

RZ/A3UL, RZ/A3M

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

RZ/A3M



Contents

1. Introduction	1
1.1 Overview	
1.2 Introduction to FSP	
1.2.1 Purpose	
1.2.2 e2 studio IDE	
1.3 Limitations	
1.3.1 Hardware Initial Setup	
2. Starting Development Introduction	
2.1 e2 studio setup	5
2.1.1 What is e2 studio?	5
2.1.2 e2 studio Prerequisites	5
2.1.3 e2 studio installation for Windows PC	5
2.1.4 e2 studio installation for Linux PC	16
2.2 FSP setup	22
2.2.1 Installation of FSP Packs using Platform Installer	22
2.2.2 Installation of FSP Packs using Package Installer	27
2.2.3 Installation of FSP Packs using Package Zip file	30
3. Set Up a Target Board	30
3.1 Supported Debugger	
3.2 Board Setup	
3.2.1 Boot Mode	
3.2.2 JTAG connection	
3.2.3 Debug Serial (console output)	
3.2.4 Power Supply	
3.2.5 How to check if your board is operational	
4. Tutorial: Your First RZ MPU Project - Blinky	38
4.1 Tutorial Blinky	38
4.2 What Does Blinky Do?	38
4.3 Creating a New Project for Blinky	39
4.3.1 Details about the Blinky Configuration	42
4.3.2 Configuring the Blinky Clocks	42
4.3.3 Configuring the Blinky Pins	42
4.3.4 Configuring the Parameters for Blinky Components	42
4.3.5 Where is main()?	
4.3.6 Blinky Example Code	
4.4 Build the Blinky Project	
4.5 Debug the Blinky Project	
4.5.1 Debug prerequisites	44



4.5.2 Debug steps	
4.6 Details about the Debug Process	51
4.6.1 Run the Blinky Project	51
5. FSP Application Launch with e2 studio	53
5.1 Create a Project	53
5.1.1 What is a Project?	53
5.1.2 Creating a New Project	55
5.2 Configuring a Project	
5.2.1 Summary Tab	59
5.2.2 Configuring the BSP	60
5.2.3 Configuring Clocks	60
5.2.4 Configuring Pins	62
5.2.5 Configuring Interrupts from the Stacks Tab	63
5.2.6 Creating Interrupts from the Interrupts Tab	63
5.2.7 Viewing Event Links	63
5.2.8 Adding Threads and Drivers	64
5.2.9 Adding and Configuring HAL Drivers	64
5.2.10 Adding Drivers to a Thread and Configuring the Drivers	65
5.2.11 Configuring Threads	67
5.3 Reviewing and Adding Components	
5.4 Debugging the Project	69
5.5 Modifying Toolchain Settings	71
5.6 Importing an Existing Project into e2 studio	71
6. Notes on development	75
6.1 Getting USB Hub to be workable with USBX	75
Revision History	76



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 and RZ/A3M.

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 to be carried out beforehand by RZ/A Initial Program Loader (hereinafter referred to as IPL). For detail on IPL, please refer to the <u>Document of IPL</u>.



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 and EK-RZ/A3M.

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

EK-RZ/A3M related information is available at EK-RZ/A3M Evaluation Kit for RZ/A3M MPU.

	Board name on GUI	Note
	screen	
RZ/A3UL Evaluation Board Kit QSPI Edition (RTK9763U02S01000BE)	RZ/A3UL Evaluation Board Kit QSPI Edition (Exec with DDR SDRAM)	If you select this, the initial program loader will transfer the entire program including the code area to DDR4.
	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In- Place)	If you select this, only the data area will be transferred to DDR4 by the initial program loader. The code area on the flash ROM is referenced during execution.
RZ/A3UL Evaluation Board Kit Octal-SPI Edition (RTK9763U02S01001BE)	RZ/A3UL Evaluation board Kit OCTAL Edition (eXecute- In-Place)	If you select this, only the data area will be transferred to OctaRAM by the initial program loader. The code area on the flash ROM is referenced during execution.
EK-RZ/A3M (RTK9EKZA3MS10001BE)	EK-RZ/A3M NOR Boot (Exec with DDR SDRAM)	If you select this, the initial program loader will transfer the entire program including the code area to Built-in DDR3.

The relationship between the board type and the board name on e^2 studio is as follows.

2.1.2.2 PC Requirements

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

- Windows 10 or Ubuntu 22.04 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. If you would like to install e2 studio and FSP at the same time, please jump to 2.2.1.

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: <u>https://www.renesas.com/e2studio</u>. 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 the e2 studio installer to launch the e2 studio installation wizard. Then, select the [Custom Install] option and click the [Next] button.

Renesas e ² studio Setup Install Type Please select the e ² studio installation type. Click here for help selecting a type and to see what features are included. Select Install Type: Image: Plant Plan	Renesas e² studio Setup	- 🗆 X
Please select the e ² studio installation type. <u>Click here</u> for help selecting a type and to see what features are included. Select Install Type: Lite Install (Recommended) Image: Installs e ² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & debugging with only important features Standard Install Image: Installs e ² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features Custom Install Image: Custom Installation of e ² studio This mode is allows you to select which features are installed	Renesas e ² studio Setup	RENESAS
Select Install Type: Lie Install (Recommended) This installs e ² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & debugging with only important features Sandard Install Paradral Installs e ² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features Puscon Install Puscon Install </td <th>nstall Type</th> <th></th>	nstall Type	
 Lite Install (Recommended) This installs e² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & debugging with only important features Standard Install This installs e² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features Custom Install Custom installation of e² studio This mode is allows you to select which features are installed 	Please select the e ² studio installation type. <u>Click here</u> for help selecting a type and to see what fe	atures are included.
 This installs e² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & debugging with only important features Standard Install This installs e² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features Custom Install Custom installation of e² studio This mode is allows you to select which features are installed 	Select Install Type:	
 This installs e² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced features Custom Install Custom installation of e² studio This mode is allows you to select which features are installed 	🖵 This installs e² studio in Lite Mode.	g with only important features
Custom Install Custom installation of e ² studio This mode is allows you to select which features are installed	(\mathbf{Q}) This installs e ² studio in Advanced Mode.	
	Custom Install Custom installation of e ² studio	
< <u>B</u> ack <u>N</u> ext > Install Cancel		
< <u>B</u> ack <u>N</u> ext > Install Cancel		
< <u>B</u> ack <u>N</u> ext > Install Cancel		
	< <u>B</u> ack <u>N</u> ext >	Install Cancel

Figure 1: e2 studio installation wizard

Note:

If you are using a multi-user environment, you may receive a prompt to confirm whether you want to install it for the current user only or for all users.

Renes	as Installer	
	Select which users to install for	
	→ All Users Install for all users on this computer Requires Administrator permissions	
	→ Current user Install for only Parts may require Administrator permissions	
	\rightarrow Cancel installation	

Figure 2: Select User for Installation



2. Welcome page

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

Note:

- 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.

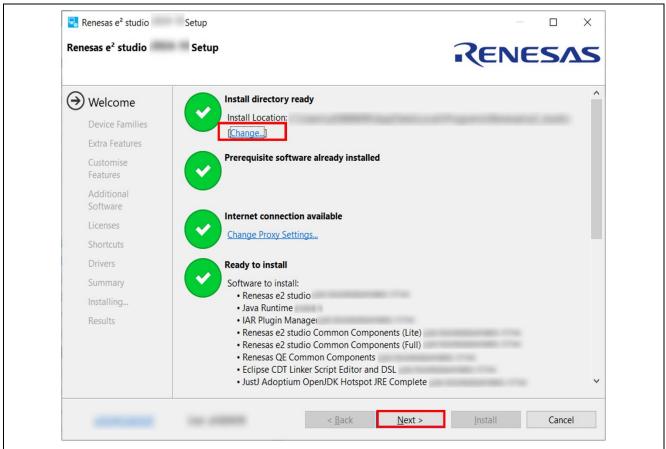


Figure 3: Installation of e2 studio – Welcome page



3. Device Families

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

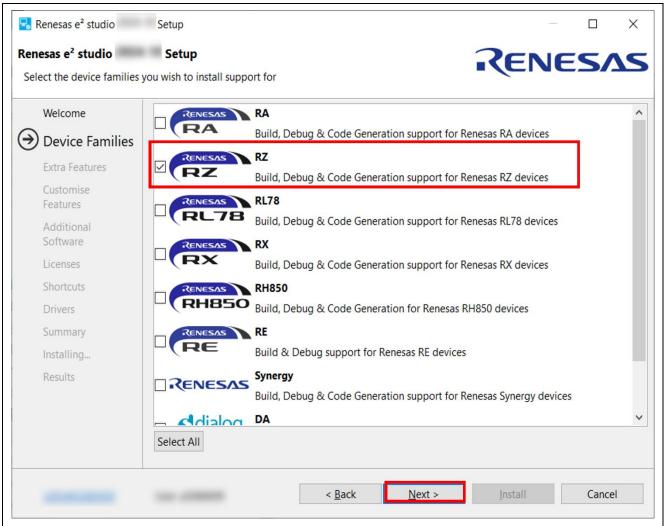


Figure 4: Installation of e2 studio – Device Families



4. Extra Features

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

Renesas e² studio	Setup		— 🗆 X
Renesas e ² studio Select the extra features y	Setur rou wish to in		RENESAS
Welcome Device Families			Japanese Language Support
→ Extra Features			Chinese (Simplified) Language Support
Customise Features			Chinese (Traditional) Language Support
Additional Software			Git Integration
Licenses Shortcuts			Git SCM Support Terminals
Drivers Summary		\bigcirc	ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
Installing Results			
Results	Select A	JI	
-			< <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel

Figure 5: Installation of e2 studio – Extra Features



5. Customize 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.

🛃 Renesas e² studio	Setup 🗌	×
Renesas e² studio	Setup RENESA	S
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing	Select the components you want to install. Renesas Ke Tamily Support to allow project generation and build or executable projects, library projects, debug Image: Components of the project sentence of the project sentenc	~
	Size of install: 938.1 MB	
	< <u>B</u> ack <u>N</u> ext > <u>I</u> nstall Cancel	

Figure 6: 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) 13.2 rel1

Select the additional softw	vare you wish to install		
		- (NESA
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	 Renesas QE Renesas AI Renesas Toolchains && Utilities Renesas Toolchains && Utilities Renesas E2 Self-Checking Program V1.02.00 Renesas E2 Lite Self-Checking Program V1.00.00 Renesas E1, E20 Self-Checking Program V1.01.00 GCC Toolchains && Utilities GCC Toolchains && Utilities GNU ARM Embedded 13.3-Rel1 GNU ARM Embedded 12.2-Rel1 GNU ARM Embedded 6 2017q2 GCC ARM A-Profile (AArch64 bare-metal) 13.2.Rel1 GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 LibGen for GNU ARM Embedded LibGen for GNU AARCH64 Embedded LibGen for GNU AARCH64 Embedded M Enesas RA FSP 	1.2024.04 1.2024.04	1000 B 1000 B

Figure 7: Installation of e2 studio – Additional Software



7. Licenses Agreement

Please read and accept the software license agreement, then click the [Next] button. Note that acceptance of the license agreement is mandatory; without it, the installation process cannot proceed.

🗟 Renesas e² studio	Setup			_		Х
Renesas e ² studio	Setup			REN	ES/	12
Welcome	Please read and accept the fo	llowing Software Agre	ements			
Device Families Extra Features Customise Features Additional Software Dicenses Shortcuts Drivers Summary Installing Results	Renesas e2 studio OpenJDK License Agreemen ARM DS-5 Toolchain Integra IAR Plugin Manager	License Terms and Con This Renesas e2 studic entity on whose behal and Renesas Electroni registered office at 3- ("Renesas"). YOU SHO CONSTITUTES A BIND The Renesas IDE Softw use by a company or co or produced for any p you intend to install th or the Renesas IDE So purpose directly or inco Otherwise, by clicking mechanism designed electronic copy of this otherwise copying or Software, you accept to	o license agreeme f you are entering cs Corporation, a 2-24, Toyosu, Koto ULD READ THIS A ING CONTRACT E vare (defined belo corporation only a rivate use or purp he Renesas IDE So ftware is expected directly, you shoul the "I accept" but to acknowledge a Agreement, or bu using all or any po-	nt ("Agreement") i j into this Agreeme Japanese company o-ku, Tokyo 135-00 AGREEMENT CARE BETWEEN CLIENT A ow) is intended for and is not designed ose. If you are an ftware on behalf of d click "No" on the ton or other butto agreement to the te y installing, accession ontion of the Renession behalf of the enti	ent ("Client") y with its 161, Japan FULLY, AS IT AND RENES/ commercia d, developed individual, c of an individ private e installer. n or erms of an ing, or ias IDE ity for which) AS. I d or ual,
	✓ I accept the terms of the So	oftware Agreements			Pr	int all
		< <u>B</u> ack	<u>N</u> ext >	<u>I</u> nstall	Cance	el

Figure 8: Installation of e2 studio – Licenses



8. Shortcuts

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

Note:

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

Renesas e ² studio	Setup										_			×
Renesas e² studio	Setup								2	E	NE	=	5/	12
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Shortcuts to important pro					in the	follo	wing	locat	tions:	đ	<mark>€ R</mark> es	store [Default
-			< <u>B</u> a	ck	N	ext >			lns	stall			Cance	9

Figure 9: Installation of e2 studio – Shortcuts



9. Summary

On the summary page, a list of components to be installed will be displayed. Please review the contents and click the [Install] button to proceed with the installation of the Renesas e2 studio IDE.

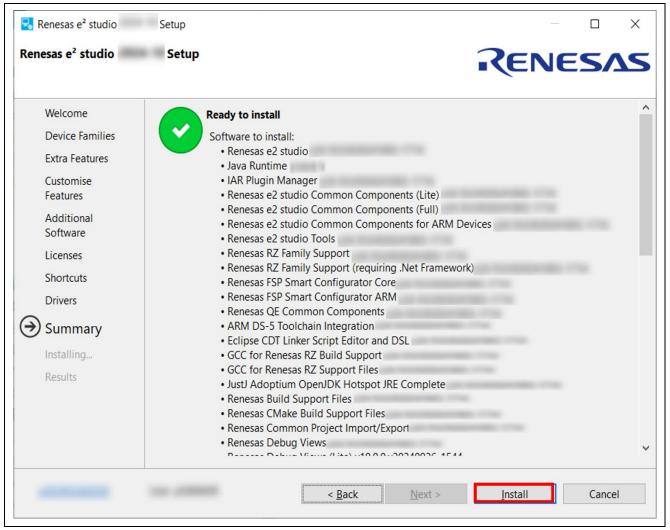


Figure 10: Installation of e2 studio – Summary

10.Installing...

The installation will proceed. Depending on selected items of the additional software, new dialog prompts may appear during the installation process. Please follow the instructions provided by the installer when this occurs.



11.Results

If the installation has been successfully completed, you should see the following information.

🗟 Renesas e² studio	Setup	— 🗆 X
Renesas e ² studio	Setup	RENESAS
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing \bigodot Results	Installation of e2 studio is complete. Please click OK to close. Launch e2 studio? View Release Notes? View What's New?	
	< <u>B</u> ack	Next > OK Cancel

Figure 11: 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 V8.10a: 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: <u>https://www.renesas.com/e2studio</u>.

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 the commands stated below:

```
$ sudo dpkg -i JLink_Linux_V810a_x86_64.deb
```

If the previous installation fails with unmet dependencies, please retry as follows:

```
$ sudo apt-get -f install
$ sudo dpkg -i JLink Linux V810a x86 64.deb
```

• e2 studio

- 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. User needs to select Install Type as shown below. In this material, it is expected that Custom Install is selected. Then, click [Next >] to continue.

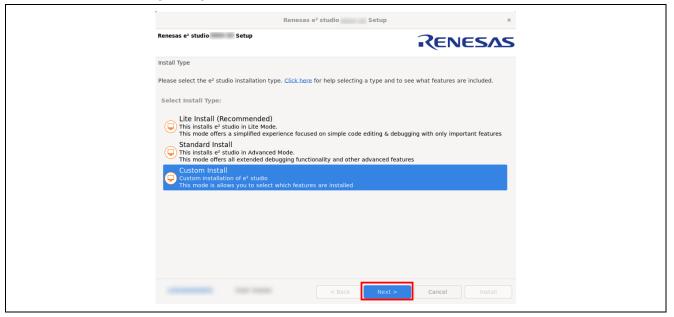


Figure 12: Selection of Install Type



3. User can change the install folder by clicking [Change...]. Click [Next] to continue.

Note:

- 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 Setu	RENESA
→ Welcome	Install directory ready
Device Families	install Eocation.
Extra Features	Change
Customise Features	Prerequisite software already installed
Additional Software	
	Internet connection available
	Change Proxy Settings
Shortcuts	
Summary	Ready to install
Installing	Software to install.
Results	Renesas e2 studio Java Runtime IAR Plugin Manager Renesas e2 studio Common Components (Lite) Renesas e2 studio Common Components (Full) Renesas QE Common Components Eclipse CDT Linker Script Editor and DSL

Figure 13: Installation of e2 studio – Welcome page

3. Device Families

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

Renesas e² studio	Setup	2	
Select the device families	you wish to install	support for	15
Welcome	RENESAS		
 Device Families 		Build, Debug & Code Generation support for Renesas RA devices	-
Extra Features	RENESAS		
Customise	~	Build, Debug & Code Generation support for Renesas RZ devices	
Features Additional	RL78		
Software	RENESAS	RX	
Licenses Shortcuts		Build, Debug & Code Generation support for Renesas RX devices	
Summary	RENESAS		
Installing		Build, Debug & Code Generation for Renesas RH850 devices	
Results		RE	
	RE	Build & Debug support for Renesas RE devices	
	RENESAS	Linux on Renesas RZ	
	Select All		
		< Back Next > Cancel Inst.	

Figure 14: 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 e ² studio Select the extra features	Setup you wish to install	RENESAS
Welcome Device Families Extra Features Customise		Japanese Language Support Chinese (Simplified) Language Support
Additional Software Licenses Shortcuts Summary Installing		Chinese (Traditional) Language Support Git Integration Git SCM Support Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
	Select All	

Figure 15: 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	Setup	RENESAS
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results	Select the components you want to install. Renesas RH850 family support Renesas DA Family Support (24.10.0.R20241003-17 Renesas DA Family Support lallow project generation. butous the sense of the sen	uild & debug 40-30-1428) 40924-1500) 1714) 1714) 1003-1714) ssk repositories

Figure 16: Installation of e2 studio – Features



6. Customize Features

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) 13.2 rel1

Renesas e² studio	Renesas e ² studio Setup	
Select the additional softw	are you wish to install	RENESAS
Welcome Device Families Extra Features Customise Features Moditional Software Licenses Shortcuts Summary Installing Results	 Renesas QE Renesas AI Renesas AI Renesas AI Navigator AI Transfer Learning Tool Plugin AI Model Conversion Tool Plugin Renesas Toolchains & & Utilities GCC Toolchains & Utilities GNU ARM Embedded 13.3-Rel1 GCC ARM A-Profile (AArch64 bare-metal) 13 GCC CARM A-Profile (AArch64 bare-metal) 10 Renesas RA FSP 	

Figure 17: 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	Setup	RENESA
Welcome	Please read and accept	the following Software Agreements
Device Families Extra Features Customise Features Additional Software	Renesas e2 studio OpenJDK License Agreem ARM DS-5 Toolchain Integ 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 REA 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 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, accessin or otherwise copying or using all or any portion of the Renesas
	I accept the terms of t	he Software Agreements Print a

Figure 18: Installation of e2 studio – Licenses



8. Shortcuts

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

Image: Provide a state of the state of
Welcome Shortcuts to important programs and files will be created in the following locations: Device Families In the application launcher Extra Features In the application launcher Additional Software Licenses Shortcuts Summary Installing

Figure 19: 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.

Welcome Device Families Extra Features Customise Features Ready to install Software to install: · Renesas e2 studio · Java Runtime · IAR Plugin Manager · Renesas e2 studio Common Components (Lite) · Renesas e2 studio Common Components (Full)	Device Families Software to install:
Software • Renesas 62 studio Common Components for AKM Devices Licenses • Renesas 82 Family Support Shortcuts • Renesas FSP Smart Configurator Core Summary • Renesas FSP Smart Configurator ARM Installing • Relipse CDT Linker Script Editor and DSL Results • GCC for Renesas RZ Sulid Sulid Support Files • Just J Adoptium OpenJDK Hotspot JRE Complete • Renesas SUlid Support Files • Results • GCC for Renesas RZ Sulid Support Files • Just J Adoptium OpenJDK Hotspot JRE Complete • Renesas SUlid Support Files • Renesas CMake Build Support Files • Renesas CMake Build Support Files v10.0.0.v20240909-0854	i Java Runtime Customise Features Additional Software Licenses Renesas e2 studio Common Components (Lite) Renesas e2 studio Common Components (Full) Additional Software Licenses Renesas e2 studio Tools Renesas R2 Family Support Shortcuts Renesas FSP Smart Configurator Core Renesas QE Common Components AdM DS-5 Toolchain Integration Installing Results GCC for Renesas RZ Suidi Support Results GCC for Renesas RZ Support Files Justj Adoptium OpenJDK Hotspot JRE Complete

Figure 20: 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 2.1.3.2 for more detailed information.

11.Results

If the installation is successfully done, you should see the following information.

Renesas e² studio	RENESAS
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results	

Figure 21: Summary Page



2.2 FSP setup

In this section, 3 ways of FSP installation are described. However, at this moment, platform installer won't be available and so, please install FSP based on either 2.2.2 or 2.2.3.

2.2.1 Installation of FSP Packs using Platform Installer

This section describes how to install FSP using Platform Installer **setup_rzafsp_v3_5_0_e2s_v2025-01.exe** showcased at <u>here</u>.

 Double-click setup_rzafsp_v3_5_0_e2s_v2025-01.exe, select either [Quick Install] or [Custom Install] and click [Next >] when the installation wizard is shown. When you choose [Quick Install], you can jump to 6. Licenses.

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 RZ/A Flexible Software Package	(FSP) with e ² studio Setup	CENESAS
Install Type		
Select Install	Туре:	
Quick Default	Install installation of e ² studio, FSP & GCC ARM Embedded	
	m Install installation of e ² studio, FSP & GCC ARM Embedded	
	< <u>B</u> ack <u>N</u> ext > :	インストール Cancel

Figure 22: FSP Platform Installation Wizard

2. Welcome page

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

Note:

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



	Install directory ready Install Location:	→ Welcome
	[Change.]	Extra Features Customise
	Prerequisite software already installed	Features
	V	Additional Software
	Internet connection available	Licenses
_	Change Proxy Settings	Shortcuts
	Change i fory settingsin	Drivers
	Ready to install	Summary
	Software to install:	Installing
	Renesas e2 studio Java Runtime	Results
	Renesas FSP Smart Configurator Core	
~	GCC for Renesas RZ Build Support Renesas RZ Common	

Figure 23: FSP Platform Installer – Welcome page

3. 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, click the [Next] button to continue.

Welcome		P	Japanese Language Support
Extra Features Customise Features			Chinese (Simplified) Language Support
Additional Software		P	Chinese (Traditional) Language Support
Licenses Shortcuts		P	Git Integration Git SCM Support
Drivers Summary		P	Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
Installing Results			
	Select	t All	

Figure 24: FSP Platform Installer – Extra Features



4. Customize Features

Essential features have already been selected. If you would like to install additional features, please check those and then click [Next >].

Welcome Extra Features Customise Features Additional Software	Select the components you want to install.	Î
Licenses Shortcuts Drivers Summary Installing	Renesas RZ Common Renesas RZ Debug Support Renesas RZ Debug Support Renesas RZ Debug Support Files Renesas RZ Debug Support Files Renesas RZ Debug Support Files Renesas e2 studio Common Components Common components for Renesas e2 studio	v
Results	Select All Optional Deselect All Optional	Size of install: 763.1 MB

Figure 25: FSP Platform Installer – Features



5. Additional Software

All the software is selected by default. click [Next >].

Welcome Device Families Extra Features Customise Features Additional Software Drivers Summary Installing Results	 ✓ Renesas RZ FSP ✓ Renesas FSP for RZ/A ✓ Renesas RZ Toolchains ✓ GNU ARM Embedded 12.2-Re11 ✓ GNU ARM Embedded 13.3-Re11 ✓ GCC ARM A-Profile (AArch64 bare-metal > □ Renesas QE 	12.2-Rel1 13.3-Rel1 13.2-Rel1 13.2-Rel1	
			0 Bのダウンロードが必要です。

Figure 26: FSP Platform Installer – Additional Software

6. Licenses

Please read and accept Software License Agreements to be listed and click [Next >].

Welcome	Please read and accept th	ne following Software Agreements
Extra Features Customise	Renesas e2 studio	License Terms and Conditions for RENESAS e2 studio This Renesas e2 studio license agreement ("Agreement") is between the
Features Additional Software	1	entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-2-4, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"), YOU Client Dead Turk Conference on C
Licenses Shortcuts		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
Drivers Summary		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
Installing Results		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
	accept the terms of t	

Figure 27: FSP Platform Installer – Licenses

7. 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 it to distinguish it from the other e2 studio(s).



Welcome Extra Features Customise	Shortcuts to important pro	eated in the following local	ions:
Features Additional Software Licenses Shortcuts Drivers Summary Installing_ Results			Sestore Defau

Figure 28: FSP Platform Installer – Shortcuts

8. Summary

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

Welcome	Ready to install	^
Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Shortcuts Drivers Summary Installing_ Results	Software to install: • Renesse e2 studio • Java Runtime • Renessa FSP Smart Configurator Core • GCC for Renesas RZ Bulld Support • Renesas RZ Debug Support • JustJ Adoptium OpenJDK Hotspot JRE Complete • openftx.media.feature • openftx.media.feature • openftx.swing.feature • openftx.web.feature • openftx.web.feature • openftx.web.feature • dditional software to install: • Renesas FSP for RZ/A • Renesas FSP for RZ/A	

Figure 29: FSP Platform Installer – 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.



Welcome Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Installation of Renesas RZ/A Flexible Software Package (FSP)v3.3.0 with e ² studio 2024-10 is complete. Please click OK to close. Uaunch e2 studio? View Release Notes? View Release Notes? View What's New? View Renesas FSP User Manual? Useful Links:

Figure 30: Installation Results of FSP Platform Installer

2.2.2 Installation of FSP Packs using Package Installer

Package Installer **RZA_FSP_Packs_v3.5.0.exe** is showcased at <u>here</u>. Please note that it's for Windows Host PC only.

Here is the procedure:

- 1. Exit e2 studio.
- 2. Invoke RZA_FSP_Packs_v3.5.0.exe.
- 3. Click [Next >] to start the installation.

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/rza/<version: The source for FSP can be found at https://github.com/renesas/rza-fsp</version: </install_directory>	
Next > Cancel	

Figure 31: FSP Package Installer

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



RENESAS	RZA FSP Please review the license terms before installing Renesas R FSP	RZA
https://qithub.com/renesa	is/rza-fsp/blob/master/LICENSE.md	-
If you accept all terms of th	he agreement, click I Agree.	
	< Back I Agree Can	ncel

Figure 32: FSP License Term

5. Specify e2 studio installation folder (e.g., C:\Renesas\e2studio) and click [Install].

RENESAS	Choose Install Location Choose the folder in which to install Renesas RZA FSP	
	point to the root of the e2 studio installation (e.g. Please make sure e2 studio is closed before installation.	
Browse to folder where C:¥Renesas¥e2_studie		
Space required: 47.8 MB Space available: 10.7 GB		
	< <u>B</u> ack Install Cancel	

Figure 33: FSP Installation

6. Click [Finish] to complete the installation.



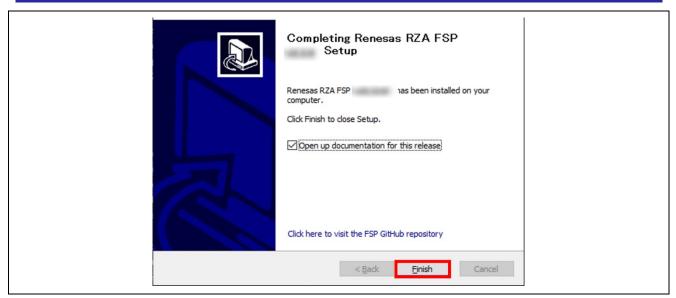


Figure 34: Completion of FSP Installation

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.



2.2.3 Installation of FSP Packs using Package Zip file

No Package installer is available for Linux Host PC and therefore, you need to install FSP Packs with **RZA_FSP_Packs_v3.5.0.zip**. This section describes how to do install it. Please note that the same installation procedure is valid for Windows Host PC.

- 1. Download RZA_FSP_Packs_v3.5.0.zip from here.
- 2. Extract the zip file to e^2 studio installation directory.
 - If the FSP Packs are successfully extracted, **rz_fsp/rza/packs** directory is placed at the location below:
 - <e² studio installation directory>/internal/projectgen

 e2_studio > internal > projectgen > rz_fsp > rza > packs Amazon.FreeRTOS-Kernelpack Amazon.FreeRTOS-Plus-TCPpack FreeRTOS.FreeRTOS_plus_FATpack Microsoft.FileXpack Microsoft.GUIXpack Microsoft.NetXDuopack Microsoft.ThreadXpack Microsoft.USBXpack Microsoft.USBXpack Renesas.RZApack Renesas.RZA_board_custompack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_freertos_blinkypack Renesas.RZA_freertos_blinkypack Renesas.RZA_threadx_blinkypack 	
Amazon.FreeRTOS-Plus-TCP. ,pack FreeRTOS.FreeRTOS_plus_FAT. ,pack Microsoft.FileX. ,pack Microsoft.GUIX. ,pack Microsoft.NetXDuo. ,pack Microsoft.ThreadX. ,pack Microsoft.USBX. ,pack Renesas.RZA. ,pack Renesas.RZA. ,pack Renesas.RZA_board_custom. ,pack Renesas.RZA_board_custom_octal_xip. ,pack Renesas.RZA_board_rza3ul_smarc_octal_xip. ,pack Renesas.RZA_board_rza3ul_smarc_ospi_xip. ,pack Renesas.RZA_board_rza3ul_smarc_ospi_xip. ,pack Renesas.RZA_board_rza3ul_smarc_ospi_xip. ,pack Renesas.RZA_config. ,pack Renesas.RZA_freertos_blinky. ,pack	ez_studio > Internal > projectgen > rz_isp > rza > packs
FreeRTOS.FreeRTOS_plus_FAT. .pack Microsoft.FileX. .pack Microsoft.GUIX. .pack Microsoft.NetXDuo. .pack Microsoft.ThreadX. .pack Microsoft.USBX. .pack Microsoft.USBX. .pack Renesas.RZA. .pack Renesas.RZA_baremetal_blinky. .pack Renesas.RZA_board_custom. .pack Renesas.RZA_board_rcastur_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack	Amazon.FreeRTOS-Kernelpack
 Microsoft.FileXpack Microsoft.GUIXpack Microsoft.NetXDuopack Microsoft.ThreadXpack Microsoft.USBXpack Renesas.RZApack Renesas.RZA_board_custompack Renesas.RZA_board_custom_octal_xippack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Amazon.FreeRTOS-Plus-TCPpack
 Microsoft.GUIXpack Microsoft.NetXDuopack Microsoft.ThreadXpack Microsoft.USBXpack Renesas.RZApack Renesas.RZA_board_custompack Renesas.RZA_board_custom_octal_xippack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_freertos_blinkypack 	FreeRTOS.FreeRTOS_plus_FAT
 Microsoft.NetXDuopack Microsoft.ThreadXpack Microsoft.USBXpack Renesas.RZApack Renesas.RZA_baremetal_blinkypack Renesas.RZA_board_custompack Renesas.RZA_board_custom_octal_xippack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Microsoft.FileXpack
 Microsoft.ThreadXpack Microsoft.USBXpack Renesas.RZApack Renesas.RZA_baremetal_blinkypack Renesas.RZA_board_custompack Renesas.RZA_board_custom_octal_xippack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Microsoft.GUIX
 Microsoft.USBX. .pack Renesas.RZA. .pack Renesas.RZA_baremetal_blinky. .pack Renesas.RZA_board_custom. .pack Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Microsoft.NetXDuopack
 Renesas.RZApack Renesas.RZA_baremetal_blinkypack Renesas.RZA_board_custompack Renesas.RZA_board_custom_octal_xippack Renesas.RZA_board_rza3ul_smarc_octal_xippack Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspi_xippack Renesas.RZA_board_rza3ul_smarc_qspi_xippack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Microsoft.ThreadX
 Renesas.RZA_board_custom. .pack Renesas.RZA_board_custom_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi_xip. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Microsoft.USBX
 Renesas.RZA_board_custom. .pack Renesas.RZA_board_custom_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi_xip. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Renesas.RZApack
 Renesas.RZA_board_custom_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi_xip. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Renesas.RZA_baremetal_blinky
 Renesas.RZA_board_rza3ul_smarc_octal_xip. .pack Renesas.RZA_board_rza3ul_smarc_qspi. .pack Renesas.RZA_board_rza3ul_smarc_qspi_xip. .pack Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Renesas.RZA_board_custompack
 Renesas.RZA_board_rza3ul_smarc_qspipack Renesas.RZA_board_rza3ul_smarc_qspi_xippack Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Renesas.RZA_board_custom_octal_xip.
 Renesas.RZA_board_rza3ul_smarc_qspi_xip. Renesas.RZA_config. .pack Renesas.RZA_freertos_blinky. .pack Renesas.RZA_mcu_rza3ul. .pack 	Renesas.RZA_board_rza3ul_smarc_octal_xip.
 Renesas.RZA_configpack Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack 	Renesas.RZA_board_rza3ul_smarc_qspi.
Renesas.RZA_freertos_blinkypack Renesas.RZA_mcu_rza3ulpack	Renesas.RZA_board_rza3ul_smarc_qspi_xip.
Renesas.RZA_mcu_rza3ulpack	Renesas.RZA_configpack
	Renesas.RZA_freertos_blinkypack
Renesas.RZA_threadx_blinkypack	Renesas.RZA_mcu_rza3ulpack
	Renesas.RZA_threadx_blinkypack
SEGGER.JLinkpack	SEGGER.JLinkpack

Figure 35: FSP Packs in e2 studio installation directory

3. At the 1st invocation of e² studio after you carry out the above procedure, FSP Packs should be installed automatically.



4. You can check if the installation is successfully done by the following procedure: Click **Help > CMSIS Packs Management > Renesas RZ/A**.

		vorkspace - e² studio			_	. 🗆 🗙		
File Edit Source Refactor Navi	igate Search Project Re	enesas Views Run Rene	sas Al Window	Help				
	* + 9 +			🚳 Welcome				
Project Explorer 🗙 📮 🗆				Help Conter	nts			
				😵 Search				
E 🕏 🍸 🕴				Show Context H				
There are no projects in your				Show Active Ke			Shift+Ctrl+L	
workspace. To add a project:				Cheat Sheets				
Create a new Makefile project				Renesas Help			•	
in a directory containing				💋 Toolchain H			+	
existing code				CMSIS Packs M			•	Renesas RZ/A
Create a new C or C++				💋 Add Renesa				Renesas RZ/G
Create a project				Eclipse Use	-		•	Renesas RZ/N
Import projects				😚 Perform Set				Renesas RZ/T
imporc projects				🍫 Check for U				Renesas RZ/V
				🚯 Install New				
				🔹 Eclipse Mar				
				a IAR Embedo		plugin manage	er	
				🔯 About e² stu	ıdio			
	🖹 Problems 🗙 📮 Conse	ole 🔲 Properties 虆 Smar	Browser 🐺 Sm	art Manual	7 8			
	0 items							
	Description		Resource	Path	Location	Туре		
						_		
						_		
0 items selected								
0 items selected					· 🦡 📖 🎓	20		

If FSP is successfully installed, 3.5.0 should be listed under FSP as shown below:

CMSIS Packs Management - Renesas RZ/A		- 0	×
Packs location:			
Show in System Explorer			
Available Packs			(i) 🕇
Category V S FSP N 3.5.0 S FSP Companions Search Search	Version	Status	ose



3. Set Up a Target Board

Below is an example of a typical system configuration.

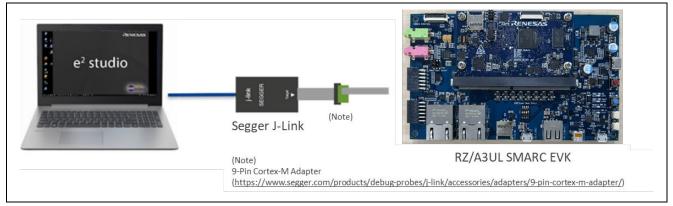


Figure 36: System Configuration Example – SMARC EVK

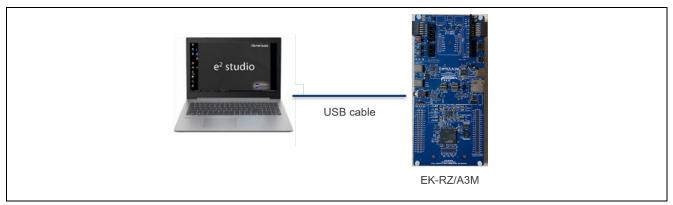


Figure 37: System Configuration Example – EK-RZ/A3M

3.1 Supported Debugger

• SEGGER J-Link

For details on SEGGER J-Link, please see <u>J-Link Debug Probes by SEGGER – the Embedded Experts</u>. In the case of EK-RZ/A3M, the on-board debug functionality is provided using Renesas RA4M2 Debug MPU and SEGGER J-Link® firmware as J-Link OB. For detail, please refer to 3.2.2.



3.2 Board Setup

3.2.1 Boot Mode

For the SMARK EVK, set up the SW11 as follows to configure Boot Mode 3 (QSPI or OCTA Boot (1.8V) Mode).

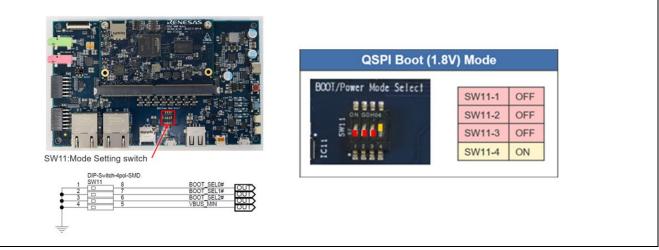


Figure 38: Boot MODE – SMARC EVK

For the EK-RZ/A3M, set up the SW5 as follows to configure Boot Mode 4 (Boot from 3.3V QSPI NOR flash).

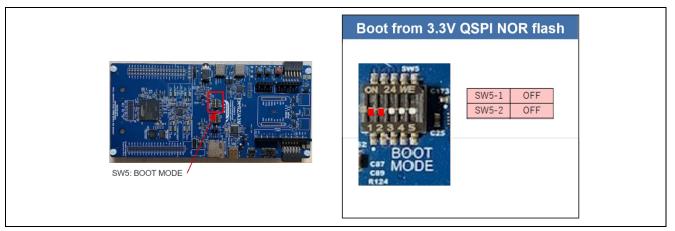


Figure 39: Boot MODE – EK-RZ/A3M



3.2.2 JTAG connection

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

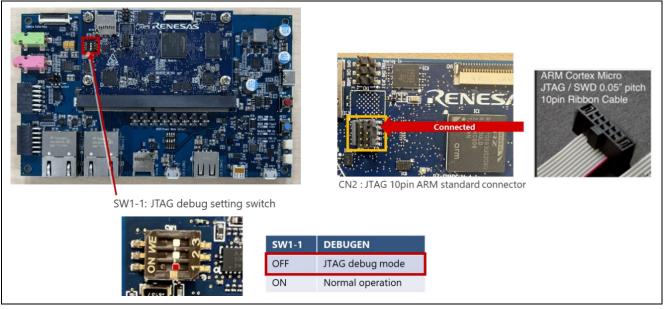


Figure 40: JTAG connection – SMARC EVK

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/

In the case of EK-RZ/A3M, the on-board debug functionality is supported. User can debug by connecting DEBUG1 connector on EK-RZ/A3M with PC. When use on-board debug functionality, you must set the J9 as follows.

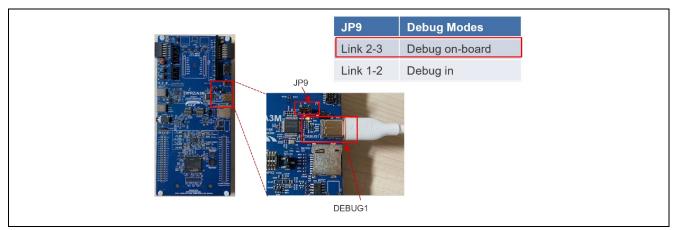


Figure 41: JTAG connection – EK-RZ/A3M



3.2.3 Debug Serial (console output)

For SMARC EVK, debug serial uses CN14. The baud rate is 115200bps.

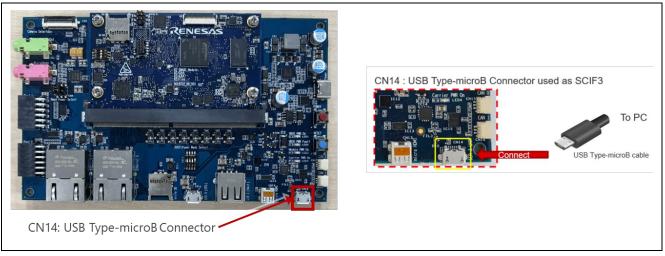


Figure 42: Debug Serial (console output) – SMARC EVK

For EK-RZ/A3M, on-board debug functionality supports debug serial output. You must configure the J-Link OB to enable the Virtual COM-Port after connection the board to your PC. And the baud rate is 115200bps.

	Connected via USB:								Auto refresh (On)
filmt	2	Product	Nickname	SN	USB Identification	Host Firmw	Configuration	×	
Real	0	J-Link-OB-RA4M2-Full V1.00		-			General		
		o Link ob tornic fail (199	Configure				Product	J-Link-OB-RA4M2-F	
and the second second			Update firmware	>			SN		
			Replace firmwar				Nickname		
			Copy info				Max. SWO speed [kHz	1 0000	
5=							Max. SWO speed [km2	3 [0000	
Trace							Target reference volt	age (VTref)	Select none
J-11-man	Connected via TCP/	P:					Automatic		
and the second second	2	Product	Nickname	SN	IP Address	MAC A	The connected probe support setting a fixe	e/ programmer does not d VTref.	rmware F
							Virtual COM-Port		
						4	Enable	O Disable	
							J-Link COM Port	PC COM Port	
Flasher							VCOMO	COM21	
Eller an	<						100000	000021	>
									Select none
E	Los:						USB Driver (Windows)		
	SEGGER J-Link Co	onfiguration						WinUSB	
ALLENS	Logging started	0						K Cancel	
							Update firmware	of selected probes and program	mers Close
Ready							1 probe/ pros	ranmar found	
Ready							1 probe/ prog	rammer tound	

Figure 43: Debug Serial (console output) – EK-RZ/A3M



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)

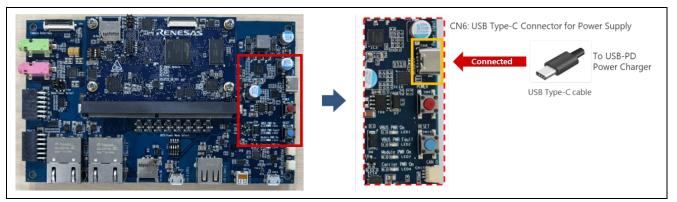


Figure 44: Power Supply

For power supply, please follow the following procedure:

SMARC EVK

- 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.
- 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. Also, the power button should be pressed and held for 2 seconds for turning off the power.
- 3. If the power supply is successful, LED4 (Carrier PWR On) should light up.

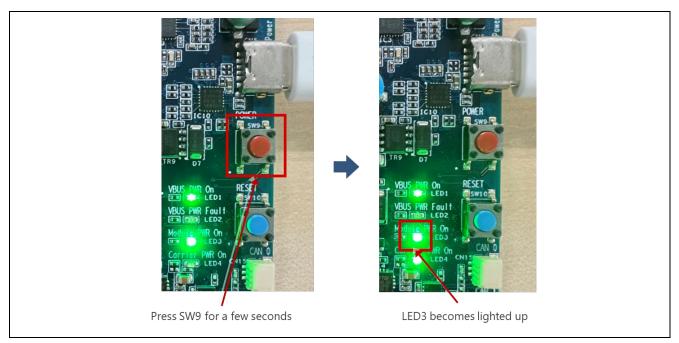


Figure 45: LED Status after Turning on EVK – SMARC EVK



EK-RZ/A3M

 Connect DEBUG1(J10) connector of EK-RZ/A3M with PC. When powered, the white LED near the center of the board (the "dash" in the EK-RZ/A3M name) will illuminate.

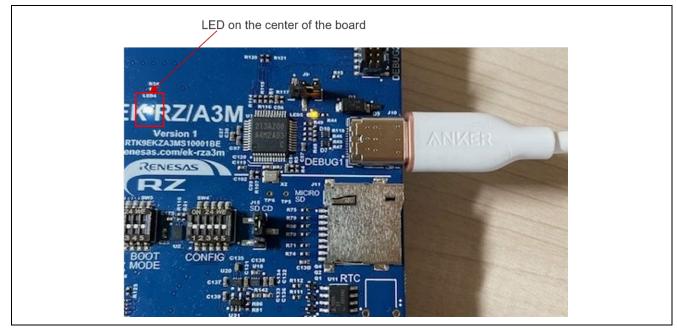


Figure 46: LED Status after Turning on – EK-RZ/A3M



3.2.5 How to check if 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 in the figure below:

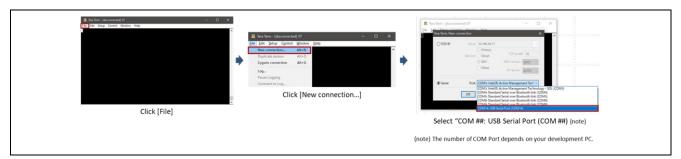


Figure 47: Establishment of connection between Target board and Development PC

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 Flash memory by default.



Figure 48: Message on your terminal at the 1st power-up

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 set up correctly and builds a working executable image for your chip.
- The debugger has been 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) interruption is intentionally fired and GPIO is properly controlled.

Note:

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



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

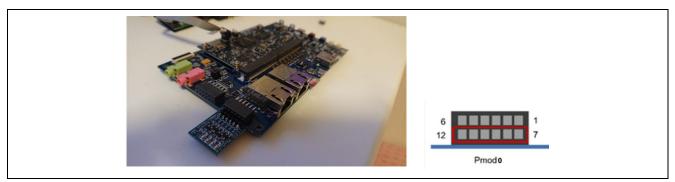


Figure 49: Connection Pmod LED module (410-076)

This module is not included on the SRMAC EVK board and so, please prepare it beforehand. In the case of EK-RZ/A3M, there are User LEDs on the board.

4.3 Creating 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].

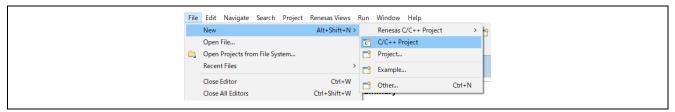


Figure 50: New C/C++ Project

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

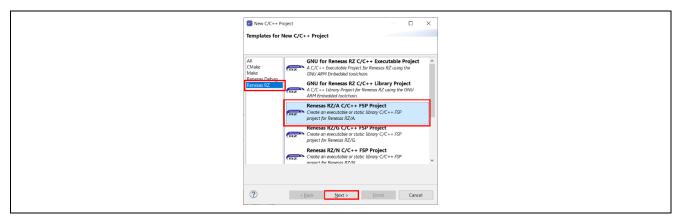


Figure 51: Renesas RZ/A C/C++ FSP Project

- 3. Assign a name for 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/A C/C++ FSP Project	- 🗆 X
Renesas RZ/A C/C++ FSP Project	
Project Name and Location	
Project name	
Blinky	
Use default location	
Location: C:\workspace\Blinky	Browse
You can download more Renesas packs here	
0	< <u>B</u> ack <u>N</u> ext > Einish Cancel

Figure 52 : e2 studio Project Configuration window (part 1)

5. Select the board support package corresponding to the package you would like to use, GCC ARM A-Profile (AArch64 bare-metal) and 13.2.1 from the Device Selection drop-down list, Toolchains and Version Selection drop-down list respectively. Then, Click [Next].

	Z/A C/C++ FSP Project /A C/C++ FSP Project			×
Device and T	Tools Selection			\square
Device Select	tion			
FSP Version: Board: Device:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXect. >			
Core:	R9A07G063U02GBG	Device Details		
		TrustZone Pins Processor	No 361 Cortex-A55	
Toolchains		Debugger		
GCC ARM A	-Profile (AArch64 bare-metal)	J-Link ARM		~
13.2.1.202310	009 V Manage Toolchains	< <u>B</u> ack	Next > Einish	Cancel

Figure 53 : e2 studio Project Configuration window (part 2)



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

	tenesas RZ/A C/C++ FSP Project Build Artifact and RTOS Selection
RTOS Selection No RTOS	Build Artifact Selection

Figure 54 : e2 studio Project Configuration window (part 3)

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

2 Renesas RZ/A C/C++ FSP Project	×
Renesas RZ/A C/C++ FSP Project	-
Project Template Selection	4
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.RZA] pack] Bare Metal - Minimal Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RZA]pack]	
Code Generation Settings Use Renesas Code Formatter	
Cance Seck Next > Einish Cance	:el

Figure 55 : 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.



Summary			Generate Project Content
Project Summar	У		
Board: Device: Core: Toolchain: Toolchain Version: FSP Version: Project Type: Location:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place) R9A070063U02GBG Core 0 GCC for Renesas RZ (A-Profile AArch64 bare-metal) 13.2.1.20021009 Flat C/workspace/Blinky 🛃	RENE	SAS
RZA Configuration Board Support Pa I/O Port Memory Managen General Timer Board support pac Board support pac Board support pac	that blinks an LED. No RTOS included. Template File ckage Common Files nent Unit ckage for R9A07G063U02GBG		

Figure 56 : 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 5.2.9.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 calling "R_BSP_PinWrite((bsp_io_port_pin_t) pin, pin_level)" for writing to the GPIO pin 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.

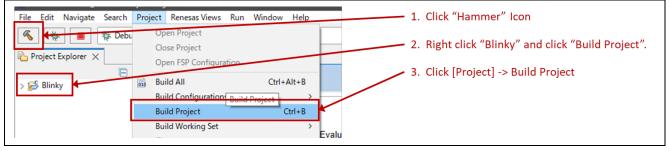


Figure 57 : e2 studio Project Explorer window

Once the build is completed, a message shown below 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-sizeformat=berkeley "Blinky.elf"
text data bss dec hex filename
11924 6736 2164980 2183640 2151d8 Blinky.elf

Figure 58 : e2 studio Project Build console



4.5 Debug the Blinky Project

4.5.1 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.

4.5.2 Debug steps

To debug the Blinky application, follow these steps:

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

Renesas Debug Tools > Run Ctrl+F11 Debug F11 Run History > Run As > Run Configurations Debug History	Run Ctrl+F11 Debug F11 Run History > Run As > Run Configurations	Run Ctrl+F11 1/2 Debug F11 Run History > Run As > Run Configurations > Debug History >	Run	un Window Help	
Image: bebug F11 Run History > Run As > Run Configurations	Image: bebug F11 Run History > Run As > Run Configurations > Debug History >	Image: body of the second s		Renesas Debug Tools	>
Run History > Run As > Run Configurations	Run History > Run As > Run Configurations > Debug History >	Run History > Run As > Run Configurations > Debug History > X Debug As	Q	Run Ctrl+	I+F11
Run As > Run Configurations	Run As > Run Configurations > Debug History >	Run As > Run Configurations > Debug History > * Debug As	杨	6 Debug	F11
Run Configurations	Run Configurations Debug History	Run Configurations Debug History > Configurations >	1	Run History	>
	Debug History >	Debug History >	-		>
Debug History >		The Debug As >	1	Run Configurations	
	称 Debug As >		1	Debug History	>

Figure 59 : e2 studio Debug icon

or by selecting the drop-down menu next to the bug icon and selecting [Debug Configurations...].

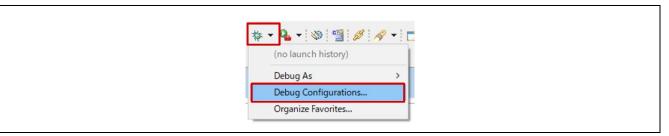


Figure 60 : e2 studio Debugger Configurations selection option



 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** for your **Blinky** project should be displayed.

Image: Search Project. Image: Startup Image: Startup </th <th>Create, manage, and run configu</th> <th>rations</th> <th>Ú.</th>	Create, manage, and run configu	rations	Ú.
	type filter text C //C++ Application C //C++ Remote Application EASE Script G GDB Hardware Debugging C GDB Simulator Debugging (I Java Applet Java Application Remote Java Application Remease GDB Hardware Debug_Flat	Main Startup Common Source Project: Blinky Blinky C/C++ Application: Debug/Blinky.elf Variables Search Project Build (if required) before launching Suild Configuration: Use Active O Enable auto build O Disable auto build	Browse

Figure 61 : e2 studio Debugger Configurations window with Blinky project (1)

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

Edit Configuration	\times
Edit Renesas GDB Hardware Debugging configuration Blinky Debug_Flat for Debu	g
Lourse Configuration Names Plinks Debug Elet	
Launch Configuration Name: Blinky Debug_Flat	
Main Pebugger Startup Common Source	
■ Main ^参 Debugger ► Startup [□] Common [↓] Source	

Figure 62 : e2 studio Debugger Configurations window with Blinky project (2)



4. Select the Connection Settings tab inside the Debugger tab.

Name: Blinky Debug_Flat	
Debug hardware: J-Link ARM	
GDB Settings Connection Settings	Debug Tool Settings
✓ J-Link	

Figure 63 : e2 studio Debugger Configurations window with Blinky project (3)

5. Change **Reset after download** to **Yes**.

Main 🏇 Debugger 🕨 Startup 🦆 Source	Common	
Debug hardware: J-Link ARM $$	e: R9A07G063U02GBG	
GDB Settings Connection Settings Debug Tool	Settings	
✓ J-Link		
Туре	USB	~
J-Link Serial	(Auto)	
Settings File	\${workspace_loc:/\${P	ojN
Script File		
Log File	\${workspace_loc:/\${P	ojN
Low Power Handling	No	~
V IP Connection		
Connection Method	IP via LAN	~
Host Name/IP Address[:port number]		
Identifier		
Tunnel Server		
Port Number		
Password		
✓ Interface		
Туре	SWD	~
Speed (kHz)	15000	~
V JTAG Scan Chain		
Multiple Devices	No	~
IRPre	0	
DRPre	0	
 Connection 		
Register initialization	No	~
Reset at the beginning of connection	Yes	~
Reset at the end of connection	No	~
Reset before download	No	~
Reset after download	Yes	~
ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFF	2222

Figure 64 : e2 studio Debugger Configurations window with Blinky project (4)



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

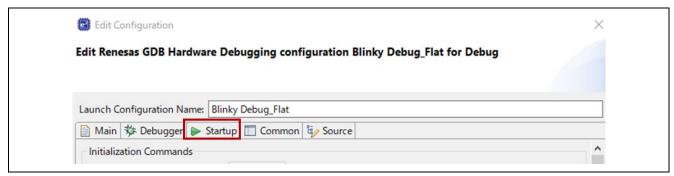


Figure 65 : e2 studio Debugger Configurations window with Blinky project (5)

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

t Renesas GDB Hardware	Debugging configur	ation Blinky D	Debug_Flat for Debug		Edit Renesas GDB Hardware	Debugging configura	ation Blinky I	Debug_Flat for Debu	9
unch Configuration Name: Bl	inky Debug_Flat				Launch Configuration Name: Bl	inky Debug_Flat			
Main 🕸 Debugger 🐌 Star	rtup 🔲 Common 🦉	Source			Main 🕸 Debugger 🕨 Star	tup 🔲 Common 🧤	Source		
Reset and Delay (seconds):	3				Reset and Delay (seconds):	3			
Halt				^ ~	Halt				
Halt				Ĵ	Load image and symbols				
pad image and symbols	Load type	Offset (hex)	On connect	Add	Load image and symbols Filename	Load type	Offset (hex)	On connect	Add
pad image and symbols		Offset (hex)	On connect Yes		Load image and symbols Filename	Symbols only 🗸	Offset (hex)	On connect Yes	
		Offset (hex)		Add Edit Remove	Load image and symbols Filename	Symbols only Image and Symbols Image only	Offset (hex)		Add Edit Remove
pad image and symbols		Offset (hex)		Edit	Load image and symbols Filename	Symbols only v Image and Symbols	Offset (hex)		Edit

Figure 66 : e2 studio Debugger Configurations window with Blinky project (6)



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.
 * If you have selected "RZ/A3UL Evaluation Board Kit OCTAL Edition (eXecute-In-Place)" in board selection, use "rza3ul_smarc_octal_ipl.srec" instead.

Launch Configuration Name: B	linky Debug_Flat]	
📄 Main 🕸 Debugger 🕨 Sta	artup 🛄 Common 💺	Source				
Initialization Commands						
Reset and Delay (seconds) Halt	: 3					e Add download module
Hait				^		Specify download module name:
				¥		Variables Search Project Workspace File System.
Load image and symbols			1.25			OK Cancel
Filename Program Binary [Blinky	Load type	Offset (hex)	On connect Yes	Add		OK Cancel
Program Binary (Blinky	, symbols only	•	Tes	Edit		
				Remove		
				Move up		Add download module
				Move down		Select a workspace resource
						✓ Selinky > ⊗ .settings
						> 😕 Debug
	2 A	dd download mod	tule	×		✓ ➢ ipl ing rza3ul_smarc_qspi_ipl.elf
	Specif	fy download mod	ule name:			rza3ul_smarc_qspi_ipl.srec
			ky¥ipl¥rza3ul_smarc_qs	:pi_ipl.srec}		> 🥭 rza_cfg
	Var	iables Sear	rch Project Work	space File System		> > > rza_gen
						> 🧽 src i .api_xml
			(OK Cancel		X .cproject
						.project .secure_azone
						.secure_xml

Figure 67 : e2 studio Debugger Configurations window with Blinky project (7)



9. 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: B	llinky Debug_Flat								
Main 🕸 Debugger 🕨 Sta	artup 🛄 Common 🍹	Source							
Initialization Commands									
Reset and Delay (seconds)	: 3					Add downlos	ad markets		~
Halt									^
				n		Specify downloa	d module name:		
					L 🔺 .				
				~		Variables	Search Project	Workspace	File System
Load image and symbols					· ·	Partourcaus	osoren rojetta.		the system
Filename	Load type	Officet (here)	On connect					OK	Cancel
Program Binary [Blinky		Offset (nex)	Ves Ves	Add				() MARK	Correct
rza3ul_smarc_qspi_ipl.s		0	Yes	Edit					
	,,			Remove					
				and the second sec		Add downlo			ο×
				Move up		Add downlo	ad module		U X
-				Move down		Select a worksp	ace resource		
						🗸 😂 Blinky			^
						Concetti			
	e Ad	d download mod	lule	>	č.	✓ ⊘ Debu > ⊘ 12			
) 🇁 rz	a_gen		
		download mod				> 🗁 💷			
	\${work	kspace_loc:#Blin	ky¥Debug¥Blinky.si	ec}	4	8 BI	inky.bin inky.elf		
	Varia	ables Sea	ch Project W	orkspace File System			inky.elf.in		
	hannakita				1		inky.hex		
				OK Cancel			inky.map		
							inky.srec		
							akefile		
							akefile.init emory_regions.ld		
						08 11	enory_regions id		Ŷ
								5	
						?		ОК	Cancel

Figure 68: e2 studio Debugger Configurations window with Blinky project (8)

10. Add the "set breakpoint enable-memread 0" to Initialization Commands.

/
^
\sim

Figure 69: e2 studio Debugger Configurations window with Blinky project (9)



11. Check the **Set breakpoint at:** check box, then, click **Debug** button.

					1	
Name: Blinky Debug_Flat						
📄 Main 🏇 Debugger 🕟 Star	rtup 🔲 Common 🍹	Source				
Initialization Commands						
Reset and Delay (seconds):	3					
Halt						
set breakpoint enable-memrea	ad 0				^	
					~	
Load image and symbols						
Filename	Load type	Offset (hex)	On connect			
Program Binary [Blinky	Symbols only		Yes		Add	
✓ rza3ul_smarc_qspi_ipl.sr			Yes		Edit	
Blinky.srec [C:\work\a3u	Image and Symbols	0	Yes	F	Remove	
					Move up	
					Move down	
Runtime Options						
Set program counter at (hex)	a.					
Set breakpoint at:	main	-				
Resume	main					
Run Commands						
					~	
					~	
				Revert	Apply	
				Debug	Close	

Figure 70: e2 studio Debugger Configurations window with Blinky project (10)

12. Debug session is now started.

e Progress Information	— D X
Launching Blinky Debug_Flat	
Configuring GDB	
	Cancel Details >>

Figure 71: e2 studio Debugger Configurations window with Blinky project (11)

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

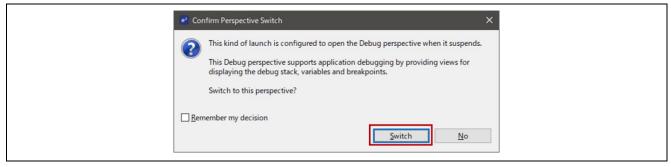


Figure 72: e2 studio Debugger Configurations window with Blinky project (12)



4.6 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.

This section describes the detail on the debug process of Blinky Project.

4.6.1 Run the Blinky Project

Click [Run] > [Resume] or click on the Play icon shown below:



Figure 73 : e2 studio Debugger Play icon

Make sure the box **Set breakpoint at:** is checked and specify **main** as its value, Program Counter should be stopped at main() function.

nitialization Commands Reset and Delay (seconds):	3		
Halt			
.oad image and symbols			
Filename	Load type	Offset (hex)	On connect
Program Binary [barem			Yes
✓ rza3ul_smarc_qspi_ipl.s		0	Yes
baremetal_blinky_new.s		0	Yes
Runtime Options		-	
Set program counter at (hex)	:		
Set breakpoint at:	main		

Figure 74 : Set breakpoint at: option



(Blinky) FSP Co	onfiguration © 0x12018 @ hal_entry.c @ main.c ×	- 8
1 2	<pre>/* generated main source file - do not edit */ #include "hal data.h"</pre>	· ^ ·
3 4 20022018		
 5 20022020 6 20022024 	hal_entry (); return 0;	
8	}	

Figure 75 : Blinky project in Debug Mode

After that LED should start to blink when clicking [Run] > [Resume] or Play icon again.



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 in the project's root directory.

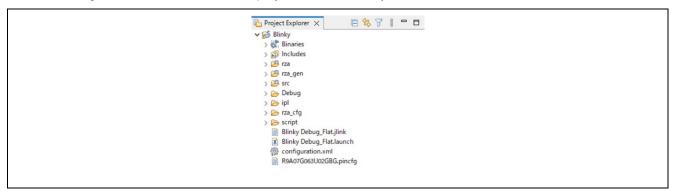


Figure 76 : e2 studio FSP Configuration Perspective

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 77 : 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.

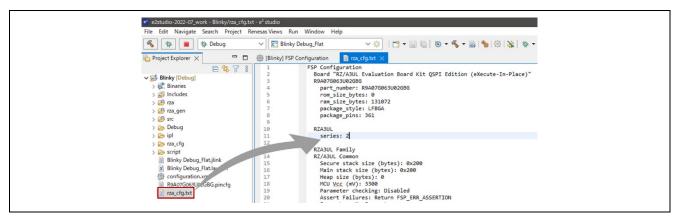


Figure 78 : RZ Project Report

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.

Summary		Generate Pro	
During			
Project Summar		2	^
Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)	RENESAS	
Device:	R9A07G063U02GBG		
Core:	Core 0		
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)		
Toolchain Version:			
FSP Version:			
Project Type:	Flat		
Location:	C:/workspace/tmp/Blinky 😓		
Selected software c	omponents		
Simple application	that blinks an LED. No RTOS included.		
RZA Configuration			
	ckage Common Files		
I/O Port	anage common mes		
Memory Managen	ent Unit		
General Timer			
	kage for R9A07G063U02GBG		
Board support pad			
	kage for RZ/A3UL (RZ/A3UL) - FSP Data		
	on Board Kit QSPI Edition (eXecute-In-Place) Board Support Files		\sim
I Supp			
Cummany DCD Cla	cks Pins Interrupts Event Links Stacks Components		

Figure 79 : 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].

File Edit Navigate Search Project Renesas Views Run Window Help New Alt+Shift+N Renesas C/C++ Project Image: Search I	Eile.	Diffs - Marchaet	Court	Devices	D \/	Desa	Mendam Hala		_
Open File C Q/C++ Project Open Projects from File System Project Recent Files Project Close Editor Ctrl+W Other Ctrl+N	File		e search	Project		_			
Open Projects from File System Recent Files Close Editor Ctrl+W Other Ctrl+N		New			Alt+Shift+N	•	Renesas C/C++ Project	>	2
Recent Files Close Editor Ctrl+W Ctrl+N		Open File				C	C/C++ Project		
Close Editor Ctrl+W Other Ctrl+N	 È,	Open Projects f	rom File S	ystem		2	Project		H
🖸 Other Ctrl+N		Recent Files					Example		
Close All Editors Ctrl+Shift+W		Close Editor			Ctrl+W		Other	Ctrl+N	E.
		Close All Editors	5		Ctrl+Shift+W		,		

Figure 80 : New RZ MPU Project

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

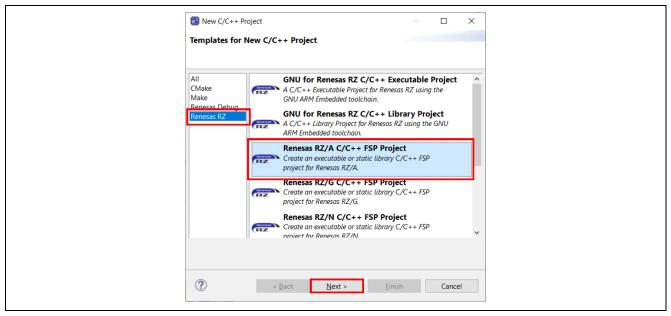


Figure 81 : New Project Templates



3. Select a project name and location.

Renesas RZ/A C/C++ FSP Project	- D >	
Renesas RZ/A C/C++ FSP Project		
Project Name and Location		
<u>P</u> roject name		
Blinky		
Use gefault location		
Location: C\workspace\Blinky	Browse	
You can download more Renesas packs here		
0	< <u>B</u> ack <u>N</u> ext > Einish Cancel]

Figure 82 : 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**.
- 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. (Please refer 2.1.2.1 for more information about the RZ MPU Evaluation Kit.)
- 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. The Toolchain selection defaults to GCC ARM A-Profile (AArch64 bare-metal).
- 5. Select the Toolchain version. This should default to the installed toolchain version.
- 6. Select the Debugger. The J-Link Arm Debugger is preselected.
- 7. Click Next.

	Z/A C/C++ FSP Project /A C/C++ FSP Project				
Device and T	fools Selection				2
Device Select	tion				
FSP Version:	~ ·	Board Description			
Board:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-1 🗸				
Device:	R9A07G063U02GBG				
Core:	Core 0 ~	Device Details TrustZone	No		
Language:	● C ○ C++	Pins Processor	361 Cortex-A55		
Toolchains		Debugger			
GCC ARM A	-Profile (AArch64 bare-metal)	J-Link ARM			~
13.2.1.202310	009 V Manage Toolchains				
?		< <u>B</u> ack	<u>N</u> ext > Ein	ish	Cancel

Figure 83 : 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 Renesas RZ/A C/C++ FSP Project	×
Build Artifact and RTOS Selection	
Build Artifact Selection Build Artifact Selection Project builds to an executable file Static Ubrary Project builds to a static library file Executable Using an RZ/A Static Library Project builds to an executable file Project uses an existing RZ/A static library project	RTOS Selection No RTOS V
0	< gack Next > Einith Cancel

Figure 84 : 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. To add threads, select **RTOS**, or **No RTOS** if an RTOS is not being used. Once you have selected the appropriate template, click **Finish**.

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.

Rennes R2A CC++ 12P Project Registration R2A CC++ 12P Project		Argent Transact XLA, SCI - 1- KP Project Argent Transact XLA, SCI - 1- KP Argent Tran
C Constant Metal - Manimal Annual Service Service and Service Service and Service and Service and Service and Service Service and Service Serv	Preetros - Binky - Static Allocation Interview pages that is called by a static and the transmission of the state is the transmission of the state is a single transmission of	or
cols Generation Settings IP to It Revises Lifet formation	Loto desantos tomaja El for Annoa Colo Farrate	Cade Servershot settings -/Use Reneals Cade Formation
C Carro Nonto Bern Carrol	(7) (8000 Hints Faid (7)	() <tax not=""> Balls Carrot</tax>

Figure 85 : 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.



Summary		Generate Project Contr
Project Summar	у	
		RENESAS
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:	13.2.1.20231009	
FSP Version:		
Project Type:	Flat	
Location:	C:/workspace/Blinky 🛃	
Selected software c	omponents	
Simple application	that blinks an LED. No RTOS included.	
RZA Configuration		
Board Support Pa	ckage Common Files	
I/O Port		
Memory Manager	nent Unit	
General Timer		
Board support pad	ckage for R9A07G063U02GBG	
Board support pad	ckage for RZA3UL	
Board support pad	ckage for RZ/A3UL (RZ/A3UL) - FSP Data	
RZ/A3UL Evaluati	on Board Kit QSPI Edition (eXecute-In-Place) Board Support Files	

Figure 86 : RZ MPU Project Editor and available editor tabs

- With the **Summary** tab, you can see all the 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 the 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

Summary			
		Generate Pro	oject Content
Project Summary			^
		RENESAS	
	Board Kit QSPI Edition (eXecute-In-Place)		
Device: R9A07G063U02GBG			
Core: Core 0			
	(A-Profile AArch64 bare-metal)		
Toolchain Version: 13.2.1.20231009			
FSP Version:			
Project Type: Flat			
Location: C:/workspace/tmp/Blin	iky 😔		
Selected software components			
Simple application that blinks an LED. No	RTOS included.		
RZA Configuration Template File			
Board Support Package Common Files			
I/O Port			
Memory Management Unit			
General Timer			
Board support package for R9A07G063U	02GBG		
Board support package for RZA3UL			
Board support package for RZ/A3UL (RZ/	A3UL) - FSP Data		
	on (eXecute-In-Place) Board Support Files		\sim
Support			
Summary BSP Clocks Pins Interrupts	Constitute Charles Constant		

Figure 87 : 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.

Problem	s Console Properties X Smart Browser Smart Manual	
RZ/A3UL	Evaluation Board Kit QSPI Edition (Exec with DDR SDRAM)	
Settings	Property	Value
securys	✓ R9A07G063U02GBG	
	part_number	R9A07G063U02GBG
	rom_size_bytes	0
	ram_size_bytes	131072
	package_style	LFBGA
	package_pins	361
	✓ RZ/A FSP Common	
	Secure stack size (bytes)	0x200
	Main stack size (bytes)	0x2000
	Heap size (bytes)	0x2000
	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 88 : 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/cbsp_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.



RZ/A3UL, RZ/A3M

nky] FSP Config						
s Configu	ration					Generate Project Co
						E. Restore D
MHz	PLL1 1GHz			> ICLK Div /1	✓ → ICLK 1GHz	
	PLL2 1600MHz		Div /2 400MHz			
				1		
	PLL2 533MHz	-		SDOCLK Sel: 533MHz	SDOCLK 533MHz	
			→ Div /2 266500kHz	SD1CLK Sel: 533MHz	✓ → SD1CLK 533MHz	
			→ Div /8 100MHz	→ POCLK Div /1	✓ → POCLK 100MHz	
			Div /10 80MHz	}	TSUCLK 80MHz	
	PLL3 1600MHz	→ Div /2 800MHz	Div /2 400MHz]	ATCLK 400MHz	
			Div /4 200MHz	12CLK Div /1	 ✓→I2CLK 200MHz 	
				> P1CLK Div /1	✓ → P1CLK 200MHz	
					MOCLK 200MHz	
				Div /2 100MHz	ZTCLK 100MHz	
				P2CLK Div /1	✓ → P2CLK 100MHz	
	PLL3 533MHz	PLL3_3 Sel: 533MHz V	→ PLL3_3 Div /2 v		→ SPIOCLK 133250kHz	
	> PLL3 400MHz	-			\$PI QSPI0_SPCLK 66625kF	
				¥ Div /2 66625kHz	→ SPI1CLK 66625kHz	
			→ Div /2 266500kHz	}	M2CLK 266500kHz	
		> PLL3_5 Sel: 400MHz >	→ PLL3_5 Div /1 v	→ Div /2 200MHz		
					Octa QSPI0_SPCLK 100MI	
	→ Div /1000			↓ Div /2 100MHz		
	PLL4 1600MHz	→ Sel: OSC Div / 1000 ~	→ Div /2 12kHz]	SOCLK 12kHz	
	> PLLS 1500MHz	→ DSI_A Div /2 v	→ Div /1 750MHz]	M3CLK 750MHz	
	> PLL5 S00MHz	→ Div /2 250MHz				
	PLL6 SOOMHz	Div /2 250MHz	> PLL6_2 Sel: 250MHz (fi ~		HPCLK 250MHz	
			→ PLL6_2 Sel: 250MHz (fi ~	1	HPCLK 250MHz	

Figure 89 : Configuration Clocks tab

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

OSC 24MHz	> PLL1 1GHz	
	PLL2 1600M Only when a	es not support this configuration change. hanging the setting in IPL, edit PLL frequency depending on IPL setting.

Figure 90 : 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:

Г

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 pins are configured, it is shown as green in the **FSP Visualization** view.

Pin Configuration			Gen	erate Project Content		🤊 🖻 🖻 Type pin	function	Pin Function 👻	Module Na	me 💌
Select Pin Configuration		Export to CSV file	Configure	Pin Driver Warnings		000		00000000		
RZA3UL-SMARC-QSPI-XIP	✓ Manage configura	tions						000000000000000000000000000000000000000		0000
Generate data: g_bsp_pin_cfg										800
Contraction of the second s	And the second second second						000	000000000	0000	000
Pin Selection $\blacksquare \boxdot \boxdot \downarrow^a_z$	Pin Configuration			Cycle Pin Group		-666	2	RENESAS	6	
Type filter text	Name	Value	Lock	Link				0 0 0 0	0_0_0_0_0 00000000000	000
> Connectivity:CAN A	Pin Group Selection	Mixed					8	0 0 0 0	0	
> ✓ Connectivity:CAN	Operation Mode	Disabled					Θ		0 0	
> ✓ Connectivity:RIIC	✓ Input/Output			< <u> </u> >			-	0 0 0 0	0	000-
> Connectivity:RSPI	SCIF4_RXD	None	10	25		-0-0	0		0 0	000
> Connectivity:SCI	SCIF4 SCK	None	18	100		· 🐵 🖸 😔	8	0 0 0 0	9 9	
S Connectivity:SCI ✓ ✓ Connectivity:SCIF	SCIF4_TXD	None		100		· 🐵 📾 🔯	0		0 0	000
✓ ✓ Connectivity:SCIP ✓ SCIF0							0	0000	9	
SCIF1								0 0 0 0	0 0	
SCIF1 SCIF2							Θ		Θ	000
SCIF2 SCIF3							0		0	-000
SCIF3 SCIF4						.000	000	000000000	0000	000
	<			>		- 🖯 🖶 🕤				000-
> V Connectivity:USB	Module name: SCIF4					- 🕞 😔 😔				600-
> Debug:JTAG/SWD	Module name: SCIF4							888888888		00000
< Interrupt:IRO Y								000000000	LONNE	0000
Pin Function Pin Number						-0000		<u>ododd</u> edd	Contractor	0000
Summary BSP Clocks Pins Interrup	ots Event Links Stacks Compon	ents			Legend					

Figure 91 : 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 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_1 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.

Select Pin Configuration Export to CSV file Configurations RZASUL-SMARC-CSPI-XIF Manage configurations Generate data: g.bp.pin_cfg Pin Selection g.bp.pin_cfg Pin Selection g.bp.pin_cfg Pin Selection g.bp.pin_cfg Very Pripherals ConcettinytRift AudioSS SciFi, C15 ConnectivityRift SciFi, C15 ConnectivityRift SciFi, SCK SciFi, SCK None SciFi, SCK None	*[Blinky] FSP Configuration × Pin Configuration			Gen	O erate Proje	ct Content
Generate data:	Select Pin Configuration		Export to CSV file	Configure l	Pin Driver	Warnings
Type filter text Name Value Lock Link Pin Group Selection Mixed Occursion Occursion <td></td> <td></td> <td><u></u></td> <td></td> <td></td> <td></td>			<u></u>			
Pin Group Selection Mixed Pin Group Selection Mixed AnalogADC Operation Mode Custom > ✓ AnalogADC Pin Group Selection Mixed > ✓ AnalogADC Pin Group Selection Mixed > ✓ Connettivity/RDH SDF1, RTS None Impl/Output > ✓ Connettivity/RDF SDF1, RTS None Impl/Output > Connettivity/SDF SDF1, RTS None Impl/Output > Connettivity/SDF SDF1, RTS None Impl/Output © Connettivity/SDF SDF1, RTS None Impl/Output © Connettivity/SDF SDF1, RTO None Impl/Output © SDF1, SCK None Impl/Output Impl/Output © Connettivity/SDF SDF1, RTO None Impl/Output © SDF1 © Connettivity/SDF Impl/Output Impl/Output Impl/Output © SDF1 © Connettivity/SDF Impl/Output Impl/Output Impl/Output Impl/Output © Connettivity/SDF © Connettivity/SDF Impl/Output Impl/Output	Pin Selection $\exists \exists \pm \Box \downarrow_z^a$	Pin Configuration		,	🕑 Cycle P	in Group
SCIF3	Peripherals AnalogADC AnalogADC AnalogADC AnalogADC Connectivity(TAN Connectivity(TAN Connectivity(SD) Connectivity(SD)	Pin Group Selection Operation Mode V Input/Output SCIF1_CTS SCIF1_RTS SCIF1_RXD SCIF1_SCK	Mixed Custom None None Vone * P13_1 None	-		>
	Pin Function Pin Number					
Pin Function Pin Number	Summary BSP Clocks 😣 Pins Inte	errupts Event Links Stacks Compone	nts			

Figure 92 : 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.

		^w HAL/Common ^g g_ioport I/O Port Driver on r_ioport ^g g_iioport I/O Port Driver on r_ioport ^g g_iioport I/O Port ^g g_timer0 Timer Driver on r_gtm ^g g_timer0 Timer Driver on r_gtm
	ОЬ	jects 🕢 New Object > 🔊 Remove
c	> Sum	mary BSP Clocks Pins Interrupts Event Links Stacks Components
Proper	rties 🗙 🖹 Problems 🁒 スマート・ブラウナ	tf 📑 🐨 🖓 🔝 🛷 🕴 🗖 🛅 🌇 Pin Conflicts 📮 Console 🔯 Debug 🗙
g_timer	0 Timer Driver on r_gtm	
	0 Timer Driver on r_gtm	Value
g_timer	0 Timer Driver on r_gtm Property > Common	Value
g_timer	0 Timer Driver on r_gtm Property → Common ✓ Module g_timer0 Timer Driver on r_g	Value
g_timer	0 Timer Driver on r_gtm Property > Common	Value
g_timer	0 Timer Driver on r_gtm Property → Common ✓ Module g_timer0 Timer Driver on r_g	Value
g_timer	0 Timer Driver on r_gtm Property Common Module g_timer0 Timer Driver on r_g General	Value
g_timer	0 Timer Driver on r_gtm Property > Common • Module g_timer0 Timer Driver on r_g > General • Interrupts	gtm
g_timer	0 Timer Driver on r_gtm Property Common Module g_timer0 Timer Driver on r_g General Interrupts Callback	gtm timer_isr

Figure 93 : 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 the New User Event button.

Interrupts	Configuration	Generate Project C	ontent
User Events		🔊 New User Event > 🔊 Ret	move
Event		ISR	
ID:78 GTM0 I	NT (GTM0 Interrupt)	gtm_int_isr	
Allocations			
Interrupt 0	Event ID:78 GTM0 INT (GTM0 Interrupt)	ISR gtm_int_isr	
			_

Figure 94 : Configuring interrupt in Interrupt Tab

5.2.7 Viewing Event Links

RZ/A FSP doesn't 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.

Stacks Configuration		Generate Project Content	
Intends New Thread Remove ✓ MAL/Common	HAL/Common Stacks g_ioport I/O Port Driver on r_ioport Gruent on r_gtm	Graphics > 🔂 125 D Input > 🔂 SPI D Networking > 🔂 UAR	Aaster Driver on r_nic_master Niver on r_ssi Niver on r_ssi
Dbjects New Object > Remo			• • • • • • • • • • • • • • • • •

Figure 95 : 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 the 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.

Stacks	Configuration			
	🖗 New Thread 🛛 🙀 Remove	New Thread Stacks	-	
Threads	Carlos Common C	Add stacks to the (above), or by pas	selected thr	
Summary	BSP Clocks Pins Interrupts Even			
Settings	Property	Value		
J-	✓ Common			
	> General > Hooks			
	> HOOKS > Stats			
	> Memory Allocation			
	> Timers			
	> Optional Functions		Enter Symbol Na	ime of your
	> Logging		thread here. Example: my_thr	read
	✓ Thread		example: my_th	
	Symbol	new thread0	Enter Name of yo	our thread here.
	Name	New Thread	Example: My Thre	
	Ca / L	4000		

Figure 96 : 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.

Thread: New Thread Remove New Thread Stacks New Stack Connectivity Description Image: Second Control Diver on r_loport <

Figure 97 : 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.

Stacks Configuration Centerate Project Content Threads New Thread Remove Image: Stacks Image: Stacks Image: Stacks Image: Stacks <t< th=""><th></th><th></th><th></th><th></th><th></th><th>-</th></t<>						-
Stacks Image: Common of Composition	Stacks	Configuration			Gener	ate Project Cont
		IAL/Common 9 giopart I/O Port Driver on r_jopart 9 freeRTOS Port (m_freertos_port) ew Thread 9 g_external_ing0 External IRQ Driver on r 2	Stacks g_external_irq0 External IRQ Driver on		Stack > 👙 Extend S	
	Proble			e 77 (ki 🖋 § 🗖 (Pin Conflicts	Console
Settings Property Value	Problem	ms みスマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_int Property	tc_irq		Pin Conflicts	Console X
Settings Property Value Common Parameter Checking Default (BSP) Module g.external.irq0 External IRQ Driver on r_intc_irq Resternal.irq0 Channel	Problem	ms	tc_irq Value ver on r_intc_irq g_ext 0 Risin NULL	e uit (BSP) ternal_irq0 g	Pin Conflicts	Console 🕇

Figure 98 : Configuring Module or Driver properties



6. When you press the Generate Project Content button for the example above, e2 studio creates 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 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:

Problem	ns 🛞 スマート・ブラウザー 🔲 Properties 🗙	🛃 🔚 🏹 🎑 🛷 🕴 🗖 🗖
New Thr	ead	
Settings	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 99 : New Thread Properties

The Properties view contains settings which are 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**.

Threads	Ā	New Thread	Remove	New Thread Stacks		New S	tack N	4 Fyte
✓ 📽 HAL/Co ⊕ g_iop ⊕ FreeR ✓ 😵 New Thr	mmon ort I/O Port Driver TOS Port (rm_freer	on r_ioport tos_port)		g_external_irq External IRQ E r_intc_irq	0 Driver on	V New 3		
	sage_buffer0 Mess							
g_new_mes	sage_buffer0 Mess	age Buffer		s				
g_new_mes		age Buffer	s Stacks Cor	s		j 📻 V	G A	000
G_new_mes Gummary BSP C Problems ♀	locks Pins Intern	age Buffer	<s con<br="" stacks="">×</s>	s		9 E 7	I., <i>1</i>	000
Gummary BSP C	locks Pins Intern » スマート・ブラウザー ge_buffer0 Me	age Buffer	<s con<br="" stacks="">×</s>			1 1	G A	000
g_new_mes g_new_mes g_new_mes g_new_messa g_new_messa Settings Prop	locks Pins Intern ッスマート・ブラウザー ge_buffer0 Me erty	age Buffer	<s con<br="" stacks="">×</s>	Value			R. 19	000
g_new_mes	locks Pins Intern » スマート・ブラウザー ge_buffer0 Me	age Buffer	<s con<br="" stacks="">×</s>	Value	message_bu		R. 19	000

Figure 100 : 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.

★ *[test] FSP Configuration ×				
Components Configuration		Gen	oerate Project Conter	nt
		Group by: Vendor V Filter: All V	Search]
Component	Version	Description	Variant	^
✓ ♣ Renesas				
🗸 💸 BSP				
V 🖗 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 101 : Components Tab

By clicking the **Generate Project Content** button, the .c and .h files for each selected component are copied 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**.

夺	• 💁 • 🖾 🛷 •	
	1 FSP_project Debug	
	Debug As	
	Debug Configurations	
	Organize Favorites	

Figure 102 : Select of Debug Configurations



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

e Debug Configurations			-	
Create, manage, and run configurations				Ť.
🖹 🖻 ն 🗎 🗶 🖻 🍸 🗸	Name: MyProject Debug_Flat			
type filter text	📄 Main 🕸 Debugger 🝺 Startup 🧤 So	urce 🔲 Common		
C/C++ Application C/C++ Remote Application	Project:			
EASE Script	MyProject			Browse
C GDB Hardware Debugging	C/C++ Application:			
 GDB OpenOCD Debugging GDB Simulator Debugging (RH850) 	Debug/MyProject.elf			
 Java Applet Java Application Launch Group Remote Java Application 	Build (if required) before launching Build Configuration: Use Active	⊻ariables	Searc <u>h</u> Project	Browse
C Renesas GDB Hardware Debugging C MyProject Debug_Flat Renesas Simulator Debugging (RX, RL78)	Enable auto build Use workspace settings	O Disable aut Configure Wo	to build orkspace Settings	
Filter matched 13 of 15 items			Re <u>v</u> ert	Apply
?			<u>D</u> ebug	Close

Figure 103 : Debug Configuration Window

3. Please set load images and set **Reset after download** setting to **Yes** as shown below:

Initialization Commands Reset and Delay (seconds): Halt	3			Main 🗱 Debugger 🍉 Startup 😨 Source 🛄 Common Debug hardware Link ARM V Target Device: R9A07G063U02GBG GDB Settings Connection Settings Debug Tool Settings
				Register initialization No
L				Reset at the beginning of connection No
Load image and symbols				Reset at the end of connection No
Filename	Load type	Offset (hex)	On connect	Reset before download No Reset after download Yes
Program Binary [My_Pr	Image only		Yes	ID Code (Bytes) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
✓ rza3ul_smarc_qspi_ipl.s	Image and Symbols	0	Yes	Hold reset during connect No
My_Project.srec [C:¥W	Image and Symbols	0	Yes	Set CPSR/Sbit) after download No

Figure 104 : Debug Setting

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 the 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.

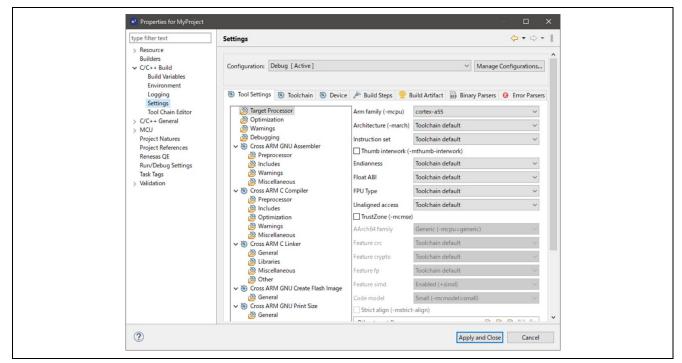


Figure 105 : e2 studio Project toolchain settings

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 controls 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_qspi_xip.ld).

5.6 Importing an Existing Project into e2 studio

- 1. Launch 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.

e ² Eclipse Launcher		×
Select a directory as workspace		
e ² studio uses the workspace directory to store its prefer	ences and development artifacts.	
Workspace: C\Users\ <user_name\e2studio\workspace< td=""><td>. ✓ <u>B</u>rowse</td><td>e</td></user_name\e2studio\workspace<>	. ✓ <u>B</u> rowse	e
Use this as the default and do not ask again		
<u>R</u> ecent Workspaces		
	Launch Cance	el

Figure 106 : 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
e ² studio uses the workspace directory to store its preferences and development artifacts.
Workspace: C:\Users\ <username>\e2studio\new_workspace > Browse</username>
Use this as the default and do not ask again
<u>R</u> ecent Workspaces
Launch Cancel

Figure 107 : Workspace Launcher dialog - Select Workspace

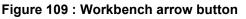
- c. Click [Launch].
- d. When the workspace is opened, you may see the Welcome Window. Click on the **Workbench** arrow button to proceed past the Welcome Screen as seen in the following figure.

RENESAS Welcome to e ² studio	Workbench
--	-----------

Figure 108 : 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:

	spWorkspace - FSPproject/confi
	File Edit Navigate Search Proje ← Menu bar ★ ■ ★ Debug ← Tool bar ↓ ▼ ☆ ▼ ☆ ▼ ☆ ▼ ☆



4. Click [Import] on the [File] menu or "Import project" on Project Explorer, as shown in the following figure:

File	e Edit Navigate Search Projec		tu	Project Explorer 💥 📃 🗖
9	New Open File Open Projects from File System Recent Files	Alt+Shift+N >	5 1	Project Explorer 🛛 🗖 🗖 🕞 🖓 🖇 There are no projects in your workspace. To add a project:
	Close Editor Close All Editors	Ctrl+W Ctrl+Shift+W		Create a new C or C++ project
	Save Save As	Ctrl+S		Create a project Create a project Import projects
	Save All Revert	Ctrl+Shift+S	or	
67	Move Rename Refresh	F2 F5		
6	Convert Line Delimiters To	>		
<u>مە</u>	Print Import	Ctrl+P		
	Export			
	Properties Switch Workspace	Alt+Enter		
	Restart Exit			

Figure 110 : 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.

Figure 111 : 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**. First, choose **Select root directory** as shown below:

🕲 Import	– 🗆 X
Import Projects	
Select a directory to search for existing Eclipse projects.	
Select root directory:	✓ Browse
O 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
	Cancel
·	

Figure 112 : Import Existing Project dialog 1 - Select root directory



- 8. Click [Browse...].
- 9. Choose the directory of the project you would like to import to specify the directory as **root directory**.
- 10. Select the project for import.
- 11.Click [Open].
- 12. Select the project to import from the list of [Projects:] as shown below:

e Import	-	- 0	×
Import Projects Select a directory to sear	rch for existing Eclipse projects.		7
O Select archive file:	Co'Nworv (1874-01) II "ImpYeZchucio-2022-07 wr. v		
Projects:	ZASUUtangAczetacja pocz organikkBlany)	Select A	JI
		Deselect A	

Figure 113 : Import Existing Project dialog 2 – Select the project to be imported

13. Click [Finish] to import the project.



6. Notes on development

6.1 Getting USB Hub to be workable with USBX

When using USB Hub with USBX, please follow the pro shown below:

- 1. Copy rza/microsoft/azure-rtos/usbx/common/core/inc/ux_user_sample.h to the directory rza/fsp/src/rm_usbx_port and rename it to ux_user.h.
- 2. Add the definitions listed below to ux_user.h.

#define	UX_MAX_CLASS_DRIVER	3
#define	UX_MAX_ED	80
#define	UX_MAX_TD	128
#define	UX MAX ISO TD	128

3. Add the following definition to rza_azurertos_sample/rza/fsp/src/rm_usbx_port/ux_port.h.

#define UX_INCLUDE_USER_DEFINE_FILE



Revision History

		Description			
Rev.	Date	Page	Summary		
3.5.0	May.08.2025	1, 5	Added RZ/A3M as supported device.		
		22, 27, 30,	Updated the description and figure based on the latest		
		46 to 50	development environment.		
		47	Removed the debug step that sets Flash Bus Type and Flash		
			Memory Type since it is not required from this version.		
		32 to 37,	Added the information of EK-RZ/A3M.		
		39			
3.4.0	Feb.21.2025	6 to 31,	Updated the description and figure based on the latest		
		37 to 39,	development environment.		
		43, 48, 52,			
		54, 56, 57			
3.3.0	Dec.20.2024	6 to 31,	Updated the description and figure based on the latest		
		37 to 39,	development environment.		
		52, 54, 56			
3.2.0	Sep.30.2024	22 to 31,	Updated the description and figure based on the latest		
		37, 39	development environment.		
3.1.0	Jul.31.2024	6 to 29,	Updated the description and figure based on the latest		
		34, 37,	development environment.		
		44 to 45,			
		50, 52, 54 to 55			
		40 to 42	Corrected the debug step when using the OCTAL Edition of		
		40 10 42	the SMARC EVK board.		
3.0.0	Apr.26.2024	6 to 29,	Updated the description and figure based on the latest		
		32 to 36,	development environment.		
		47 to 53			
2.0.2	Feb.29.2024	15, 36, 72,	Added 2.2 to install Arm GNU toolchain.		
		73	Added Step 4 and 5 to 4.5.2.		
			Removed 6.1 Unexpected termination of GDB connection.		
			Removed 6.3 to describe the way to fix the building error.		
2.0.1	Sep.30.2023	-	Updated the versions.		
2.0.0	Jun.30.2023	63, 67	Added 6.3 to describe the way to fix the building error.		
			Added 6.4 to describe the way to use USBX.		
1.21	Apr.07.2023	22, 23	Added 2.2.3 to describe the way to install FSP with the zipped Packs.		
1.20	Dec.26.2022	-	Added the instructions for installing FSP using Platform		
			Installer.		
1.10	Sep.30.2022	-	Added the info on "RZ/A3UL Evaluation Board Kit Octal-SPI		
1.00			Edition.		
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 is supplied until the power is supplied until the power reaches the level at which reseting 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 systemevaluation test for the given product.

Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

Trademarks

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

Contact information

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