

RZ/A3UL

Getting Started with RZ/A Flexible Software Package

Introduction

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

Target Device

RZ/A3UL



Contents

1. Introduction	4
1.1 Overview	
1.2 Introduction to FSP	
1.2.1 Purpose	4
1.2.2 e2 studio IDE	4
1.3 Limitations	4
1.3.1 Hardware Initial Setup	
2. Starting Development Introduction	
2.1 e2 studio setup	
2.1.1 What is e2 studio?	
2.1.2 e2 studio Prerequisites	
2.1.3 e2 studio installation for Windows PC	
2.1.4 e2 studio installation for Linux PC	
2.2 Arm GNU Toolchain setup	
2.2.1 Arm GNU Toolchain installation for Windows PC	
2.2.2 Arm GNU Toolchain installation for Linux PC	
2.2.3 Arm GNU Toolchain registration to e2 studio	
2.3 FSP setup	
2.3.1 Installation of FSP Packs using Platform Installer	
2.3.2 Installation of FSP Packs using Package Installer	
2.3.3 Installation of FSP Packs using Package Zip file	
3. Set up an SMARC EVK	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	33
4. Tutorial: Your First RZ MPU Project - Blinky	
4.1 Tutorial Blinky	
4.2 What Does Blinky Do?	
4.3 Create a New Project for Blinky	
4.3.1 Details about the Blinky Configuration	
4.3.1 Details about the Blinky Configuration 4.3.2 Configuring the Blinky Clocks	
4.3.2 Configuring the Blinky Clocks	
4.3.4 Configuring the Parameters for Blinky Components	
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	
4.5.2 Debug steps	
4.6 Details about the Debug Proce	ss
4.6.1 Run the Blinky Project	
5. FSP Application Launch with	e2 studio
5.1 Create a Project	
5.1.1 What is a Project?	
5.1.2 Creating a New Project	
5.2 Configuring a Project	
5.2.1 Summary Tab	
5.2.2 Configuring the BSP	
5.2.3 Configuring Clocks	
5.2.4 Configuring Pins	
5.2.5 Configuring Interrupts from th	e Stacks Tab58
5.2.6 Creating Interrupts from the In	nterrupts Tab58
5.2.7 Viewing Event Links	
5.2.8 Adding Threads and Drivers	
5.2.9 Adding and Configuring HAL	Drivers59
5.2.10 Adding Drivers to a Thread an	nd Configuring the Drivers60
5.2.11 Configuring Threads	
5.3 Reviewing and Adding Compo	nents
5.4 Debugging the Project	
5.5 Modifying Toolchain Settings	
5.6 Importing an Existing Project in	nto e2 studio67
6. Notes on development	
6.1 Unexpected Update of Program	n placed on Flash ROM71
6.2 Getting USB Hub to be workab	le with USBX72
Revision History	
-	



1. Introduction

1.1 Overview

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

1.2 Introduction to FSP

1.2.1 Purpose

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

1.2.2 e2 studio IDE

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

1.3 Limitations

1.3.1 Hardware Initial Setup

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

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

	Board name on GUI screen	Note
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.
Octal-SPI Edition (RTK9763U02S01001BE)	RZ/A3UL Development board Dialog OCTAL Boot (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.

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

2.1.2.3 Licensing

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

2.1.3 e2 studio installation for Windows PC

This chapter describes how to install the e2 studio IDE on Windows PC. 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
Renesas e ² studio 2024-01.1 Setup	• (ENESAS
Install Type	
Please select the e ² studio installation type. <u>Click here</u> for help selecting a type and to see wha	at features are included.
Select Install Type:	
Lite Install (Recommended) This installs e ² studio in Lite Mode. This mode offers a simplified experience focused on simple code editing & d	debugging with only important features
Standard Install This installs e ² studio in Advanced Mode. This mode offers all extended debugging functionality and other advanced f	features
Custom Install Custom installation of e ² studio This mode is allows you to select which features are installed	

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.

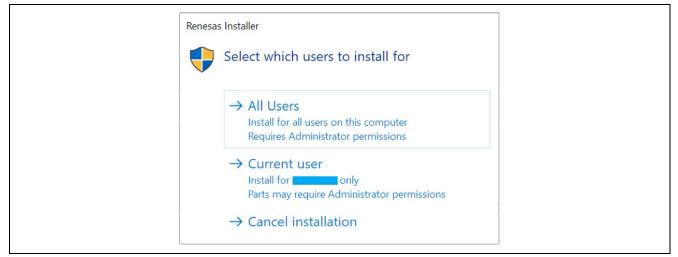


Figure 2: Select User for Installation



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.

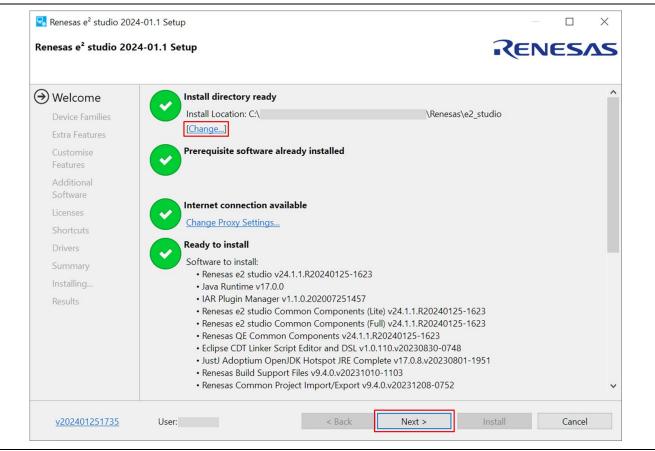


Figure 3: Installation of e2 studio – Welcome page



3. Device Families

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

Select the device families you wish	to install support for
	RA Build, Debug & Code Generation support for Renesas RA devices
	RZ Build, Debug & Code Generation support for Renesas RZ devices
	RL78 RL78 Build, Debug & Code Generation support for Renesas RL78 devices
	RX Build, Debug & Code Generation support for Renesas RX devices
	RH850 Debug support for Renesas RH850 devices
Drivers	RE Build & Debug support for Renesas RE devices
Summary Installing	ENESAS Synergy Build, Debug & Code Generation support for Renesas Synergy devices
Results	ARemana Company A Remana Company Build & Debug support for Renesas DA devices
	RISC-V MCU Build & Debug support for Renesas RISC-V MCU devices
Select	All

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.

Select the extra features	you wish	to install	
Welcome		P	Japanese Language Support
Device Families			Chinese (Simplified) Language Support
Customise Features			Chinese (Traditional) Language Support
Additional Software			Git Integration Git SCM Support
Licenses			Terminals
Shortcuts		P	ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
Drivers			
Summary			
Installing			
Results			
	Selec	t All	

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 20	024-01.1 Setup RENESA	S
WelcomeDevice FamiliesExtra FeaturesCustomiseFeaturesAdditionalSoftwareLicensesShortcutsDriversSummaryInstallingResults	Select the components you want to install. Renesas RX family support requiring .Net Framework	*
v202401251735	Available: 176.4 GB	

Figure 6: Installation of e2 studio – Features



6. 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.

enesas e² studio 2	oza o na setup	RENESAS
Welcome	Please read and accept the follo	wing Software Agreements
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Renesas e2 studio OpenJDK License Agreement ARM DS-5 Toolchain Integration IAR Plugin Manager	License Terms and Conditions for RENESAS e2 studio This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS. The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer. Otherwise, by clicking the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of this Agreement, or by installing, accessing, or otherwise copying or using all or any portion of the Renesas IDE Software, you accept this Agreement on behalf of the entity for which you are authorized to act (e.g., an employer) and acknowledge that such entity is legally bound by this Agreement and you represent and
	< > > ✓ > ✓ I accept the terms of the Softw	warrant that you have the right, power and authority to act on behalf of and vare Agreements Print all,

Figure 8: Installation of e2 studio – Licenses



7. 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 e² studio 2024	4-01.1 Setup				– 🗆 X
Renesas e² studio 202	4-01.1 Setup			RE	
Welcome Device Families Extra Features Customise Features Additional Software Licenses Orivers Summary Installing Results	Shortcuts to important programs and In start menu group: Renesas Ele		in the following locations		€ Restore Default
<u>v202401251735</u>	User:	< Back	Next >	Install	Cancel

Figure 9: Installation of e2 studio – Shortcuts



8. 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.

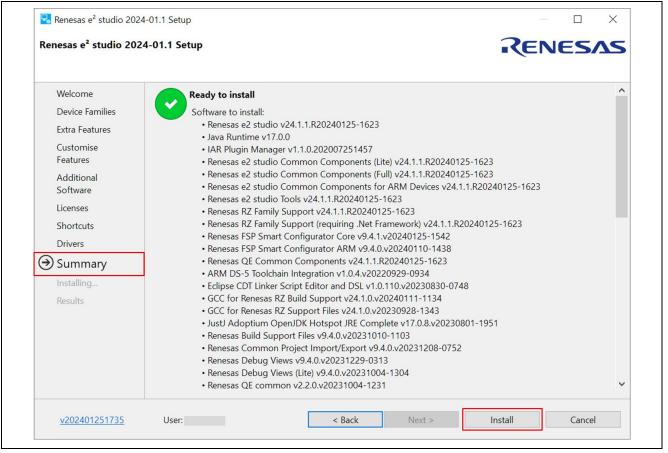


Figure 10: Installation of e2 studio – Summary

9. 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.



10.Results

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

Figure 1: Results Page



2.1.4 e2 studio installation for Linux PC

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

2.1.4.1 Prerequisite

Please download the development tool related stuff:

 SEGGER J-Link driver
 Please download the driver V7.92c or after from: https://www.segger.com/downloads/jlink/JLink Linux x86 64.deb

• e2 studio IDE installer

The latest e2 studio IDE installer package can be downloaded from Renesas website for free. Please check detailed information from: <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_V782c_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 V782c 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.

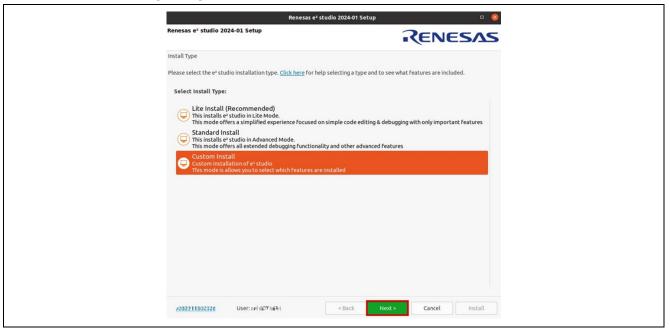


Figure 2: Selection of Install Type



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.



Figure 3: Installation of e2 studio – Welcome page

3. Device Families

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

Patersa of studio 2024-01 Setup Concention Select the device families you wish to install support for RA Welcome RA Device Families Build, Debug & Code Generation support for Renesas RA devices Customise RA Features Build, Debug & Code Generation support for Renesas RA devices Customise RE Build, Debug & Code Generation support for Renesas RL78 devices Oftware RE Licenses Build, Debug & Code Generation support for Renesas RL78 devices Software RE Installing Resolts Results RE Processor RE Build A Debug support for Renesas RL78 devices Curvessor RE Debug support for Renesas RL950 devices Curvessor RE Build & Debug support for Renesas RE devices Curvessor License RE Build & Debug support for Linux on Renesas RZ devices Curvessor Da Build & Debug support for Renesas DA devices
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results Results Results Results Cremesson RE Build, Debug & Code Generation support for Renesas RZ devices Cremesson RX Build, Debug & Code Generation support for Renesas RX devices Cremesson RE Build & Debug support for Renesas RH850 devices Cremesson RE Build & Debug support for Renesas RE devices Cremesson RE Build & Debug support for Renesas RE devices Cremesson RE Build & Debug support for Linux on Renesas RZ devices Cremesson Dal

Figure 4: Installation of e2 studio – Device Families



4. Extra Features

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

Renesas e² studio 2024	-01 Setup	Renesas e ³ studio 2024-01 Setup 0
Select the extra features y	you wish to install	
Welcome Device Families		Japanese Language Support
Extra Features		Chinese (Simplified) Language Support
Customise Features Additional Software	• •	Chinese (Traditional) Language Support
Licenses		Git Integration
Shortcuts		Git SCM Support
Summary		Terminals
Installing		ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
Results		
	Select All	
4202311302526	User: milicomcaa4	<back next=""> Cancel Install</back>

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 that "Renesas FSP Smart Configurator Core" is certainly selected.

Welcome Select the components you want to install. Becker Brainlies Extra Features O Customise Features Castomise Features Additional software Kiss Damise State Software Kiss Damise Configurator (24:1.0.R20231130-2205) Reneass RHSD Family Support (24:1.0.R20231130-2205) Reneass RHSD Family Support (24:1.0.R20231130-2205) Reneass RHSD Family Support (24:1.0.R20231130-2205) Common components for Reneass RHSD Family Support (24:1.0.R20231130-2205) Reneass RHSD Family Support (24:1.0.R20231130-2205) Common components for Reneass GE Installing Results @ Reness RHSD Family Support (24:1.0.R20231130-2205) Common components for Reneass GE Installing Results @ Reness RHSD Family Support (24:1.0.R20231130-2205) Common components for Reneass GE Calditional Tools (24:0. @ Reness RHSD Family Support (4:0.10.R20231130-2205) Common components for Reneass GE Calditional Tools (24:0. @ Reness RHSD Family Support (4:0.10.R20231130-2205) Common components for Reneas GE Calditional Tools (24:0.R20231130-2205) Common components for Reneas GE @ Chinese (Simplified) Language Support (4:5.100) @ Chinese (Simplified) Language Support (4:5.100) @ Chinese (Simplified) Language Support (4:5.100) @ Chinese (Inditional) Language Support (4:5.100) @ Chinese (Inditional)
Wetcome Device Families Extra Features Customise Features Customise Carecass RHSD Family Support (24.1.0.R20231130-2205) Common components for Renesa SPE Smart Configurator Care(0.4.0.20231130-1713) Common components for Renesa SPE Smart Configurator Care(0.4.0.20231130-2205) Common components for Renesa SPE Smart Configurator Care(0.4.0.20231130-2205) Common components for Renesa SPE Smart Configurator Care(0.4.0.0.20230050840-1) Customic Components (24.1.0.R20231130-2205) Common components (24.1.0.R20231130-2205) Common components (24.1.0.R20231130-2205) Common components (24.1.0.R20231130-2205) Common components (24.1.0.R202311

Figure 6: Installation of e2 studio – Features



6. 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.

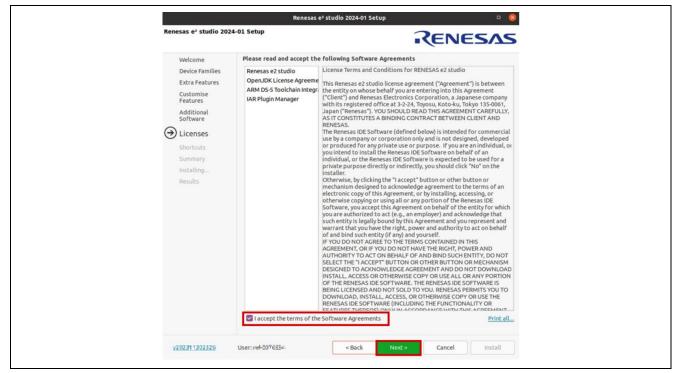


Figure 7: Installation of e2 studio – Licenses

7. Shortcuts

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

Nercease of studio 2022+01 Setup Vielcome Device Raillies Customise Pactrues Official Software Licenses Summary Installing Results
Device Families Extra Features Customise Features Additional Software Licenses € Shortcuts Summary Installing

Figure 8: Installation of e2 studio – 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.

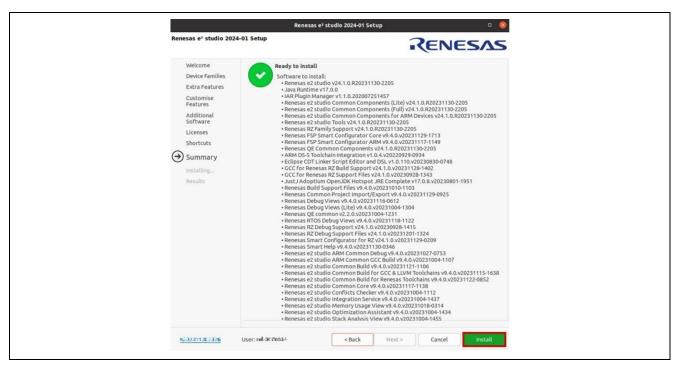


Figure 9: Installation of e2 studio – Summary

9. 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.

10.Results

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

Figure 10: Summary Page



2.2 Arm GNU Toolchain setup

2.2.1 Arm GNU Toolchain installation for Windows PC

1. Download installer

Download arm-gnu-toolchain-13.2.rel1-mingw-w64-i686-aarch64-none-elf.exe from the following URL:

https://developer.arm.com/downloads/-/arm-gnu-toolchain-downloads

2. Install

Launch the downloaded file and complete the installation.

2.2.2 Arm GNU Toolchain installation for Linux PC

1. Download the tarball from URL:

```
$ wget --content-disposition https://developer.arm.com/-
/media/Files/downloads/gnu/13.2.rel1/binrel/arm-gnu-toolchain-13.2.rel1-
x86_64-aarch64-none-
elf.tar.xz\?rev\=a05df3001fa34105838e6fba79ee1b23\&hash\=D63F63D13F01626D2070
19956E7122B5
```

2. Adjust the **e2studio_root** in the command below to correspond with the e2studio installation directory, and then extract its contents into the **[e2studio_root]/toolchains/gcc_arm_aarch64/13_2023_10** directory:

```
$ mkdir -p "[e2studio_root]/toolchains/gcc_arm_aarch64/13_2023_10"
$ tar -xf arm-gnu-toolchain-13.2.rel1-x86_64-aarch64-none-elf.tar.xz -C
"[e2studio root]/toolchains/gcc arm aarch64/13 2023 10"
```



2.2.3 Arm GNU Toolchain registration to e2 studio

Select Help -> Add Renesas Toolchains on e2 studio.

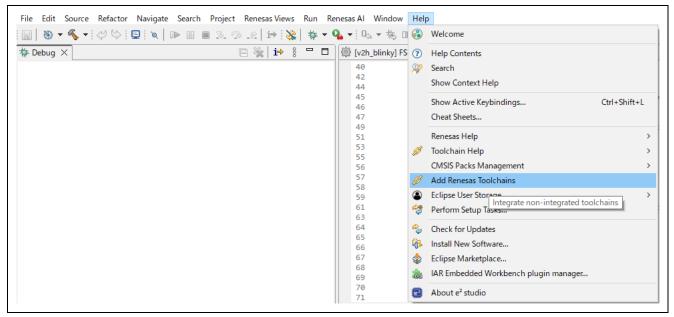


Figure 11: Add Renesas Toolchains Menu

Confirm **13.2.1.20231009** exists under **GCC ARM A-Profile** in the table. If it does not exist, press Add button and select the directory you have installed Arm GNU Toolchain.

Preferences		>
type filter text	Renesas Toolchain Management	↓ ↓
> FSP ^ Launch Settings	Scan for installed toolchains on startup	
Logging	Toolchain Type	Installation Path
Module Download	LLVM Embedded Toolchain for Arm	
My Renesas	V GNU ARM Embedded	
Reality AI Authentical	12.2.1.arm-12-mpacbti-34	C:¥Program Files (x86)¥Arm GNU Toolchain arm-none-eabi¥12.2 mpacbti-rel1¥
Renesas QE	10.3.1.20210824	C:¥Program Files (x86)¥GNU Arm Embedded Toolchain¥10 2021,10¥
Renesas Toolchain Ma	9.3.1.20200408	C:¥Program Files (x86)¥GNU Arm Embedded Toolchain¥9 2020-q2-update¥
Smart Browser	9.2.1.20191025	C:¥Program Files (x86)¥GNU Tools ARM Embedded¥9 2019-q4-major¥
> Smart Configurator	6.3.1.20170620	C:¥Program Files (x86)¥GNU Tools ARM Embedded¥6 2017-q2-update¥
Smart Demo	KPIT GNUARM-NONE-EABI Toolchain	
> Smart Manual	xPack GNU ARM Embedded	
Support Folders	✓ 🔽 GCC ARM A-Profile (AArch64 bare-metal)	
Tracealyzer	13.2.1.20231009	C:¥Program Files (x86)¥Arm GNU Toolchain aarch64-none-elf¥13.2 Rel1¥
TraceX	10.3.1.20210621	C:¥Renesas¥e2_studio_2024-01¥toolchains¥gcc_arm_aarch64¥10_2021_07¥gcc-arm-10.3-2021.07-mingw-w64-i686-aarch64-none-elf¥
> Run/Debug		
Scripting		
Terminal		
> TextMate		
Tracing		
Validation		
> Version Control (Team)		
< XML <		Download Scan Add Remove
? 눱 🖆 🖲		Apply and Close Cancel

Figure 12: Renesas Toolchain Management



2.3 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 2.3.2 or 2.3.3.

2.3.1 Installation of FSP Packs using Platform Installer

This section describes how to install FSP using Platform Installer **setup_rzafsp_v2_0_2_e2s_v2024-01.exe** showcased at <u>here</u>.

 Double-click setup_rzafsp_v2_0_2_e2s_v2024-01.exe, select either [Quick Install] or [Custom Install] and click [Next >] when the installation wizard is shown. When you chose [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.



Figure 13: 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 new folder here.
- 2. Multi-byte characters cannot be used for e2 studio installation folder name.

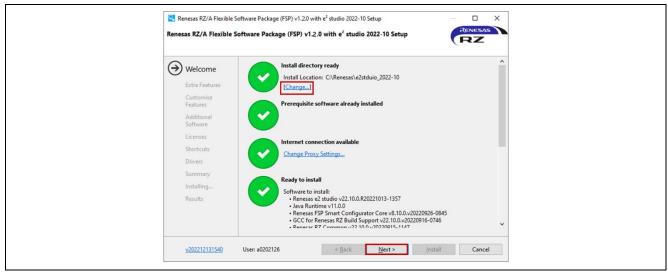


Figure 14: FSP Platform Installer – Welcome page



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.

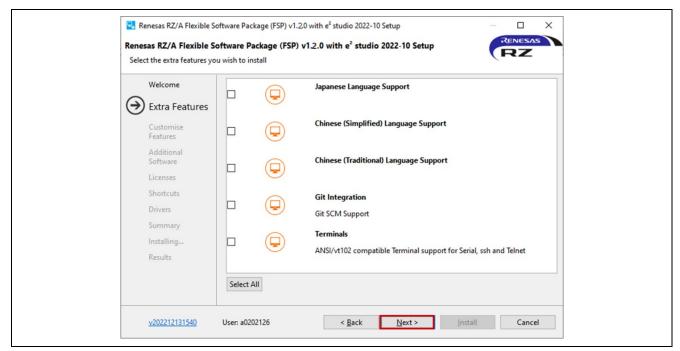


Figure 15: FSP Platform Installer – Extra Features

4. Customize Features

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

	Software Package (FSP) v1.2.0 with e ² studio 2022-10 Setup – \square × Software Package (FSP) v1.2.0 with e ² studio 2022-10 Setup
Welcome Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Select the components you want to install. Select the components you want to install. Common components for Renesa vel studio Image: Select the component for Renesa vel studio Image: Select the component for Renesa vel studio Image: Select the component for Renesa vel studio Image
<u>v202212131540</u>	User: a0202126 < <u>Back Next > Install</u> Cancel

Figure 16: FSP Platform Installer – Features



5. Additional Software

All the software is selected by default. Be sure to select at least GCC ARM A-Profile (AArch64 baremetal) 10.3-2021.07 and click [Next >].

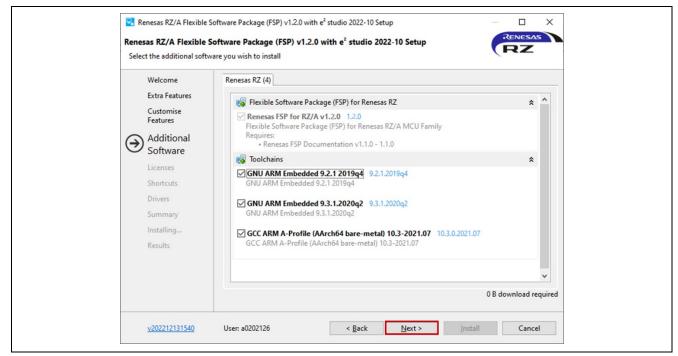


Figure 17: FSP Platform Installer – Additional Software

6. Licenses

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

Welcome	· · · · · · · · · · · · · · · · · · ·	ollowing Software Agreements	
Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Renesas e2 studio OpenJDK License Agreeme ARM DS-5 Toolchain Integr Renesas FSP for RZ/A v1.1.1 GCC ARM A-Profile (AArch Renesas Common Library E	This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREMENT 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 is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer. Otherwise, by clicking the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of this Agreement, or by installing, accessing, or otherwise copying or using all or any portion of the Renesas IDE Software. you accent this Agreement on behalf of the entity for which	~

Figure 18: 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 to distinguish from the other e2 studio(s).

Renesas RZ/A Flexible	Software Package (FSP) v1	.2.0 with e ² studio 2022	2-10 Setup	(RENESAS
Welcome Extra Features	Shortcuts to important pro		ated in the following	locations:	
Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results				e	Bestore Default
<u>v202212131540</u>	User: a0202126	< <u>B</u> ack	<u>N</u> ext >	Install	Cancel

Figure 19: 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.

	oftware Package (FSP) v1.2.0 with e ² studio 2022-10 Setup oftware Package (FSP) v1.2.0 with e ² studio 2022-10 Setup Ready to install	NESAS
Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Software to install: • Renesas e2 studio v22.10.0.R20221013-1357 • Java Runtime v11.0.0 • Renesas F2P Smart Configurator Core v8.10.0.v20220926-0845 • GCC for Renesas RZ Build Support v22.10.0.v20220916-0746 • Renesas RZ Debug Support v22.10.0.v20220915-1147 • OpenJDK v11.0.15.v20220714-1112 • openJfX.standard.feature v17.0.2.202205100908 • openJfX.standard.feature v17.0.2.202205100908 • openJfX.swin.feature v17.0.2.202205100908 • openJfX.swin.feature v17.0.2.202205100908 • openJfX.swin.feature v17.0.2.202205100908 • openJfX.swi.feature v17.0.2.20205100908 • openJfX.swi.feature v17.0.2.20205100908 • openJfX.swi.feature v17.0.2.20205100908 • openJfX.swi.feature v17.0.2.20205100908 • openJfX.swi.feature v17.0.2.20205100908 • openJfX.swi.featurev	
<u>v202212131540</u>	User: a0202126 < <u>B</u> ack <u>Next</u> > <u>Install</u>	Cancel

Figure 20: 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. Please note that the link to the installation place of GCC ARM A-Profile (AArch64 bare-metal) 10.3.2021.07 you selected as Additional Software is shown at "Useful Links:".

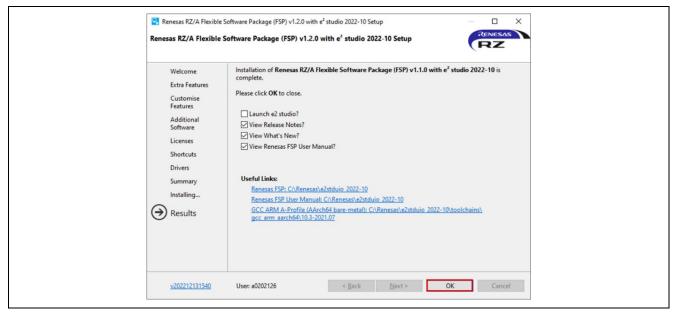


Figure 21: Installation Results of FSP Platform Installer

2.3.2 Installation of FSP Packs using Package Installer

Package Installer **RZA_FSP_Packs_v2.0.2.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_v2.0.2.exe.
- 3. Click [Next >] to start the installation.

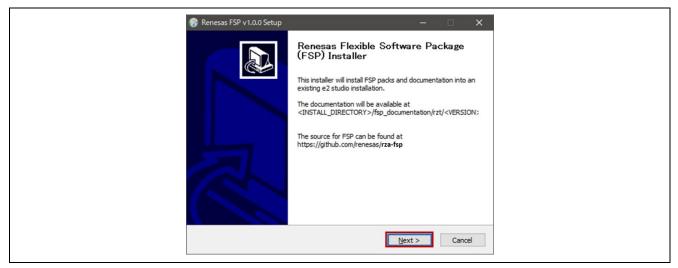


Figure 22: FSP Package Installer



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

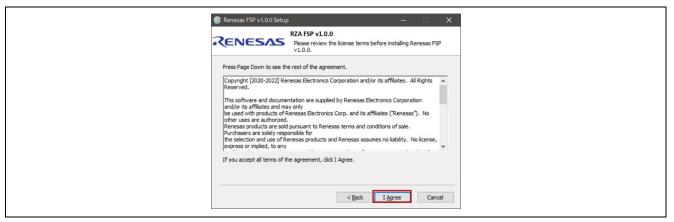


Figure 23: FSP License Term

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

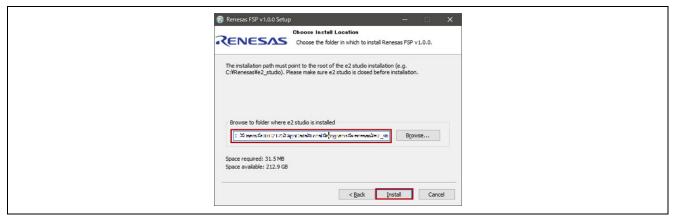


Figure 24: FSP Installation

6. Click [Finish] to complete the installation.

😴 Renesas FSP v1.0.0 Setup	×
	Completing Renesas FSP v1.0.0 Setup
	Renesas FSP v1.0.0 has been installed on your computer. Click Finish to close Setup.
	Open up documentation for this release)
	Click here to visit the FSP GitHub repository
	< Back Einish Cancel

Figure 25: 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.



2.3.3 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_v2.0.2.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_v2.0.2.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

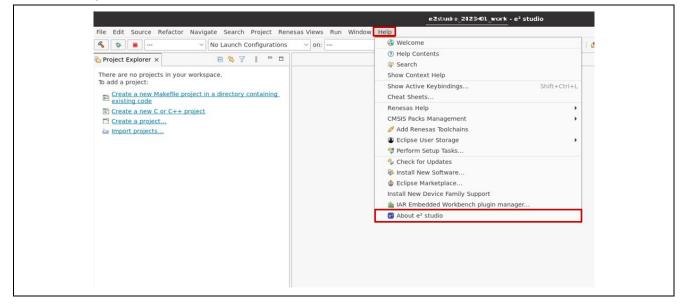
⊘ Recent	Name	▼ Size	Modified
	Amazon.FreeRTOS-Kernel.10.4.3-LTS.Patch.1+fsp.1.2.1.pack	263.9 kB	火
🚖 Starred	Amazon.FreeRTOS-Plus-TCP.2.3.2-LTS.Patch.1+fsp.1.2.1.pack	250.4 kB	火
	FreeRTOS.FreeRTOS_plus_FAT.1.2.1.pack	290.6 kB	火
Documents	🖌 Renesas.RZA.1.2.1.pack	1.5 MB	火
_	Renesas.RZA_baremetal_blinky.1.2.1.pack	2.6 kB	火
↓ Downloads	Renesas.RZA_board_custom.1.2.1.pack	4.8 kB	火
႕ Music	Renesas.RZA_board_custom_octal_xip.1.2.1.pack	4.9 kB	火
Pictures	Renesas.RZA_board_rza3ul_smarc_octal_xip.1.2.1.pack	72.3 kB	火
Videos	Renesas.RZA_board_rza3ul_smarc_qspi.1.2.1.pack	70.5 kB	火
	Renesas.RZA_board_rza3ul_smarc_qspi_xip.1.2.1.pack	70.7 kB	火
间 Trash	Renesas.RZA_config.1.2.1.pack	5.5 kB	火
thinclient_drives	Renesas.RZA_freertos_blinky.1.2.1.pack	4.9 kB	火
	🖌 Renesas.RZA_mcu_rza3ul.1.2.1.pack	287.2 kB	火
+ Other Locations	SEGGER.JLink.7.82.3.pack	29.4 MB 18	3 11月 2022

Figure 33. 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.

You can check if the installation is successfully done by the following procedure:

1. Click Help > About e² studio.





2. Click Installation Details.

	About e² studio 🛛 😣
	Renesas e ² studio
A ²	Version: 2023-01 (23.1.0) Build Id: R20230106-1556
E	Parts Copyright (C) 2010-2023 Renesas Electronics Corp. All rights reserved.
	e ² studio IDE is an extension of software developed for eclipse.org.
	e ² studio IDE is based on Eclipse Platform 4.25 (2022-09) and CDT version 10.7.0.
@ ●	e² 🔇 🌲 🔁 👙 🚿 🌐
⑦ Inst	tallation Details Close

 Click the link to e² studio support area. If the installation is successfully done, a file manager should be invoked, and the directory specified as e² studio support area should be shown there.



- 4. Check if FSP Packs are placed in the directory below:
- <e² studio support area>/internal/projectgen/rz_fsp/rza/packs

✓ > .eclipse c	om.renes159947646 internal projectgen rz_fsp rza packs 🔻		
⊘ Recent	Name		
-	Amazon.FreeRTOS-Kernel.10.4.3-LTS.Patch.1+fsp.1.2.1.pack		
🚖 Starred	Amazon.FreeRTOS-Plus-TCP.2.3.2-LTS.Patch.1+fsp.1.2.1.pack		
습 Home	FreeRTOS.FreeRTOS_plus_FAT.1.2.1.pack		
Documents	Renesas.RZA.1.2.1.pack		
	Renesas.RZA_baremetal_blinky.1.2.1.pack		
↓ Downloads	Renesas.RZA_board_custom.1.2.1.pack		
J Music	Renesas.RZA_board_custom_octal_xip.1.2.1.pack		
D Pictures	Renesas.RZA_board_rza3ul_smarc_octal_xip.1.2.1.pack		
	Renesas.RZA_board_rza3ul_smarc_qspi.1.2.1.pack		
Videos	Renesas.RZA_board_rza3ul_smarc_qspi_xip.1.2.1.pack		
🗐 Trash	Renesas.RZA_config.1.2.1.pack		
thinclient_drives	Renesas.RZA_freertos_blinky.1.2.1.pack		
	Renesas.RZA_mcu_rza3ul.1.2.1.pack		
+ Other Locations	SEGGER. JLink. 7.82.3. pack		

- 5. Check if xml files are placed in the folders below:
 - <e² studio support area>/internal/projectgen/rz_fsp/rza/modules
 - <e² studio support area>/internal/projectgen/rz_fsp/rza/2.0

Contraction of the second s	om.renes159947646	internal	projectgen	rz_fsp	rza 🔻
⊙ Recent	Name				
🛊 Starred	in 1.2				
습 Home	🛅 packs				
Documents					
↓ Downloads					
🗃 Music					
Pictures					
Videos					
🗐 Trash					
thinclient_drives					
+ Other Locations					



3. Set up an SMARC EVK

Below is an example of a typical system configuration.

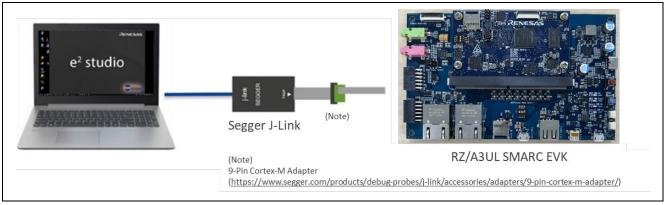


Figure 26: System Configuration Example – SMARC EVK

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>.

3.2 Board Setup

3.2.1 Boot Mode

Set up the SW11 as follows to configure Boot Mode 3 (QSPI or OCTA Boot (1.8V) Mode)/

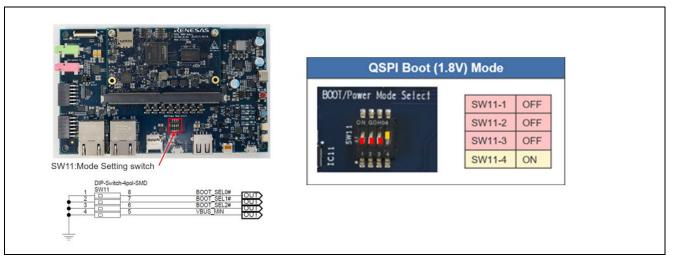


Figure 27: Boot MODE



3.2.2 JTAG connection

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

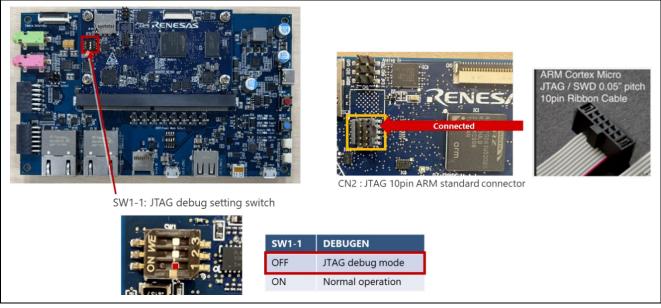


Figure 28: JTAG connection

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

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

3.2.3 Debug Serial (console output)

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

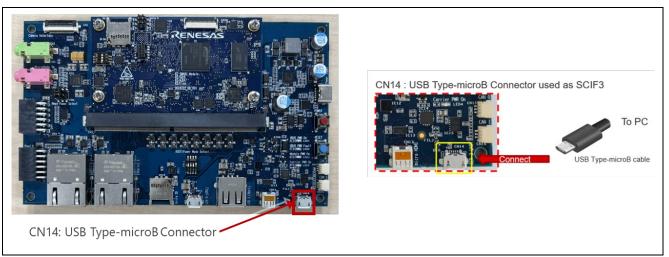


Figure 29: Debug Serial (console output)



3.2.4 Power Supply

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

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

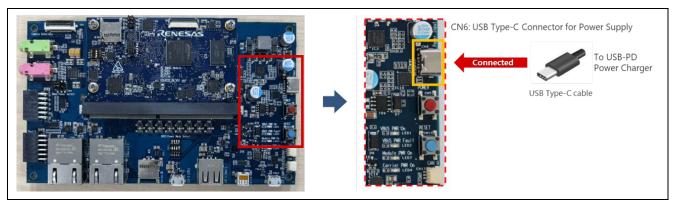


Figure 30: Power Supply

For power supply, please follow the following procedure:

- 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.

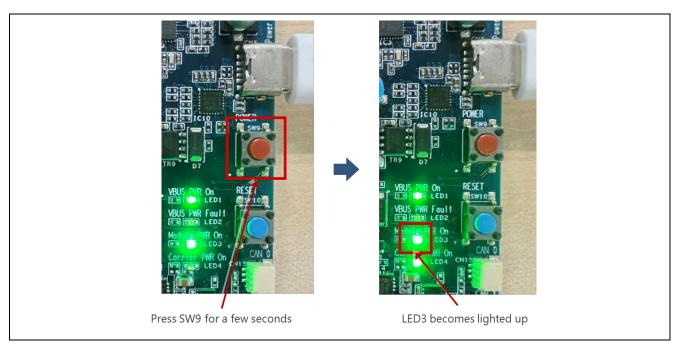


Figure 31: LED Status after Turning on EVK



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 figure below:



Figure 32: Establishment of connection between EVK 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 QSPI Flash or OctaFlash by default.

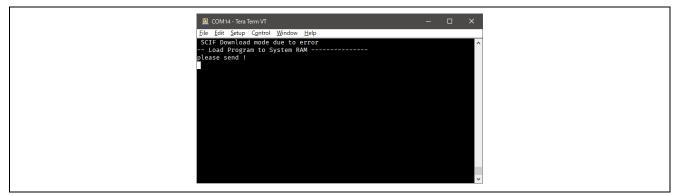


Figure 33: Message on your terminal at the 1st power-up of EVK

4. Tutorial: Your First RZ MPU Project - Blinky

4.1 Tutorial Blinky

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

4.2 What Does Blinky Do?

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

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

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

Note:

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



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

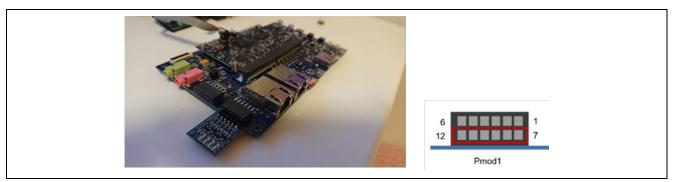


Figure 34: Connection Pmod LED module (410-076)

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

4.3 Create a New Project for Blinky

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

Follow these steps to create an RZ MPU project:

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

e² e	2studio	-2022-07_w	ork - e ²	studio					
File	Edit	Navigate	Search	Project	Renesas Views	Run	Window	Help	
	New				Alt+Shift+N	>	Renesas (C/C++ Project	>
	Open	File				C	C/C++ Pr	roject	
Ċ,	Open	Projects fro	m File S	ystem		2	Project		
	Recent	t Files				> 📑	Example.		
	Close	Editor			Ctrl+W	-9	Other		Ctrl+N
	Close	All Editors			Ctrl+Shift+W	-			

Figure 35: New C/C++ Project

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

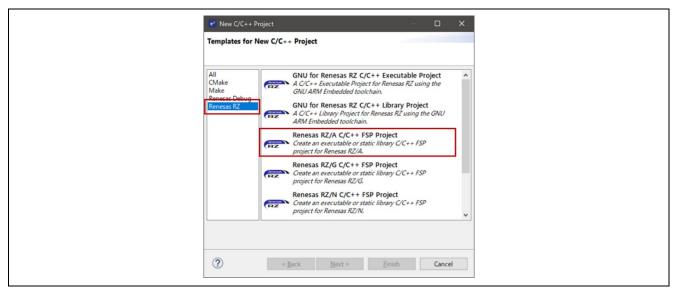


Figure 36: Renesas RZ/A C/C++ FSP Project



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

Renesas RZ/G C/C++ FSP Project	– D X
Renesas RZ/G C/C++ FSP Project Project Name and Location	Ż
Project name Blinky Use default location	
Location: C:\work\RZG2L\tmp\e2studio_2022-01_work\Blinky Choose file system: default </td <td>Browse</td>	Browse
You can download more Renesas packs here	
? < Back Next >	

Figure 37 : e2 studio Project Configuration window (part 1)

5. Select the board support package corresponding to the package you would like to use, GNU 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].

Renesas RZ/A C/C++ FSP Project Device and Tools Selection		Ď
Device Selection FSP Version: 2.0.1 Board: RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place) Device: R9A07G063U02GBG Core: Core 0	Board Description	
Language:	Device Details TrustZone No Pins 361 Processor Cortex-A55	
Toolchains GCC ARM A-Profile (AArch64 bare-metal) 13.2.1.20231009 V	Debugger J-Link ARM	~

Figure 38 : e2 studio Project Configuration window (part 2)



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

	Renesas RZ/A C/C++ FSP Project
	Build Artifact and RTOS Selection
RTOS Selection	Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file

Figure 39 : e2 studio Project Configuration window (part 3)

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

Project Template Selection	
Bare Metal - Bli	nky
Bare metal FSP project the C runtime enviror	t that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and ment.
[Renesas.RZA3UL.1.0	0.pack]
Bare metal FSP project	t that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment.
[Renesas.RZA3UL.1.0	0.pack]
	0.pack]
[Renesas.RZA3UL1.0	0.pack]

Figure 40 : 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.

ummary			Generate Project Content
Project Summar	y .		
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:	10.3.1.20210621		
FSP Version:	1.0.0		
Project Type:	Flat		
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky		
Selected software c	omponents		
Simple application	that blinks an LED. No RTOS included.	v1.0.0	
Application configu	uration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0	
Board Support Pa	ckage Common Files	v1.0.0	
I/O Port		v1.0.0	
MMU		v1.0.0	
General Timer		v1.0.0	
RZ/A3UL Evaluati	on Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0	
Board support pac	kage for R9A07G063U02GBG (RZ/A3UL)	v1.0.0	
Board support pac	kage for RZA3UL (RZA)	v1.0.0	
Board support pac	kage for RZ/A3UL (RZ/A3UL) - FSP Data	v1.0.0	

Figure 41 : 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.

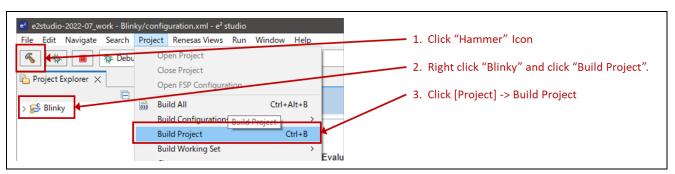


Figure 42 : e2 studio Project Explorer window

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:



Figure 43 : 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 Debug History Debug As >	Run Ctrl+F11 Debug F11 Run History > Run As > Run Configurations > Debug History >	Run	Window Help	
Image: bebug F11 Run History > Run As > Run Configurations	Image: body of the second s	Image: body with the second	1	Renesas Debug Tools	>
Run History > Run As > Run Configurations	Run History > Run As > Run Configurations Debug History > Debug As >	Run History > Run As > Run Configurations > Debug History > * Debug As	Q	Run	Ctrl+F11
 Run As Run Configurations 	Nun As > Run Configurations > Debug History > ★ Debug As >	Run As > Run Configurations > Debug History > * Debug As	椮	Debug	F11
Run Configurations	Run Configurations Debug History > As Debug As >	Run Configurations Debug History > the Debug As >		Run History	>
	Debug History >	Debug History >	-		>
Debug History >	🎋 Debug As >	The Debug As		Run Configurations	
					>

Figure 44 : e2 studio Debug icon

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

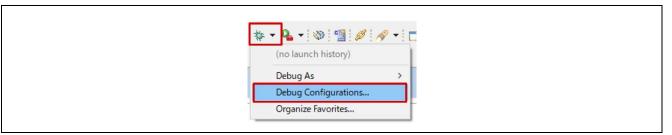


Figure 45 : e2 studio Debugger Configurations selection option



 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.

Figure 46 : e2 studio Debugger Configurations window with Blinky project (1)

3. If you have selected the QSPI Edition of the SMARC EVK board, skip to Step 8. If you have selected the OCTAL Edition of the SMARC EVK board, Select the debug configuration for the generated project and select the **Debugger** tab.

	uduana Dahumaina aan	finunation Blinks Dal	hun Elet fen Dehun
cult Kenesas GDB H	ardware Debugging con	liguration Blinky Der	bug_riat for Debug
	me: Blinky Debug_Flat		
🗎 Main 🎋 Debugger 🕨	Startup 🗉 Common 🦻 Sou	rce	
Dobug bardware I-Lir	k ARM 🛛 👋 Target Device: 🛛	(9A0/G0630	

Figure 47 : e2 studio Debugger Configurations window with Blinky project (2)



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

Aain 🕸 Debugger 🕨 Startup 🧐 Source 🔲 Com	hmon
Debug hardware: J-Link ARM V Target Device: R	
GDB Settings Connection Settings Debug Tool Settin	gs
✓ J-Link	^

Figure 48 : 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 V Target De	vice: R9A07G063U02GBG	
GDB Settings Connection Settings Debug Too	ol Settings	
✓ J-Link		^
Туре	USB	~
J-Link Serial	(Auto)	
Settings File	\${workspace_loc:/s	Restored and
Script File		
Log File	\${workspace_loc:/	\${ProjN
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	JTAG	~
Speed (kHz)	15000	~
 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	~

Figure 49 : e2 studio Debugger Configurations window with Blinky project (4)



٦

6. Select the **Debug Tool Settings tab** inside the **Debugger** tab.

📴 Edit Config	guration	×
Edit Renesa	s GDB Hardware Debugging configuration Blinky Debug_Flat for De	bug
Laurah Caufin	netice News Rivers Delever Flat	
	uration Name: Blinky Debug_Flat	
🖹 Main 🏂 Deb	bugger 🕨 Startup 🔲 Common 🦻 Source	
Debug hardv	vare: J-Link ARM \checkmark Target Device: R9A07G063U	
GDB Setting	s Connection Settings Debug Tool Settings	

Figure 50 : e2 studio Debugger Configurations window with Blinky project (5)

7. Change Flash Bus Type to OctaBus (At this time, Flash Memory Type is automatically changed to OctaFlash).

Edit Configuration		>
dit Renesas GDB Hardware Debugging	g configuration Blinky Debug_Flat fo	or Debug
unch Configuration Name: Blinky Debug_Fla	t	
Main 🏇 Debugger 🕨 Startup 🖽 Common	Source	
Debug hardware: J-Link ARM 🗡 Target De	vice: R9A07G063U	
GDB Settings Connection Settings Debug	Tool Settings	
 ✓ IO 	oor settings	
Use Default IO Filename	Yes	~
IO Filename	\${support_area_loc}	
✓ General Debug		
Reset After Reload	Yes	~
✓ Memory		
Endian	Little Endian	~
❤ Break		
Use Flash Breakpoints	Yes	~
Allow Simulation	No	~
✓ Flash		
Flash Bus Type	OctaBus	~
Flash Memory Type	OctaFlash	~
WorkRam Start		

Figure 51 : e2 studio Debugger Configurations window with Blinky project (6)



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

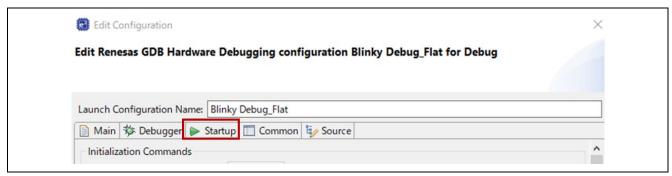


Figure 52 : e2 studio Debugger Configurations window with Blinky project (7)

9. 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 [Debug_Flat for <u>D</u> ebug	9	Edit Renesas GDB Hardware	Debugging configura	ation Blinky I	Debug_Flat for Debu	g
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				Ŷ	Load image and symbols				
oad image and symbols Filename	Load type	Offset (hex)		Add	Load image and symbols Filename	Load type Symbols only	Offset (hex)	On connect Vec	Add
oad image and symbols Filename		Offset (hex)	On connect Yes	Add Edit	Load image and symbols Filename	Symbols only v Image and Symbols	Offset (hex)	On connect Yes	Add Edit
pad image and symbols		Offset (hex)			Load image and symbols Filename	Symbols only Image and Symbols Image only Symbols only	Offset (hex)		
oad image and symbols		Offset (hex)		Edit	Load image and symbols Filename	Symbols only Image and Symbols Image only	Offset (hex)		Edit

Figure 53 : e2 studio Debugger Configurations window with Blinky project (8)



10. 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:	Blinky Debug_Flat]	
📄 Main 🕸 Debugger 🕨 St	tartup 🛄 Common 💺	Source				
Initialization Commands						
Reset and Delay (seconds	s): 3					Add download module
Hait				~		Specify download module name:
				×		Variables Search Project Workspace File System
Load image and symbols						
Filename	Load type	Offset (hex)		Add		OK Cancel
Program Binary [Blinky.	Symbols only	~	Yes	Edit		
				Remove		
				Move up		Add download module
				Move down		Select a workspace resource
						V 😂 Blinky
						> 🦢 .settings > 😕 Debug
	e 4	dd download mod	tule	×		✓ ➢ ipl in rza3ul_smarc_qspi_ipl.elf
						rza3ul_smarc_qspi_ipl.srec
		fy download mod rkspace_loc:¥Blinl	ule name: ky¥ipl¥rza3ul_smarc_q:	spi_ipl.srec}		> 🤄 rza > 🍃 rza_cfg
						> 😓 rza_gen
	Va	iables Sear	rch Project Work	space File System		> 🦢 src
				OK Cancel		i .api_xml x .cproject
						x .project
						 .secure_azone .secure_xml

Figure 54 : e2 studio Debugger Configurations window with Blinky project (9)



11. 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**.

e ² Edit Configuration		×	
Edit Renesas GDB Hardware Debug	ging configuration Blinky Debug_Flat for <u>D</u> ebug		
Launch Configuration Name: Blinky De	ug_Flat	_	
🔝 Main 🕸 Debugger 🕨 Startup			
Initialization Commands			
Reset and Delay (seconds): 3			Add download module X
Halt			Add download module X
		0	Specify download module name:
		-	Variables Search Project Workspace File System
Load image and symbols			vanabies Search Project Pric System
		_	OK: Cancel
Filename Load			Caricel
rza3ul_smarc_qspi_ipl.s Imagi			
El resoluter deployees mag	Remov		•
	Move	2	R Add download module - C X
	Move do	vn	Select a workspace resource
			V 😸 Blinky
			settings
	Add download module	×	✓ ⊘ Debug
) 🦢 rza_gen
	Specify download module name:		> 🦢 src
	\${workspace_loc:#Blinky#Debug#Blinky.srec}	_ 4	M Blinky.bin Blinky.elf
	Variables Search Project Workspace File Sys	m	Blinkyeltin
	Insurance and In		Blinkychex
	OK Canci	10	Blinky.map
			Blinky.srec
			Co makefile
			makefile.init memory_regions.ld
			III memory_regions.ia
			CK Cancel

Figure 55: e2 studio Debugger Configurations window with Blinky project (10)

12. Then, click **Debug** button.

Create, manage, and run configurations					Ś		
	Name: Blinky Debug_Flat						
type filter text C/C++ Application C/C++ Remote Application C/C++ Remo	Main (\$p Debugger Dawn) Common (\$p Source Initialization Commands Reset and Debug (seconds): Hait						
7 Java Application	Load image and symbols				×		
Remote Java Application C Renesas GD8 Hardware Debugging	Filename	Load type	Offset (hex)	On connect			
C Blinky Debug_Flat	Program Binary [Blinky			Yes	Add		
rzv2l_cm33_rpmsg_demo Debug_Flat	✓ rza3ul_smarc_qspi_ipl.s			Yes			
rzv2l_cm33_rpmsg_demo Release_Flat test Debug_Flat	Blinky.srec [C:\work\R	Image and Symbols	0	Yes	Edit		
twte Debug_Flat					Remove		
C Renesas Simulator Debugging (RX, RL78)					Move up		
					Move down		
	Runtime Options Stranger Stranger Stranger Run Commands C	de [× •		
Filter matched 17 of 19 items				Revert	Apply		
0				Debu	g Close		

Figure 56: e2 studio Debugger Configurations window with Blinky project (11)



13.Debug session is now started.

Configuring GDB	🤨 Progress Information	- D X
	Launching Blinky Debug_Flat	
	Configuring GDB	
		Cancel Details >>

Figure 57: e2 studio Debugger Configurations window with Blinky project (12)

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

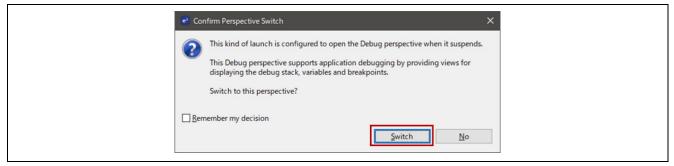


Figure 58: e2 studio Debugger Configurations window with Blinky project (13)

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:



Make sure the box Set breakpoint at: is checked and specify main as its value, Program Counter should be

stopped at main() function.



) Main 🕸 Debugger 🕟 Star	tup 🔲 Common 🦌	Source	
Initialization Commands			
Reset and Delay (seconds):	3		
Halt			
Load image and symbols		1	
Filename	Load type	Offset (hex)	On connect
Program Binary [barem	Symbols only		Yes
✓ rza3ul_smarc_qspi_ipl.s	Image and Symbols	0	Yes
baremetal_blinky_new.s	Image and Symbols	0	Yes
Runtime Options Set program counter at (hex)	:	1	
Set breakpoint at:	main		
Resume			

Figure 60 : Set breakpoint at: option

- E
^

Figure 61 : Blinky project in Debug Mode

After that LED on the Pmod LED marked LD0, LD1, LD2 and LD3 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 located in the project's root directory.

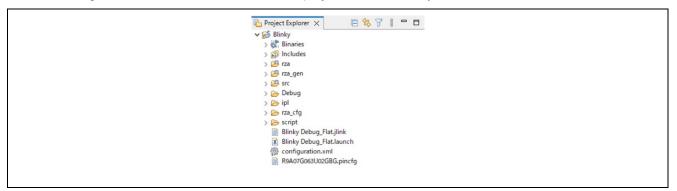


Figure 62 : e2 studio FSP Configuration Perspective

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 63 : 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.

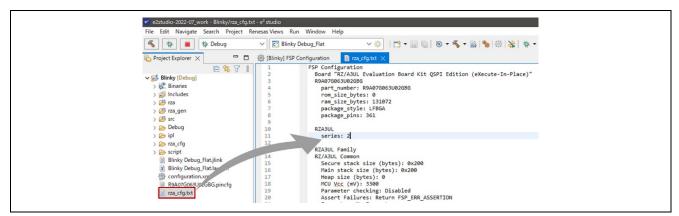


Figure 64 : RZ Project Report

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 Project Content
Project Summary	Generate Project Content
2ENESAS	^
board. REASOL Evaluation board Kit QSFT Edition (execute-in-riace)	·
Device: R9A07G063U02GBG Core: Core 0	
Core: Core U Toolchain: GCC for Renesas RZ (A-Profile AArch64 bare-metal)	
Toolchain Version: 10.3.120210621	
FSP Version: 1.0.0	
Project Type: Flat	
Location: C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky	
Selected software components	
Simple application that blinks an LED using FreeRTOS. v1.0.0	
Board Support Package Common Files v1.0.0	
Application configuration settings (Virtual memory settings, code RAM expansion rules) v1.0.0	
VO Port v1.0.0	
MMU v1.0.0	
General Timer v1.0.0	
FreeRTOS Port v1.0.0	
FreeRTOS v1.0.0	
FreeRTOS - Memory Management - Heap 4 v1.0.0 RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place) v1.0.0	
Board support package for R9A07G063U02GBG (RZ/A3UL) v1.0.0	
Board support package for RZA3UL (RZA) v1.0.0	

Figure 65 : 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].

e ² (2studio-2022-	7_work - e	² studio				
File	Edit Navig	ite Searc	h Project	Renesas Views	Run	Window Help	
	New			Alt+Shift+N	>	Renesas C/C++ Project	>
	Open File				Ċ	C/C++ Project	
	Open Project	from File	System			Project	
	Recent Files				° 🖻	Example	
	Close Editor			Ctrl+W	1	Other	Ctrl+N
	Close All Edit	ors		Ctrl+Shift+W	-	,	

Figure 66 : New RZ MPU Project

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

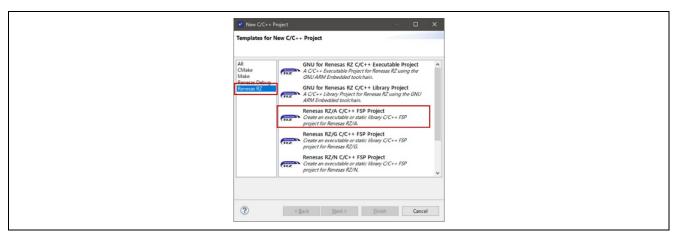


Figure 67 : New Project Templates



3. Select a project name and location.

I	🔮 Renesas RZ/A C/C++ FSP Project —	□ ×
	Renesas RZ/A C/C++ FSP Project Project Name and Location	
	Project name Blinky Use gefault location ocation: Choose file system: default You can download more Renesas packs here	Browse

Figure 68 : 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. To add threads, select RTOS, or No RTOS if an RTOS is not being used.
- 5. The Toolchain selection defaults to GNU ARM A-Profile (AArch64 bare-metal).
- 6. Select the **Toolchain version**. This should default to the installed toolchain version.
- 7. Select the **Debugger**. The J-Link Arm Debugger is preselected.
- 8. Click Next.

	Z/A C/C++ FSP Project		-	0 X
Renesas RZ/F	A C/C++ FSP Project			Ď
Device Selecti	ion			
FSP Version:	1.0.0 ~	Board Description		
Board:	[RZ/A3UL Evaluation Board Kit QSPI Edition (eXecut] \sim			
Device:	R9A07G063U02GBG			
Core:	Core 0 ~			
Language:		Device Details	1.4.1	
		TrustZone Pins Processor	No 361 cortex-a55	
Toolchains	-Profile (AArch64 bare-metal)	Debugger J-Link ARM		~
OCC ARM A	Prome (AArcho4 bare-metal)	J-LINK ARM		× .
10.3.1.202106	521 ~			
?		< <u>B</u> ack	Next > Einish	Cancel

Figure 69 : RZ MPU Project Generator (Screen 2)



5.1.2.2 Selecting a Project Template

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

e* Renesas RZ/A C/C++ FSP Project	— C	o x
Renesas RZ/A C/C++ FSP Project Build Artifact and RTOS Selection		Ď
Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file	RTOS Selection No RTOS	
0	< <u>B</u> ack <u>N</u> ext > Einish C	Cancel

Figure 70 : RZ MPU Project Generator (Screen 3)

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

Note:

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.

🖌 Renesas RZ/A C/C++ FSP Project — 🔲 🗙	Renesas RZ/A C/C++ FSP Project	- 0
enesas RZ/A C/C++ FSP Project roject Template Selection Project Template Selection Bare Metal - Blinky	Renesas RZ/A C/C -+ FSP Project Project Template Selection Project Template Selection Project Template Selection Propert OS - Blinky - Dynamic Allocation Tempert To Project mil Inside the MCU using the BSR PreeBTOS will also be initiated and a sin [Renesas 2023.03.10.00, 203]	onfigured for dynamic memory allocation. right thread to blink the LEDs will be started
Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RZA3UL.1.0.0.pack] Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RZA3UL.1.0.0.pack]	FreeRTOS - Blinky - Static Allocation FreeRTOS is proceed that includes BS and will blink LIDs if available, freeRIOS is proceeding in the DCD using the RB FreeRIOS will be add using the RB FreeRIOS will be add using the RB FreeRIOS - Binky Static Allocation FreeRIOS - Minimal - Static Allocation FreeRIOS - Minimal - Static Allocation MOV using the RB [Renewas R2A3UL1.100.pack]	
Code Generation Settings 2] Use Renesas Code Formatter	Cade Generation Settings	
(Back Next > Einish Cancel	(?)	st > Einish Cancel

Figure 71 : 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.



ummary				Generate Project Content
Project Summary	/			^
Board: Device:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place) R9A07G063U02GBG		RENESAS	
Core:	Core 0			
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)			
Toolchain Version:	10.3.1.20210621			
FSP Version:	1.0.0			
Project Type:	Flat			
Location:	C:/work/RZA3UL/tmp/e2studio-2022-07_work/Blinky 🔄			
Selected software co	omponents			
Simple application	that blinks an LED using FreeRTOS.	v1.0.0		
Board Support Pag	ckage Common Files	v1.0.0		
Application configu	ration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0		
I/O Port		v1.0.0		
MMU		v1.0.0		
General Timer		v1.0.0		
FreeRTOS Port		v1.0.0		
FreeRTOS		v1.0.0		
	ory Management - Heap 4	v1.0.0		
RZ/A3UL Evaluation	on Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0		
	kage for R9A07G063U02GBG (RZ/A3UL)	v1.0.0		
	kage for RZA3UL (RZA)	v1.0.0		
Board support pac	kage for RZ/A3UL (RZ/A3UL) - FSP Data	v1.0.0		*
Support	(<u></u>			

Figure 72 : RZ MPU Project Editor and available editor tabs

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

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

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



5.2 Configuring a Project

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

5.2.1 Summary Tab

Summary		Gen	erate Project Content
Project Summar	y	RENESAS	^
Board: Device: Core:	RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place) R9A07G063U02GBG Core 0		
Toolchain: Toolchain Version: FSP Version:	GCC for Renesas RZ (A-Profile AArch64 bare-metal) 10.3.1.20210621 1.0.0		
Project Type: Location:	Flat C:/work/RZA3UL/tmp/e2studio-2022-07 work/Blinky 🔄		
Selected software c			
	that blinks an LED using FreeRTOS.	v1.0.0	
	ckage Common Files uration settings (Virtual memory settings, code RAM expansion rules)	v1.0.0 v1.0.0 v1.0.0	
MMU General Timer FreeRTOS Port		v1.0.0 v1.0.0 v1.0.0	
FreeRTOS FreeRTOS - Mem	ory Management - Heap 4 on Board Kit QSPI Edition (eXecute-In-Place)	v1.0.0 v1.0.0 v1.0.0	
Board support page	kage for R9A07G063U02GBG (RZ/A3UL) kage for RZA3UL (RZA)	v1.0.0 v1.0.0	~
Support	(E)		

Figure 73 : 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.

III Hopen	ties 🗙 💽 Problems 🆓 スマート・ブ	505- 🛃 🖇 🗖	
RZ/A3UI	Evaluation Board Kit QSPI E	dition (eXecute-In-Place)	
Settings	Property	Value	
	part_number	R9A07G063U02GBG	
	rom_size_bytes	0	
	ram_size_bytes	131072	
	package_style	LFBGA	
	package_pins	361	
	✓ RZ/A3UL Common		
	Secure stack size (bytes)	0x200	
	Main stack size (bytes)	0x200	
	Heap size (bytes)	0	
	MCU Vcc (mV)	3300	
	Parameter checking	Disabled	
	Assert Failures	Return FSP_ERR_ASSERTION	
	Error Log	No Error Log	
	PFS Protect	Enabled	
	C Runtime Initialization	Enabled	

Figure 74 : 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/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

s Configu	ation			0
Connigu				Generate Project
				Restore
4MHz	PLL1 1GHz	ICLK Div /1	$\sim \longrightarrow$ ICLK 1GHz	
	PLL2 1600MHz	→ Dív /2 800MHz → Dív /2 400MHz		
	PLL2 IGOUNIPIZ			
	> PLL2 533MHz	SDOCLK Set: 533MH	iz → SDOCLK 533MHz	
		Div /2 266500kHz SD1CLK Sel: 533MH	Iz → 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	
	PELS IDDOMINE		A ROCK HOMITZ	
		Div /4 200MHz > I2CLK Div /1	✓ → I2CLK 200MHz	
		→ P1CLK Div /1	✓ → P1CLK 200MHz	
			MOCLK 200MHz	
		Div /2 100MHz	> ZTCLK 100MHz	
		P2CLK Div /1	V	
		Faces by / i	Track with	
	→ PLL3 533MHz	PLL3_3 Sel: 533MHz V > PLL3_3 Div /2 V Div /2 133250kHz		
		-	↓ 	
	> PLL3 400MHz		SPI QSPI0_SPCLK 66625kF	
		V Div /2 66625kHz	→ SPI1CLK 66625kHz	
		→ Div /2 266500kHz	→ M2CLK 266500kHz	
		→ PLL3_5 Sel: 400MHz V → PLL3_5 Div /1 V → Div /2 200MHz		
			Octa QSPI0_SPCLK 100MH	
	> Div / 1000	Div /2 100MHz	OC1CLK 100MHz	
	017/1000		CCICCK IOWIN2	
	PLL4 1600MHz	→ Sel: OSC Div / 1000 ~ → Div /2 12kHz	SOCLK 12kHz	
	-			
	PLL5 1500MHz	→ DSLA Div /2 V → Div /1 750MHz	→ M3CLK 750MHz	
	PLL5 500MHz	→ Div /2 250MHz		
	> PLL6 500MHz	→ Div /2 250MHz → PLL6_2 Sel: 250MHz (fr ∨	HPCLK 250MHz	

Figure 75 : 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	s not support this configuration change. hanging the setting in IPL, edit PLL frequency depending on IPL setting.

Figure 76 : 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	O nerate Project Conten	📫 🍂 🌮 🖻 🖻 Type pin function 📰 Pin Function 👻 🔤 Module Name 🔻
Select Pin Configuration		Export to CSV file	Configure	Pin Driver Warnings	
RZA3UL-SMARC-QSPI-XIP	Manage configurat	tions			
Pin Selection $\exists \exists \vdots \exists z = \downarrow^a_z$	Pin Configuration			Cycle Pin Group	
Type filter text	Name	Value	Lock	Link	
> Connectivity:CAN A	Pin Group Selection	Mixed			
> V Connectivity:ETHE	Operation Mode	Disabled		100071000	
> 🗸 Connectivity:RIIC	✓ Input/Output SCIF4_RXD	None	0.0401		000 0 00000 0 000
> Connectivity:RSPI	SCIF4_SCK	None	1000	100	
Connectivity:SCI ✓ Connectivity:SCIF	SCIF4_TXD	None	1	-	
✓ ✓ Connectivity:SCIP ✓ SCIF0					
SCIF1					
SCIF2					
SCIF3					
SCIF4	<			>	-888
> Connectivity:USB > Debug:JTAG/SWD > Interrupt:IRO *	Module name: SCIF4				
Pin Function Pin Number					
Summary BSP Clocks Pins Interrup	. In succession of the				▶ Legend

Figure 77 : 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 Image configurations RZABUL-SMARC-QSPI-XDF Manage configurations Image configurations Image configurations Image configurations <	*[Blinky] FSP Configuration × Pin Configuration			Gen	O erate Projec	ct Content
Type filter text Name Value Lock Link V @ Periphenis Operation Mixed Outom Operation Mixed Outom Image: Section Image: Se	RZABUL-SMARC-QSPI-XIP			Configure	Pin Driver V	Warnings
V @ Prinjhenis Pin Group Selection Mixed > AnalogADC Operation Mode Custom > ConnectivityGFH SCIF1_CTS None ConnectivityGFH > ConnectivityGFH SCIF1_RSN Operation Operation > ConnectivityGFH SCIF1_RSN None ConnectivityGFH > SCIF1 SCIF1 None ConnectivityGFH	Pin Selection $\exists \exists t \models t \models \downarrow^a_{t \ge t}$	Pin Configuration			😲 Cycle Pi	in Group
	Peripherals AnalogADC AnalogADC AnalogADC AnalogADC ConnectivityCAN ConnectivityCAN ConnectivitySIC ConnectivitySI ConnectivitySI ConnectivitySI ConnectivitySI SOF0 SOF0	Pin Group Selection Operation Mode Input/Output SCIF-LCTS SCIF-LRTS SCIF-LRTS SCIF-LRTS SCIF-LRTS SCIF-LRTS SCIF-LTDO	Mixed Custom None None (20 * P13_1 None			>
	Pin Function Pin Number Summary BSP Clocks (3) Pins Inte	errupts Event Links Stacks Compone	nts			

Figure 78 : 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.

		HAL/Common General g_ioport I/O Port Driver on r_ioport g_timer0 Timer Driver on r_gtm General g_ioport I/O Port Driver on r_ioport Driver on r_ioport
	> Summar	s New Object > Remove
	rties × 💽 Problems 🌸 スマート・ブラウザー	📑 🔚 🏹 🗔 🔗 🖇 🗖 🗖 🌇 Pin Conflicts 📮 Console 🔯 Debug 🗙
g_timer	0 Timer Driver on r_gtm	Value
	 Common Module g_timer0 Timer Driver on r_gtm 	
	 General 	
	✓ Interrupts	
	Callback	timer_isr
	Underflow Interrupt Priority	5
	Generate at Start	Disable

Figure 79 : Configuring Interrupts in the Stacks tab

5.2.6 Creating Interrupts from the Interrupts Tab

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

-	Configuration	Generate Project Content
User Events		💿 New User Event > 🕼 Remove
Event		ISR
ID:78 GTM0	INT (GTM0 Interrupt)	gtm_int_isr
Allocations		
Interrupt	Event	ISR
0	ID:78 GTM0 INT (GTM0 Interrupt)	gtm_int_isr

Figure 80 : 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.

Threads	HAL/Common Stacks	New Stack > 🐣 Extend Stack > 🙀 Remove	000000
	g joport VO Port Driver on r_joport		- r_rspi on r_scif_uart
		Timers > Transfer >	
Objects 🕢 New Object > 🔬 Remove			000 0

Figure 81 : 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		
Threads	뒿 New Thread 🛛 🙀 Remove	New Thread Stacks	€.
✓ ▲ H ⊕ ⊕ ∞ N <	AL/Common G. g_ioport I/O Port (r_ioport) G. g_mmu MMU Driver on r_mmu FreeRTOS Port (rm_freertos_port New Thread	Add stacks to the (above), or by pa	
Summary	BSP Clocks Pins Interrupts Even	t Links Stacks Components	5
Propert	ties 🗙 🔝 Problems 🏟 Smart Br	owser 📮 Console	
New Th	read		
Settings	Property	Value	
	✓ Common		
	> General		
	> Hooks		
	> Stats		
	> Memory Allocation		
	> Timers		Enter Symbol Name of your
	> Optional Functions		thread here.
	> Logging Thread		Example: my_thread
	Symbol	new thread0	
	Name	New Thread	Enter Name of your thread here. Example: My Thread
	Charle size (huter)	4000	Example: wy fineau

Figure 82 : Adding a new RTOS Thread on the Stack tab

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

Note:

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

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

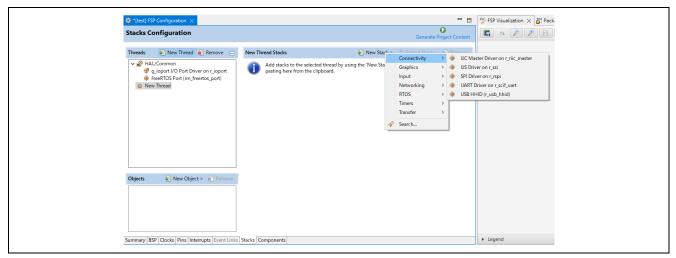


Figure 83 : Adding Modules and Drivers to a thread



- 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.

	FSP Configuration $ imes$				=	
	Configuration			Generate	O Project Con	
4 4 •	HAL/Common # g.jopot I/O Port Driver on r_joport # reeRTOS Port (m_freertos_port) www Thread # g_external_irq0 External IRQ Driver on r x	nal_irq0 External IRQ Driver on r_inte_irq g_external_irq0 External IRQ Driver on r_inte_irq	New Sta Remove	ck > <u>≗</u> Extend Stack		
-	BSP Clocks Pins Interrupts Event Links Stacks Co		8	Pin Conflicts	Console	á. Deb
Proble	ems 🦚 スマート・ブラウザー 🔲 Properties 🗙		00 -	Pin Conflicts	Console 3	S Debu
Proble g_exter	ems ④ スマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_intc_irq	C 🖩 7 🗔 🔗	8 = 0	Pin Conflicts	Console	‡ Debu
Proble	ems ④ スマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_intc_irq Property		8 -	Pin Conflicts	Console 3	\$ Deb
g_exter	ms ④ スマート・ブラクザー □ Properties × nal_irq0 External IRQ Driver on r_intc_irq Property v Common	📑 📴 🖓 🗔 🔗 Value	00 P	Pin Conflicts	Console 3	\$ Deb
Proble g_exter	ems	다 📴 🏹 🖾 🛷 Value Default (BSP)	00 P	Pin Conflicts	Console 3	‡ Deb
Proble g_exter	ens ④ スマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_intc_irq Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r,	Value Uefault (BSP)	8	Pin Conflicts	Console	‡ Deb
Proble g_exter	ens 金 スマート・ブラクザー Properties × nal_irq0 External IRQ Driver on r_intc_irq Property Common Parameter Checking Module g_external inQ External IRQ Driver on r, Name	Value Uefault (BSP) g_external_irq0	8 -	Pin Conflicts 😨	Console	‡ Deb
Proble g_exter	ems 、スマート・ブラウザー 「 Properties × nal_irq0 External IRQ Driver on r_intc_irq Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r Name Channel	Value Value 	8 8 8	Pin Conflicts	Console 3	‡ Deb
Proble g_exter	AR-h-ブラウザ ─ Properties × nal_irq0 External IRQ Driver on r_intc_irq Property Common Parameter Checking Module g_external iRQ Driver on r, Name Channel Tingger	Value Value Lintc_irq g_external_irq0 0 Rising		Pin Conflicts 🖸	Console	‡ Deb
g_exter	ems シスマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_intc_irq Property × Common Parameter Checking × Module g_external_irq0 External IRQ Driver on r. Name Channel Trigger Callback	Value Value Default (BSP) g_external_irq0 0 Rising NULL		Pin Conflicts 🖘	Console	‡÷ Deb
Proble g_exter	ms 、スマート・ブラウザー 「 Properties × nal_irq0 External IRQ Driver on r_intc_irq Property Common Parameter Checking Module g_external_irq0 External IRQ Driver on r, Name Channel Trigger Callback Pin Interrupt Priority	Value Value Lintc_irq g_external_irq0 0 Rising		Pin Conflicts	Console 3	a Debu
Proble g_exter	ems シスマート・ブラウザー Properties × nal_irq0 External IRQ Driver on r_intc_irq Property × Common Parameter Checking × Module g_external_irq0 External IRQ Driver on r. Name Channel Trigger Callback	Value Value Default (BSP) g_external_irq0 0 Rising NULL		Pin Conflicts 💿	Console 1	\$ Debu

Figure 84 : 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 an RTOS, the Stacks tab can be used to simplify the creation of RTOS threads, semaphores, mutexes, and event flags. The components of each thread can be configured from the **Properties** view as shown below:



શ Probler	ns 🛞 スマート・ブラウザー 🧮 Properties 🗙	📑 🔚 7 🗔 🛷 i 🗖 🗖
New Thr	ead	
Settings	Property V Common > General	Value
	Hooks Stats Memory Allocation Timers	
	Optional Functions Logging	
	✓ Thread Symbol Name	new_thread0 New Thread
	Stack size (bytes) Priority Thread Context	4096 1
	Memory Allocation	NULL Static

Figure 85 : 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**.

✓ MAL/Common ♥ g_ioport I/O Port Driver on r_ioport ♥ FreeRTOS Port (m_freertos_port) ♥ @ New Thread ♥ g_external_irq0 External IRQ Driver on r_intc_irq Ø g_new_message_buffer0 Message Buffer Ø g_new_message_buffer0 Message Buffer Ø g_new_message_buffer0 Message Buffer	Threads	🔄 New Thread 👔 Remove 😑	New Thread	d Stacks	New Stack >	🐣 Extend
Summany PSD Clocks Dins Intersunt Suppl Links Stacks Components	 g_ioport I/O Port D FreeRTOS Port (rm_ New Thread g_external_irq0 Ext 	freertos_port) ernal IRQ Driver on r_intc_irq @ New Object > Remove	Exte	rnal IRQ Driver on		
Summary DSP Clocks Prins Interrupts Event Links Stacks Components	Summary BSP Clocks Pins II	nterrupts Event Links Stacks Compon	ents			
Problems (%) スマート・ブラウザー □ Properties × Image: Problems (%) スマート・ブラウザー □ Properties × Image: Problems (%) スマート・ブラウザー □ Properties ×						
g_new_message_buffer0 Message Buffer						4

Figure 86 : 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.

Components Configuration		Gen	erate Project Conter	nt
		Group by: Vendor v Filter: All v	Search]
Component	Version	Description	Variant	^
Renesas				
V 🐼 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	1.000	External IRQ Driver(IRQ)		~

Figure 87 : 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	r 💁 🔹 🥖 🔹	
	1 FSP_project Debug	
	Debug As	\rightarrow
	Debug Configurations	
	Organize Favorites	

Figure 88 : 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.
📑 🖻 🍋 🗎 🗶 🖻 🍸 🗸	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
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 C Reness GDB Hardware Debugging	Build (if required) before launching Build Configuration: Use Active	¥ariables	Searc <u>h</u> Project	Browse
C MyProject Debug_Flat	O Enable auto build	O Disable aut		
Renesas Simulator Debugging (RX, RL78) Filter matched 13 of 15 items	Use workspace settings	<u>Contigure wo</u>	Re <u>v</u> ert	Apply
0	1		<u>D</u> ebug	Close

Figure 89 : 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 32 Debugger Startup 52 Source Common Debug hardware: J-Link ARM Target Device: R8A07G063J02GBG GDB Settings GDB Settings Register initialization No
				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			Yes	ID Code (Bytes) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
rza3ul_smarc_qspi_ipl.s		0	Yes	Hold reset during connect No
My_Project.srec [C:¥W			Yes	Set CPSR(5bit) after download No

Figure 90 : 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 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.

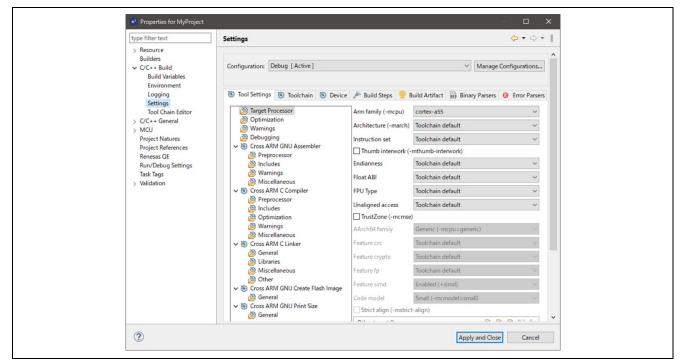


Figure 91 : e2 studio Project toolchain settings

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

The settings for the linker which control the location of the various memory sections are contained in a script file specific for the device being used. This script file is included in the project when it is created and is found in the created project. (for example, script/rza3ul_smarc_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	s preferences and development artifact	s.	
Workspace: C:\Users\ <user_name\e2studio\wo< td=""><td>rkspace</td><td>✓ <u>B</u>rowse</td><td></td></user_name\e2studio\wo<>	rkspace	✓ <u>B</u> rowse	
Use this as the default and do not ask again			
<u>R</u> ecent Workspaces			
	Launci	Cancel	

Figure 92 : 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 X 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 ► Recent Workspaces
Launch Cancel

Figure 93 : 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 94 : 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:

۵,	Edit Navigate Search Project New Open File Open Projects from File System Recent Files	Renesas Views R Alt+Shift+N >	21 21 2	Project Explorer 🔀 🗖 🗖 🗖 🔁 🔂 There are no projects in your workspace.
	Close Editor Close All Editors Save Save As	Ctrl+W Ctrl+Shift+W Ctrl+S		To add a project: C create a new C or C++ project Create a project Import projects
	Save All Revert Move Rename Refresh	Ctrl+Shift+S F2 F5	or	
ð	Convert Line Delimiters To Print Import	Ctrl+P		
	Export Properties Switch Workspace Restart	Alt+Enter		
	Exit		J	

Figure 96 : 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.

Select an import wizard: type filter text ✓ Seneral ● Archive File ● CMSIS Park- ● Existing Projects into Workspace ● File System ● File System ● Projects from Folder or Archive ● Projects from Folder or Archive ● Projects from Folder or Archive ● Renease CS+ Project for CA780K0(X78K0) ● Renease CS+ Project for CC-RX and CC-RL ● Renease CS+ Project for CC-RX and CC-RL ● Renease GitHub FreeRTOS (with Iol Tibraries) Project ● Source Generator > ● Code Generator > ● Comph > ● Run/Debug > ● Taxing > ● XML	Import Select Create new projects from an archive file or directory.	· ×	
	Select an import wizard: Type filter text Sype filter text General Gen		

Figure 97 : 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:

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

Figure 98 : 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	-		×
Import Projects Select a directory to sear	rch for existing Eclipse projects.		7
O Select archive file:	CoYwarv (1846-3011 Yhttp://w2shudia-2622-67 wid >	Browse	
Projects:	ZASU(MangMassado 2022.07_work//Blinsy)	Select All	
		Deselect All Refresh	

Figure 99 : Import Existing Project dialog 2 – Select the project to be imported

13. Click [Finish] to import the project.



6. Notes on development

6.1 Unexpected Update of Program placed on Flash ROM

If the program written to Flash ROM isn't updated expectedly even though there is no failure in downloading your program via e2studio, please check the settings of Flash Bus Type and Flash Memory Type shown in the figure below:

Edit Configuration			×
Edit Renesas GDB Hardware Debugging configu	uration test_bustype Debug_Flat for		
Launch Configuration Name: rza_blinky Debug_Flat			
🗈 Main 🎋 Debugger 🕨 Startup 🦻 Source 🗉 Common			
Debug hardware: J-Link ARM 🗡 Target Device: R9A0	D7G063U		
GDB Settings Connection Settings Debug Tool Setting	as		
× 10			^
Use Default IO Filename Ye	/es	~	
IO Filename \${	{support_area_loc}		
✓ General Debug			
Reset After Reload Ye	/es	~	
✓ Memory			
Endian Li	ittle Endian	~	
✓ Break			
Use Flash Breakpoints Ye	/es	~	
Allow Simulation N	No	~	
✓ Flash		_	
		~	
	DctaFlash	~	
WorkRam Start			
WorkRam End			
Erase on-chip program flash before download N		~	
		~	
		~	
	x0		E.A.

Figure 100 : Flash related settings in Debug Tool Settings

When Flash Bus Type and/or Flash Memory Type don't match the configuration of board shown in the following table, your program won't be downloaded properly.

Board Edition	Expected Flash Settings
QSPI Edition (RTK9763U02S01000BE)	Flash Bus Type : SPIBSC
	Flash Memory Type : SerialFlash
Octal-SPI Edition (RTK9763U02S01001BE)	Flash Bus Type : OctaBus
	Flash Memory Type : OctaFlash



6.2 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	
1.00	Jul.28.2022	-	First edition issued	
1.10	Sep.30.2022	-	Added the info on "RZ/A3UL Evaluation Board Kit Octal-SPI Edition.	
1.20	Dec.26.2022	-	Added the instructions for installing FSP using Platform Installer.	
1.21	Apr.07.2023	22, 23	Added 2.2.3 to describe the way to install FSP with the zipped Packs.	
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.	
2.0.1	Sep.30.2023	-	Updated the versions.	
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.	



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 resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a systemevaluation test for the given product.

Notice

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

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

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

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

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