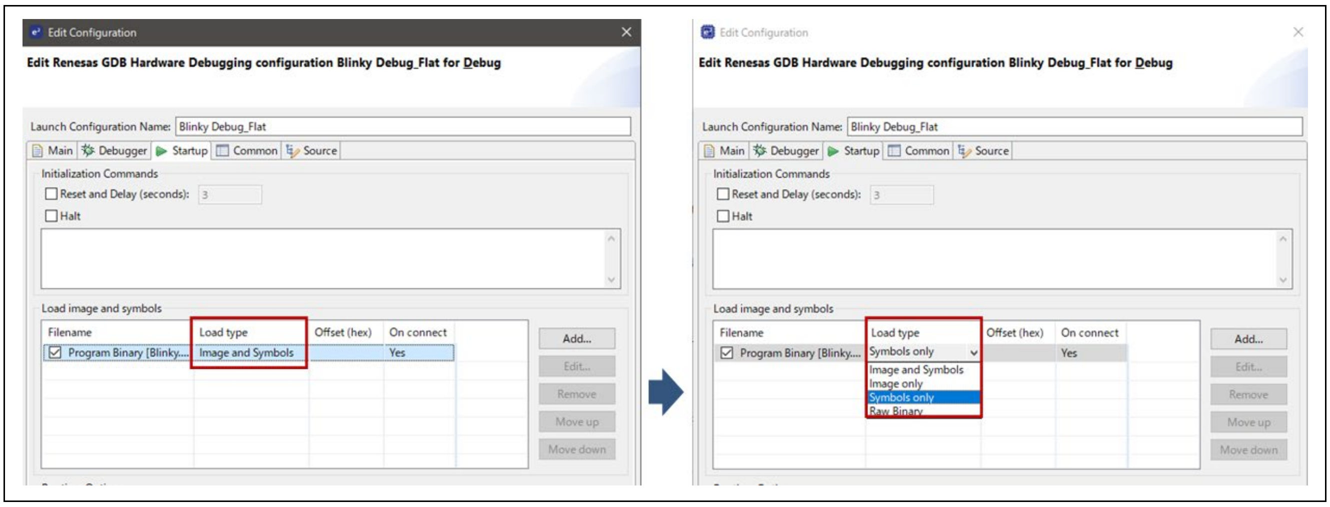
Launch Configuration Name: Blinky Debug_Flat

Main | Debugger | Startup | Common | Source

Initialization Commands

Figure 43 : e2 studio Debugger Configurations window with Blinky project (2)

4. Be sure to change the setting in **Load type** field of **Program Binary [Blinky...** raw from **Image and Symbols** to **Symbols only**.


Edit Configuration

Edit Renesas GDB Hardware Debugging configuration Blinky Debug_Flat for Debug

Launch Configuration Name: Blinky Debug_Flat

Initialization Commands
Reset and Delay (seconds): 3
Halt

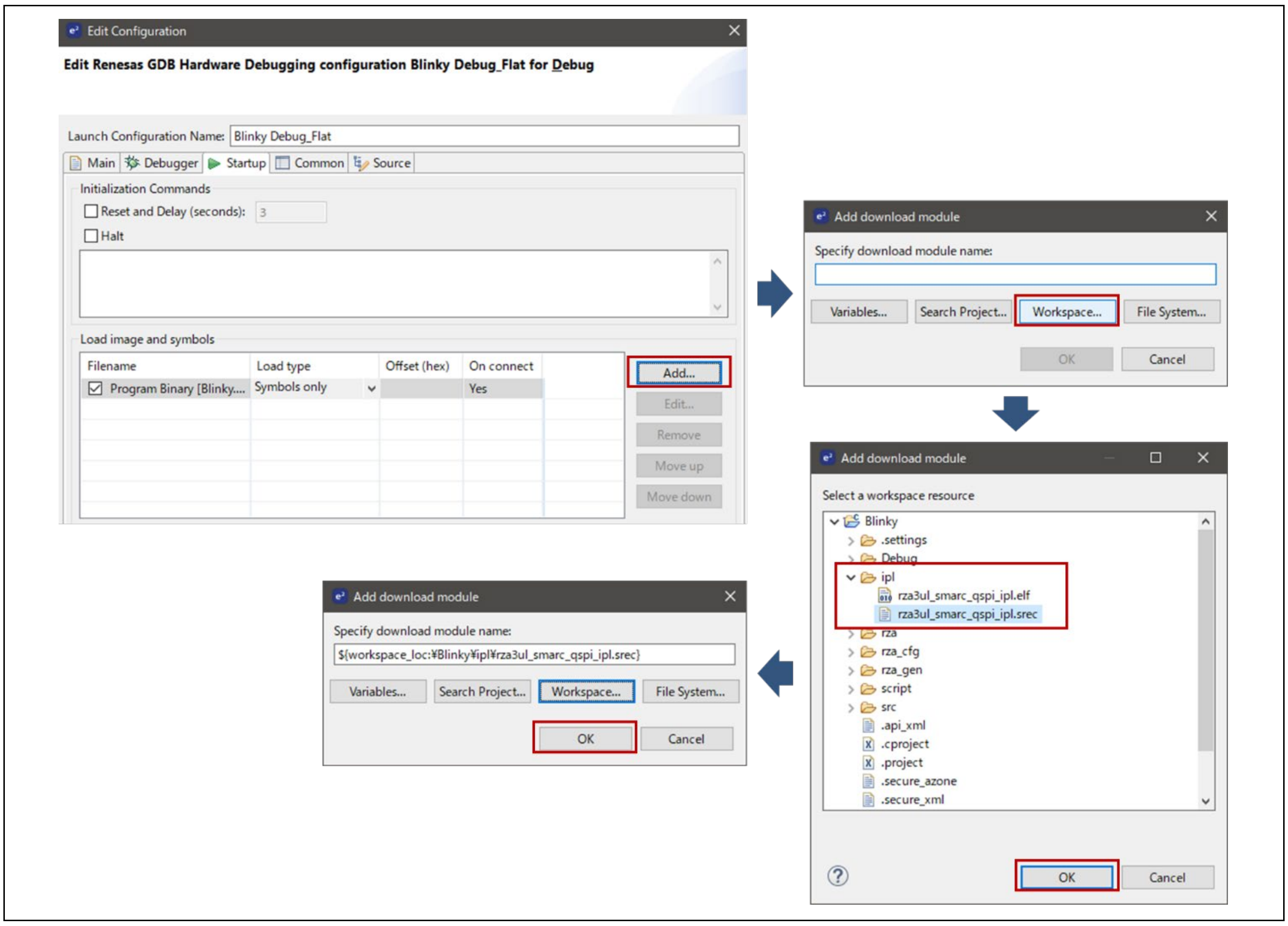
Load image and symbols

Filename	Load type	Offset (hex)	On connect
Program Binary [Blinky....	Image and Symbols		Yes

Filename	Load type	Offset (hex)	On connect
Program Binary [Blinky....	Symbols only		Yes

Figure 44 : e2 studio Debugger Configurations window with Blinky project (3)

5. Click on **Add...** to launch **Add download module** window. Then click **Workspace...**, choose **rza3ul_smarc_qspi_ipl.srec** as module to be downloaded and finally click on **OK**.


Edit Configuration

Edit Renesas GDB Hardware Debugging configuration Blinky Debug_Flat for Debug

Launch Configuration Name: Blinky Debug_Flat

Initialization Commands
Reset and Delay (seconds): 3
Halt

Load image and symbols

Filename	Load type	Offset (hex)	On connect
Program Binary [Blinky....	Symbols only		Yes

Add download module

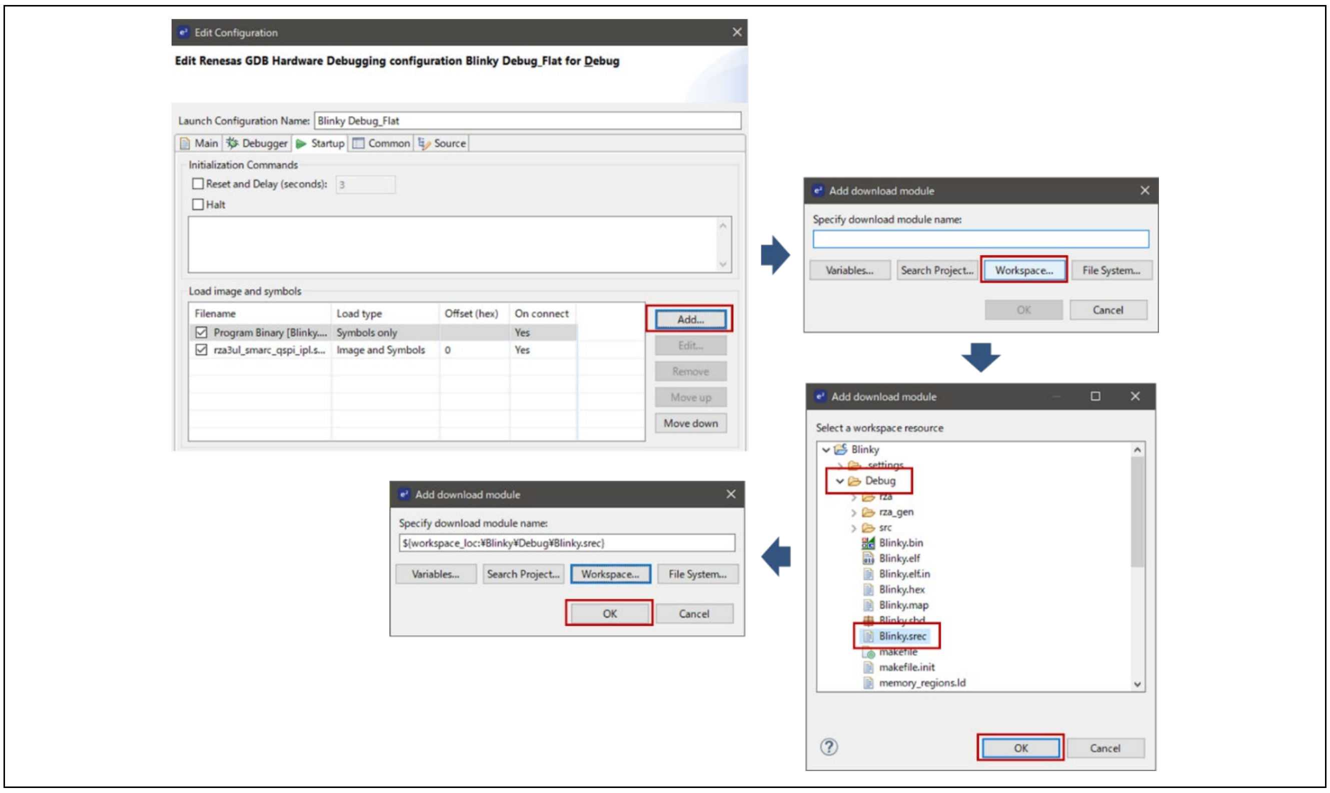
Specify download module name:

Select a workspace resource

Specify download module name: ${workspace_loc:¥Blinky¥ipl¥rza3ul_smarc_qspi_ipl.srec}

Figure 45 : e2 studio Debugger Configurations window with Blinky project (4)

6. Again, click on **Add...** to launch **Add download module** window. Then click **Workspace...**, choose **BLINKY.srec** as module to be downloaded and finally click on **OK**.



Launch Configuration Name: Blinky Debug_Flat

Initialization Commands
- [ ] Reset and Delay (seconds): 3
- [ ] Halt

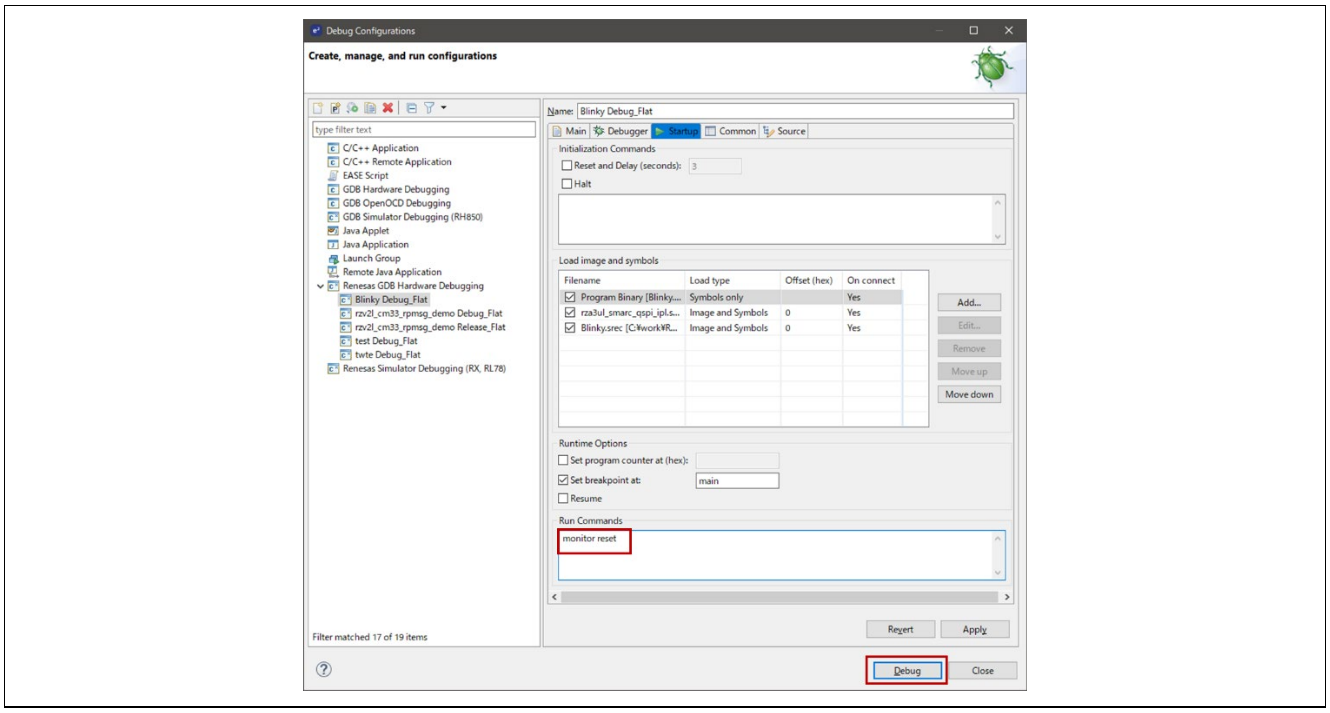
Load image and symbols

Filename	Load type	Offset (hex)	On connect
☑ Program Binary [Blinky....	Symbols only		Yes
☑ rza3ul_smarc_qspi_ipl.s...	Image and Symbols	0	Yes

Specify download module name: ${workspace_loc:¥Blinky¥Debug¥Blinky.srec}

Figure 46: e2 studio Debugger Configurations window with Blinky project (5)

7. Enter the command "**monitor reset**" to **Run Commands** at the bottom of **Startup** tab and then, click **Debug** button.



Name: Blinky Debug_Flat

Initialization Commands
- [ ] Reset and Delay (seconds): 3
- [ ] Halt

Load image and symbols

Filename	Load type	Offset (hex)	On connect
☑ Program Binary [Blinky....	Symbols only		Yes
☑ rza3ul_smarc_qspi_ipl.s...	Image and Symbols	0	Yes
☑ Blinky.srec [C:¥work¥R...	Image and Symbols	0	Yes

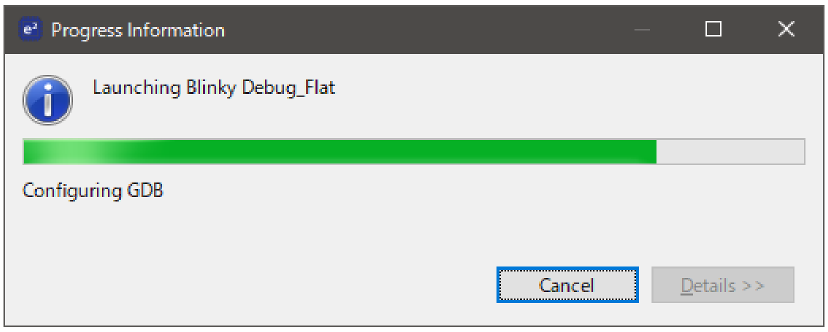
Runtime Options
- [ ] Set program counter at (hex):
- [x] Set breakpoint at: main
- [ ] Resume

Run Commands

```
monitor reset
```

Figure 47: e2 studio Debugger Configurations window with Blinky project (6)

8. Debug session is now started.



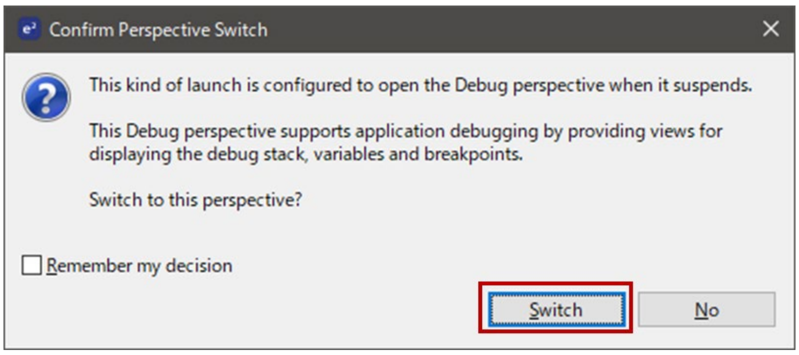
Progress Information

Launching Blinky Debug_Flat

Configuring GDB

Figure 48: e2 studio Debugger Configurations window with Blinky project (7)

9. If you see the following window, please click **Switch** to continue.



Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

☐ Remember my decision

Figure 49: e2 studio Debugger Configurations window with Blinky project (9)

4.8 Details about the Debug Process

In debug mode, e2 studio executes the following tasks:

1. Downloading the application image to QSPI/OctaFlash ROM or DDR SDRAM.
2. Setting a breakpoint at main().
3. Setting the stack pointer register to the stack.

4.9 Run the Blinky Project

Click **Run** > **Resume** or click on the **Play** icon.

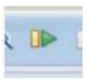
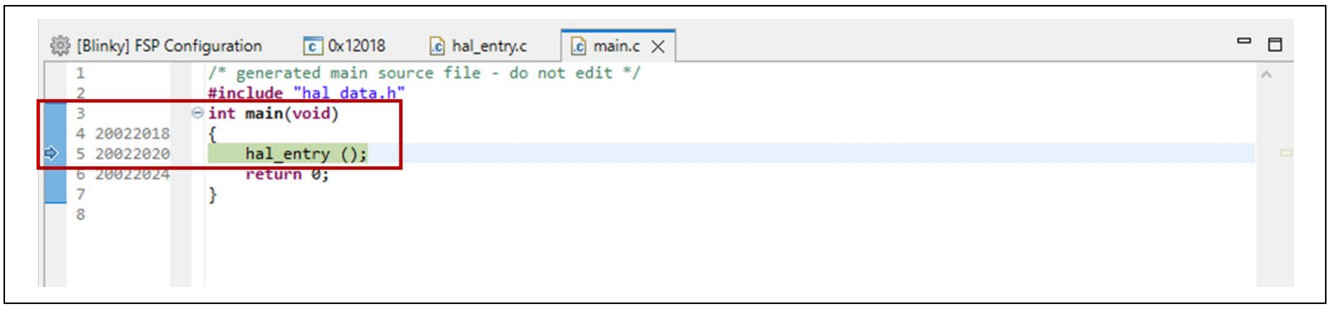


Figure 50 : e2 studio Debugger Play icon

Then, Blinky should stop at main().



```
1           /* generated main source file - do not edit */
2           #include "hal_data.h"
3           int main(void)
4 20022018  {
5 20022020      hal_entry ();
6 20022024      return 0;
7           }
8
```

Figure 51 : Blinky project in Debug Mode

Click Run > Resume or Play icon again. Then, the LEDs

After that, please click Run > Resume or Play icon again. You should see LEDs on the Pmod LED marked LD0, LD1, LD2 and LD3 are blinking.

5. FSP application launch with e2 studio

5.1 Create a Project

5.1.1 What is a Project?

In e2 studio, all FSP applications are organized in RZ MPU projects. Setting up an RZ MPU project involves:

1 **Create a Project**
2 **Configuring a Project**

These steps are described in detail in the next two sections. When you have existing projects already, after you launch e2 studio and select a workspace, all projects previously saved in the selected workspace are loaded and displayed in the **Project Explorer** window. Each project has an associated configuration file named configuration.xml, which is located in the project's root directory.

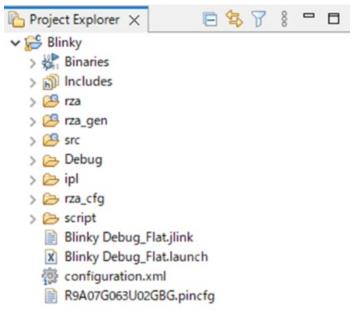


Figure 52 : e2 studio Project Configuration file

Double-click on the configuration.xml to open the RZ MPU Project Editor. To edit the project configuration, make sure that the **FSP Configuration** perspective shown below is selected in the upper right-hand corner of the e2 studio window. Once selected, you can use the editor to view or modify the configuration settings associated with this project.

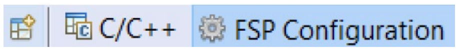
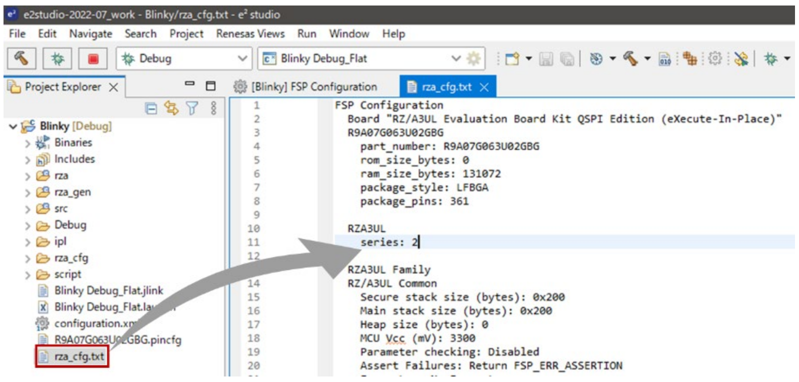


Figure 53 : e2 studio FSP Configuration Perspective

Note:

Whenever the RZ project configuration (that is, the configuration.xml file) is saved, a verbose RZ Project Report file (rza_cfg.txt) with all the project settings is generated. The format allows differences to be easily viewed using a text comparison tool. The generated file is located in the project root directory.



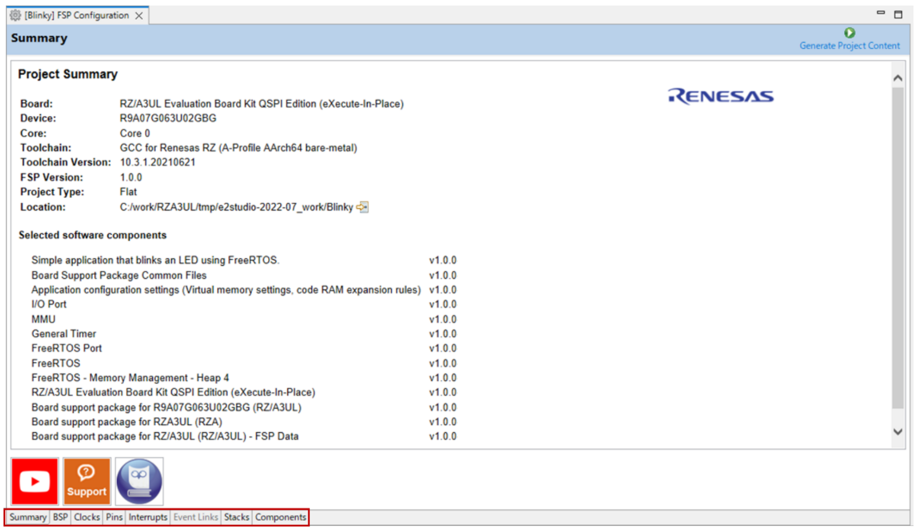
```
FSP Configuration
  Board "RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)"
  R9A07G063U02GBG
    part_number: R9A07G063U02GBG
    rom_size_bytes: 0
    ram_size_bytes: 131072
    package_style: LFBGA
    package_pins: 361

  RZA3UL
    series: 2

  RZA3UL Family
  RZ/A3UL Common
    Secure stack size (bytes): 0x200
    Main stack size (bytes): 0x200
    Heap size (bytes): 0
    MCU Vcc (mV): 3300
    Parameter checking: Disabled
    Assert Failures: Return FSP_ERR_ASSERTION
```

Figure 54 : RZ Project Report

The RZ Project Editor has several tabs. The configuration steps and options for individual tabs are discussed in the following sections.

Note:

The tabs available in the RZ Project Editor depend on the e2 studio version and the layout may vary slightly, however the functionality should be easy to follow.



Project Summary

Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)
Device:	R9A07G063U02GBG
Core:	Core 0
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)
Toolchain Version:	10.3.1.20210621
FSP Version:	1.0.0
Project Type:	Flat
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky

Selected software components

Simple application that blinks an LED using FreeRTOS.	v1.0.0
Board Support Package Common Files	v1.0.0
Application configuration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0
I/O Port	v1.0.0
MMU	v1.0.0
General Timer	v1.0.0
FreeRTOS Port	v1.0.0
FreeRTOS	v1.0.0
FreeRTOS - Memory Management - Heap 4	v1.0.0
RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0
Board support package for R9A07G063U02GBG (RZ/A3UL)	v1.0.0
Board support package for RZA3UL (RZA)	v1.0.0
Board support package for RZ/A3UL (RZ/A3UL) - FSP Data	v1.0.0

Summary | BSP | Clocks | Pins | Interrupts | Event Links | Stacks | Components

Figure 55 : RZ Project Editor tabs

5.1.2 Creating a New Project

For RZ MPU applications, generate a new project using the following steps:

1. Click on **File** > **New** > **C/C++ Project**.

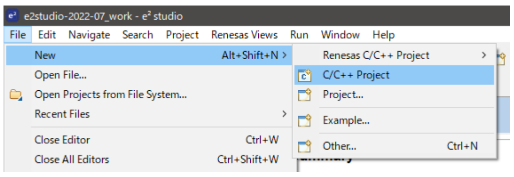
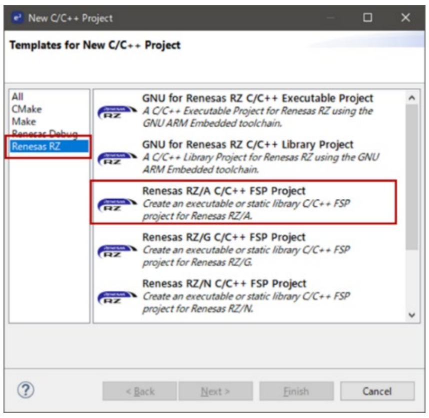

e2studio-2022-07_work - e² studio
File Edit Navigate Search Project Renesas Views Run Window Help
New Alt+Shift+N >
Open File...
Open Projects from File System...
Recent Files >
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Renesas C/C++ Project >
C/C++ Project
Project...
Example...
Other... Ctrl+N

Figure 56 : New RZ MPU Project

2. Click on the **Renesas RZ/A C/C++ FSP Project** template for the type of project you are creating.



GNU for Renesas RZ C/C++ Executable Project
A C/C++ Executable Project for Renesas RZ using the GNU ARM Embedded toolchain.

GNU for Renesas RZ C/C++ Library Project
A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain.

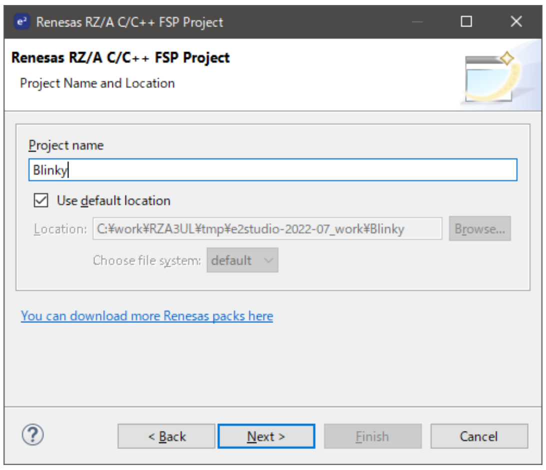
Renesas RZ/A C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/A.

Renesas RZ/G C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/G.

Renesas RZ/N C/C++ FSP Project
Create an executable or static library C/C++ FSP project for Renesas RZ/N.

Figure 57 : New Project Templates

3. Select a project name and location.



Renesas RZ/A C/C++ FSP Project

Renesas RZ/A C/C++ FSP Project

Project Name and Location

Project name: Blinky

[✓] Use default location

Location: C:¥work¥RZA3UL¥tmp¥e2studio-2022-07_work¥Blinky

Choose file system: default

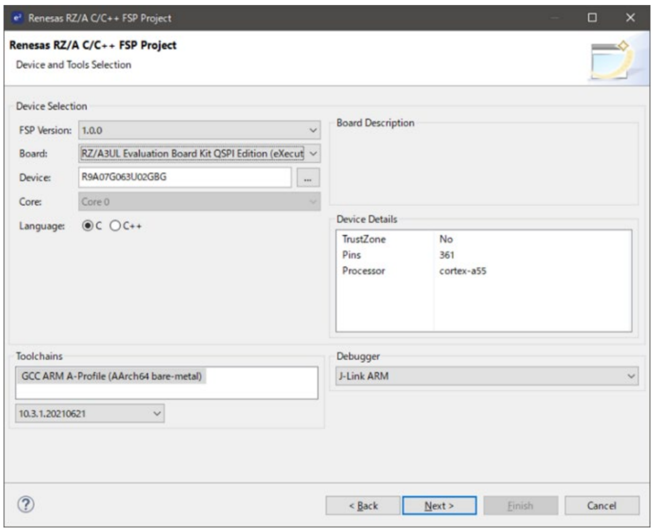
Figure 58 : RZ MPU Project Generator (Screen 1)

4. Click **Next**.

5.1.2.1 Selecting a Board and Toolchain

In the Project Configuration window select the hardware and software environment:

1. Select the **FSP version**.
2. Select the **Board** for your application. You can select an existing RZ MPU Evaluation Kit or **Custom User Board** for any of the RZ MPU devices with your own BSP definition.
3. Select the **Device**. The **Device** is automatically populated based on the **Board** selection. Only change the **Device** when using the **Custom User Board QSPI Boot (eXecute-In-Place)** board selection.
4. To add threads, select **RTOS**, or **No RTOS** if an RTOS is not being used.
5. The **Toolchain** selection defaults to **GNU ARM A-Profile (AArch64 bare-metal)**.
6. Select the **Toolchain version**. This should default to the installed toolchain version.
7. Select the **Debugger**. The J-Link Arm Debugger is preselected.
8. Click **Next**.



Renesas RZ/A C/C++ FSP Project

Renesas RZ/A C/C++ FSP Project

Device and Tools Selection

Device Selection

FSP Version: 1.0.0

Board: RZ/A3UL Evaluation Board Kit QSPI Edition (eXecut

Device: R9A07G063U02GBG

Core: Core 0

Language: [✓ C] C++

Board Description

Device Details

TrustZone	No
Pins	361
Processor	cortex-a55

Toolchains: GCC ARM A-Profile (AArch64 bare-metal)

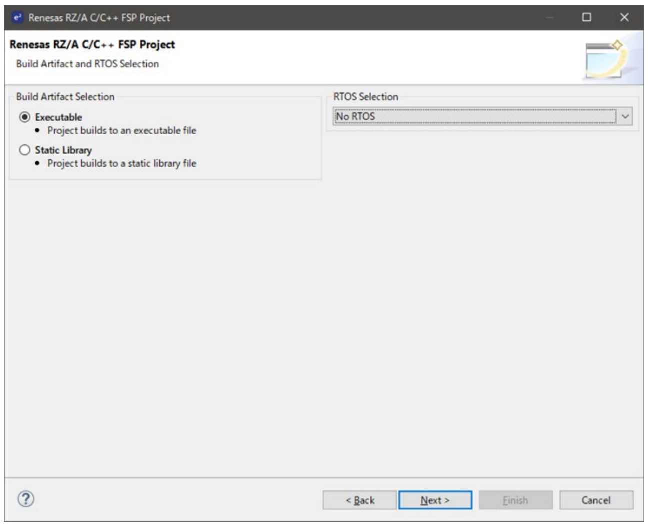
10.3.1.20210621

Debugger: J-Link ARM

Figure 59 : RZ MPU Project Generator (Screen 2)

5.1.2.2 Selecting a Project Template

In the next window, select the build artifact and **RTOS**.



Renesas RZ/A C/C++ FSP Project
Build Artifact and RTOS Selection

Build Artifact Selection
- (●) Executable
  - Project builds to an executable file
- ( ) Static Library
  - Project builds to a static library file

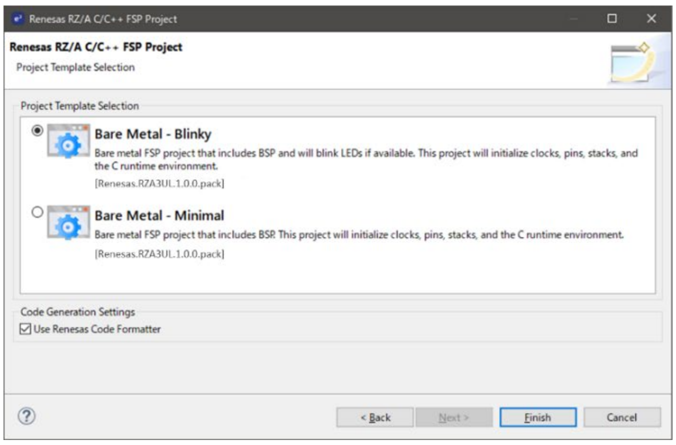
RTOS Selection: No RTOS

Figure 60 : RZ MPU Project Generator (Screen 3)

In the next window, select a project template from the list of available templates. By default, this screen shows the templates that are included in your current RZ/A MPU Pack. Once you have selected the appropriate template, click **Finish**.

Note:

If you want to develop your own application, select the basic template for your board, **Bare Metal - Minimal** or **FreeRTOS - Minimal**.

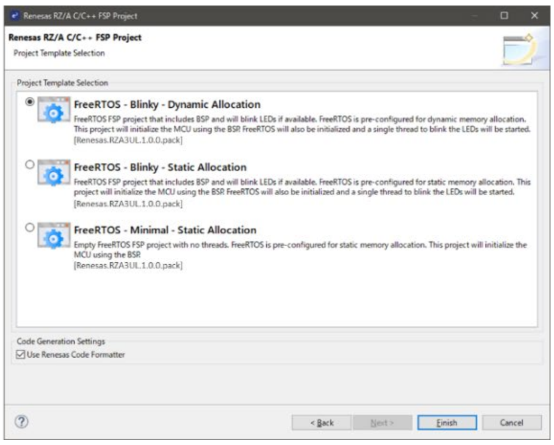


Renesas RZ/A C/C++ FSP Project
Project Template Selection

Project Template Selection
- (●) Bare Metal - Blinky
  Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment.
  [Renesas.RZA3UL.1.0.0.pack]
- ( ) Bare Metal - Minimal
  Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment.
  [Renesas.RZA3UL.1.0.0.pack]

Code Generation Settings
☑ Use Renesas Code Formatter

or



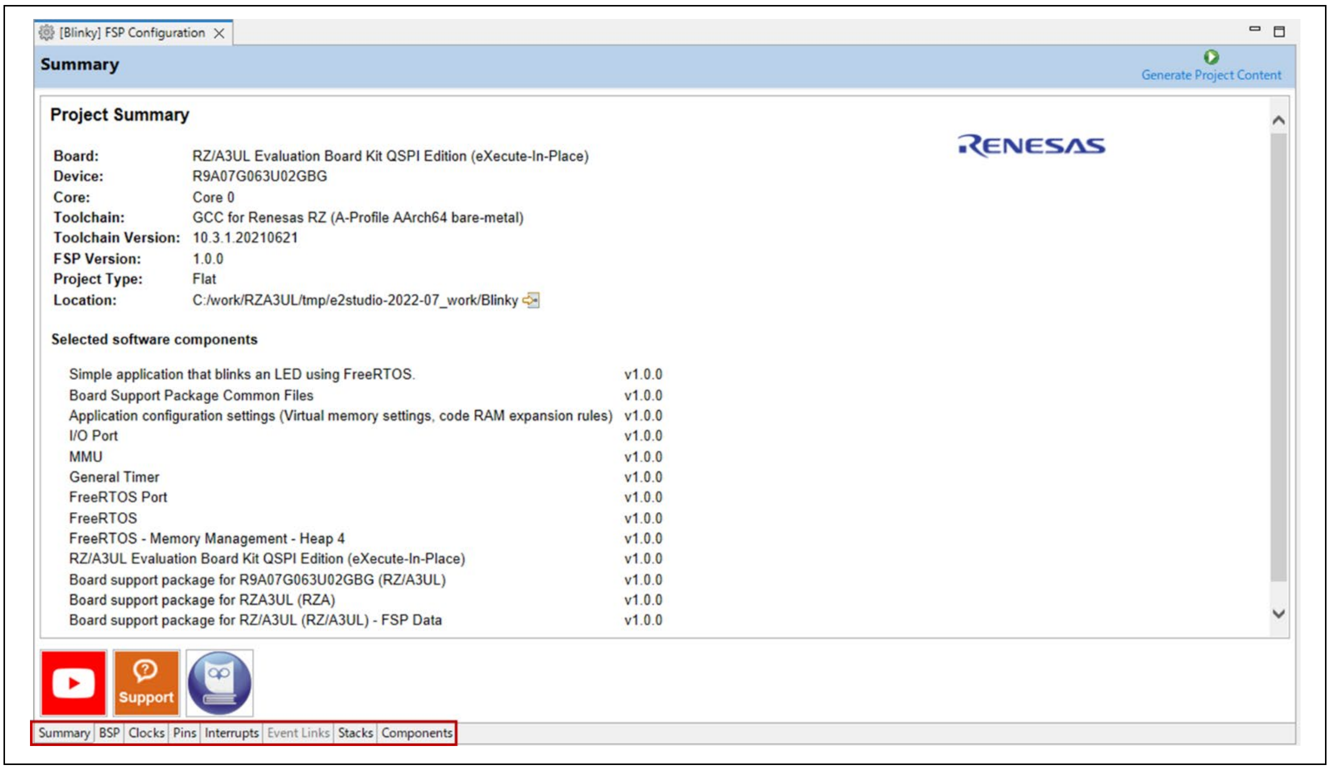
Renesas RZ/A C/C++ FSP Project
Project Template Selection

Project Template Selection
- (●) FreeRTOS - Blinky - Dynamic Allocation
  FreeRTOS FSP project that includes BSP and will blink LEDs if available. FreeRTOS is pre-configured for dynamic memory allocation. This project will initialize the MCU using the BSP. FreeRTOS will also be initialized and a single thread to blink the LEDs will be started.
  [Renesas.RZA3UL.1.0.0.pack]
- ( ) FreeRTOS - Blinky - Static Allocation
  FreeRTOS FSP project that includes BSP and will blink LEDs if available. FreeRTOS is pre-configured for static memory allocation. This project will initialize the MCU using the BSP. FreeRTOS will also be initialized and a single thread to blink the LEDs will be started.
  [Renesas.RZA3UL.1.0.0.pack]
- ( ) FreeRTOS - Minimal - Static Allocation
  Empty FreeRTOS FSP project with no threads. FreeRTOS is pre-configured for static memory allocation. This project will initialize the MCU using the BSP.
  [Renesas.RZA3UL.1.0.0.pack]

Code Generation Settings
☑ Use Renesas Code Formatter

Figure 61 : RZ MPU Project Generator (Screen 4)

When the project is created, e2 studio displays a summary of the current project configuration in the RZ MPU Project Editor.



Project Summary

Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)
Device:	R9A07G063U02GBG
Core:	Core 0
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)
Toolchain Version:	10.3.1.20210621
FSP Version:	1.0.0
Project Type:	Flat
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky

Selected software components

Simple application that blinks an LED using FreeRTOS.	v1.0.0
Board Support Package Common Files	v1.0.0
Application configuration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0
I/O Port	v1.0.0
MMU	v1.0.0
General Timer	v1.0.0
FreeRTOS Port	v1.0.0
FreeRTOS	v1.0.0
FreeRTOS - Memory Management - Heap 4	v1.0.0
RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0
Board support package for R9A07G063U02GBG (RZ/A3UL)	v1.0.0
Board support package for RZA3UL (RZA)	v1.0.0
Board support package for RZ/A3UL (RZ/A3UL) - FSP Data	v1.0.0

Figure 62 : RZ MPU Project Editor and available editor tabs

On the bottom of the RZ MPU Project Editor view, you can find the tabs for configuring multiple aspects of your project:

- With the **Summary** tab, you can see all they key characteristics of the project: board, device, toolchain, and more.
- With the **BSP** tab, you can change board specific parameters from the initial project selection.
- With the **Clocks** tab, you can configure the MCU clock settings for your project.
- With the **Interrupts** tab, you can add new user events/interrupts.
- With the **Stacks** tab, you can add and configure FSP modules. For each module selected in this tab, the **Properties** window provides access to the configuration parameters, interrupt selections.
- The **Components** tab provides an overview of the selected modules. Although you can also add drivers for specific FSP releases and application sample code here, this tab is normally only used for reference.

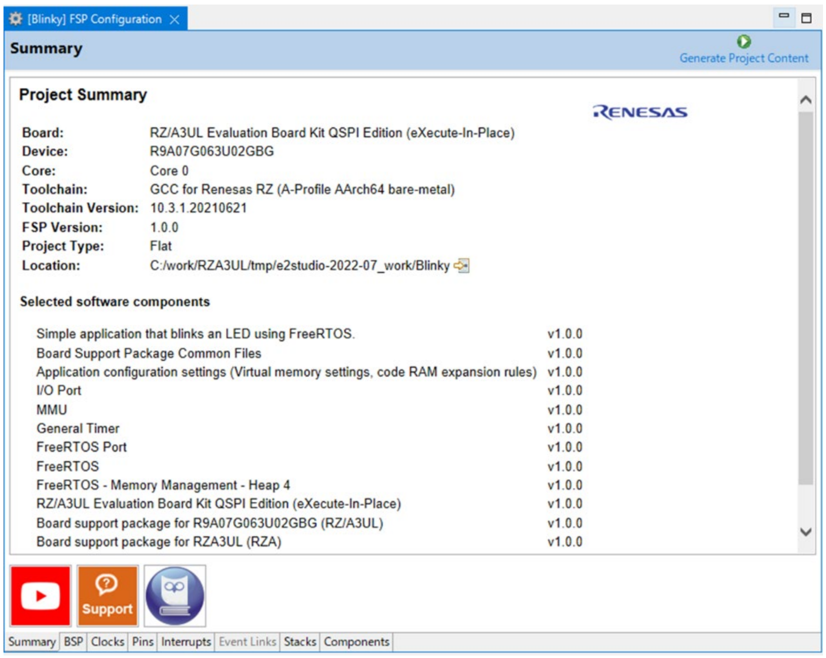
The functions and use of each of supported tabs is explained in detail in the next section.

Please note that RZ/A FSP doesn’t support **Event Links** tab and so, those tabs are grayed out as shown above.

5.2 Configuring a Project

Each of the configurable elements in an FSP project can be edited using the appropriate tab in the RZ Configuration editor window. Importantly, the initial configuration of the MPU after reset and before any user code is executed is set by the configuration settings in the **BSP** tab. When you select a project template during project creation, e2 studio configures default values that are appropriate for the associated board. You can change those default values as needed. The following sections detail the process of configuring each of the project elements for each of the associated tabs.

5.2.1 Summary Tab



Project Summary

Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)
Device:	R9A07G063U02GBG
Core:	Core 0
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)
Toolchain Version:	10.3.1.20210621
FSP Version:	1.0.0
Project Type:	Flat
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky

Selected software components

Simple application that blinks an LED using FreeRTOS.	v1.0.0
Board Support Package Common Files	v1.0.0
Application configuration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0
I/O Port	v1.0.0
MMU	v1.0.0
General Timer	v1.0.0
FreeRTOS Port	v1.0.0
FreeRTOS	v1.0.0
FreeRTOS - Memory Management - Heap 4	v1.0.0
RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0
Board support package for R9A07G063U02GBG (RZ/A3UL)	v1.0.0
Board support package for RZA3UL (RZA)	v1.0.0

Figure 63 : Configuration Summary tab

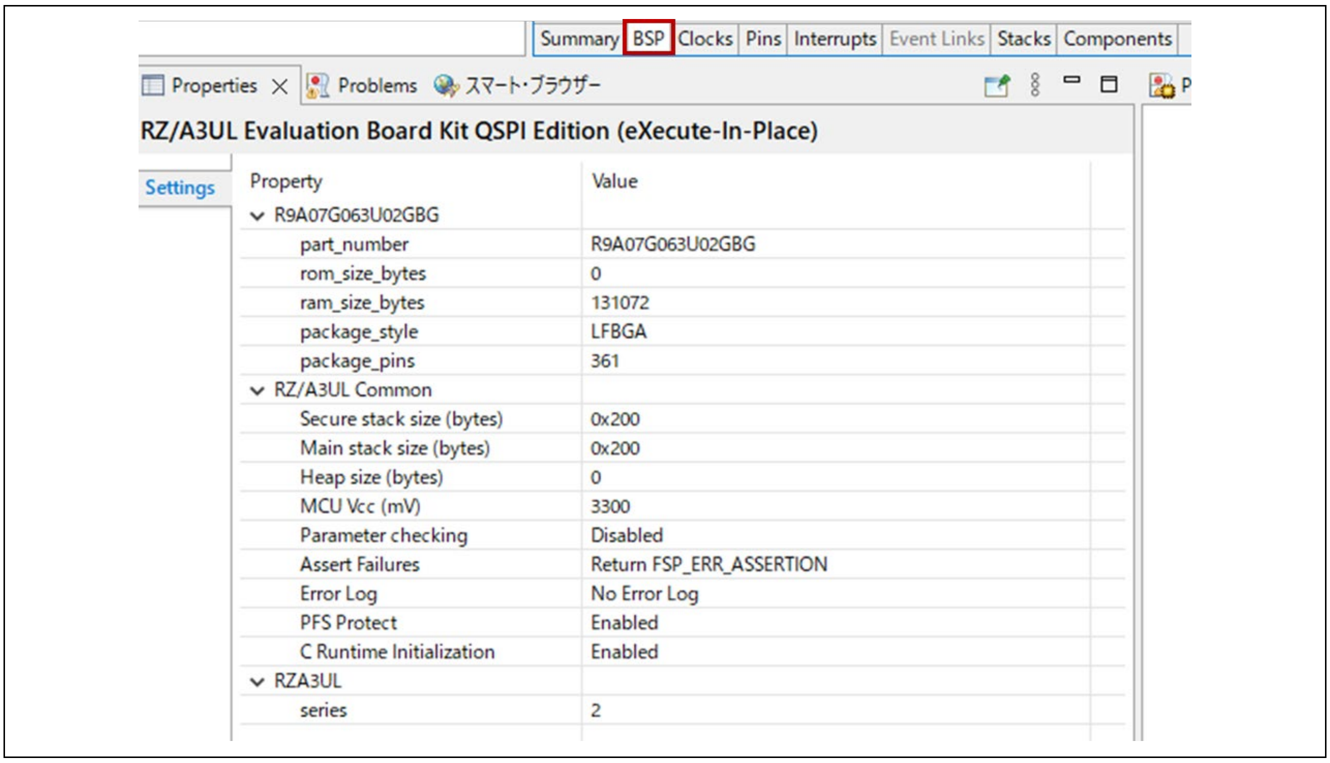
The **Summary** tab, seen in the above figure, identifies all the key elements and components of a project. It shows the target board, the device, toolchain and FSP version. Additionally, it provides a list of all the selected software components and modules used by the project. This is a more convenient summary view when compared to the **Components** tab.

5.2.2 Configuring the BSP

The **BSP** tab shows the currently selected board (if any) and device. The Properties view is located in the lower left of the Project Configurations view as shown below.

Note:

If the Properties view is not visible, click **Window > Show View > Properties** in the top menu bar.



RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)

Property	Value
R9A07G063U02GBG	
part_number	R9A07G063U02GBG
rom_size_bytes	0
ram_size_bytes	131072
package_style	LFBGA
package_pins	361
RZ/A3UL Common	
Secure stack size (bytes)	0x200
Main stack size (bytes)	0x200
Heap size (bytes)	0
MCU Vcc (mV)	3300
Parameter checking	Disabled
Assert Failures	Return FSP_ERR_ASSERTION
Error Log	No Error Log
PFS Protect	Enabled
C Runtime Initialization	Enabled
RZA3UL	
series	2

Figure 64 : Configuration BSP tab

The **Properties** view shows the configurable options available for the BSP. These can be changed as required. The BSP is the FSP layer above the MPU hardware.

When you click the **Generate Project Content** button, the BSP configuration contents are written to rza_cfg/fsp_cfg/bsp/bsp_cfg.h This file is created if it does not already exist.

Warning:

Do not edit this file as it is overwritten whenever the Generate Project Content button is clicked.

5.2.3 Configuring Clocks

The **Clocks** tab presents a graphical view of the MPU's clock tree, and each HAL driver uses the settings for dedicated numerical calculation. For example, scif_uart driver calculates the communication rate from the settings in Clocks tab. Please note that PLLs should be configured by IPL and therefore, PLL settings should be consistent with those in IPL.

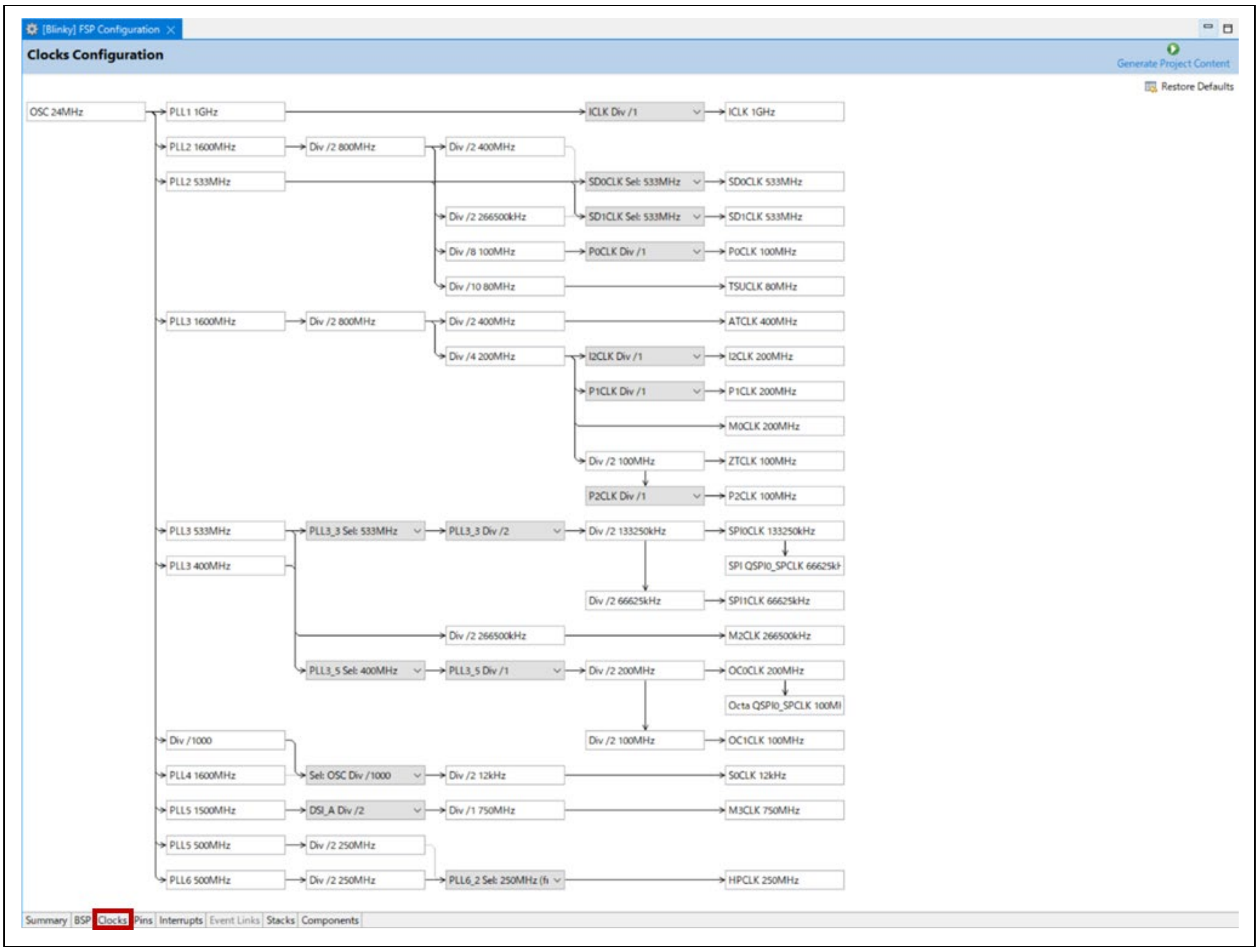
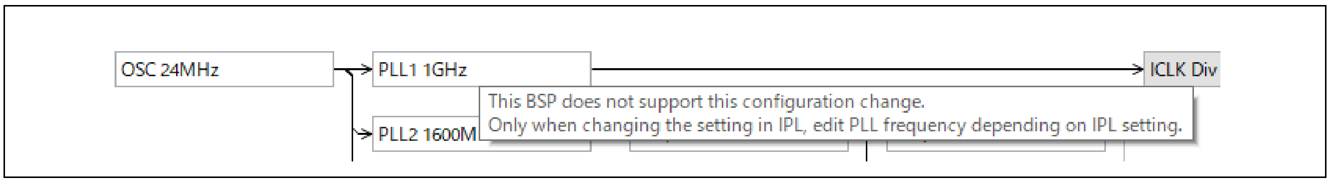


Figure 65 : Configuration Clocks tab

When mousing over the blocks of PLLs on clocks tab, you should see the pop-up message describing this precaution.



This BSP does not support this configuration change.
Only when changing the setting in IPL, edit PLL frequency depending on IPL setting.

Figure 66 : Precautions for PLL settings

When you click the **Generate Project Content** button, the clock configuration contents are written to: rza_gen/bsp_clock_cfg.h

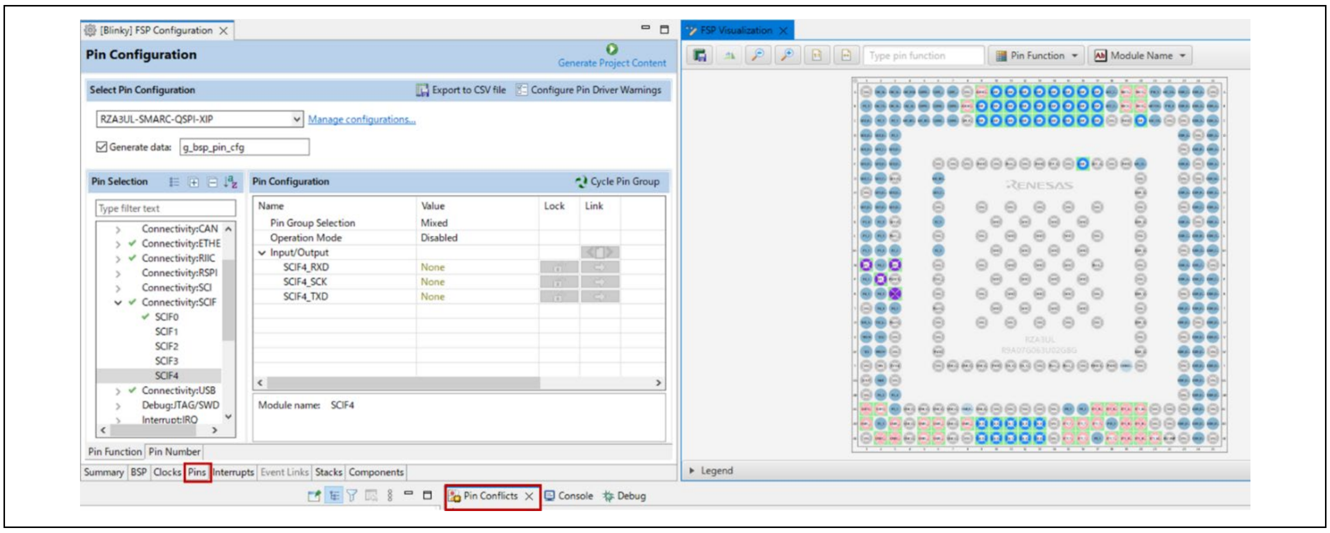
This file will be created if it does not already exist.

Warning:

Do not edit this file as it is overwritten whenever the **Generate Project Content** button is clicked.

5.2.4 Configuring Pins

The pins tab provides flexible configuration of the MPU's pins. As many pins can provide multiple functions, they can be configured on a peripheral basis. For example, selecting a serial channel via the SCIF peripheral offers multiple options for the location of the receive and transmit pins for that module and channel. Once a pins is configured, it is shown as green in the **FSP Visualization** view.



[Blinky] FSP Configuration

Pin Configuration

Select Pin Configuration: RZA3UL-SMARC-QSPI-XIP

Generate data: g_bsp_pin_cfg

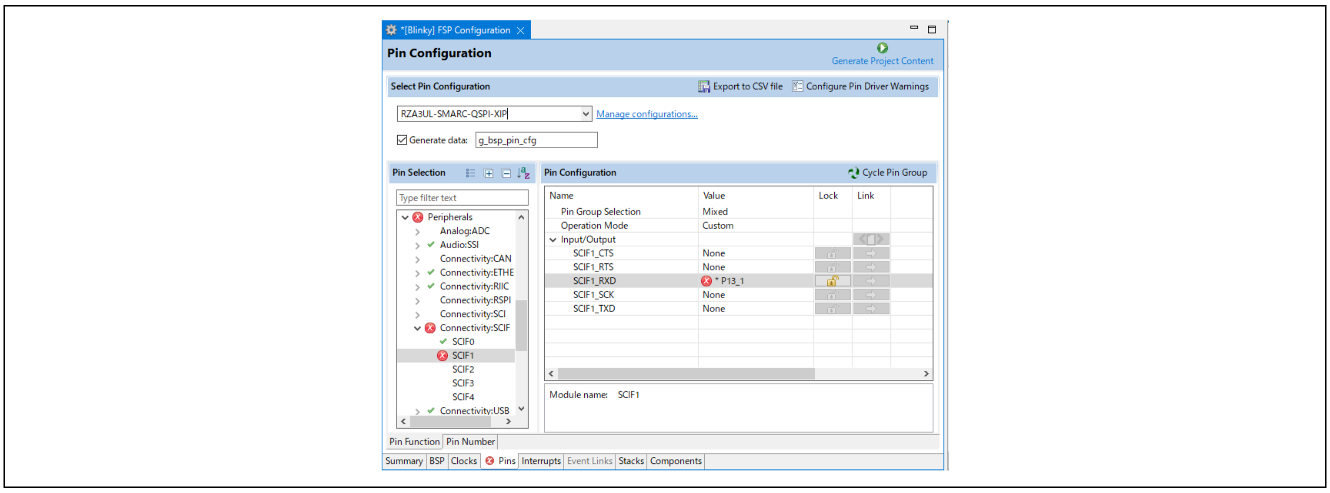
Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Disabled		
Input/Output			
SCIF4_RXD	None		
SCIF4_SCK	None		
SCIF4_TXD	None		

Module name: SCIF4

Figure 67 : Pin Configuration

The pin configurator includes built-in conflict checker, so if the same pin is allocated to another peripheral or I/O function, the pin will be shown as red in the **FSP Visualization** view and also with white cross in a red square in the **Pin Selection** pane and **Pin Configuration** pane in the main **Pins** tab.

In the example shown below, port P13_2 is already used by the Display, and the attempt to connect to this pin to the Serial Communication Interface with FIFO (SCIF) results in dangling connection error. To fix this error, select another port from the pin drop-down list or disable the Display.



*[Blinky] FSP Configuration

Pin Configuration

Select Pin Configuration: RZA3UL-SMARC-QSPI-XIP

Generate data: g_bsp_pin_cfg

Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Custom		
Input/Output			
SCIF1_CTS	None		
SCIF1_RTS	None		
SCIF1_RXD	* P13_1		
SCIF1_SCK	None		
SCIF1_TXD	None		

Module name: SCIF1

Figure 68 : e2 studio Pin Configurator

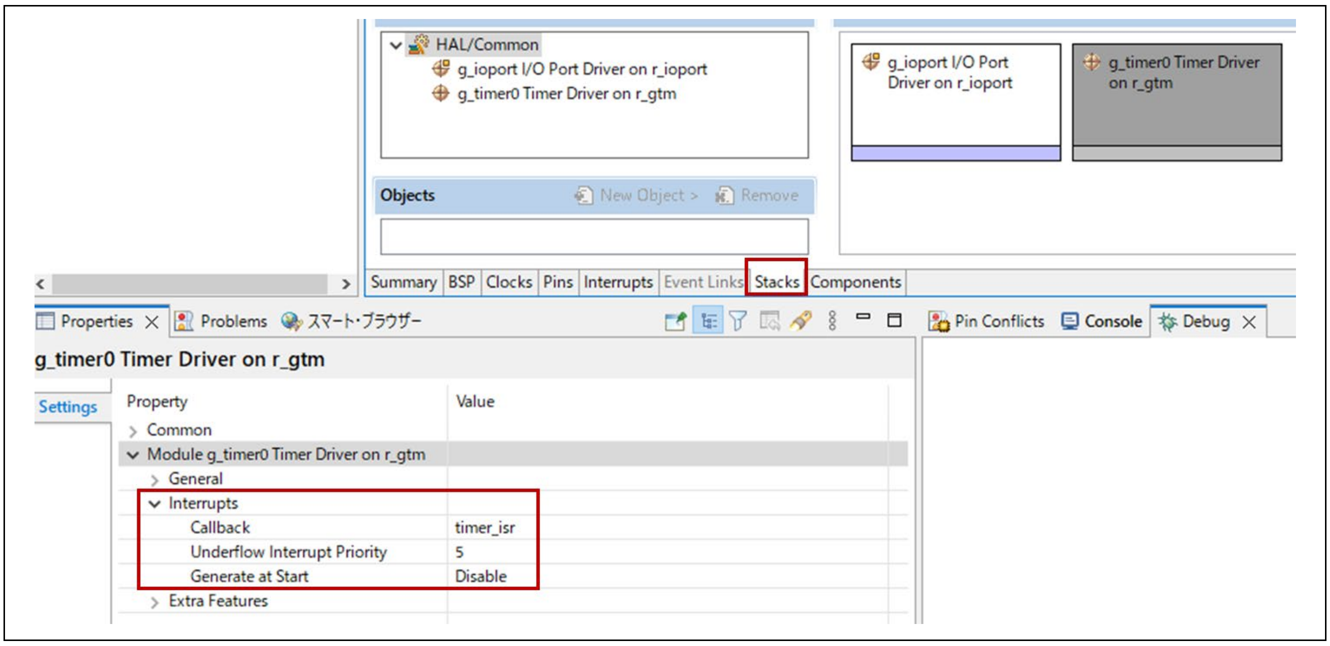
When you click the **Generate Project Content** button, the pin configuration contents are written to: ra_gen\bsp_pin_cfg.h. This file will be created if it does not already exist.

Warning:

Do not edit this file as it is overwritten whenever the **Generate Project Content** button is clicked.

5.2.5 Configuring Interrupts from the Stacks Tab

You can use the **Properties** view in the **Stacks** tab to enable interrupts by setting the interrupt priority. Select the driver in the **Stacks** pane to view and edit its properties.



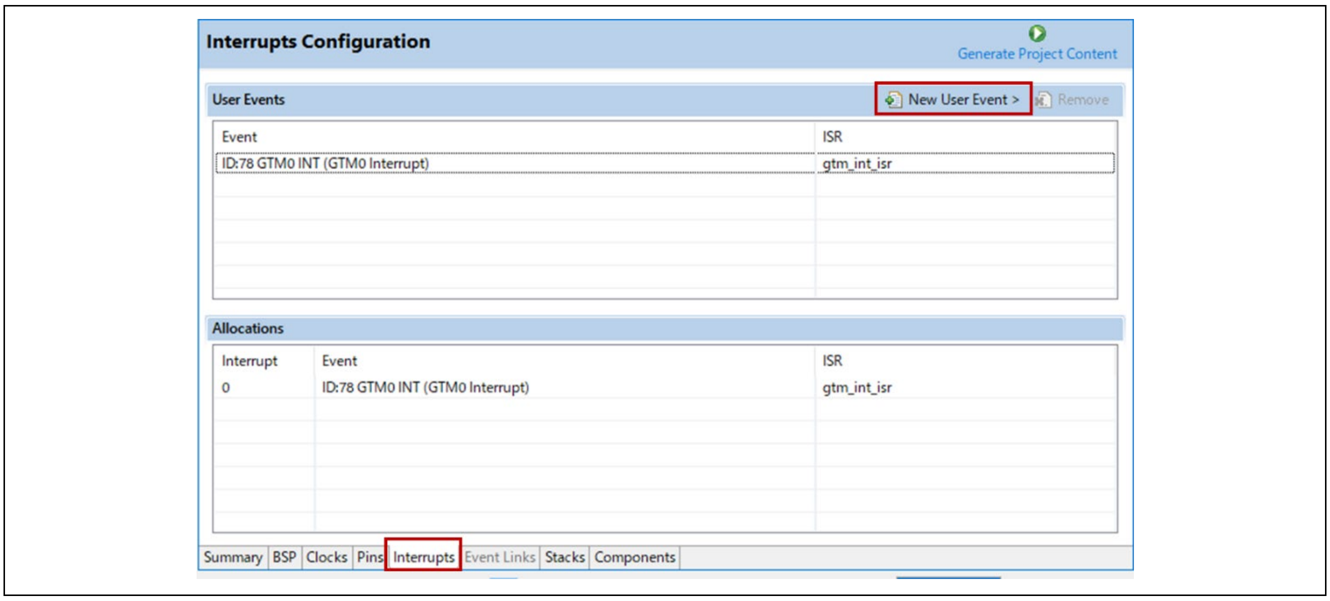
g_timer0 Timer Driver on r_gtm

Property	Value
Common	
Module g_timer0 Timer Driver on r_gtm	
General	
Interrupts	
Callback	timer_isr
Underflow Interrupt Priority	5
Generate at Start	Disable
Extra Features	

Figure 69 : Configuring Interrupts in the Stacks tab

5.2.6 Creating Interrupts from the Interrupts Tab

On the **Interrupts** tab, the user can bypass a peripheral interrupt set by the FSP by setting a user-defined ISR. This can be done by adding a new event via New User Event button.



Event	ISR
ID:78 GTM0 INT (GTM0 Interrupt)	gtm_int_isr

Interrupt	Event	ISR
0	ID:78 GTM0 INT (GTM0 Interrupt)	gtm_int_isr

Figure 70 : Configuring interrupt in Interrupt Tab

5.2.7 Viewing Event Links

RZ/A FSP does not support **Event Links** tab, and it is grayed out.

5.2.8 Adding Threads and Drivers

Every RTOS-based RZ/A FSP Project includes at least one RTOS Thread and a stack of FSP module running in that thread. The **Stacks** tab is a graphical user interface which helps you to add the right modules to a thread and configure the properties of both the threads and the modules associated with each thread. Once you have configured the thread, e2 studio automatically generates the code reflecting your configuration choices.

For any driver, or, more generally, any module that you add to a thread, e2 studio automatically resolves all dependencies with other modules and creates the appropriate stack. This stack is displayed in the **Stacks** pane, which e2 studio populates with the selected modules and module options for the selected thread.

The default view of the **Stacks** tab includes a Common Thread called **HAL/Common**. This thread includes the driver for I/O control (IOPORT). The default stack is shown in the **HAL/Common Stacks** pane. The default modules added to the HAL/Common driver are special in that the FSP only requires a single instance of each, which e2 studio then includes in every user-defined thread by default.

In applications that do not use an RTOS or run outside of the RTOS, the HAL/Common thread becomes the default location where you can add additional drivers to your application.

For a detailed description on how to add and configure modules and stacks, see the following sections:

Adding and Configuring HAL Drivers
Adding Drivers to a Thread and Configuring the Drivers

Only you have added a module either to HAL/Common or to a new thread, you can access the driver's configuration options in the **Properties** view. If you added thread objects, you were able to access the objects configuration options in the **Properties** view in the same way.

5.2.9 Adding and Configuring HAL Drivers

For applications that run outside or without the RTOS, you can add additional HAL drivers to your application using the HAL/Common thread. To add drivers, follow these steps:

1. Click on the HAL/Common icon in the **Stacks** pane. The Modules pane changes to **HAL/Common** Stacks.
2. Click New Stack to see a drop-down list of HAL level drivers available in the FSP.
3. Select a driver from the menu **New Stack > Driver**.

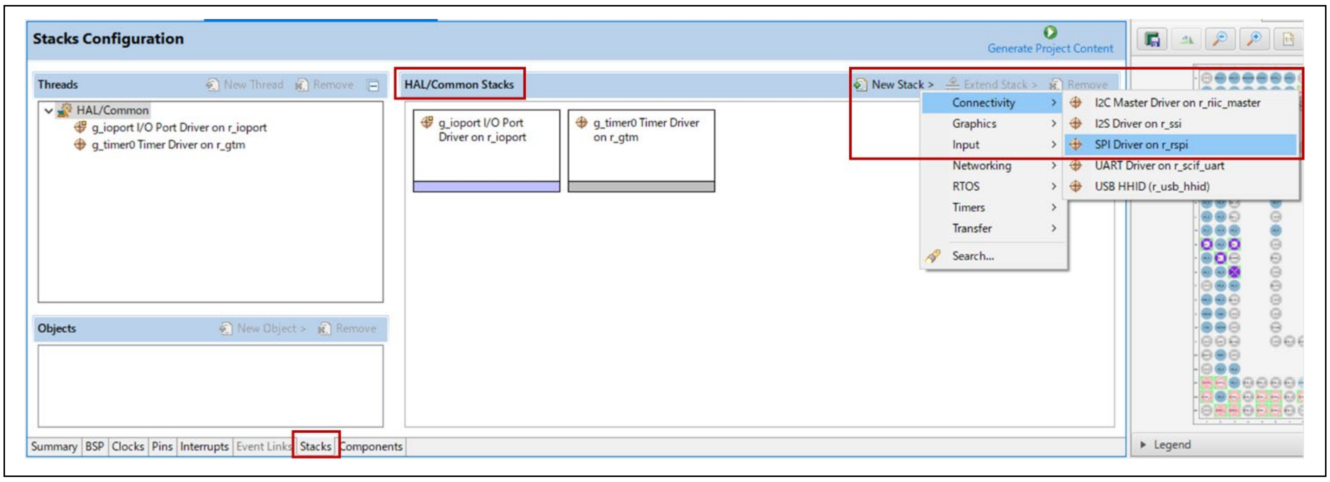


Figure 71 : e2 studio Project configurator - Adding drivers

4. Select the driver module in the **HAL/Common Modules** pane and configure the driver properties in the **Properties** view.

e2 studio adds the following files when you click the **Generate Project Content** button:

- The selected driver module and its files to the rza/fsp directory
- The main() function and configuration structures and header files for your application as shown in the table below.

File	Contents	Overwritten by Generate Project Content?
rza_gen/main.c	Contains main() calling generated and user code. When called, the BSP has already initialized the MPU	Yes
rza_gen/hal_data.c	Configuration structures for HAL Driver only modules	Yes
rza_gen/hal_data.h	Header file for HAL driver only modules	Yes
src/hal_entry.c	User entry point for HAL Driver only code. Add your code here	No
src/mmu_page_table.c	Virtual memory page table settings	No
src/sections.c	Rules for section transfer from ROM to RAM	No
src/syscalls.c	Low-level processing stub for file I/O functions	No

The configuration header files for all included modules are created or overwritten in this folder:

```
rza_cfg/fsp_cfg
```

5.2.10 Adding Drivers to a Thread and Configuring the Drivers

For an application that uses the RTOS, you can add one or more threads, and for each thread at least one module that runs in the thread. You can select modules from the Driver dropdown menu. To add modules to a thread, follow these steps:

1. In the **Threads** pane, click **New Thread** to add a Thread.

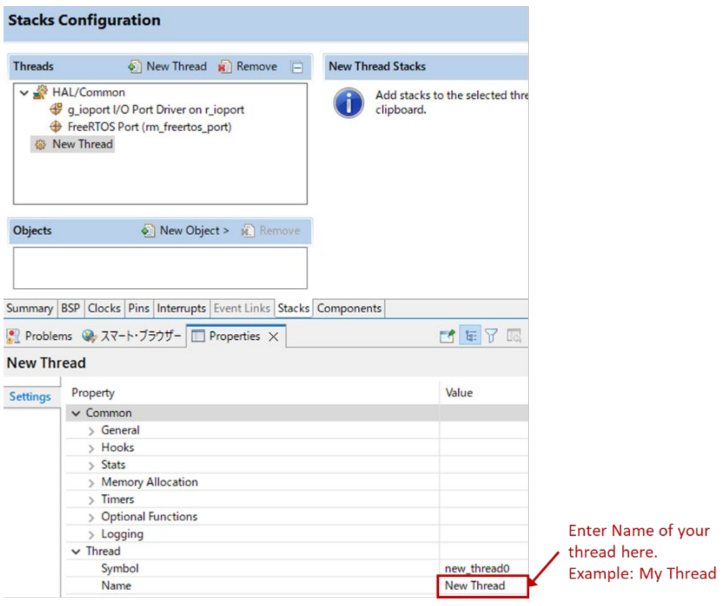


Figure 72 : Adding a new RTOS Thread on the Stack tab

2. In the properties view, click on the Name and Symbol entries and enter distinctive name and symbol for the new thread.

Note:

e2 studio updates the name of the thread stacks pane to **My Thread Stacks**.

3. In the **My Thread Stacks** pane, click on **New Stack** to see a list of modules and drivers.

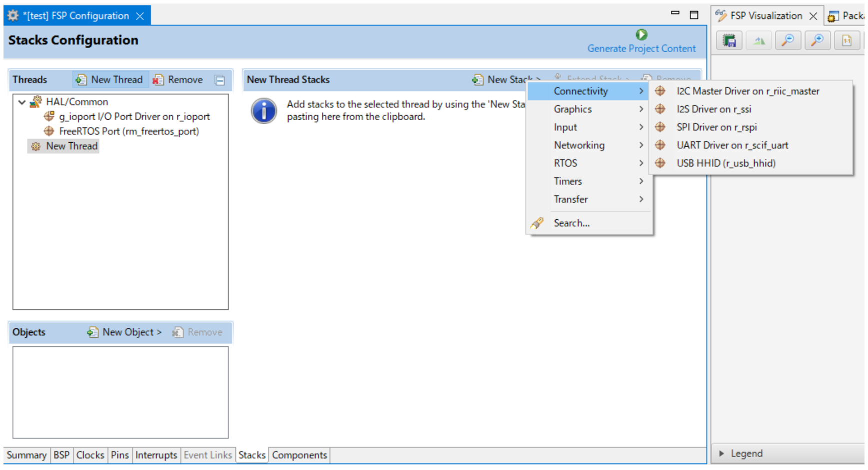
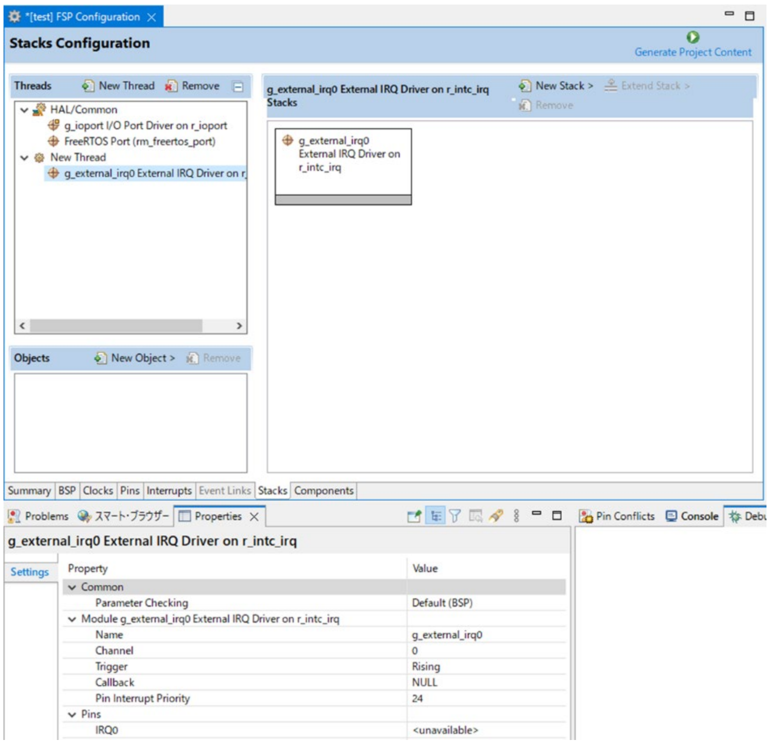


Figure 73 : Adding Modules and Drivers to a thread

4. Select a module or driver from the list.
5. Click on the added driver and configure the driver as required by the application by updating the configuration parameters in **Properties** view. To see the selected module or driver and be able to edit its properties, make sure the Thread containing the driver is highlighted in the **Threads** pane.



g_external_irq0 External IRQ Driver on r_intc_irq

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_external_irq0 External IRQ Driver on r_intc_irq	
Name	g_external_irq0
Channel	0
Trigger	Rising
Callback	NULL
Pin Interrupt Priority	24
Pins	
IRQ0	<unavailable>

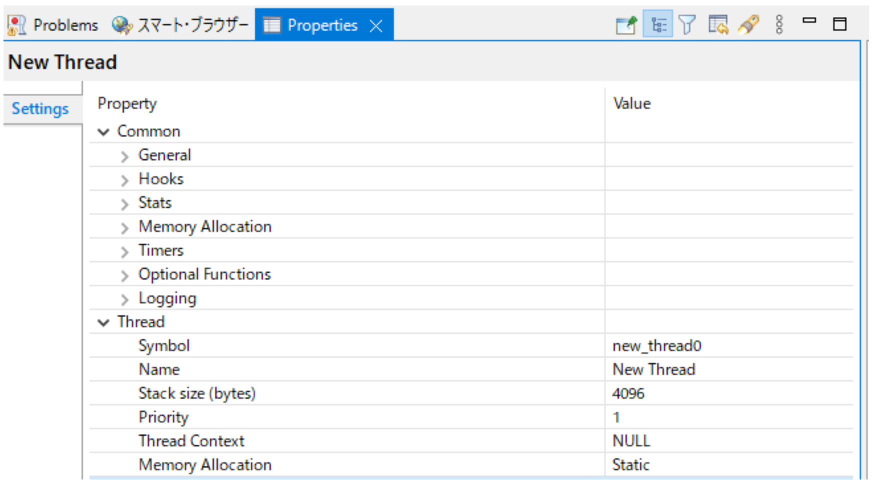
Figure 74 : Configuring Module or Driver properties

6. When you press the Generate Project Content button for the example above, e2 studio created the files as shown in the following table:

File	Contents	Overwritten by Generate Project Content?
rza_gen/main.c	Contains main() calling generated and user code. When called, the BSP will have initialized the MPU.	Yes
rza_gen/my_thread.c	Generated thread “my_thread” and configuration structures for modules added to this thread.	Yes
rza_gen/my_thread.h	Header file for thread “my_thread”	Yes
rza_gen/hal_data.c	Configuration structures for HAL Driver only modules.	Yes
rza_gen/hal_data.h	Header file for HAL driver only modules.	Yes
src/hal_entry.c	User entry point for HAL Driver only code. Add your code here.	No
src/my_thread_entry.c	User entry point for thread “my_thread”. Add your code here.	No

5.2.11 Configuring Threads

If the application uses an RTOS, the Stacks tab can be used to simplify the creation of RTOS threads, semaphores, mutexes, and event flags. The components of each thread can be configured from the **Properties** view as shown below:



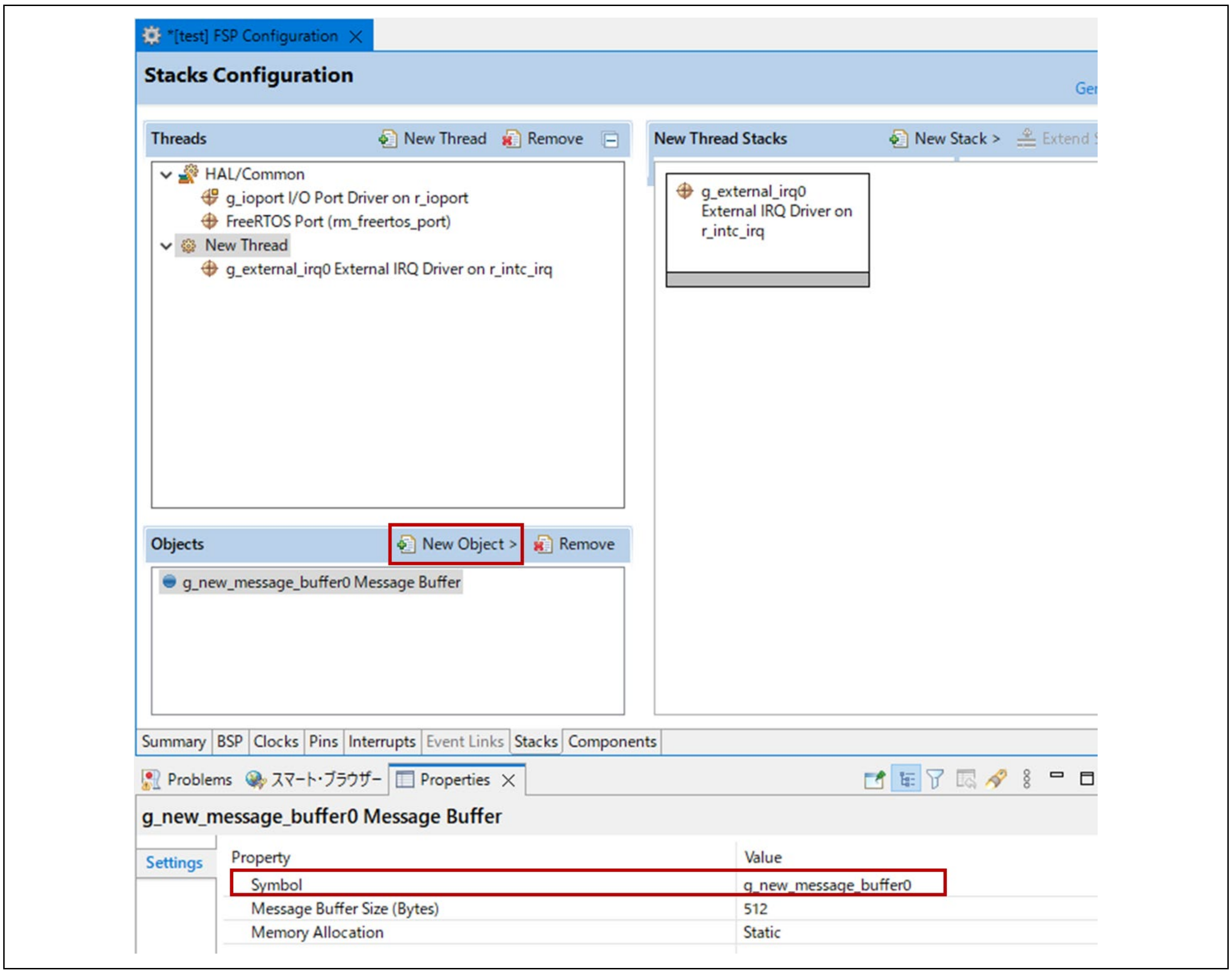
Property	Value
Common	
General	
Hooks	
Stats	
Memory Allocation	
Timers	
Optional Functions	
Logging	
Thread	
Symbol	new_thread0
Name	New Thread
Stack size (bytes)	4096
Priority	1
Thread Context	NULL
Memory Allocation	Static

Figure 75 : New Thread Properties

The Properties view contains settings common for all Threads (**Common**) and settings for this particular thread (**Thread**).

For this thread instance, the thread's name and properties (such as priority level or stack size) can be easily configured. e2 studio checks that the entries in the property field are valid. For example, it will verify that the field **Priority**, which requires an integer value, only contains numeric values between 0 and 9.

To add RTOS resources to a Thread, select a thread and click on **New Object** in the Thread Objects pane. The pane takes on the name of the selected thread, in this case **My Thread Objects**.


*[test] FSP Configuration
Stacks Configuration
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port Driver on r_ioport
FreeRTOS Port (rm_freertos_port)
New Thread
g_external_irq0 External IRQ Driver on r_intc_irq
New Thread Stacks
New Stack >
Extend
g_external_irq0 External IRQ Driver on r_intc_irq
Objects
New Object >
Remove
g_new_message_buffer0 Message Buffer
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components
Problems
スマート・ブラウザー
Properties
g_new_message_buffer0 Message Buffer
Settings

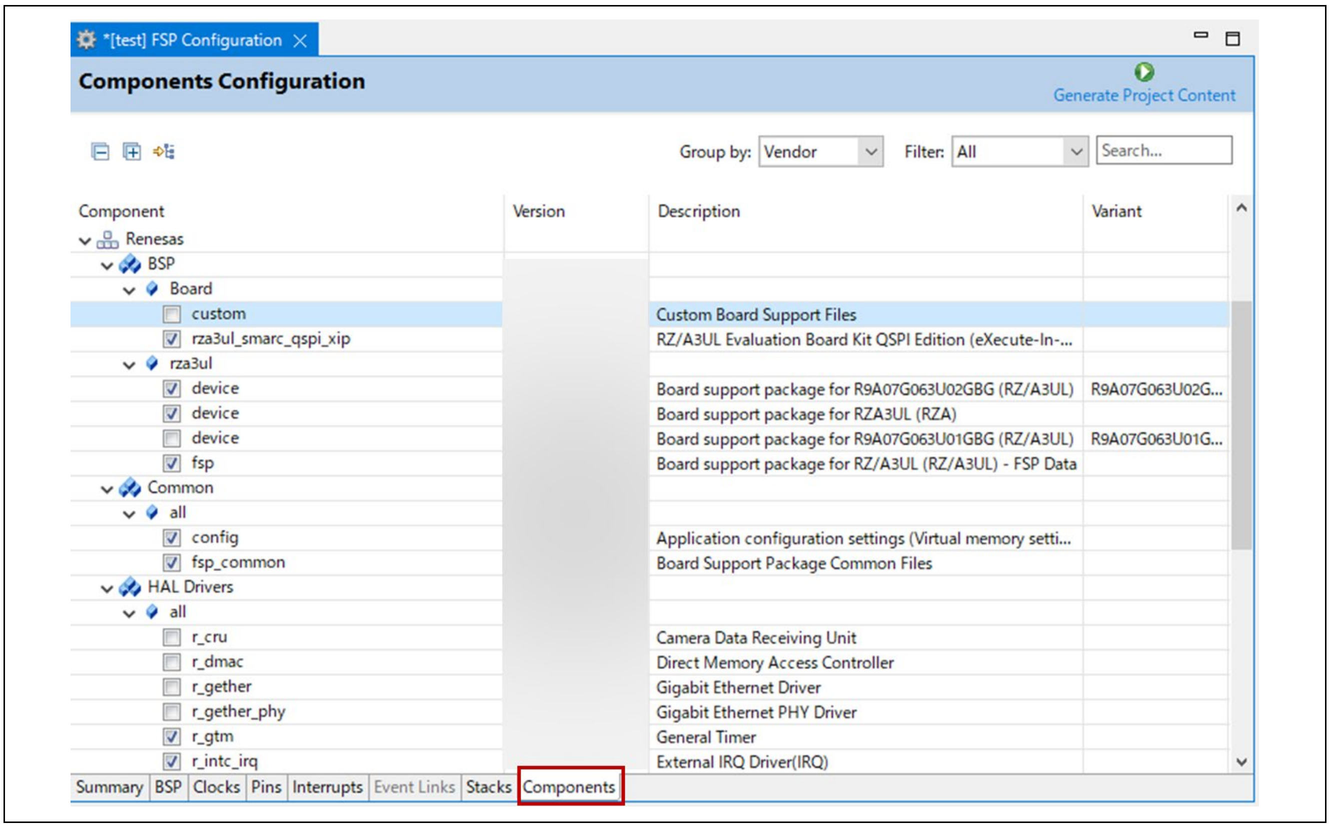
Property	Value
Symbol	g_new_message_buffer0
Message Buffer Size (Bytes)	512
Memory Allocation	Static

Figure 76 : Configuring Thread Object Properties

Make sure to give each thread object a unique symbol by updating the **Symbol** entries in the **Properties** view.

5.3 Reviewing and Adding Components

The **Components** tab enables the individual modules required by the application to be included or excluded. Modules common to all RZ/A MPU projects are preselected. All modules that are necessary for the modules selected in the **Stacks** tab are included automatically. You can include or exclude additional modules by ticking the box next to the required component.



Component	Version	Description	Variant
Renesas			
BSP			
Board			
☐ custom		Custom Board Support Files	
☑ rza3ul_smarc_qspi_xip		RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-...	
rza3ul			
☑ device		Board support package for R9A07G063U02GBG (RZ/A3UL)	R9A07G063U02G...
☑ device		Board support package for RZA3UL (RZA)	
☐ device		Board support package for R9A07G063U01GBG (RZ/A3UL)	R9A07G063U01G...
☑ fsp		Board support package for RZ/A3UL (RZ/A3UL) - FSP Data	
Common			
all			
☑ config		Application configuration settings (Virtual memory setti...	
☑ fsp_common		Board Support Package Common Files	
HAL Drivers			
all			
☐ r_cru		Camera Data Receiving Unit	
☐ r_dmac		Direct Memory Access Controller	
☐ r_gether		Gigabit Ethernet Driver	
☐ r_gether_phy		Gigabit Ethernet PHY Driver	
☑ r_gtm		General Timer	
☑ r_intc_irq		External IRQ Driver(IRQ)	

Figure 77 : Components Tab

Clicking the **Generate Project Content** button copies the .c and .h files for each selected component into the following folders:

- rza/fsp/inc/api
- rza/fsp/inc/instances
- rza/fsp/src/bsp
- rza/fsp/src/<Driver_Name>

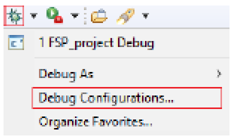
e2 studio also creates configuration files in the rza_cfg/fsp_cfg folder with configuration options set in the **Stacks** tab.

5.4 Debugging the Project

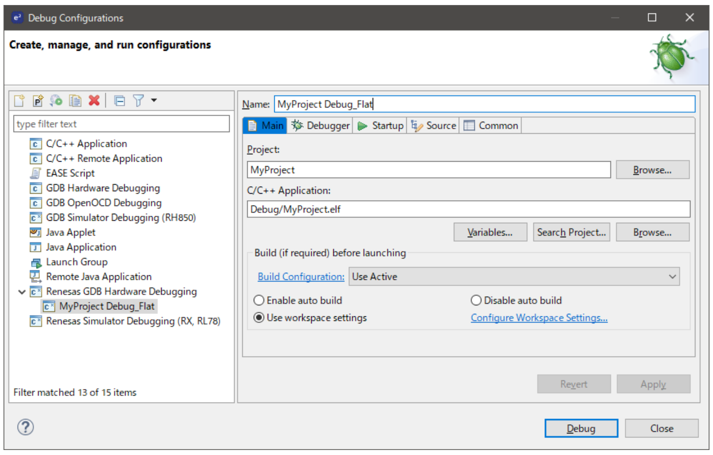
Once your project builds without errors, you can use the Debugger to download your application to the board and execute it.

To debug an application, follow these steps:

1. On the drop-down list next to the debug icon, select **Debug Configurations**.



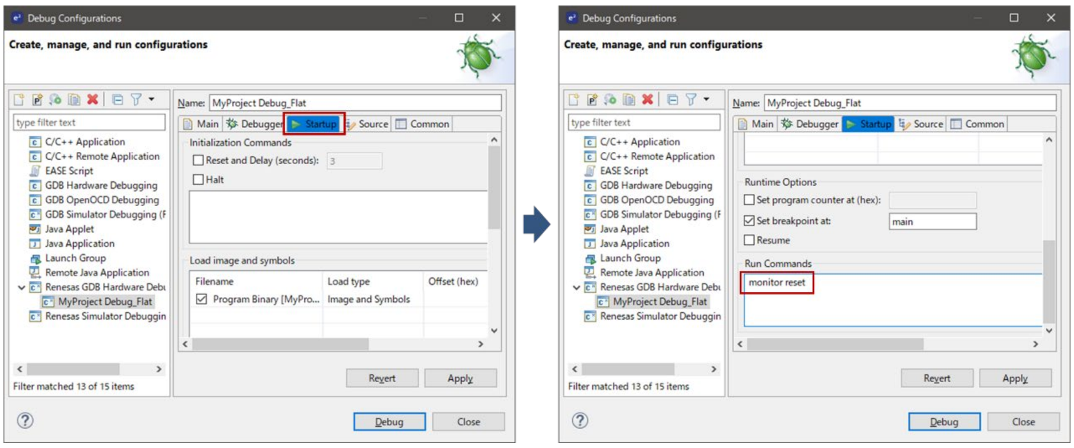
2. In the **Debug Configurations** view, click on your project listed as **MyProject Debug_Flat**.



Name: MyProject Debug_Flat
Project: MyProject
C/C++ Application: Debug/MyProject.elf
Build Configuration: Use Active
● Use workspace settings

3. Choose **Startup** tab and enter the following command in Run Commands located at the bottom of the window:

```
monitor reset
```



Name: MyProject Debug_Flat
Reset and Delay (seconds): 3
☑ Set breakpoint at: main
Run Commands: monitor reset

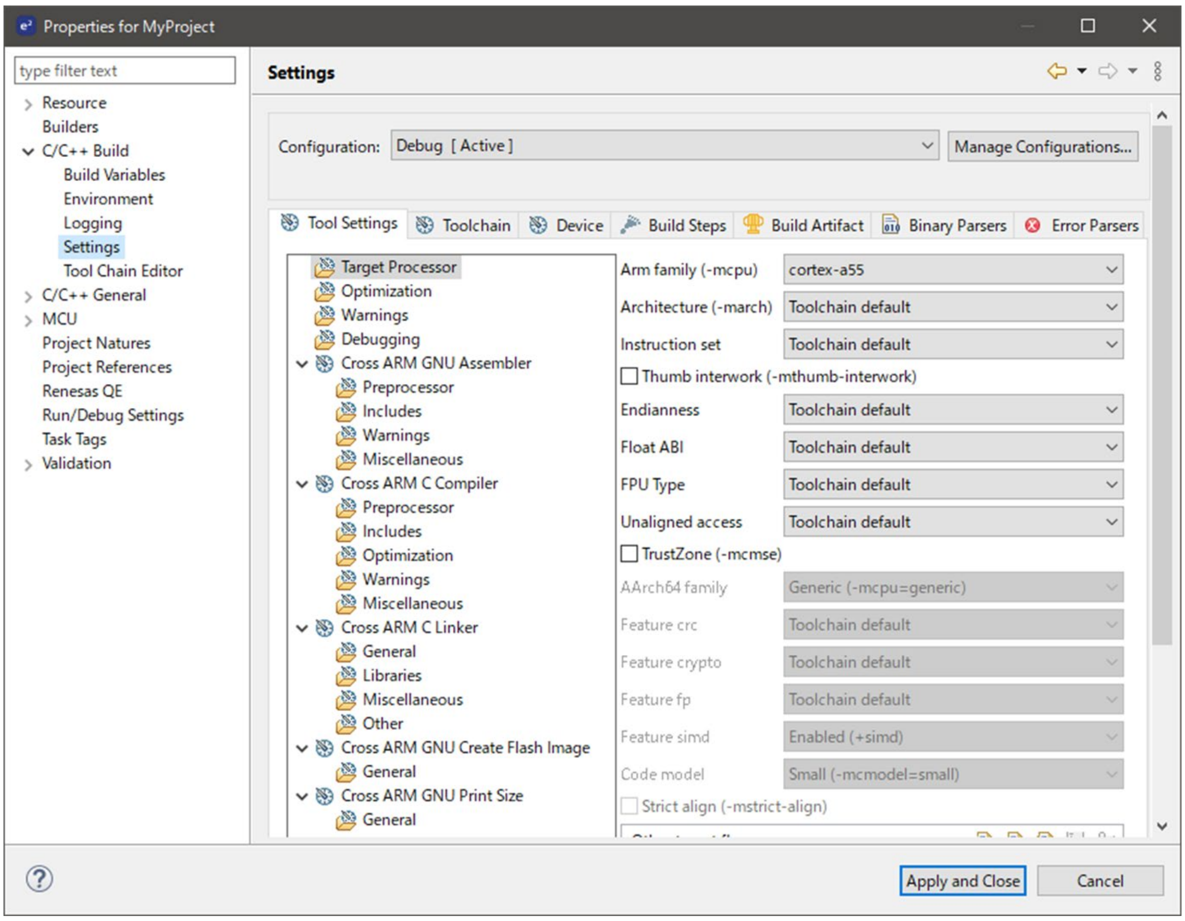
4. Connect the board to your PC via a standalone Segger J-Link debugger and click **Debug**.

Note:

For details on using J-Link and connecting the board to the PC, see 3.2.2.JTAG connection.

5.5 Modifying Toolchain Settings

There are instances where it may be necessary to make changes to the toolchain being used (for example, to change optimization level of the compiler or add a library to the linker). Such modifications can be made within e2 studio through the menu **Project > Properties > Settings** when the project is selected. The following screenshot shows the settings dialog for the GNU Arm toolchain. This dialog will look slightly different depending upon the toolchain being used.



Properties for MyProject

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Environment
  - Logging
  - Settings
  - Tool Chain Editor
- C/C++ General
- MCU
- Project Natures
- Project References
- Renesas QE
- Run/Debug Settings
- Task Tags
- Validation

Settings

Configuration: Debug [Active] Manage Configurations...

Tool Settings | Toolchain | Device | Build Steps | Build Artifact | Binary Parsers | Error Parsers

- Target Processor
- Optimization
- Warnings
- Debugging
- Cross ARM GNU Assembler
  - Preprocessor
  - Includes
  - Warnings
  - Miscellaneous
- Cross ARM C Compiler
  - Preprocessor
  - Includes
  - Optimization
  - Warnings
  - Miscellaneous
- Cross ARM C Linker
  - General
  - Libraries
  - Miscellaneous
  - Other
- Cross ARM GNU Create Flash Image
  - General
- Cross ARM GNU Print Size
  - General

Arm family (-mcpu)	cortex-a55
Architecture (-march)	Toolchain default
Instruction set	Toolchain default
☐ Thumb interwork (-mthumb-interwork)	
Endianness	Toolchain default
Float ABI	Toolchain default
FPU Type	Toolchain default
Unaligned access	Toolchain default
☐ TrustZone (-mcmse)	
AArch64 family	Generic (-mcpu=generic)
Feature crc	Toolchain default
Feature crypto	Toolchain default
Feature fp	Toolchain default
Feature simd	Enabled (+simd)
Code model	Small (-mcmodel=small)
☐ Strict align (-mstrict-align)	

Apply and Close Cancel

Figure 78 : e2 studio Project toolchain settings

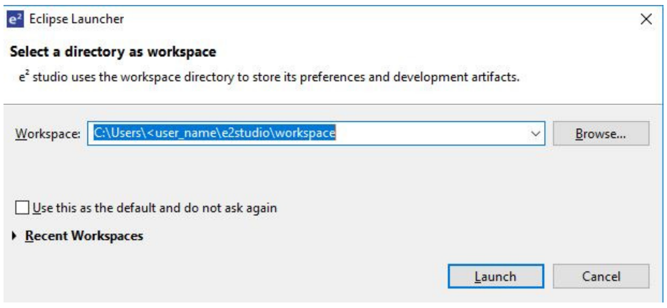
The scope for the settings is project scope which means that the settings are valid only for the project being modified.

The settings for the linker which control the location of the various memory sections are contained in a script file specific for the device being used. This script file is included in the project when it is created and is found in the created project. (for example, script/rza3ul_smarc_qsip_xip.ld).

5.6 Importing an Existing Project into e2 studio

1. Start by opening e2 studio.
2. Open an existing Workspace to import the project and skip to step d. If the workspace does not exist, proceed with the following steps:

   a. At the end of e2 studio startup, you will see the Workspace Launcher Dialog box as shown in the following figure.



Eclipse Launcher

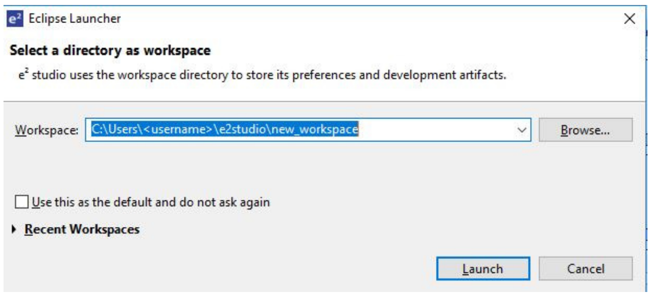
Select a directory as workspace

e² studio uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\<user_name\e2studio\workspace

Figure 79 : Workspace Launcher dialog

   b. Enter a new workspace name in the Workspace Launcher Dialog as shown in the following figure. e2 studio creates a new workspace with this name.



Eclipse Launcher

Select a directory as workspace

e² studio uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\<username>\e2studio\new_workspace

Figure 80 : Workspace Launcher dialog - Select Workspace

   c. Click **Launch**.
   d. When the workspace is opened, you may see the Welcome Window. Click on the **Workbench** arrow button to proceed past the Welcome Screen as seen in the following figure.

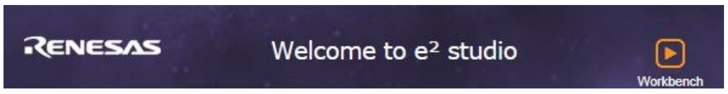
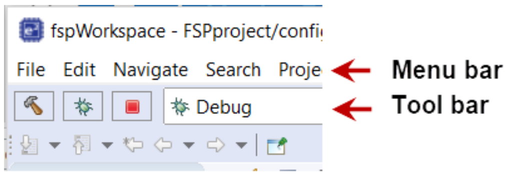


Figure 81 : Workbench arrow button

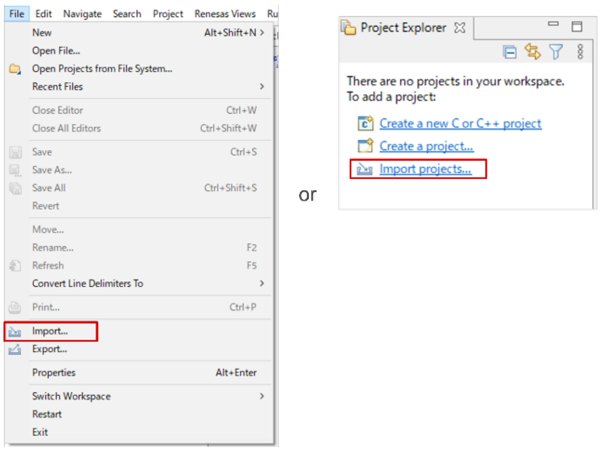
3. You are now in the workspace that you want to import the project into. Click the **File** menu in the menu bar, as shown in the following figure.



fspWorkspace - FSPproject/confi
File Edit Navigate Search Proje
Debug
Menu bar
Tool bar

Figure 82 : Menu and tool bar

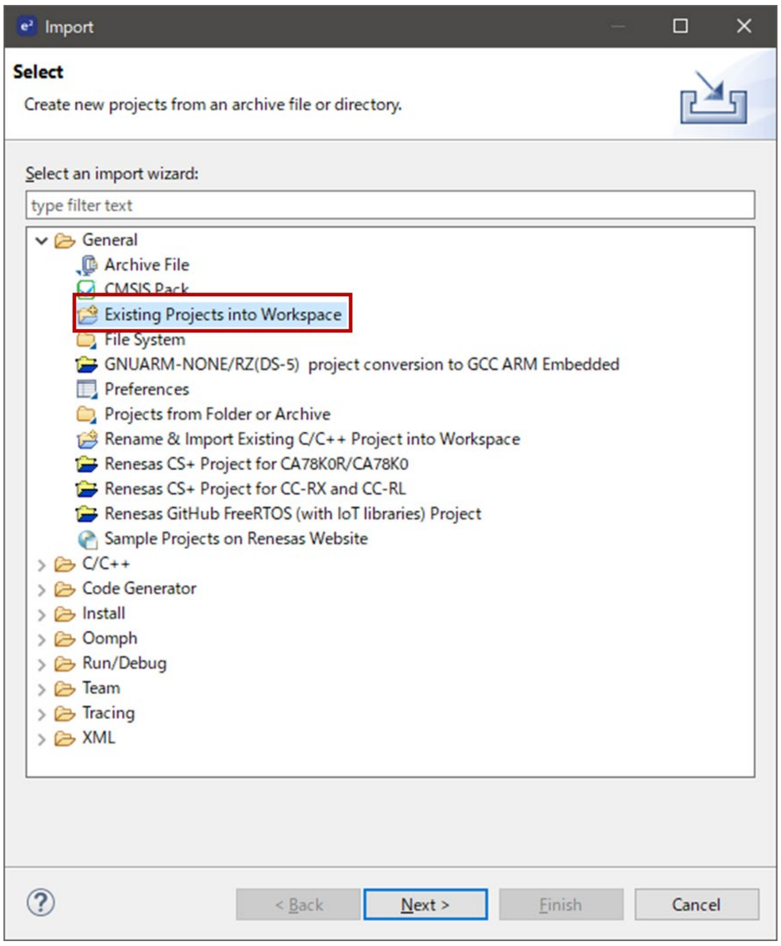
4. Click **Import** on the **File** menu or “Import project” on Project Explorer, as shown in the following figure.



There are no projects in your workspace.
To add a project:

Figure 83 : File drop-down menu

5. In the **Import** dialog box, as shown in the following figure, choose the **General** option, then **Existing Projects into Workspace**, to import the project into the current workspace.


Import
Select
Select an import wizard:
type filter text
General
Archive File
CMSIS Pack
Existing Projects into Workspace
File System
GNUARM-NONE/RZ(DS-5) project conversion to GCC ARM Embedded
Preferences
Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace
Renesas CS+ Project for CA78K0R/CA78K0
Renesas CS+ Project for CC-RX and CC-RL
Renesas GitHub FreeRTOS (with IoT libraries) Project
Sample Projects on Renesas Website
C/C++
Code Generator
Install
Oomph
Run/Debug
Team
Tracing
XML
< Back
Next >
Finish
Cancel

Create new projects from an archive file or directory.

Figure 84 : Project Import dialog with "Existing Projects into Workspace" option selected

6. Click **Next**.
7. To import the project, use either **Select archive file** or **Select root directory**.

a. Click **Select root directory** file as shown in the following figure.

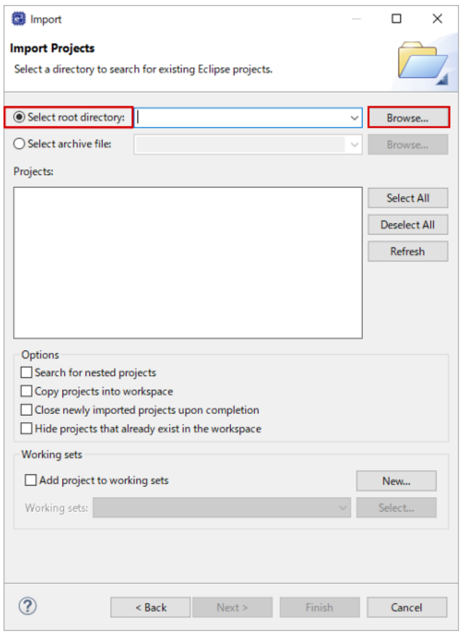

Import
Import Projects
Select a directory to search for existing Eclipse projects.
Select root directory:
Browse...
Select archive file:
Browse...
Projects:
Select All
Deselect All
Refresh
Options
Search for nested projects
Copy projects into workspace
Close newly imported projects upon completion
Hide projects that already exist in the workspace
Working sets
Add project to working sets
New...
Working sets:
Select...
< Back
Next >
Finish
Cancel

Figure 85 : Import Existing Project dialog 1 - Select root directory

8. Click **Browse**.
9. For **Select root directory**, browse to the project folder that you want to import.
10. Select the file for import.
11. Click **Open**.
12. Select the project to import from the list of **Projects**, as shown in the following figure.

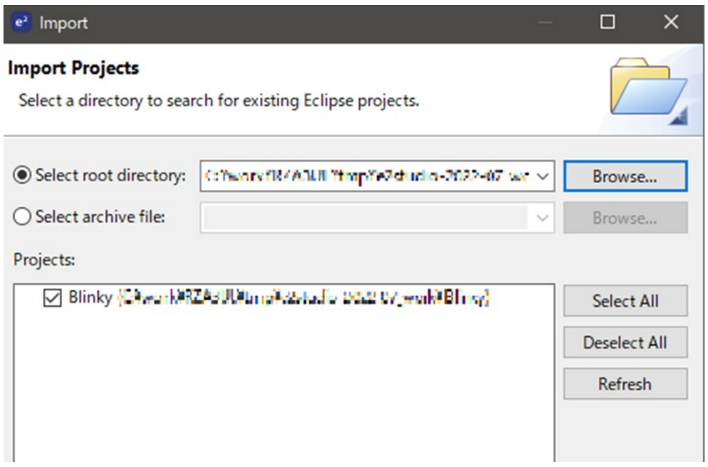


Figure 86 : Import Existing Project dialog 2

13. Click **Finish** to import the project.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.28.2022	-	First Edition issued.

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.