

RZ/G2L RZ/G2LC RZ/G2UL RZ/G3S

Getting Started with Flexible Software Package

Introduction

This material describes how to use the Renesas Flexible Software Package (FSP) for writing applications for the RZ microprocessor series.

Target Device

RZ/G2L RZ/G2LC RZ/G2UL RZ/G3S



Contents

1. I	ntroduction	4
1.1	Overview	4
1.2	Introduction to FSP	4
1.2.1	Purpose	4
1.2.2	e2 studio IDE	4
1.3	Limitations	4
1.3.1	Peripherals and pins assignment	4
1.3.2	RAM Initialization	4
2. 5	Starting Development Introduction	5
2.1	e2 studio setup	5
2.1.1	What is e2 studio?	5
2.1.2	e2 studio Prerequisites	5
2.1.3	e2 studio installation for Windows PC	5
2.1.4	e2 studio installation for Linux PC	. 13
2.2	FSP setup	. 20
2.2.1	Installation of FSP using Package Installer	. 20
2.2.2	Installation of FSP Packs using Package Zip file	. 22
3. 8	Set up an SMARC EVK	.23
3.1	RZ/G2L SMARC EVK	. 23
3.1.1	Supported Debugger	. 23
3.1.1 3.1.2	Supported Debugger Board Setup	
		. 23
3.1.2	Board Setup	. 23 . 27
3.1.2 3.2	Board Setup RZ/G3S SMARC EVK	. 23 . 27 . 27
3.1.2 3.2 3.2.1 3.2.2	Board Setup RZ/G3S SMARC EVK Supported Debugger	. 23 . 27 . 27 . 27 . 27
3.1.2 3.2 3.2.1 3.2.2 4. T	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup	. 23 . 27 . 27 . 27 . 33
3.1.2 3.2 3.2.1 3.2.2 4. 7 4.1	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky	. 23 . 27 . 27 . 27 . 33 . 33
3.1.2 3.2 3.2.1 3.2.2 4. 7 4.1	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky	. 23 . 27 . 27 . 27 . 33 . 33 . 33
3.1.2 3.2 3.2.1 3.2.2 4. 7 4.1 4.2	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do?	. 23 . 27 . 27 . 27 . 33 . 33 . 33 . 34
3.1.2 3.2 3.2.1 3.2.2 4. 1 4.1 4.2 4.3	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 34 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 1 4.1 4.2 4.3 4.3.1	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky Details about the Blinky Configuration	. 23 . 27 . 27 . 27 . 33 . 33 . 33 . 33 . 34 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 4.1 4.2 4.3 4.3.1 4.3.2	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky Details about the Blinky Configuration Configuring the Blinky Clocks	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 33 . 38 . 38 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 4.1 4.2 4.3 4.3.1 4.3.2 4.3.3	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Utorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky Details about the Blinky Configuration Configuring the Blinky Clocks Configuring the Blinky Pins	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 34 . 38 . 38 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 1 4.1 4.2 4.3 4.3.1 4.3.2 4.3.3 4.3.4	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky Details about the Blinky Configuration Configuring the Blinky Clocks Configuring the Blinky Pins Configuring the Blinky Pins	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 33 . 38 . 38 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 1 4.1 4.2 4.3 4.3.1 4.3.2 4.3.3 4.3.4 4.3.5	Board Setup RZ/G3S SMARC EVK Supported Debugger Board Setup Tutorial: Your First RZ MPU Project - Blinky Tutorial Blinky What Does Blinky Do? Create a New Project for Blinky Details about the Blinky Configuration Configuring the Blinky Clocks Configuring the Blinky Pins Configuring the Blinky Pins Configuring the Parameters for Blinky Components Where is main()?	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 33 . 33 . 38 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 4.1 4.2 4.3 4.3.1 4.3.2 4.3.3 4.3.4 4.3.5 4.3.6	Board Setup	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 33 . 38 . 38 . 38
3.1.2 3.2 3.2.1 3.2.2 4. 4.1 4.2 4.3 4.3.1 4.3.2 4.3.3 4.3.4 4.3.5 4.3.6 4.4	Board Setup	. 23 . 27 . 27 . 33 . 33 . 33 . 33 . 33 . 33 . 38 . 38



RZ/G2L RZ/G2LC RZ/G2UL RZ/G3S

4.8 Details about the Debug Process	
4.9 Run the Blinky Project	
5. FSP application launch with e2 studio	12
5.1 Create a Project	
5.1.1 What is a Project?	
5.1.2 Creating a New Project	
5.1.3 Duplication of Resources	
5.2 Configuring a Project	
5.2.1 Summary Tab	
5.2.2 Configuring the BSP	
5 5	
5.2.4 Configuring Pins	
5.2.5 Configuring Interrupts from the Stacks Tab	
5.2.6 Creating Interrupts from the Interrupts Tab	
5.2.7 Viewing Event Links	
5.2.8 Adding and Configuring HAL Drivers	
5.3 Reviewing and Adding Components	
5.4 Debugging the Project	
5.5 Modifying Toolchain Settings	
5.6 Importing an Existing Project into e2 studio	
6. Migration from previous version	60
6.1 How to update RZ/G2L, RZ/G2LC, RZ/G2UL project	
6.2 How to update RZ/G3S project	
6.2.1 Procedure to update the project for CM33 without FPU Core	
6.2.2 Procedure to update the project for CM33 with FPU Core	
Revision History	



1. Introduction

1.1 Overview

This application note describes how to use the Renesas Flexible Software Package (FSP) running on the Cortex®-M33 (hereinafter referred to as CM33) incorporated on RZ/G2L, RZ/G2LC, RZ/G2UL and RZ/G3S.

1.2 Introduction to FSP

1.2.1 Purpose

The Renesas 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 Peripherals and pins assignment

RZ/G2L, RZ/G2LC, RZ/G2UL and RZ/G3S has a multi-core configuration of Cortex-A55 (hereinafter referred to as CA55) and CM33. It is possible to use each peripheral and GPIO from each core. This package provides drivers for the peripheral, and it is expected that peripherals, channels and pins to be used in the package can be occupied by FSP.

1.3.2 RAM Initialization

Initialization of DDR SDRAM is always carried out in CA55 bootstrap regardless of the selection of boot CPU, meanwhile Internal SRAM is initialized in the bootstrap of boot CPU.



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 the software for RZ MPUs, e2 studio hosts the Renesas Flexible Software Package (FSP). 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/G FSP, start with Evaluation Board Kit for each RZ/G Series.

Start-up guide of RZ/G2L, RZ/G2LC, RZ/G2UL Evaluation Board Kit is available at <u>SMARC EVK of RZ/G2L, RZ/G2UL, RZ/V2L, and RZ/Five Start-up Guide</u>.

2.1.2.2 PC Requirements

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

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

2.1.2.3 Licensing

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

2.1.3 e2 studio installation for Windows PC

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

2.1.3.1 Download

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

- Double-click on e2 studio installer to invoke the e2 studio installation wizard page. First, you need to select Install Type. In this material, it is expected that Custom Install is selected. Then, click [Next >] to continue.
- Note: If e2 studio was installed in your PC, the option to modify, remove the existing version or install e2 studio to a different location will be displayed



Nenesas e² studio 2024-01.1 Setup	- 🗆 X
Renesas e² studio 2024-01.1 Setup	RENESAS
Install Type	
Please select the e ² studio installation type. <u>Click here</u> for help selecting a type and to se	ee what features are included.
Select Install Type:	
Lite Install (Recommended) This installs e ^a studio in Lite Mode. This mode offers a simplified experience focused on simple code editing	& debugging with only important features
Standard Install This installs e ² studio in Advanced Mode. This mode offers all extended debugging functionality and other advance	ed features
Custom Install Custom installation of e ² studio This mode is allows you to select which features are installed	

Figure 1: Installation of e2 studio – Install Type

2. Welcome page

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

Notes: 1. If you would like to have multiple versions of e2 studio, please specify the new folder here.

2. Multi-Byte characters cannot be used for e2 studio installation folder name.

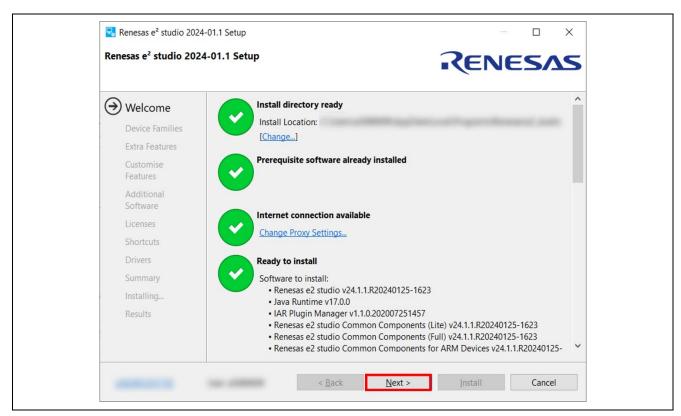


Figure 2: Installation of e2 studio – Welcome page



3. Device Families

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

Renesas e ² studio 2024- Renesas e ² studio 2024 Select the device families	I-01.1 Setup	pport for
Welcome Device Families Extra Features		RA Build, Debug & Code Generation support for Renesas RA devices RZ
Customise Features Additional Software	RL78	Build, Debug & Code Generation support for Renesas RL78 devices
Licenses Shortcuts		RX Build, Debug & Code Generation support for Renesas RX devices RH850 Debug support for Renesas RH850 devices
Drivers Summary Installing	RE	RE Build & Debug support for Renesas RE devices
Results		S Synergy
		< <u>Back Next > Install</u> Cancel

Figure 3: Installation of e2 studio – Device Families

4. Extra Features

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

Welcome Image: Constant of the sector of	Renesas e ² studio 2024-01.1 S Renesas e ² studio 2024-01.1 Select the extra features you wis	.1 Setup	
Customise Features Chinese (Traditional) Language Support Additional Git Integration Software Git SCM Support Licenses Git SCM Support Shortcuts Ferminals Drivers ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet Summary Installing Results Example Chinese] 🖵 Ja	panese Language Support
Features Chinese (Traditional) Language Support Additional Software Software Git Integration Licenses Git SCM Support Shortcuts Ferminals Drivers ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet Summary Installing Results Image: Support	→ Extra Features	_ 🖵 ^{Ch}	ninese (Simplified) Language Support
Software Git Integration Licenses Git SCM Support Shortcuts Terminals Drivers ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet Summary Installing Results Installing	Features	_ 🖵 ^{Ch}	ninese (Traditional) Language Support
Shortcuts Terminals Drivers ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet Summary Installing Results Installing	Software	(L)	-
Summary Installing Results	Shortcuts	те 🕞 Те	rminals
Results	Summary		
Calaat All			
Select All	Se	Select All	

Figure 4: Installation of e2 studio – Extra Features



5. Customize Features

Select the components to install and click the [Next >] to continue. Be sure that Renesas FSP Smart Configurator Core and Renesas FSP Smart Configurator ARM are selected.

Renesas e ² studio 20	24-01.1 Setup	12
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Select the components you want to install. Renesas RX family support requiring .Net Framework. Renesas RH850 family Support (24.1.1.R20240125-1623) Renesas RE family Support (24.1.1.R20240125-1623) Renesas Synergy family Support (24.1.1.R20240125-1623) Renesas Synergy family Support to allow project generation, build & debug Renesas DA family Support to allow project generation, build & debug Renesas Synergy Family Support to allow project generation, build & debug Renesas SPS Smart Configurator Core (94.1.1.V20240125-1542) Common ABM components for Renesas FSP Smart Configurator Renesas RISC-V MCU Support (24.1.1.R20240125-1623) Renesas RISC-V MCU Support (24.1.1.R20240125-1623) Renesas SR SC-V MCU Support Core (94.1.V20240125-1542) Common ABM components for Renesas FSP Smart Configurator Chenesas RISC-V MCU Support (24.1.1.R20240125-1623) Renesas RISC-V MCU Support (24.1.1.R20240125-1623)	~
	Size of install: 857.9 MB	

Figure 5: Installation of e2 studio – Customise Features

6. Additional Software

Select additional software (i.e., compilers, utilities, QE...) to be installed. Be sure to select the following item and click [Next >] to continue.

• GNU ARM Embedded 10.3 2021.10

Figure 6: Installation of e2 studio – Additional Software

For more details on the installation of Additional Software, please see section 2.1.3.3.



7. Licenses

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

Renesas e ² studio 20	24-01.1 Setup	RENESA
Welcome	Please read and accept the	following Software Agreements
Device Families Extra Features Customise Features Additional Software Contects Drivers Summary Installing Results	Renesas e2 studio OpenJDK License Agreem ARM DS-5 Toolchain Inter 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 Software Agreements

Figure 7: Installation of e2 studio – Licenses

8. Shortcuts

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

Note: If e2 studio has already been installed in another location, it is recommended to rename the shortcut to distinguish from the other e2 studio(s).

🛃 Renesas e² studio 2024-	4-01.1 Setup	– 🗆 X
Renesas e² studio 2024	4-01.1 Setup	
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Shortcuts to important programs and files will be created i	in the following locations:

Figure 8: Installation of e2 studio – Shortcuts



9. Summary

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

enesas e² studio 2024-01.1 Se	
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing Results	Ready to install: Software to install: • Renesas e2 studio v24.1.1.R20240125-1623 • Java Runtime v17.0.0 • IAR Plugin Manager v1.1.0.202007251457 • Renesas e2 studio Common Components (Lite) v24.1.1.R20240125-1623 • Renesas e2 studio Common Components (Full) v24.1.1.R20240125-1623 • Renesas e2 studio Common Components for ARM Devices v24.1.1.R20240125-1623 • Renesas e2 studio Tools v24.1.1.R20240125-1623 • Renesas e2 studio Tools v24.1.1.R20240125-1623 • Renesas RZ Family Support (requiring .Net Framework) v24.1.1.R20240125-1623 • Renesas FSP Smart Configurator Core v9.4.1.v20240125-1542 • Renesas FSP Smart Configurator ARM v9.4.0.v20240110-1438 • Renesas QE Common Components v24.1.1.R20240125-1623 • ARM D5-5 Toolchain Integration v1.0.4.v20220929-0934 • Eclipse CDT Linker Script Editor and DSL v1.0.110.v20230830-0748 • GCC for Renesas RZ Build Support v24.1.0.v20230928-1343 • Just J Adoptium OpenJDK Hotsport JRE Complete v17.0.8.v20230801-1951

Figure 9: Installation of e2 studio – Summary

8. Installing...

The installation is performed. Depending on selected items of additional software, new dialog prompts may appear during the installation process. Please see section 2.1.3.3 for more detailed information.

Welcome Please wait while e2 studio is installed. Device Families Extra Features Extra Features Installing IUs Customise Installing openjfx.web.win32_64 Features Installing openjfx.web.win32_64 Additional Software Licenses Shortcuts Drivers Summary Installing Results	Renesas e ² studio 2024 Renesas e ² studio 2024		RENESA	× \S
	Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary \overleftrightarrow Installing	Installing IUs		

Figure 10: Installation of e2 studio – Installing...



9. Results

Click the **OK** button to complete the installation.

Renesas e² studio 2024-01.1 Setup Installation of e2 studio is complete. Welcome Installation of e2 studio is complete. Device Families Please click OK to close. Extra Features □ Launch e2 studio? Customise □ View Release Notes? Additional □ View What's New? Software □ Useful Links: Licenses Jorivers Summary Installing Istalling Istalling Istalling Istalling Istalling Istall tinks:	🔜 Renesas e² studio 2024	4-01.1 Setup	- 🗆 X
Device Families Please click OK to close. Extra Features	Renesas e ² studio 202	4-01.1 Setup	RENESAS
	Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Drivers Summary Installing	Please click OK to close.	2021.10. To retry re-run installer and select Modify

Figure 11: Summary Page

2.1.3.3 Installation of Additional Software

As mentioned in section 2.1.3.2, the additional software listed below is essential for RZ/G FSP.

GNU ARM Embedded 10.3 2021.10

In this section, the detailed procedure for installing these tools.

(1) GNU ARM Embedded Toolchain 10.3 2021.10

If it was selected in the Additional Software pane of e2 studio, you will see the installation wizard for the GNU ARM Embedded Toolchain during the installation process.



RZ/G2L RZ/G2LC RZ/G2UL RZ/G3S

Getting Started with Flexible Software Package

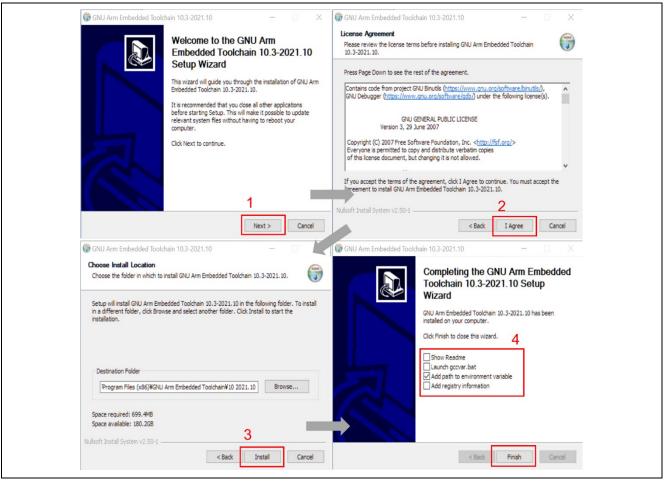


Figure 12: Installation of GNU ARM Embedded Toolchain



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 choose the version V7.94h or after and download Linux 64-bit DEB Installer at the URL below: <u>https://www.segger.com/downloads/jlink/</u>

GNU ARM Embedded Toolchain

Please download gcc-arm-none-eabi-10.3-2021.10-x86_64-linux.tar.bz2 in the Arm Developer site: https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads

e2 studio IDE installer

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.

1. SEGGER J-Link driver

Open a terminal window and enter commands stated below:

```
$ sudo dpkg -i JLink Linux V794h x86 64.deb
```

If the previous install fails with unmet dependencies, retry it as follows:

```
$ sudo apt-get -f install
$ sudo dpkg -i JLink Linux V794h x86 64.deb
```

2. GNU ARM Embedded Toolchain

Enter commands below on terminal. Note that GNU ARM Embedded Toolchain is expected to be placed at ~/Downloads.

```
$ sudo mkdir -p /opt
$ cd /opt
$ sudo tar jxvf ~/Downloads/gcc-arm-none-eabi-10.3-2021.10-x86 64-linux.tar.bz2
```

3. Installation of e2 studio IDE

Invoke the commands below to run the e2 studio IDE Installer. Note that it is expected that the installer is placed at ~/Downloads.

```
$ cd ~/Downloads
$ chmod 755 e2studio_installer-2024-01_1_linux_host.run
$ ./e2studio_installer-2024-01_1_linux_host.run
```



Then, the installation should be started. Please follow the following procedure:

• Install Type

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.

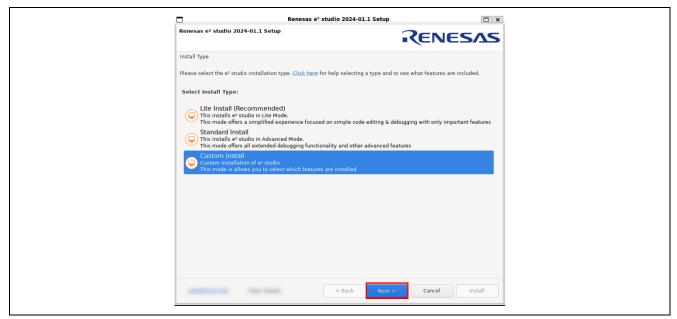


Figure 15: Selection of Install Type

• Welcome

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

Renesas e² studio 2024-01.1 Setup	RENESAS
⊖ Welcome	Install directory ready
Device Families	Install Location: [Change]
Extra Features	
Customise Features	Prerequisite software already installed
Additional Software	
	Internet connection available
Shortcuts	Change Proxy Settings
Summary	Ready to install
Installing	Software to install:
Results	 Renesas e2 studio v24.1.1.R20240125-1623 Java Runtime v17.0.0 IAR Plugin Manager v1.1.0.202007251457 Renesas e2 studio Common Components (Lite) v24.1.1.R20240125-1623 Renesas e2 studio Common Components (Lite) v24.1.1.R20240125-1623 Renesas e2 studio Common Components (Lite) v24.1.1.R20240125-1623 Eclipse CDT Linker Script Eclifor and DSL v1.0.110.v20230830-0748 Justj Adoptium OpenjOk Hotopot (R1K) v4.0.v20231010-1103 Renesas Build Support Files v9.4.0.v20231010-1103 Renesas Common Project Import/Export v4.0.4.v20231208-0752 Renesas Common Project v9.4.0.v20231204-033 Renesas Common V2.2.0.v20231204-1334 Renesas CDE common v2.4.0.v2023104-1304 Renesas Sant Heip v4.0.v2023104-131 Renesas Sant Heip v4.0.v2023104-1304 Renesas 2 studio Common Build v9.4.0.v2024104-018 Renesas 2 studio Common Build v9.4.0.v2024104-018

Figure 16: Installation of e2 studio – Welcome page

Notes: 1. If you would like to have multiple versions of e2 studio, please specify another directory here.

2. Multi-byte characters cannot be used for e2 studio installation directory name.



• Device Families

Select Devices Families to install. Click [Next >] to continue.

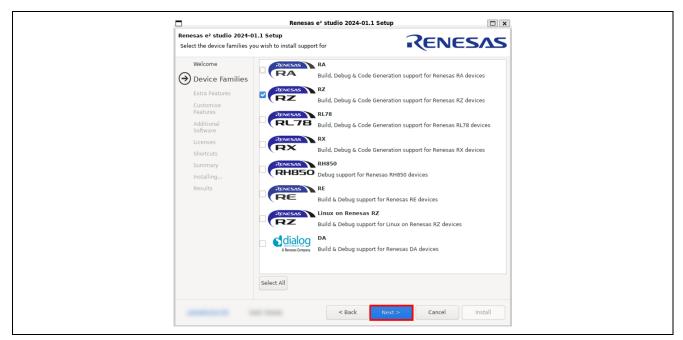


Figure 17: Installation of e2 studio – Device Families

Extra Features

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

Figure 18: Installation of e2 studio – Extra Features



Customize Features
 Select the components to install and click the [Next >] to continue. Be sure that Renesas FSP
 Smart Configurator Core and Renesas FSP Smart Configurator ARM are selected.

Welcome Select the components you want to install. Welcome perice families Extra Features Select the components you want to install. OC Lotomise Features Select the components you want to install. Additional Softwares Select the components you want to install. Mease RH850 family Support (24.1.1.R20240125-1623) mease RH850 family Support (24.1.1.R20240125-1623) Because RH850 family support (24.1.1.R20240125-1623) mease RH850 family support (24.1.1.R20240125-1623) Software Penesas RF9 Smart Configurator Components (04.1.V20240125-15423) Common components for Renesas PSP Smart Configurator Penesas RF9 Smart Configurator Components (24.1.1.R20240125-1623) Summary Sit integration for Eclipse (6.7.0.202309050840-1) Wersoning with GL, Integration with Gert. Gitflow, and Task repositories Calditional Tooling Benesities Sit integration for Eclipse (6.7.0.202309050840-1) Wersoning with GL, Integration with Gert. Gitflow, and Task repositories Additional Tooling Benesities Sit integration for Eclipse (6.7.0.202309050840-1) Wersoning with GL, Integration with Gert. Gitflow, and Task repositories Additional Tooling Benesities Sit integration for Eclipse (6.7.0.202309050840-1) Merice Integraton with Gert. Gitflow, and Task repositories
Welcome projects, debug only projects and IAR projects. Device Families Renesas RH850 family Support (24.1.1.R20240125-1623) Extra Features Penesas DA Family Support (24.1.1.R20240125-1623) Customise Renesas Ch4 Solve support to allow project quantaria. Additional Renesas FSP Smart Configurator RAN (06.40.v20240110-1438) Software Renesas FSP Somat Configurator RAN (06.40.v20240110-1438) Licenses Renesas QE Common Components for Renesas FSP Smart Configurator RAN (06.40.v20240110-1438) Software Renesas QE Common Components for Renesas FSP Smart Configurator RAN (06.40.v20240110-1438) Summary Renesas QE Common Components for Renesas (FSP Smart Configurator RAN (06.40.v20240125-1623) Summary Git integration for Eclipse (6.70.202309050840-r) Versioning with GL, Integration with Gert. Gitflow, and Task repositories Additional Tool (24.1.1.R20240125-1623) Add

Figure 19: Installation of e2 studio – Customise Features

Additional Software

Select the additional software (e.g., GCC Toolchains, Utilities and so on) to be installed and then, click [Next >] to continue. When following the procedure described in this material, there is no need to choose any software.

Figure 20: Installation of e2 studio – Customise Features



• Licenses

Read and accept the software license agreements listed below. Then, click [Next >] to continue. Note that the user must accept the license agreement, otherwise installation cannot be continued.

	Renesas e² studio 20	024-01.1 Setup	
Renesas e² studio 202	4-01.1 Setup	REN	Ε
Welcome	Please read and accept the fo	llowing Software Agreements	
Device Families Extra Features Customise Features Additional software Contects Summary Installing Results	OpenJDK License Agreer ARM DS-S Toolchain Inte IAR Plugin Manager IAR Plugin Manager INIS A CONTR design If you Softwar is expe you sh an elec or othe a cknow and you authori F you Softwar Agreen Contr IS A CONTR IS A CO	DO NOT AGREE TO THE TERMS CONTA MENT, OR IF YOU DO NOT HAVE THE RI RITY TO ACT ON BEHALF OF AND BIND	spreement*) is entering into this i Corporation, a i Corporation, a i Corporation, a i Corporation, a VITS-A BINDING intended for only and is not ate use or purpose. If the Renesas IDE software linetly or indirectly, in other button or nent to the terms of stalling, accessing, on of the Renesas sohalf of the entity imployer) and by this Agreement the right, power anc tity (if any) and NIND IN THIS IGHT, POWER AND
	< Ba	ck Next > Cancel	Install

Figure 21: Installation of e2 studio – Licenses

• Shortcuts

If you would like to create the shortcut for e2 studio, check the **In the application launcher** as shown below and click [Next >] to continue.

■ Renesas e² studio 2024-01.1 Setup ● Shortcuts ■ Summary Installing Results ■ Results
Device Families Extra Features Customise Features Additional Software Licenses Summary Installing

Figure 22: Installation of e2 studio – Shortcuts



- Summary
 - List of software to install is shown as follows. Click [Install] to start the installation.

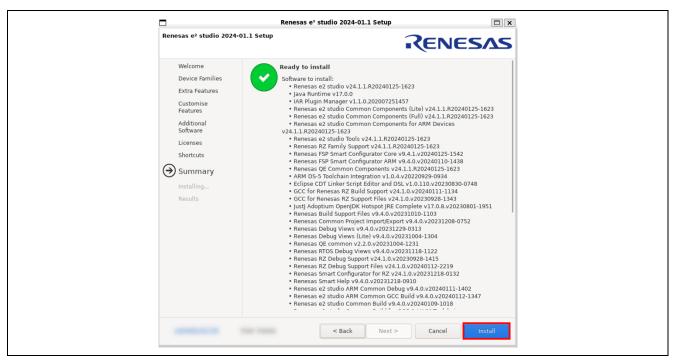


Figure 23: Installation of e2 studio – Summary

Installing...

Installation should be performed as shown below.

Renesas e² studio 2024-01.1 Setup Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Postalling Results
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing

Figure 24: Installation of e2 studio – Installing...

Results

Click [OK] to complete the installation.

Figure 25: Installation of e2 studio – Results

2.1.4.3 Add GNU ARM Embedded Toolchain on e2 studio IDE

1. Launch e2 studio

Specify the workspace path and launch the e2 studio.

Figure 26: e2 studio Launcher

2. Add Renesas Toolchains

Select [Help] -> [Add Renesas Toolchains] then you can see the [Preferences] window. After that Click [Add] button.

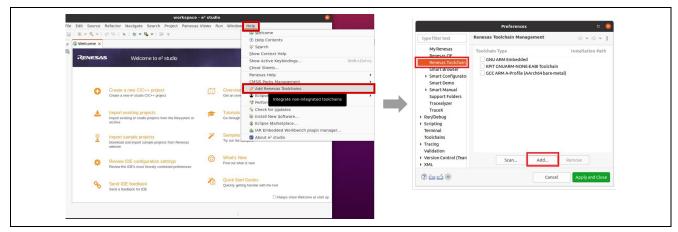


Figure 27: Add Renesas Toolchains



- 3. Add New Toolchain
 - Specify the path of the GNU ARM Embedded Toolchain. Click [OK]

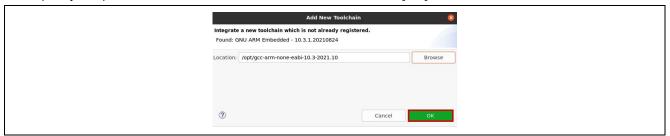


Figure 28: Add New Toolchain

4. Apply

Confirm that the GNU ARM Embedded is checked. Click [Apply and Close].

	Preferences	8
type filter text	Renesas Toolchain Management	⇔ ≠ ⇔ ≠ §
Renesas Toolchai	Toolchain Type	Installation Path
Smart Browser Smart Configurat Smart Demo Smart Manual Support Folders Tracealyzer TraceX Run/Debug Scripting Terminal TextMate Toolchains Toolchains	GNU ARM Embedded G10.3.1.20210824 KPIT GNUARM-NONE-EABI Toolchain XPack GNU ARM Embedded GCC ARM A-Profile (AArch64 bare-metal)	/opt/gcc-arm-none-e
Validation Version Control (Tea XML	Download Scan Add	Remove
? 2 2 0	Cancel	Apply and Close

Figure 29: Apply New Toolchain

2.2 FSP setup

2.2.1 Installation of FSP using Package Installer

Package Installer **RZG_FSP_Packs_v2.0.1.exe** is showcased at <u>here</u>. This section describes the procedure for installation. Note that it's for Windows Host PC only.

- 1. Quit e2 studio.
- 2. Invoke RZG_FSP_Packs_v2.0.1.exe.
- 3. Click [Next >] to start the installation.

Renesas Flexible Software Package (FSP) Installer This installer will install FSP packs and documentation into an existing e2 studio installation. The documentation will be available at clisTAL_DIRECTORY./fgp_documentation/rzg/ <version: The source for FSP can be found at https://gthub.com/renesas/rzg-fsp</version: 	(💮 Renesas RZ/G FSP v2.0.1 Setup	р —		×
This installer will installers packs and documentation into an existing e2 studio installation. The documentation will be available at <install_directory>/fsp_documentation/izg/<version. The source for FSP can be found at</version. </install_directory>			Renesas Flexible Software Pac (FSP) Installer	ckage	
The source for FSP can be found at			existing e2 studio installation. The documentation will be available at		
			The source for FSP can be found at	g/ <versio< td=""><td>N:</td></versio<>	N:

Figure 30: FSP Package Installer



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

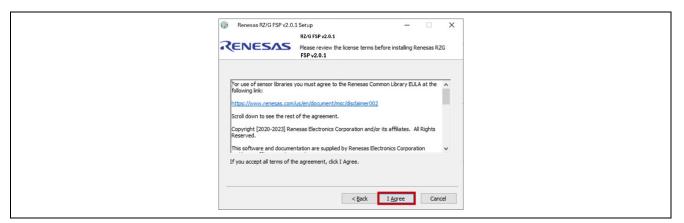


Figure 31: FSP License Term

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

Renesas RZ/G FSP v2.0.1 Setup - X Choose Install Location Choose the folder in which to install Renesas RZG FSP v2.0.1
The installation path must point to the root of the e2 studio installation (e.g. C:\Renesas\e2_studio). Please make sure e2 studio is closed before installation.
Browse to folder where e2 studio is installed
Space required: 38.3 MB Space available: 49.3 GB

Figure 32: Browse to the folder where e2 studio is installed

6. Click [Finish] to complete the installation.

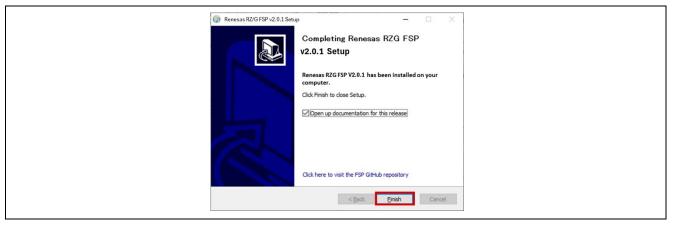


Figure 33: Completion of FSP Installation

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



2.2.2 Installation of FSP Packs using Package Zip file

No package installer is available for Linux Host PC. Thus, you need to install FSP with the zip file **RZG_FSP_Packs_v2.0.1.zip**. This section describes the procedure for installation.

- 1. Download RZG_FSP_Packs_v2.0.1.zip from here.
- 2. Extract the zip file to e2 studio installation directory. If it's successfully extracted, **rz_fsp/rzg/packs** should be placed at **<e2 stduio installation directory>/Internal/projectgen**.

File Hone Share Vex
← → ∨ ↑ 🖡 « renesas > e2,studio > internal > projectgen > rz,sp > rzg > packs ∨ D
Amizon/reeRIOS-Kernel.10.4.8+tp.2.0.1.pack
Am.CMSIS5.5.9.0+fsp.2.0.1.pack
Linaro.OpenAMP.10.0+tsp.20.1.pack
Renesas RZG2.0.1 pack
Renesas RZG_baremetal_bilrky-20.1.pack
RenesasRZG, baremetal, bliriky, cm33boot.2.0.1,pack
Renesas RZG, baremetal_minimal_cm33boot.2.0.1.pack
Renesas RZG_board_custom 2.0.1 pack
Renesas RZG_board_rzg2Lsmarc.20.1.pack
Renesas RZG, board, rzg2lc, smarc.2.0.1, pack
Renesas RZG_board_rzg2uLsmarc.2.0.1 pack
Renesas RZG_board_rzg3s_smarc.2.0.1,pack
Renepsis RZG_froertos, bilnioj 2.0.1 pack
Renesas RZG_freertos_bilnig_cm33boot.2.0.1 pack
Renetas RZG, freetos, minimalcm33boot.2.0.1,pack
Reneas RZG_mcu_zzg212.0.1.pack
RenesasRZG_mcu_zg2ul2.0.1 pack
RenessERZG_mcu_rzg3s2.0.1.pack
SEGGER/Link 7.94&pack

Figure 34: FSP Packs on e2studio installation directory

- 3. At the 1st invocation of e2 studio after the extraction, FSP should be automatically installed.
- 4. You can check if the installation is successfully done by the procedure below:
 - Click Help > CMSIS Packs Management > Renesas RZ/G

le Edit Source Refactor Navigate Search		Welcome Help Contents	
RENESAS Welcome to e ²	studio	% Search Show Context Help Show Active Keybindings Shold Active Keybindings Cheat Sheets	
0	Create a new C/C++ project Create a new e ¹ studio C/C++ project	Renesas Help CMSIS Packs Management CMSIS Packs Management	Renesas RZ/A Renesas RZ/G Renesas RZ/N Renesas RZ/T
÷	Import existing projects Import existing e ¹ studio projects from the filesys	Scheck for Updates	Renesas RZ/V
2	Import sample projects Download and import sample projects from Renes.	 lAR Embedded Workbench plugin manager About e² studio 	
0	Review IDE configuration settings Review the IDE's most flercely contested preferen	nes	
D	Open an existing file Open a file from the filesystem		
∞	Send IDE feedback		

Figure 35: CMSIS Packs Management (1)

• If FSP is successfully installed, 2.0.1 should be listed under FSP as shown below:

CMSIS Packs Ma	anagement - Renesas RZ	/G	
Packs location:			
/home/masao/.eclipse/com.renesas.p	olatform_1646807469/intern	al/projectgen,	/rz_fsp/rz
Show in System Explorer			
Available Packs		E E	i 🕂
Category	Version	Status	
▼ 🔀 FSP			
▶ III 2.0.1			
🕨 🗟 Generic			

Figure 36: CMSIS Packs Management (2)



3. Set up an SMARC EVK

3.1 RZ/G2L SMARC EVK

Below is an example of a typical system configuration.

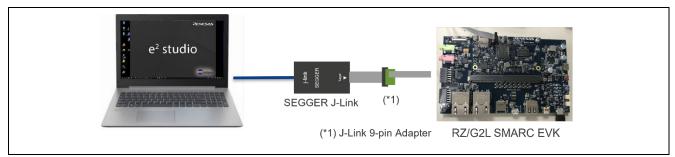


Figure 37: System Configuration Example – RZ/G2L SMARC EVK

3.1.1 Supported Debugger

SEGGER J-Link

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

3.1.2 Board Setup

3.1.2.1 Boot MODE

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

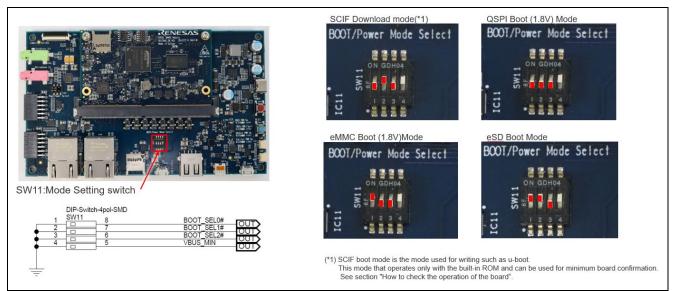


Figure 38: Boot MODE



3.1.2.2 JTAG connection

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

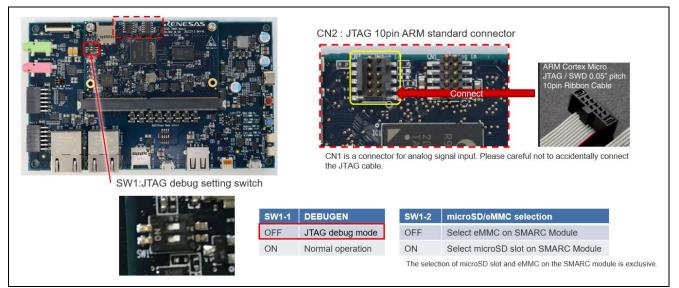


Figure 39: JTAG connection

Please note that RZ/G2L 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/

3.1.2.3 Debug Serial (console output)

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

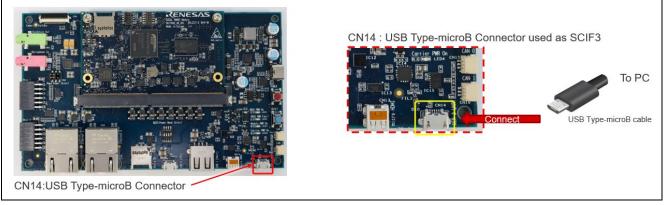


Figure 40: Debug Serial (console output)



3.1.2.4 Power Supply

Here are the 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)

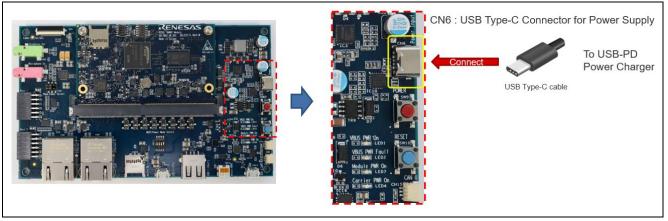


Figure 41: Power Supply

Connect USB-PD Power Charger to USB Type-C Connector. Then LED1(VBUS PWR On) and LED3 (Module PWR On) lights up. Press SW9 to turn on the power. Then LED4(Carrier PWR On) lights up.

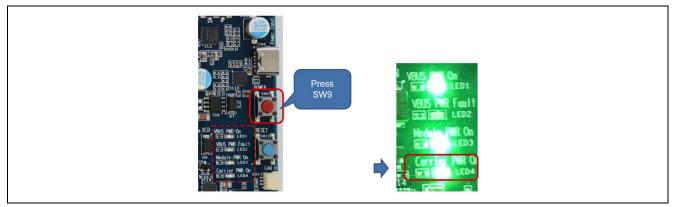


Figure 42: LED Status after Turning on EVK



3.1.2.5 How to check the operation of the board

First, check the board for problems. There are two ways to do this. Please check with either.

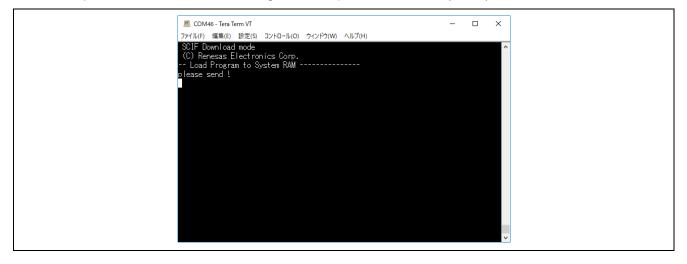
BOOT MODE: QSPI Boot (1.8V) Mode

If u-boot is written to the serial flash, When the power is turned on, the following will be output to the console (CN14).

🔟 COM46 - Tera Term VT	-	×	
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)			
NOTICE: BL2: v2.3():66ba71f35 NOTICE: BL2: Built : 17:54:22, Feb 10 2021 NOTICE: BL2: Booting BL31		^	
NOTICE: BL31: v2.3():66ba71f35 NOTICE: BL31: Built : 17:54:22, Feb 10 2021			
U-Boot 2020.10 (Feb 11 2021 - 10:19:36 +0000)			
CPU: Renesas Electronics E rev 16.15 Model: SMARC-RZG2L			
DRAM: 2 GiB			
MMC: sh-sdhi: 0, sh-sdhi: 1 Loading Environment from MMC OK			
In: scif@1004b800			
Out: scif@1004b800 Err: scif@1004b800			
Net: No ethernet found.			
Hit any key to stop autoboot: 0 Failed to load 'Image'			
Failed to load 'r9a07g0441-smarc-rzg21.dtb' Pad Linux APM04 Junan marial		_	
Bad Linux ARM64 Image magic! =>		~	

BOOT MODE: SCIF Download Mode

When the power is turned on, the following will be output to the console (CN14).





3.2 RZ/G3S SMARC EVK

Below is an example of a typical system configuration.

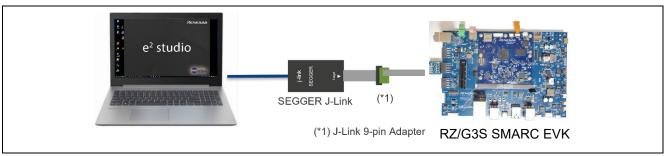


Figure 43: System Configuration Example – RZ/G3S SMARC EVK

3.2.1 Supported Debugger

SEGGER J-Link

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

3.2.2 Board Setup

3.2.2.1 Boot MODE

Set the boot mode using the two DIP SWITCHs shown in the figure below.

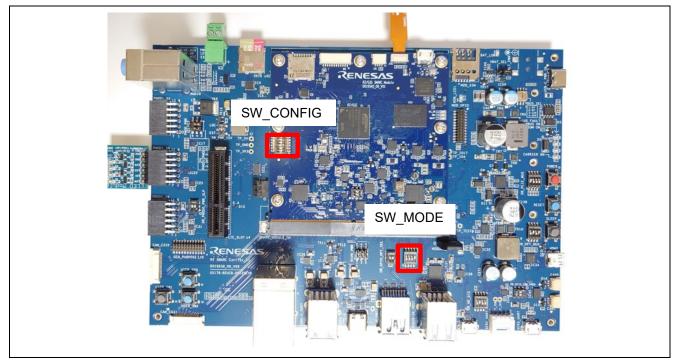


Figure 44: Boot MODE

In SW_CONFIG, select the boot CPU with the following settings.

1	2	3	4	5	6
OFF	OFF	ON	OFF	OFF	ON: CM33 boot
					OFF: CA55 boot



In SW_MODE, select the boot device with the following settings.

• Boot Mode 1: Booting from eMMC

ſ	1	2	3	4
	ON	OFF	OFF	ON

Boot Mode 2: Booting from serial flash memory

1	2	3	4
OFF	OFF	OFF	ON

• Boot Mode 3: Booting from the program downloaded through the serial communications with FIFO (SCIF)

1	2	3	4
OFF	ON	OFF	ON

3.2.2.2 JTAG connection

For JTAG connection, connect the included "SMARC JTAG ADAPTOR" to the board.

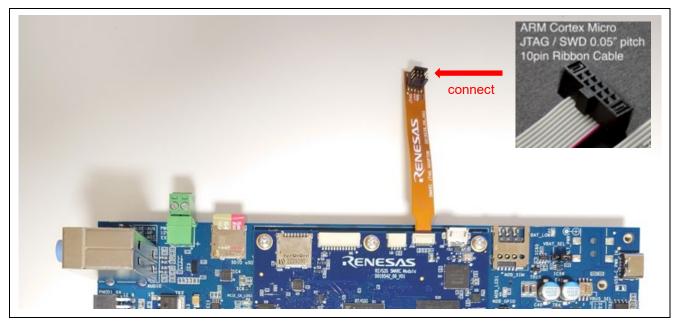


Figure 45: JTAG connection

Please note that RZ/G3S 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/



3.2.2.3 Debug Serial (console output)

Debug serial uses SER3_UART(SCIFA ch0). The baud rate is 115200bps.

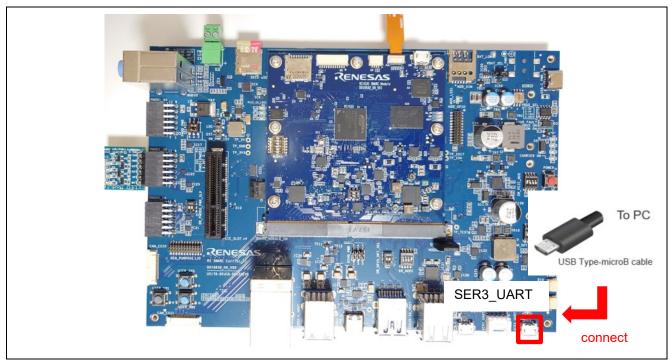


Figure 46: Debug Serial(console output)



3.2.2.4 Power Supply

Here are the 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)

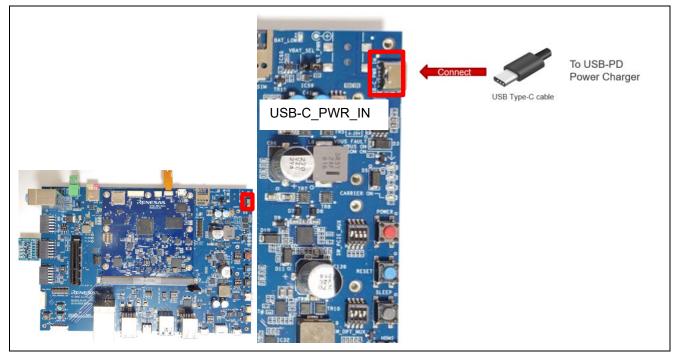


Figure 47: Power Supply

Connect USB-PD Power Charger to USB Type-C Connector. Then VBUS ON and SOM ON lights up. Press POWER to turn on the power. Then CARRIER ON lights up.

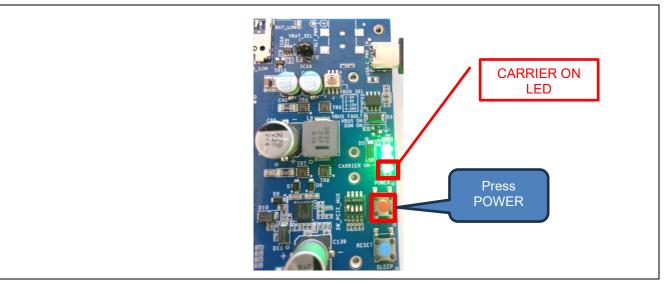


Figure 48: LED Status after Turning on EVK



3.2.2.5 How to check the operation of the board

First, check the board for problems. There are three ways to do this. Please check with any of them.

BOOT MODE1: Booting from eMMC

If u-boot is written to the eMMC, When the power is turned on, the following will be output to the console (SER3_UART).

COM4 - Tera Term VT	×		
		<	
Eile Edit Setup Control Window KanjiCode Help			
NOTICE: BL2: v2.7(release):2.7.0/g3s_1.0.0_rc3 NOTICE: BL2: Built : 14:34:44, Nov 2 2023		î	
NOTICE: BL2: Booting BL31 NOTICE: BL31: v2.7(release):2.7.0/g3s_1.0.0_rc3			
NOTICE: BL31: V2./(TetasB):2.1/V305_10.0/F00 NOTICE: BL31: Built : 14:34:44, Nov 2 2023			
U-Boot 2021.10 (Oct 25 2023 - 08:58:16 +0000)			
CPU: Renesas Electronics CPU rev 1.0			
Model: smarc-rzg3s			
DRAM: 896 MiB MMC: sd@11c00000: 0, sd@11c10000: 1, sd@11c20000: 2			
Loading Environment from MMCOK In: serial@1004b800			
In: seria101004b800 Out: seria101004b800			
Err: serial@1004b800 Net:			
Error: ethernet@11c30000 address not set.			
No ethernet found.			
Hit any key to stop autoboot: 0			
Card did not respond to voltage select! : -110 Card did not respond to voltage select! : -110			
Couldn't find partition mmc 1:1			
Can't set block device Card did not respond to voltage select! : -110			
Couldn't find partition mmc 1:1			
Can't set block device Bad Linux APM64 Imaze mazic!			
		~	

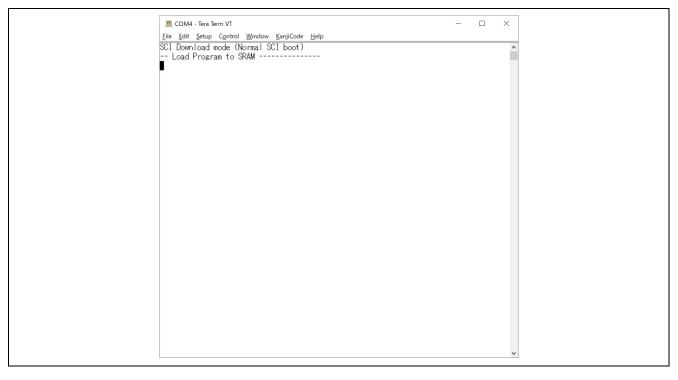
BOOT MODE2: Booting from serial flash memory

If u-boot is written to the serial flash, When the power is turned on, the following will be output to the console (SER3_UART).

COM4 - Tera Term VT	- 0	\times	
<u>Eile Edit Setup Control Window KanjiCode H</u> elp			
NOTICE: BL2: v2.7(release):2.7.0/g3s_1.0.0_rc3		^	~
NOTICE: BL2: Built : 14:34:44, Nov 2 2023 NOTICE: BL2: Booting BL31			
NOTICE: BL31: v2.7(release):2.7.0/g3s_1.0.0_rc3			
NOTICE: BL31: Built : 14:34:44, Nov 2 2023			
U-Boot 2021.10 (Oct 25 2023 - 08:58:16 +0000)			
OPU: Renesas Electronics OPU rev 1.0			
Model: smarc-rzg3s			
DRAM: 896 MiB			
MMC: sd@11c00000: 0, sd@11c10000: 1, sd@11c20000: 2			
Loading Environment from MMC OK			
In: serial@1004b800 Out: serial@1004b800			
Err: serial@1004b800			
Net:			
Error: ethernet@11c30000 address not set.			
No ethernet found.			
Hit any key to stop autoboot: 0			
Card did not respond to voltage select! : -110			
Card did not respond to voltage select! : -110			
Couldn't find partition mmc 1:1 Can't set block device			
Card did not respond to voltage select! : -110			
Couldn't find partition mmc 1:1			
Can't set block device			
Bad Linux ARM64 Image magic! => ^			
-/			
		~	



BOOT MODE3: Booting from the program downloaded through the serial communications with FIFO (SCIF) When the power is turned on, the following will be output to the console (SER3_UART).





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>

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

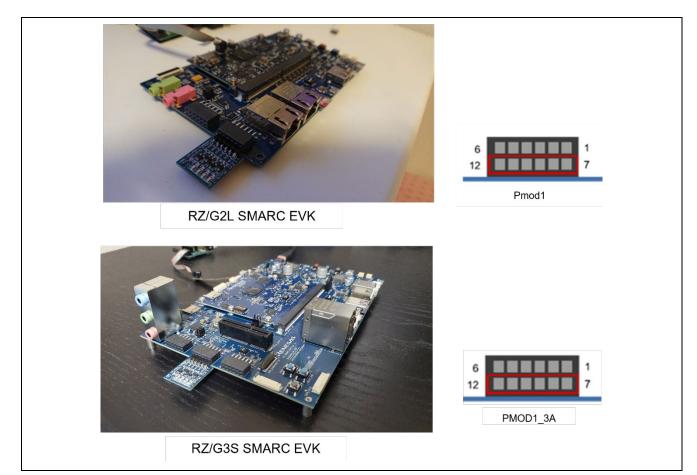


Figure 49: Connection Pmod LED module (410-076)



4.3 Create a New Project for Blinky

The creation and configuration of an RZ/G C/C++ FSP Project is the first step in the creation of an application. The base RZ/G 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.

📴 work - e² studio				
File Edit Source Refactor Navigate	Search Projec	t Re	nesas Views Run Window Help	
New	Alt+Shift+N >		Renesas C/C++ Project	> - 📑
Open File		CÅ	Makefile Project with Existing Code	
Open Projects from File System		C	C/C++ Project	(> -)
Recent Files	>		Project	

Figure 50: New C/C++ Project

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

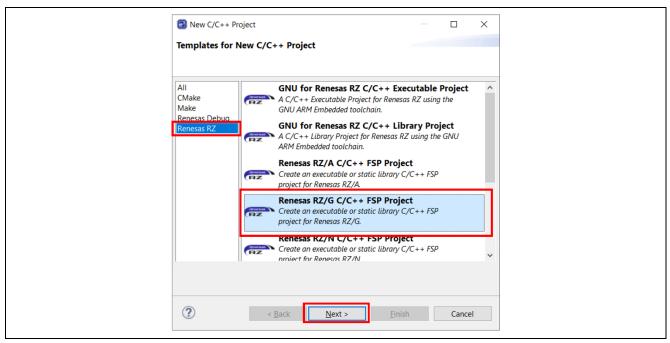


Figure 51: Renesas RZ/G C/C++ FSP Project

- 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 Project Name and Location Project name Blinky Use default location Location: C:\workspace\Blinky You can download more Renesas packs here	Browse
Blinky Use default location Location: C:\workspace\Blinky	B <u>r</u> owse
Use default location	Browse
Location: C:\workspace\Blinky	Browse
You can download more Renesas packs here	

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

5. Select the board support package by selecting the name of your board from the Device Selection dropdown list. Select **GNU ARM Embedded** in Toolchains and version is **10.3.1.20210824** and Click **Next**.

📴 Renesas R	Z/G C/C++ FSP Project			
Renesas RZ/	/G C/C++ FSP Project			
Device and T	Tools Selection			4
Device Select	tion			
FSP Version: 2.0.1		Board Description		
Board:	RZ/G3S Evaluation Kit (SMARC)			
Device:	R9A08G045S33GBG_CM33	Device Details		
Core:	Core 1(CM33) ~	TrustZone	No	
Language:		Pins Processor	359 Cortex-M33	
Toolchains		Debugger		
GNU ARM Embedded		J-Link ARM		~
10.3.1.202104	824 ~			
?		< <u>B</u> ack	Next > Einish	Cancel

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



6. Select the **build artifact** and **RTOS**. Be sure that **Secure** must be chosen at the **Sub-core start state** on the current version. Otherwise, the created project can't be built successfully.

Renesas RZ/G C/C++ FSP Project		
Build Artifact, RTOS Selection and Sub-Core Selection		1
Build Artifact Selection	RTOS Selection	
 Executable Project builds to an executable file 	No RTOS	~
 Static Library Project builds to a static library file 		
 Executable Using an RZ/G Static Library Project builds to an executable file Project uses an existing RZ/G static library project 		
Sub-core start state		
 Secure Start sub-core in secure state 		
 Non-secure Start sub-core in non-secure state 		

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

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

Renesas RZ/G C/C++ FSP Project		— C	X
Renesas RZ/G C/C++ FSP Proje	ect		
Project Template Selection			
Project Template Selection			
	oject that includes BSP and will blink LEDs if available. e C runtime environment.	This project will initialize cl	ocks,
Bare Metal - Bare metal FSP pro environment. [Renesas.RZG.2.0.0.	oject that includes BSP. This project will initialize clock	s, pins, stacks, and the C run	time
Code Generation Settings Use Renesas Code Formatter			

Figure 55 : e2 studio Project Configuration window (part 4)

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

Summary			Generate Project Content
Project Summary	y		
P			RENESAS
Board: Device:	RZ/G3S Evaluation Kit (SMARC)		
Core:	R9A08G045S33GBG_CM33		
Toolchain:	Core 1(CM33) GCC for Renesas RZ		
Toolchain Version:			
FSP Version:	2.0.1		
Project Type:	Flat		
Location:	C:/workspace/Blinky 😔		
Selected software c	omponents		
Simple application	that blinks an LED. No RTOS included.	v2.0.1	
Board Support Pa	ckage Common Files	v2.0.1	
I/O Port		v2.0.1	
General Timer		v2.0.1	
Quad Serial Peripl	heral Interface Flash on Expanded Serial Peripheral Interface	v2.0.1	
Arm CMSIS Versio	on 5 - Core (M)	v5.9.0+fsp.2.0.1	
Board support pac	kage for R9A08G045S33GBG_CM33	v2.0.1	
Board support pac	kage for RZG3S	v2.0.1	
Board support pac	kage for RZ/G3S - FSP Data	v2.0.1	
Evaluation Kit RZ/	G3S Support Files (RZ/G3S)	v2.0.1	

Figure 56 : e2 studio Project Configuration tab

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



4.3.1 Details about the Blinky Configuration

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

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

4.3.2 Configuring the Blinky Clocks

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

4.3.3 Configuring the Blinky Pins

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

4.3.4 Configuring the Parameters for Blinky Components

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

- r_gtm
- r_ioport

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

4.3.5 Where is main()?

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

4.3.6 Blinky Example Code

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

The application performs the following steps:

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



4.4 Build the Blinky Project

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

There are three ways to build a project:

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

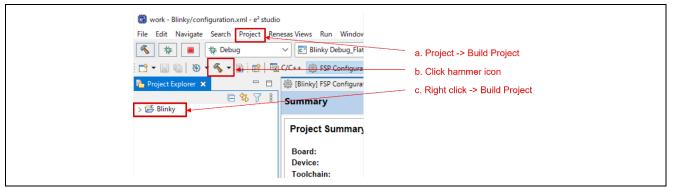


Figure 57 : e2 studio Project Explorer window

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

Pin Conflicts
CDT Build Console [Blinky]
'arm-none-eabi-gcc -mthumb -mcpu=cortex-m33+nodsp+notp -tdiagnostics-pa
'arm-none-eabi-gcc -mthumb -mcpu=cortex-m33+nodsp+nofp -fdiagnostics-pa
arm-none-eabi-gcc @"Blinky.elf.in"
arm-none-eabi-objcopy -O srec "Blinky.elf" "Blinky.srec"
arm-none-eabi-sizeformat=berkeley "Blinky.elf"
text data bss dec hex-filename
4812 2072 16784304 16791188 1003694 Blinky.elf
05:35:14 Build Finished. 0 errors, 0 warnings. (took 7s.823ms)

Figure 58 : e2 studio Project Build console



4.5 Debug the Blinky Project

4.6 Debug prerequisites

To debug the project on a board, you need

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

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

4.7 Debug steps

To debug the Blinky application, follow these steps:

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

un	Window Help	
R	Renesas Device Partition Manag	ger
	TraceX	>
ð Ti	Tracealyzer	>
R	Run	Ctrl+F11
6 D	Debug	F11
R	Run History	>
R	Run As	>
R	Run Configurations	
D	Debug History	>
≽ D	Debug As	>
D	Debug Configurations	
E	External Tools	>

Figure 59 : e2 studio Debug icon

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

1	🏘 🕶 🏊 📲 🕺 🖉 🖉 🛹 📼 🗖
	(no launch history)
1	Debug As >
	Debug Configurations
	Organize Favorites

Figure 60 : e2 studio Debugger Configurations selection option

2. Select your debugger configuration in the window. If it is not visible, then it must be created by clicking New icon in the top left corner of the window. Once selected, the **Debug Configuration** window displays the **Debug configuration** for your **Blinky** project.

Cost, mange, al dra norffenstes Cost,	Cash, nang, ad na cashpades	Cinc. Hungs, sid an estimation		
Crash, nange, ad van antigendees	Cosh, nange, sei an coshprotes:	Cinc. Hungs, sid an estimation		
Contract Contrect Contract Contract Contract Contract Contract Contract Contrac	Version Control C	Image: Section None Status Section None Image: Section None Status Section None Image: Section None None None None Image: Section None	Debug Configurations	- n x
Description Description Description Image: Control of the state of t	Importance Distance Importance Importance Importance	Constraints Constrain	Create, manage, and run configurations	- *
Description Description Description EXESTINGT Balance Balance Exesting Balance Balance	Description Description Description EXEST: File Bit Bi	Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Directore Dir	CBBBRNB7+	Name Sinty Jetus
C bedra Kapitalian 1 Oktowe 1 Ok	CO- fine deglinite Color deglinite	Construction of the second sec		
B12 Streets Streets Streets	B1 Standowski klaudyje Uriek Apprenni B2 Standowski klaudyje Uriek Apprenni B2 Backowski klaudyje Uriek Apprennik B2 Backowski klaudyje Uriek Apprennik	Bit Science, Balancia Bit Science, Bal	C/C++ Remote Application	
213 Gradua Okuping Mong Exciption 213 Gradua Okuping Mong Exciption 213 Gradua Okuping Mong Exciption 213 Gradua Okuping Texture 213 Gradua Okuping Texture 213 Gradua Okuping Construct and add	B03 Evolus Zhanging Root Teaching and the second sec	Diff State Scherger Street Material Material Material Bill State Scherger Street Fall Street Scherger Fall Street Scherger Fall Street Scherger Constrained Fall Street Scherger Fall Street Scherger Of scherger Scherger Scherger Distances Fall Street Scherger Of scherger Scherger Of scherger Scherger Scherger Scherger Distances Scherger Scherger Of scherger Scherger Of scherger Scherger Of scherger Scherger	C 608 Bardware Debugging	
En land Application End of Provided Hone's are Mary Service and Application End of Provided Hone's are Mary End of Pr	Toka Applications Fach of Provided March Stars Nature And Stranding And Andre Sans Nature Andre Sans Nature Andre Sans Nature Tokate and table Orande and table Orande and table Orande and table	If Interference	GDB Simulator Debugging (RH050)	
C Senote lea Application Indi Configuence: Use Active ✓ Senote Coll Headware Debugging Ornetie auto build Of matter auto build	C. Semote les àpplication Partir Certig minim: Une Active ✓ Semote Coll Headware Debugging Ornable auto build Ornable auto build	If were included backgroup Intelling genesis (backgroup) >>>>>>>>>>>>>>>>>>>>>>>>>>>>	T Inva Application	
Binty Debug Obtable auto build Obtable auto build	Binty Debug Obtable auto build Obtable auto build	T Simily Sings	C. Sencte Java Application	
	kulturer zweiten gestelliet (of itra)		El Stinky Debug	
Filer motified to of 16 items Reveil Apply	File matched 53 of 15 koms Apply		3	Delway Cover
The second		(2) Disc. Con	•	

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



- 3. Select the debug configuration for the generated project and select the **Debugger** tab.
- 4. Click **Debug** to begin debugging the application.
- 5. Extracting **RZ Debug**.

Configuring GD8	Progress Information -	□ ×
Configuring GDB	Preparing launch delegate	
	Configuring GDB	
Cancel Details >>		

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

4.8 Details about the Debug Process

In debug mode, e2 studio executes the following tasks:

- 1. Downloading the application image to the microprocessor and programming the image to the internal and/or external memory.
- 2. Setting a breakpoint at main().
- 3. Setting the stack pointer register to the stack.
- 4. Loading the program counter register with the address of the reset vector.
- 5. Displaying the startup code where the program counter points to.

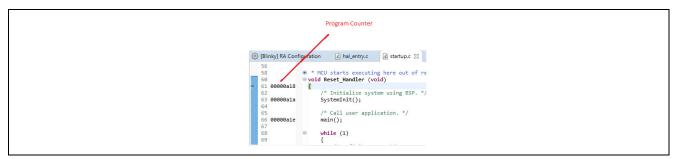


Figure 63 : e2 studio Debugger memory window

4.9 Run the Blinky Project

While in Debug mode, click **Run > Resume** or click on the **Play** icon twice.



Figure 64 : e2 studio Debugger Play icon

The LED on the Pmod LED should now be blinking.



5. FSP application launch with e2 studio

5.1 Create a Project

5.1.1 What is a Project?

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

- 1. Create a Project
- 2. Configuring a Project

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

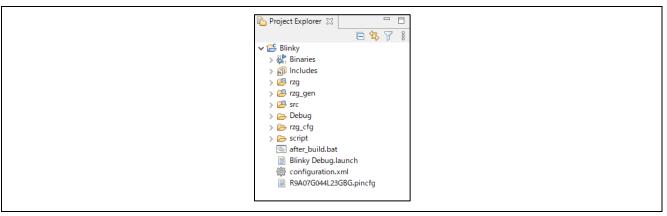


Figure 65 : e2 studio Project Configuration file

Double-click on the configuration.xml file to open the RZ MPU Project Editor. To edit the project configuration, make sure that the **FSP Configuration** perspective 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.



Note: Whenever the RZ project configuration (that is, the configuration.xml file) is saved, a verbose RZ Project Report file (rzg_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.

Project Explorer 🛛 🗖 🗖	📄 rzg_cfg.txt 🔀	
□ 🕏 🍸 🕴	1	FSP Configuration
✓ 😂 Blinky	2	Board "RZ/G2L Evaluation Kit (SMARC)" R9A07G044L23GBG
> 🖑 Binaries	4	part number: R9A07G044L23GBG
> 🔊 Includes	5	rom_size_bytes: 0
> 😕 rzg	6	ram_size_bytes: 131072
> 😕 rzg_gen	7	package_style: LFBGA
> 🐸 src	9	package_pins: 456
> 🗁 Debug	10	RZG2L
> 🗁 rzg_cfg	11	series: 2
> 🗁 script	12	prov cardle
after_build.bat	14	RZG2L Family RZ/G2L Common
Blinky Debug.launch	15	Secure stack size (bytes): 0x200
i configuration.xml	16	Main stack size (bytes): 0x200
R9A07G044L23GBG.pincfg	17	Heap size (bytes): 0
📄 rzg_cfg.txt	19	MCU Vcc (mV): 3300 Parameter checking: Disabled
	20	Assert Failures: Return FSP_ERR_ASSERTION

Figure 67 : 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.

ummary			Generate Project Content
Project Summary			
Board Support Pac I/O Port General Timer Arm CMSIS Versio Evaluation Kit RZ/C Board support pac Board support pac	2.0.1 Flat C:/workspace/Blinky - mponents that blinks an LED. No RTOS included. kage Common Files n 5 - Core (M) S2L Support Files (RZ/G2L) kage for R9A07G044L23GBG_CM33	v2.0.1 v2.0.1 v5.9.0+fsp.2.0.1 v2.0.1 v2.0.1 v2.0.1 v2.0.1	RENESAS
Supp	ort		

Figure 68 : 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.

8	work - e² studio					
File	Edit Source Refactor Navigate	Search Projec	t R	enesas Views Run Window Help		
	New	$Alt+Shift+N \succ$		Renesas C/C++ Project	>	-
	Open File		C ²	Makefile Project with Existing Code		
	Open Projects from File System		C	C/C++ Project		÷ 4
	Recent Files	>		Project		

Figure 69 : New RZ MPU Project

2. Then click on the Renesas RZ/G C/C++ FSP Project template for the type of project you are creating.

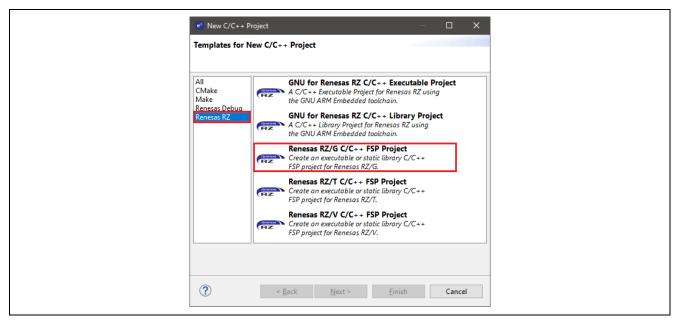


Figure 70 : New Project Templates

3. Select a project name and location.

Renesas RZ/G C/C++ FSP Project —	o x	
Renesas RZ/G C/C++ FSP Project	-	
Project Name and Location	1	
Project name		
Blinky		
Use default location		
Location: C:#works_RZG_V#work#Blinky	Browse	
Choose file system: default \vee		
You can download more Renesas packs here		
(?) (Back Next > Finish	Cancel	

Figure 71 : RZ MPU Project Generator (Screen 1)

4. Click Next.

5.1.2.1 Selecting a Board and Toolchain

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

- 1. Select the **FSP version**.
- 2. Select the **Board** for your application. You can select an existing RZ MPU Evaluation Kit or select **Custom User Board** for any of the RZ MPU devices with your own BSP definition.
- 3. Select the **Device**. The **Device** is automatically populated based on the **Board** selection. Only change the **Device** when using the **Custom User Board (Any Device)** board selection.
- 4. Select the **Core**. You can select Core 1(CM33) or Core 2(CM33_FPU) if you selected RZ/G3S for the **Device**.
- 5. To add threads, select RTOS, or No RTOS if an RTOS is not being used.
- 6. The Toolchain selection defaults to GNU Arm Embedded.
- 7. Select the Toolchain version. This should default to the installed toolchain version.
- 8. Select the **Debugger**. The J-Link Arm Debugger is preselected.
- 9. Click Next.

Renesas RZ/G C/C++ FSP Project	– 🗆 X
Renesas RZ/G C/C++ FSP Project	
Device and Tools Selection	
Device Selection	
FSP Version: 2.0.1	Board Description
Board: RZ/G3S Evaluation Kit (SMARC) ~	
Device: R9A08G045S33GBG_CM33	
Core: Core 1(CM33)	Device Details TrustZone No
Language: C C++	Pins 359
Language. Geoloci	Processor Cortex-M33
Toolchains	Debugger
GNU ARM Embedded	J-Link ARM 🗸
10.3.1.20210824	
(?)	< Back Next > Einish Cancel

Figure 72 : RZ MPU Project Generator (Screen 2-1)

If Core 2(CM33_FPU) is selected in procedure 4, you need to select the preceding project. To select the preceding project when creating the Core 2(CM33_FPU) project, it is required to prepare Core 1(CM33) before Core 2(CM33_FPU) project creation.

Renease RZ/G C/C++ FSP Project Preceding Project or Smart Bundle Selection Preceding Project: Binay Choose this option if you have access to the project source code of the preceding processor core or security context. O Smart Bundler Resolved location: Workspace Preceding Project: Binay Preceding Project: Binay Preceding Project: Statistics option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project: Board R2/G35 Evaluation Kit (SMARC) Device RAMGROMS336BG_CM33 Core CM33 Zones CM33_5
Preceding Project: Ellinky: Choose this option if you have access to the project source code of the preceding processor core or security context. Of smart Bundle: Resolved location: Workspace. File System. Variables. Orhoose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details FSP version 20.1 Toolchain GNU ARM Embedded Toolchain Project R24(2035 Evaluation Kit (SMARC) Device R34(68004533366 c.M33 Core C M33
Choose this option if you have access to the project source code of the preceding processor core or security context. Origonal Bundle: Resolved location: Workspace. File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. FSP version GNU ARM Embedded Toolchain GNU ARM Embedded Toolchain version R2/G35 Evaluation Kit (SMARC) Derive R94060053066_CM33 Core Ch83
Smart Bundle: Resolved location: Workspace. File System Variables. Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details FSP version 20.1 Toolchain GNU ARM Embedded Toolchain version 10.3.1.20210824 Board R2/G35 Evaluation Kit (SMARC) Device R9A080050533665_CM33 Core CM33
Resolved location: Workspace. File System. Variables. Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. FSP version 2.0.1 Toolchain GNU ARM Embedded Toolchain version 10.3.1.20210824 Board R 2/G35 Evaluation Kit (SMARC) Device R8A086045533665.CM33 Core CM33
Workspace File System Vanables Choose this option If you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details File System Vanables FSP version 2.0.1 Conclusion GNU ARM Embedded Toolchain version 10.3.1.202.10824 Board Packadous Sequences of CMU and Conclusion Strt (SMARC) Derice RAA080045333665_CMU and Conclusion Strt (SMARC) Derice Conce CMU and Conclusion Conce CMU and Conclusion Conce CMU and Concentration Strt (SMARC) Derice RAA08004533365_CMU and Concentration Strt (SMARC) Derice RAA08004533365 Concentration Strt (SMARC) Concentration Strt (SMARC) Derice Concentration Strt (SMARC) Concentration Strt (SMARC)
Preceding Project/Smart Bundle Details FSP version 2.0.1 Tochchain GNU ARM Embedded Tochchain version 10.3.120210824 Board R2/G3S Evaluation Kit (SMARC) Device R94080453086_CM33 Core CM83
Toolchain GNU ARM Embedded Toolchain version 10.3.1.20210824 Board RZ/G35 Evaluation Kit (SMARC) Device R9A060645533GBG_CM33 Core CM33

Figure 73 : RZ MPU Project Generator (Screen 2-2)



5.1.2.2 Selecting a Project Template

In the next window, select the build artifact, **Sub-core start state** and **RTOS**. Be sure that you select **Secure** as **Sub-core start state** in the current version.

Renesas RZ/G C/C++ FSP Project	_	п×	
Renesas RZ/G C/C++ FSP Project Build Artifact, RTOS Selection and Sub-Core Selection		\bigcirc	
Build Artifact Selection	RTOS Selection		
 Executable Project builds to an executable file 	No RTOS	~	
 Static Library Project builds to a static library file 			
Sub-core start state Secure • Start sub-core in secure state Non-secure • Start sub-core in non-secure state			
? Sack	Next > Finish	Cancel	

Figure 74 : 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/G MPU Pack. Once you have selected the appropriate template, click **Finish**.

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

** Remeas RZ/G C/C++ FSP Project Remeasa RZ/G C/C++ FSP Project	Renesas RZ/G (/C++ FSP Project Renesas RZ/G (/C++ FSP Project
Project Template Selection	Project Template Selection
Project Template Selection	Project Template Selection Image: Selection
Code Generation Settings	Code Generation Settings
Einish Cancel	(Back Next> Finish Cancel

Figure 75 : 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.

			Generate Proj	
Project Summary	1			^
Board:	RZ/G3S Evaluation Kit (SMARC)		RENESAS	
Device:	R9A08G045S33GBG_CM33			
Core:	Core 1(CM33)			
Toolchain:	GCC for Renesas RZ			
Toolchain Version:	10.3.1.20210824			
FSP Version:	2.0.1			
Project Type:	Flat			
Location:	C:/workspace/Blinky 🔄			
Selected software c	omponents			
Simple application	that blinks an LED. No RTOS included.	v2.0.1		
	ckage Common Files	v2.0.1		
I/O Port		v2.0.1		
General Timer		v2.0.1		
	neral Interface Flash on Expanded Serial Peripheral Interface			
Arm CMSIS Versio		v5.9.0+fsp.2.0.1		
Board support pac Board support pac	kage for R9A08G045S33GBG_CM33	v2.0.1 v2.0.1		
	kage for RZ/G3S - FSP Data	v2.0.1		
	G3S Support Files (RZ/G3S)	v2.0.1		

Figure 76 : RZ MPU Project Editor and available editor tabs



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

- With the **Summary** tab, you can see all they key characteristics of the project: board, device, toolchain, and more.
- With the **BSP** tab, you can change board specific parameters from the initial project selection.
- With the **Clocks** tab, you can refer to the MPU clock settings for your project. In the case of CM33 cold boot, you can configure the MPU clock settings for your project.
- With the **Pins** tab, you can configure the electrical characteristics and functions of each port pin.
- With the Interrupts tab, you can add new user events/interrupts.
- With the **Stacks** tab, you can add and configure FSP modules. For each module selected in this tab, the **Properties** window provides access to the configuration parameters, interrupt selections.
- The Components tab provides an overview of the selected modules. Although you can also add drivers for specific FSP releases and application sample code here, this tab is normally only used for reference.

The functions and use of each of the supported tabs is explained in detail in the next section. Please note that RZ/G MPU Pack doesn't support **Event Links** tab and so, that tab is grayed out as shown above.

5.1.3 Duplication of Resources

In the case of RZ/G3S Core 2(CM33_FPU) project, duplicate resources are indicated as red character in **Stacks** tab when using resources that are used in the linked Core 1(CM33) project.

The following image is the example that both of Core 1(CM33) and Core 2(CM33_FPU) projects are created with Blinky template. The duplication of r_gtm is indicated in **Stacks** tab. To avoid this duplication, please change the channel resource in **Properties** of r_gtm.

Stacks Configuration		Generate Project Content
Threads 🚯 New Thread 😰 Remove 🛅	HAL/Common Stacks	🐑 New Stack > 🚔 Extend Stack > 🎣 Remove
V ■ HAUCommon Joport (U) Port (r_Joport) P (J.Ime2 Timer (J.gm) ⊕ (J.gmpi0 CSPI (r_sp)_qsp) ⊕ (J.gmpi0 CSPI (r_sp)_qsp) ♥ Summary BSP [Clocks Pins Interrupts [Svent Links ● Stacks Comp Summary BSP [Clocks Pins Interrupts [Svent Links ● Stacks Comp Problems □ Properties × ● Smart Browser ● Console	This instance may be refer Error: Peripheral 'GTM2' is a	g.gspi0 QSPI (r_sspi.gspi) of this color indicates 'g_timer2 Timer (r_gtm)' is a Module instance. mocd by one other module instance only. allocated within a preceding project or smart bundle
g_timer2 Timer (r_gtm)		
Settings Property Common Parameter Checking Module g_time? Timer (r_gtm) General Name Channel Mode Period Period Unit Count Source Interrupts		Value Default (BSP) a timer2 GTM2 Periodic 8 Hertz POCLK

Figure 77 : Duplication of resource between Core 1(CM33) and Core 2(CM33_FPU) projects



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

Bilinky] FSP Configuration × Summary	Generate Project Co	ontent
Board Support Package Common Files 100 Pert General Timer Quad Serial Peripheral Interface Flash on Expanded Serial Peripheral Interface Am CMSIS Version 5 - Core (M) Board support package for R2A00604553306BG_CM33 Board support package for R2A05	v201 v201 v201 v201 v201 v201 v201 v500fbg201 v201 v201 v201	^
	v201	~

Figure 78 : 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.

Su Properties 🗙 🕵 Problems 🏟 Smart Browser	immary BSP Clocks Pins Interrupts Event Links Stacks Components	1 8 - D
Properties X & Problems & Smart Browser		
RZ/G2L Evaluation Kit (SMARC)		
Settings Property	Value	
✓ R9A07G044L23GBG		
part_number	R9A07G044L23GBG	
rom_size_bytes	0	
ram_size_bytes	131072	
package_style	LFBGA	
package_pins	456	
✓ RZ/G2L 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	
✓ RZG2L		
series	2	

Figure 79 : 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. e2 studio checks the entry fields to flag invalid entries. For example, only valid numeric values can be entered for the stack size.

When you click the **Generate Project Content** button, the BSP configuration contents are written to rzg_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 the clock configuration is carried out on the main core (CA55) in advance when CM33 work as sub core. Thus, clocks configuration here must align with the settings on CA55.

In the case of CM33 cold boot, BSP will configure each clock setting in start-up process according to content of **Clocks** tab. If a clock setting is invalid, the offending clock value is highlighted in red. It is still possible to generate code with this setting, but correct operation cannot be guaranteed. In the figure below, the xSPI clock SPI0CLK has been changed so the resulting clock frequency is 400 MHz instead of the required less than 267 MHz. This parameter is colored red.

德 *[Blinky] FSP Configuration	n X	- 8
Clocks Configuration	n	Generate Project Content
		Image: Restore Defaults > OCOCLK ~33.333MHz > OC1CLK ~16.667MHz
ų		SPIOCLK 400MHz SPI1CLK 200MHz
		> SOCLK 12kHz
		> I3CLK 250MHz > HPCLK 250MHz
	Div /3	>OSC2 8MHz ✓

Figure 80 : Configuration Clocks tab

When you click the **Generate Project Content** button, the clock configuration contents are written to rzg_gen/bsp_clock_cfg.h. This file will be created if it does not already exist.

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



5.2.4 Configuring Pins

The pins tab provides flexible configuration of the MPU's pins. As many pins can provide multiple functions, they can be configured on a peripheral basis. For example, selecting a serial channel via the SCIF peripheral offers multiple options for the location of the receive and transmit pins for that module and channel. The location and function of the pins are shown in the **FSP Visualization** view. For more information on the function and color coding of the pins, please check the Legend in the **FSP Visualization** view.

Pin Configuration			Gen	erate Project Content		Type pin function	Pin Func 👻
Select Pin Configuration	🔋 Load DTSi configur	ation 🔚 Export to CSV fi	ile 🔝 Configure	Pin Driver Warnings			
RZG3S-SMARC.pincfg	 Manage configuration 	ations					0.00
Generate data: g_bsp_pin_cf	g						
Pin Selection $\models = \downarrow_z^a$	Pin Configuration			😲 Cycle Pin Group			000000
Type filter text	Name	Value	Lock	Link			00
> peripheral:I3C ^	Pin Group Selection	Mixed					
> v peripheral:RIIC	Operation Mode	Disabled				0 0 0 0 0 0	· • •
> v peripheral:RIC	✓ Input/Output				· 😔 🔁	0 0 0 0 0	0000000000
 ✓ peripheral:SCIF 	SCIF4_RXD	None	- mil	\Rightarrow			
✓ ✓ peripheral:SCIF ✓ SCIF0	SCIF4_SCK	None	TT.		- 20		
SCIF0	SCIF4_TXD	None	dî .			్ల్ల్ల్ల్ల్	
SCIF1 SCIF2					- 😁 \cdots		
SCIF2 SCIF3					· •• • 💀 🔂 🖸 😔 🖯		• • • • • • • • • • • •
SCIF3						🕞 RZG3S 😁 🕞	
SCIF4 SCIF5	<			>			
> peripheral:SCIg	Module name: SCIF4				-000 0		
· · · · · · · · · · · · · · · · · · ·					· 🛥 🖂 😁		9 🕺 🛛 🖓 -
< >	Usage: Serial Comm	unication Interface with Fl	FO		00000000		
Pin Function Pin Number							

Figure 81 : 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 GPT, 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 GPT.

*[Blinky] FSP Configuration × Pin Configuration			Gene	o erate Projec	t Content
Select Pin Configuration	👔 Load DTSi configur	ation 🔛 Export to CSV file	Configure	Pin Driver	Namings
RZG3S-SMARC.pincfg	Manage configur fg	ations			
Pin Selection $\models \models \models \downarrow_z^a$	Pin Configuration			😲 Cycle P	n Group
Type filter text > peripheral320 > peripheral321 > @ > @ > @ > @ > @ > @ > @ > @ > CF10 SCIF1 SCIF1 SCIF3 SCIF4 SCIF5 SCIF5	Name Pin Group Selection Operation Mode V Input/Output SCIF1_RTS_N SCIF1_RTS_N SCIF1_RTS_N SCIF1_SCK SCIF1_TXD <	Value Mixed Custom None One None None None	e e e e e e e e e e e e e e e e e e e	Link	>
> peripheral:SCIg	Module name: SCIF1 Usage: Serial Comm	nunication Interface with FIFO			
Pin Function Pin Number					

Figure 82 : e2 studio Pin Configurator

When you click the **Generate Project Content** button, the pin configuration contents are written to: rzg_gen\pin_data.c. 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.



In the case of versions earlier than RZ/G FSP v2.0.0, It does not support **Pins** tab and If user would like to use I/O port, I/O Port setting should be applied to "src/pin_data.c" manually. For details on I/O Port setting and how to apply the setting of "src/pin_data.c" to **Pins** tab, please refer to <u>Setting GPIO with Flexible</u> <u>Software Package</u>.

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.

(Blinky) FSP Configuration 23		- 8
Stacks Configuration		Generate Project Content
Threads 🔹 New Thread 🔊 Remove 😑	HAL/Common Stacks	Stack > 🔮 Extend Stack > 👔 Remove
 ✓ MHAL/Common ④ g_ioport I/O Port Driver on r_ioport ④ g_timer0 Timer Driver on r_gtm 	g joport I/O Port Driver on r_joport	er
Objects 🕢 New Object > 😥 Remove		
Summary BSP Clocks Pins Interrupts Event Link	nks Stacks Components	
🔳 Properties 🗙 🖹 Problems 👒 スマート・ブラウ	0 7 -	1 8 - D
g_timer0 Timer Driver on r_gtm		
Settings Property	Value	
✓ Common		
Parameter Checking	Default (BSP)	
 Module g_timer0 Timer Driver on r_ 	r.c	
> General		
✓ Interrupts		
Callback	NULL	
Underflow Interrupt Priority		
Generate at Start	Disable	
✓ Extra Features		
GTM Mode	Interval timer mode	

Figure 83 : Configuring Interrupts in the Stacks tab

File Edit Navigate Search Project Reneasi Views Run Window Help 低 後 ● ● ◆ Debug · ② Binky Debug · ② ↓ ① - 物 回 ② ② ② ③ 本 物 回 ② ② ③ ③ ● • ④ - ③ ③ ● • ④ • ③ ③ ● • ④ • ③ ② ③ ⑤ CC++ ⑧ 59 Configuration : ② - 例 + ♡ - ♡ ◆ · ♡ + ⑦		
Image: Configuration Image: Configuration <td< th=""><th>work - Blinky/configuration.xml - e² studio</th><th>- 🗆 X</th></td<>	work - Blinky/configuration.xml - e ² studio	- 🗆 X
Image: Starting Configuration Image: Starting Configuration Starting Configuration Image: Starting Configuration Starting Configuration Image: Starting Configuration Image: Starting Configuration Image: Starting Config	File Edit Navigate Search Project Renesas Views Run Window Help	
Q: Yilling SP Configuration X Stacks Configuration Threads I for the Direct on r (spont) We find the Direct on r (spont) Wighting to P and the Direct on r (spont) Objects I form Direct on r (spont) Objects I form Direct on r (spont) Objects I form Direct on r (spont)	🐔 🗱 📕 🎄 Debug 🗸 💽 Blinky Debug	▼♦ 0, - 5, 10 10 12 2 2 2 2 2 2 2 4 4 4 4 4 4 5 2 4 2 2 2 2
Stacks Configuration VS Binky Threads) free Thread Control > 0 Financia V @ NAUCommon Am > port V @ NAUCommon Am > port V @ NAUCommon Am > port Ø g.lepont 10 Part Diver on r.jopont Diver Connective y > block (cg,h) Diver on r.jopont Diver on r.jopont Three Diver on r.gen Ø g.lepont 10 Part Diver on r.jopont > 0 Fort Out Table for Off On r.gong Ø g.lepont 10 Part Diver on r.jopont > 0 Fort Out Table for Off On r.gong Ø g.lepont 10 Part Diver on r.jopont > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off On r.gong Ø block control > 0 Fort Out Table for Off Off On r.gong Ø block control > 0 Fort Out Table for Off Off Off Off Off Off Off Off Off Of	😰 🔤 C/C++ 👜 FSP Configuration 🔡 = 🖓 = 🏷 😅 🗇 = 🔿 = 📑	Q
Stacks Configuration > Set Runnies Threads () Now Thread () Now Now Wath HAU Common Stacks () Now Stacks Wath H	🗱 *[Blinky] FSP Configuration 🗙	😑 🗖 🎼 Project Explorer 🖾 🛛 🗧 😫 🏹 💈 🗮 🗆
Threads MAL/Common Stacks €) New Stacks Am Port W PHAL/Common # g.jopot I/O Port Deter on r_jopot Am Port Sector Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Port Org Port	Stacks Configuration	Generate Project Content > 5 th Binaries
	✓ 👷 HAU/Cemmon ♥ g.ioport VO Port Driver on r_ioport Driver on 1_ioport	

Figure 84: Add new stack Timer (GTM)

5.2.6 Creating Interrupts from the Interrupts Tab

On the Interrupts tab, the interrupts of the driver which user selected in the Stacks tab are registered.

谢 (Blinky) FSP C	Configuration 22		- 0
Interrupts C	onfiguration		Generate Project Content
User Events			New User Event > 10 Remove
Event		ISR	
Allocations			
Interrupt	Event	ISR	
0	ID-46 GTM0 INT (GTM0 Interrupt)	gtm_int	"st
Summary BSP C	Docks Pins Interrupts Event Links Stacks Components		
annay by c	and the second s		

Figure 85 : Configuring interrupt in Interrupt Tab

Also, on the Interrupts tab, the user can add user's own peripheral interrupts. This can be achieved by adding a new event via the **New User Event** button.



5.2.7 Viewing Event Links

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

5.2.8 Adding and Configuring HAL Drivers

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

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

Stacks Configuration		O ✓	
Threads Thread Remove Thread Remove Thread Statution of State on Lippot	HAL/Common Stacks	New Stack Stack Second Sta	beg

Figure 86 : 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 rzg/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?
rzg_gen/main.c	Contains main() calling generated and user code. When called, the BSP has already initialized the MPU.	Yes
rzg_gen/hal_data.c	Configuration structures for HAL Driver only modules.	Yes
rzg_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

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



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/G 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		Generate	Project Conter	nt
		Filter All 🗸 S	earch	
Component	Version	Description	Variant	^
🗸 🔧 BSP				
🗸 🏈 Board				
custom		Custom Board Support Files		
✓ rzg2l_smarc		Evaluation Kit RZ/G2L Support Files (R		
🗸 🧳 rzg2l				
device		Board support package for R9A07G044	R9A07G04	
device		Board support package for RZG2L (RZG)		
device		Board support package for R9A07G044	R9A07G04	
device		Board support package for R9A07G044	R9A07G04	
device		Board support package for R9A07G044	R9A07G04	
device		Board support package for R9A07G044	R9A07G04	
device		Board support package for R9A07G044	R9A07G04	
🕼 fsp		Board support package for RZ/G2L (RZ		
🗸 🛷 CMSIS				
V V CMSIS5				
CoreM		Arm CMSIS Version 5 - Core (M)		
DSP		Arm DSP Library Source		
NN NN		Arm NN Library Source		ļ
🗸 💸 Common				
🗸 🏈 all				
fsp_common		Board Support Package Common Files		
V 🐼 HAL Drivers				
✓ ♀ rzg2l				
r_gpt		General PWM Timer		
▼ r_gtm		General Timer		
✓ r_ioport		I/O Port		

Figure 87 : Components Tab

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

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

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

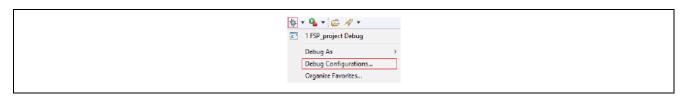


5.4 Debugging the Project

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

To debug an application, follow these steps:

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



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

Image: Second
CC++ Application CC++
Cle Hardware Debugging Cle Hardware Debugging Cle Deb
Java Applet Yana Appletion Build (of required) before launching Starch Project Browse Is anch Group Is anch Group Build Configuration: Use Active ✓ Remete Nava Application: View Active O table ando build ○ Disable ando build ✓ If Incess C0B Hordware D Image: Second Seco
Remote his Application V Interests GDB Hardners D Canable auto build Obsoble auto build

 Secure and Non-secure Vector Address are configured in the Connection Settings tab of the Debugger tab. The settings in below image are for setting the address of Secure and Non-secure Vector Offset mapped in Blinky project. Please note that these addresses vary in accordance with linker settings.

Debug Configurations				Select Symbol	×
Create, manage, and run config	gurations		- The	Filter	
Image: Constraint of the second se	Name: Blinky Debug, Flat			Symbol • Usagefault Handler, NS • Usagefault Handler, S • Worm Breet S • _Secure Vectors • _Secure Vectors • _Secure Jock Init vector	Address ^ 0x600108f1 0x72eff56d 0x1001#80 0x00010000 0x72eff511
C/C++ Remote Application EASE Script GDB Hardware Debugging	Debug hardware: J-Link ARM Target Device GDB Settings Connection Settings Debug Tool 5	ettings		bsp_clock_freq_var_init bsp_clock_init bsp_init bsp_ing_cfg	0x60010855 0x600108d9 0x6001097d 0x600108ed
GDB Simulator Debugging Launch Group C Renesas GDB Hardware Del C Blinky Debug_Flat C Renesas Simulator Debuggi	Reset at the beginning of connection Reset at the end of connection Reset before download Reset after download ID Code (Bytes) Hold reset during connect	Yes No No FFFFFFFFFFFFFFFFFFFFFFFFFFFFF No		OK	Cancel
	Set CPSR(Sbit) after download Prevent Releasing the Reset of the CM3 Con Secure Vector Address Non-secure Vector Address	&_Secure_Vectors &_Vectors		Filter: Symbol • UsageFasIL; Handler,NS • UsageFasIL; Handler,S	Address ^ 0x60010811 0x72eff56d
	Hot Plug Disconnection Mode V SWV Core clack (MHz)	No Stop 0	<u> </u>	Warm_Reset_S Secure Vectors Vectors bsp_clock_init_veneer	0x72eff51d 0x1001ff80 0x00010000 0x72eff611
< >	Set TrustZone Secure/non-secure boundarie		~ ~	bsp_clock_freq_var_init bsp_clock_init bsp_init bsp_init bsp_ing_cfg	0x60010855 0x600108d9 0x6001097d 0x600108ed ~
Filter matched 9 of 11 items		Reyert	Apply	ок	Cancel
?		Debug	Close		

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 3.1.2.2.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.

	Settings		¢	* -> * *
> Resource Builders ♥ C/C++ Build Build Variables Environment Incomment	Configuration: Debug [Active]		V Manage Configu	
Logging Settings Tool Chain Editor > C/C++ General	Tool Settings Toolchain P Build St 200 Target Processor 200 Optimization	ARM family	cortex-m4	v Parsers
> MCU	Warnings Control Con	Architecture	Toolchain default	~
Project References Renesas QE	V 🛞 GNU ARM Cross Assembler	Instruction set	Thumb (-mthumb) ork (-mthumb-interwork)	~
Run/Debug Settings > Task Repository	Preprocessor Preprocessor Preprocessor	Endianness	Toolchain default	~
Task Tags > Validation	Warnings Miscellaneous	Float ABI	FP instructions (hard)	~
	GNU ARM Cross C Compiler	FPU Type	fpv4-sp-d16	~
	🚔 Includes	Unaligned access	Toolchain default	~
	Optimization Warnings	AArch64 family	Generic (-mcpu=generic)	
	Miscellaneous Wind GNU ARM Cross C Linker	Feature crc Feature crypto	Toolchain default	~
	General Bibraries	Feature fp	Toolchain default	
	🙆 Miscellaneous	Feature simd	Enabled (+simd)	~
	Solution Sector Stream St	Code madel	Small (-mcmodel=small)	
	<			>

Figure 88 : 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/fsp.ld).



5.6 Importing an Existing Project into e2 studio

- 1. Start by opening e2 studio.
- 2. Open an existing Workspace to import the project and skip to step d. If the workspace does not exist, proceed with the following steps:
 - a. At the end of e2 studio startup, you will see the Workspace Launcher Dialog box as shown in the following figure.

🛃 Eclipse Launcher X
Select a directory as workspace e ² studio uses the workspace directory to store its preferences and development attifacts.
Workspace C\Users\-suser_name\e2studio\workspace v Browse
☐ Use this as the default and do not ask again
<u>Recent Workspaces</u>
Launch Cancel

Figure 89 : 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.

er 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 V Browse</username>
☐ Use this as the default and do not ask again ▶ Recent Workspaces
Launch Cancel

Figure 90 : Workspace Launcher dialog - Select Workspace

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

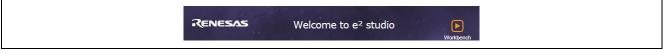


Figure 91 : Workbench arrow button

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.

C	SpWorkspace - FSPproject/confi
	File Edit Navigate Search Proje 🔶 Menu bar 🐔 🚁 🔳 🕸 Debug 🔶 Tool bar
	$ \underbrace{ \begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \end{array} \end{array}} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $

Figure 92 : Menu and tool bar



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

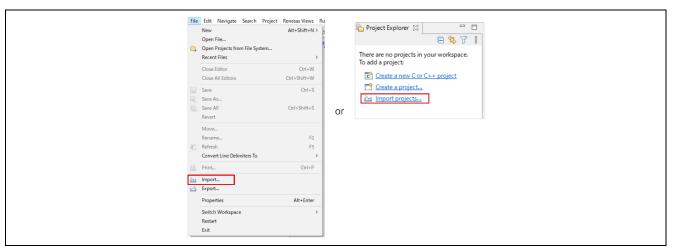


Figure 93 : File drop-down menu

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

Import - × Select	r			
Create new projects from an archive file or directory. Select an import wizard: type filter text CMSIS Pack C		Import	– 🗆 X	
type filter text			Ľ	
CMSIS Pack CMSIS Pack CMSIS Pack System GOVARM-NONE/RZ(DS-5) project conversion to GCC ARM Embedded Preferences Projects from Folder or Archive Renaes Web97(h:2052p) Reneas Web97(h:2052p) Reneas Web97(h:2052p) Reneass CGRX/CC-RL(CS+)7052p) Reneass GCRX/CC-RL(CS+)7052p) Reneass GitHub FreeRTOS (with IoT libraries)7057p) Reneasle GitHub FreeRTOS (with IoT libraries)7057p) Renease GitHub FreeRTOS (with IoT libraries)7057p)				
		✓ Constant Cons		

Figure 94 : Project Import dialog with "Existing Projects into Workspace" option selected



- 6. Click Next.
- 7. To import the project, use either **Select archive file** or **Select root directory**.
 - a. Click **Select root directory** file as shown in the following figure.

Import				×
Import Projects				5
Select a directory to sear	rch for existing Eclipse projects.			
Select root directory:			0	_
			Browse.	_
 Select archive file: 		~	Browse.	
Projects:				
			Select A	All
			Deselect	All
			Refresh	h
Options Search for nested pro	rojects			
Copy projects into w				
	ed projects upon completion			
Hide projects that al	lready exist in the workspace			
Working sets				
Add project to work	rking sets		New	
Working sets:			Select	
?	a Dudi - Marca - Marca		C 1	
	< Back Next > Finish		Cancel	

Figure 95 : Import Existing Project dialog 1 - Select root directory

- b. Click Browse.
- c. For **Select root directory**, browse to the project folder that you want to import.
- d. Select the file for import.
- e. Click Open.
- f. Select the project to import from the list of **Projects**, as shown in the following figure.

Import			×
Import Projects Select a directory to search for existing Eclipse projects.			
Select root directory: C:¥works_RZG_V¥work	~	Browse	
○ Select archive file:	~	Browse	
Projects:			
Blinky (C:¥works_RZG_V¥work¥Blinky)		Select /	All
		Deselect	All
		Refres	h

Figure 96 : Import Existing Project dialog 2

8. Click Finish to import the project.



6. Migration from previous version

When updating the FSP used in project from v2.0.0 to FSP v2.0.1, please follow the procedure described in this chapter.

As a preparation, please install the latest version of FSP pack in advance by following the procedure in section 2.2.

6.1 How to update RZ/G2L, RZ/G2LC, RZ/G2UL project

1. Display the BSP tab of Smart Configurator for the target project.

Board Supp	ort Package Configuration		Generate Project Content
1		· È	Board Details
Core: RTOS:	No RTOS	~	
< Summary BSP C	Clocks Pins Interrupts Event Links Stacks Com	ponents	>

Figure 97 : BSP tab of Smart Configurator

2. Select the version you want to change from the drop-down list in the **FSP version** field.

Board Supp	ort Package Configuration	(Generate Project Content
Device Selecti	on		
Device: Core:	2.0.1 2.0.0	× 	Board Details
C Summary BSP	Clocks Pins Interrupts Event Links Stacks Compone	ents	>

Figure 98 : Version selection



3. The following confirmation dialog will be displayed. Then select **OK**.

📴 e2 studio	\times
Warning: Changing FSP version will modify the selected set of FSP components as necessary. Any components not provided by the newly selected FSP version will be removed from the project. Changes in FSP component selection may result in the overwriting or removal of existing project files originating from FSP packs. The preceding versions of any user-modified files will be preserved in the local history store according to workspace preferences.	
Continue switching FSP version?	
OK Cancel	

Figure 99 : Confirmation dialog

4. Press the Save button to save the FSP configuration update.

File Edit Navigate Search Project Renesas View	Run Window Help
🔚 🕲 - 🔦 - 🖉 🌣 🗉 🤹 - 💁 -	
🔁 Project Explorer 🗙 📄 🔄 🕞	🌼 *[rzg2l_baremetal_blinky] FSP Configuration $ imes$
> 👺 rzg2l_baremetal_blinky	Board Support Package Configuration

Figure 100 : Save the FSP configuration

5. Press the **Generate Project Content** button to generate the code for the selected version.

🔅 [rzg2l_baremetal_blinky] FSP Configuration 🛛	E
Board Support Package Configuration	Generate Project Content
Device Selection	

Figure 101 : Project Generation

6. Finally, build the project and update the executable.



6.2 How to update RZ/G3S project

The procedure differs depending on the target core. Please refer to the procedure depending on the core targeted by the project you are modifying.

6.2.1 Procedure to update the project for CM33 without FPU Core

This section describes procedure to update the version of a project created in the following conditions:

Device Selecti	on	
FSP version:	2.0.0	\sim
Board:	RZ/G3S Evaluation Kit (SMARC) ~	2
Device:	R9A08G045S33GBG_CM33	
Core:	Core 1(CM33)	~

Figure 102 : Condition of target project on BSP tab

1. Open Smart Configurator **BSP** tab and change **FSP version** to latest.

	wo_fpu] FSP Configuration ×		0
Board Supp	oort Package Configuration	Gener	ate Project Content
Device Select	ion		
FSP version	200	~	Board Details
	2.0.1		
Board:	2.0.0		
Device:	R9A08G045S33GBG_CM33		
Core:	Core 1(CM33)	\sim	
RTOS:	No RTOS	\sim	
<			>
Summary BSP	Clocks Pins Interrupts Event Links Stacks Co	mponents	

Figure 103 : Version selection



2. The following confirmation dialog will be displayed. Then select **OK**.

 e2 studio Warning: Changing FSP version will modify the selected set of FSP components as necessary. Any components not provided by the newly selected FSP version will be removed from the project. Changes in FSP component selection may result in the overwriting or removal of existing project files originating from FSP packs. The preceding versions of any user-modified files will be preserved in the local history store according to workspace preferences.
components not provided by the newly selected FSP version will be removed from the project. Changes in FSP component selection may result in the overwriting or removal of existing project files originating from FSP packs. The preceding versions of any user-modified files will be
Continue switching FSP version?

Figure 104 : Confirmation dialog

3. Select Custom User Board (Any Device) on the Board combo.

	g_wo_fpu] FSP Configuration ×	Gener	ate Project Content
Device Select FSP version		~	Board Details
Board: Device: Core: RTOS:	RZ/G3S Evaluation Kit (SMARC) Custom User Board (Any Device) RZ/G2LC Evaluation Kit (SMARC) RZ/G2L Evaluation Kit (SMARC) RZ/G2UL Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC)		
< Summary BSP	Clocks Pins Interrupts Event Links Stacks C	omponents	>

Figure 105 : Board selection

4. The following confirmation dialog will be displayed. Then select **Save and Continue**.

Confirm device change	\times
Changing device may result in substantial changes to this configuration. FSP modules and pin settings that are incompatible with the new device will be removed. Saving this configuration before changing device will allow you to abandon the device change if the resulting changes are not acceptable. Do you want to save the configuration file before proceeding with the device change?	
Save and Continue Continue Cancel	

Figure 106 : Confirmation dialog



5. Select RZ/G3S Evaluation Kit (SMARC) on the Board combo.

	_wo_fpu] FSP Configuration ×	Gener	rate Project Content
Device Select	ion		
FSP version: Board: Device: Core: RTOS:	2.0.1 Custom User Board (Any Device) Custom User Board (Any Device) RZ/G2LC Evaluation Kit (SMARC) RZ/G2L Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) (CM33_FPU)		Board Details
< Summary BSP	Clocks Pins Interrupts Event Links Stacks (Components	>

Figure 107 : Board reselection

6. The following confirmation dialog will be displayed. Then select **Save and Continue**.

3 Confirm device change	\times
Changing device may result in substantial changes to this configuration. FSP modules and pin settings that are incompatible with the new device will be removed. Saving this configuration before changing device will allow you to abandon the device change if the resulting changes are not acceptable. Do you want to save the configuration file before proceeding with the device change?	
Save and Continue Cancel	

Figure 108 : Confirmation dialog

7. Select Core 2(CM33_FPU) and save.

Device Selection FSP version: 2.0.1 Board: RZ/G3S Evaluation Kit (SMARC) Device: R9A08G045S33GBG Core: Core 2(CM33_FPU) RTOS: No RTOS		wo_fpu] FSP Configuration ×	Gene	rate Project Content
FSP version: 2.0.1 Board: RZ/G3S Evaluation Kit (SMARC) Device: R9A08G045S33GBG Core: Core 2(CM33_FPU)	Device Select	ion		
	Board: Device: Core:	RZ/G3S Evaluation Kit (SMARC) R9A08G045S33GBG Core 2(CM33_FPU)	· 2	Board Details

Figure 109 : Core selection (1/2)



8. Select Core 1(CM33) and save.

	wo_fpu] FSP Configuration ×		Gener	Trate Project Conter	
Device Select	ion				
FSP version	2.0.1			Board Details	
Board:	RZ/G3S Evaluation Kit (SMARC)	× È			
Device:	R9A08G045S33GBG		🥼	\	
Core:	Core 1(CM33)		\sim		
RTOS:	No RTOS		\sim		
<					>
Summary BSP	Clocks Pins Interrupts Event Links Stack	s Components			

Figure 110 : Core selection (2/2)

9. Press the Generate Project Content button.

<pre>[rzg3s_cm33_</pre>	wo_fpu] FSP Configuration $ imes$		- 8
Board Supp	ort Package Configuration	Gene	rate Project Content
Device Selecti	on		
FSP version:	2.0.1	×.	Board Details
Board:	RZ/G3S Evaluation Kit (SMARC)		
Device:	R9A08G045S33GBG		
Core:	Core 1(CM33)	\sim	

Figure 111 : Project Generation

10. Build the project

Note: The exclamation mark to the right of the device combo disappears once you switch to a display other than the BSP tab.



6.2.2 Procedure to update the project for CM33 with FPU Core

This section provides procedure to update the version of a project created in the following conditions:

Device Select	ion
FSP version:	2.0.0 ~
Board:	G3S Evaluation Kit (SMARC) (CM33_FPU) ~ 🚵
Device:	R9A08G045S33GBG_CM33_FPU
Core:	~

Figure 112 : Condition of target project on BSP tab

1. Open Smart Configurator BSP tab and change FSP version to latest.

[rzg3s_cm33]	_w_fpu] FSP Configuration $ imes$		- 8
Board Sup	port Package Configuration	G	enerate Project Content
Device Selec	tion		
FSP version	: 2.0.0	~	Board Details
Board:	<mark>2.0.1</mark> 2.0.0		
Device:	R9A08G045S33GBG_CM33_FPU		
Core:		\sim	
RTOS:	No RTOS	\sim	
 Summary BSP 	Clocks Pins Interrupts Event Links Stack	s Components	>
,			

Figure 113 : Version selection

2. The following confirmation dialog will be displayed. Then select **OK**.

📴 e2 studio	\times
Warning: Changing FSP version will modify the selected set of FSP components as necessary. Any components not provided by the newly selected FSP version will be removed from the project. Changes in FSP component selection may result in the overwriting or removal of existing project files originating from FSP packs. The preceding versions of any user-modified files will be preserved in the local history store according to workspace preferences.	
Continue switching FSP version?	
OK Cancel	

Figure 114 : Confirmation dialog



3. Select Custom User Board (Any Device) on the Board combo.

-	_w_fpu] FSP Configuration ×	Generate Proje	ect Content
Device Select	ion		
FSP version: Board: Device: Core: RTOS:	2.0.1 Custom User Board (Any Device) RZ/G2LC Evaluation Kit (SMARC) RZ/G2L Evaluation Kit (SMARC) RZ/G3L Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC) CM33_FPL	Board Deta	ails
Summary 💩 B	SP Clocks Pins Interrupts Event Links Sta	cks Components	>

Figure 115 : Board selection

4. The following confirmation dialog will be displayed. Then select **Save and Continue**.

S Confirm device change	\times
Changing device may result in substantial changes to this configuration. FSP modules and pin settings that are incompatible with the new device will be removed. Saving this configuration before changing device will allow you to abandon the device change if the resulting changes are not acceptable.	
Do you want to save the configuration file before proceeding with the device change?	
Save and Continue Cancel	

Figure 116 : Confirmation dialog



5. Select RZ/G3S Evaluation Kit (SMARC) on the Board combo.

Device Selection FSP version: 2.0.1 Board: Custom User Board (Any Device) Device: RZ/G2LC Evaluation Kit (SMARC) Core: RZ/G2L Evaluation Kit (SMARC) RTOS: RZ/G3S Evaluation Kit (SMARC) (CM33_FPU)		w_fpu] FSP Configuration ×	Ge	enerate Project Content
Board: Custom User Board (Any Device) Device: RZ/G2LC Evaluation Kit (SMARC) Core: RZ/G2L Evaluation Kit (SMARC) RTOS: RZ/G3S Evaluation Kit (SMARC)			X	Board Details
	Board: Device: Core:	Custom User Board (Any Device) Custom User Board (Any Device) RZ/G2LC Evaluation Kit (SMARC) RZ/G2UL Evaluation Kit (SMARC) RZ/G3S Evaluation Kit (SMARC)		

Figure 117 : Board reselection

6. The following confirmation dialog will be displayed. Then select **Save and Continue**.

Confirm device change	\times
Changing device may result in substantial changes to this configuration. FSP modules and pin settings that are incompatible with the new device will be removed. Saving this configuration before changing device will allow you to abandon the device change if the resulting changes are not acceptable.	
Do you want to save the configuration file before proceeding with the device change?	
Save and Continue Continue Cancel	

Figure 118 : Confirmation dialog



7. Select Core 2(CM33_FPU) and save.

	w_fpu] FSP Configuration ×	G	enerate Project Content
Device Selecti	on		Devel Date 1
FSP version: Board: Device: Core: RTOS:	2.0.1 RZ/G3S Evaluation Kit (SMARC) R9A08G045S33GBG Core 1(CM33) Core 1(CM33) Core 2(CM33_FPU)	~ 2	Board Details
< Summary BSP	Clocks Pins Interrupts Event Links Stack	s Components	>

Figure 119 : Core selection

8. Press the Generate Project Content button.

<pre>[rzg3s_cm33_</pre>	w_fpu] FSP Configuration $ imes$	_	- 8
Board Supp	ort Package Configuration	G	enerate Project Content
Device Selecti FSP version: Board: Device: Core:	2.0.1	~ 	Board Details

Figure 120 : Project Generation

9. Build the project

Note: The exclamation mark to the right of the device combo disappears once you switch to a display other than the BSP tab.



Revision History

Rev.	Date	Description	
		Page	Summary
2.01	Feb.13.2.24	6 to 11, 13 to 22,	Updated the description and figure based on the latest development environment.
		45	Updated the description of project creation.
		48	Added the section of how to avoid resource duplication description in the case of RZ/G3S project creation.
		60 to 69	Added the section of how migrate the project which using previous FSP version to latest FSP version.
2.00	Jan.9.24	1	Added RZ/G3S to the target device.
		5	Updated the description of the RAM initialization section.
		6 to 11	Updated the description and figure based on the latest development environment.
		12 to 22	Updated the development setup for Linux Host PC and FSP installation.
		26 to 32	Added description and figure for RZ/G3S SMARC EVK.
		33 to 36	Updated the description and figure based on the latest development environment.
		40	Removed the steps to configure Secure Vector and Non- secure Vector from the Debug step.
		46 to 51	Added description about the Pins and Clocks tabs.
		55	Updated the instructions for configuring Secure Vector and Non-secure Vector.
1.20	Nov.30.22	13	Updated version information of SEGGER J-Link driver and Libgen Update for GNU ARM Embedded Toolchains.
		20, 24	Updated installation procedure for FSP packs.
		55	Updated the method of specifying Secure Vector Address and Non-secure Vector Address.
1.10	Apr.27.22	5 to 53	Updated the description and figure based on the latest development environment.
1.01	Dec.3.21	14 to 22	Added a section of e2 studio installation for Linux PC.
1.00	Jul.30.21	-	First Edition issued.



General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power is supplied until the power reaches the level at which reseting is specified.

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

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

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

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

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

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury 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/.