

RZ/V2L

Getting Started with Flexible Software Package

Introduction

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

Target Device

RZ/V2L



Contents

1. Introduction	4
1.1 Overview	4
1.2 Introduction to FSP	4
1.2.1 Purpose	4
1.2.2 e ² studio IDE	4
1.3 Limitations	4
1.3.1 Peripherals and pins assignment	4
1.3.2 RAM Initialization	4
2. 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
2.1.2.1 Obtaining an RZ MPU Kit	5
2.1.2.2 PC Requirements	5
2.1.2.3 Licensing	5
2.1.3 e ² studio installation for Windows PC	5
2.1.3.1 Download	5
2.1.3.2 Installation of e ² studio IDE	6
2.1.3.3 Installation of Additional Software	11
2.1.4 e ² studio installation for Linux PC	14
2.1.4.1 Download	14
2.1.4.2 Installation	14
2.1.4.3 Add GNU ARM Embedded Toolchain on e ² studio IDE	21
2.2 FSP setup	23
2.2.1 Installation of FSP Packs	23
3. Set up an RZ/V2L SMARC EVK	25
3.1 Supported Emulator	25
3.2 Board Setup	25
3.2.1 Boot MODE	25
3.2.2 JTAG connection	26
3.2.3 Debug Serial (console output)	26
3.2.4 Power Supply	27
3.2.5 How to check the operation of the board	28
4. Tutorial: Your First RZ MPU Project - Blinky	29
4.1 Tutorial Blinky	29
4.2 What Does Blinky Do?	29
4.3 Create a New Project for Blinky	29



RZ/V2L

4.3.1	Details about the Blinky Configuration	32
4.3.2	Configuring the Blinky Clocks	32
4.3.3	Configuring the Blinky Pins	32
4.3.4	Configuring the Parameters for Blinky Components	32
4.3.5	Where is main()?	32
4.3.6	Blinky Example Code	32
4.4	Build the Blinky Project	33
4.5	Debug the Blinky Project	34
4.5.1	Debug prerequisites	34
4.5.2	Debug steps	34
4.5.3	Details about the Debug Process	36
4.5.4	Run the Blinky Project	36
5. F	-SP application launch with e² studio	37
5.1	Create a Project	37
5.1.1	What is a Project?	37
5.1.2	Creating a New Project	39
5.1.2.	1 Selecting a Board and Toolchain	40
5.1.2.	2 Selecting a Project Template	40
5.2	Configuring a Project	43
5.2.1	Summary Tab	43
5.2.2	Configuring the BSP	43
5.2.3	Configuring Clocks	44
5.2.4	Configuring Pins	45
5.2.5	Configuring Interrupts from the Stacks Tab	45
5.2.6	Creating Interrupts from the Interrupts Tab	46
5.2.7	Viewing Event Links	46
5.2.8	Adding and Configuring HAL Drivers	47
5.3	Reviewing and Adding Components	48
5.4	Debugging the Project	49
5.5	Modifying Toolchain Settings	51
5.6	Importing an Existing Project into e ² studio	52
Revis	sion History	55



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/V2L.

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/V2L 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 each peripheral for CM33, but each driver can operate on the assumption that it is not used in CA55.

1.3.2 RAM Initialization

In RZ/V2L FSP, there is no implementation for initializing internal RAM and external DDR SDRAM, and it is expected that the RAMs are initialized in the software that should run before CM33 is kicked. This is because CA55 is always the primary boot core because of RZ/V2L specification and initialization of RAMs should be carried out as a part of bootstrap.



2. Starting Development Introduction

2.1 e² studio setup

2.1.1 What is e² studio?

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

When developing for RZ MPUs, e² studio hosts the 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/V2L FSP, start with RZ/V2L Evaluation Board Kit.

Start-up guide of RZ/V2L Evaluation Board Kit is available at <u>RZ/V2L SMARC EVK Start-up Guide</u>.

2.1.2.2 PC Requirements

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

- Windows 10 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

- 1. Double-click on e² studio installer to invoke the e² studio installation wizard page. Click [Install].
- **Note:** If e² studio was installed in your PC, the option to modify, remove the existing version or install e² studio to a different location will be displayed

🛃 Renesas e² studio 2023-01 Se	tup	- 🗆 X
Renesas e ² studio 2023-01 S	Setup	
e2 studio version 22.10.0.R202210	013-1357 is already installed.	
	What do you want to do? Upgrade Upgrade to version 23.1.0.R20230106-1556. Location (AReneasive2 studio) Missall to a different location.	
<u>v202301061733</u> Us	ser: All Users < <u>B</u> ack <u>N</u> ext >	Install Cancel

Figure 1: e² studio installation wizard

2. Welcome page

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

Note1: If you would like to have multiple versions of e² studio, please specify new folder here. **Note2**: Multi-byte characters cannot be used for e² studio installation folder name.

Renesas e ² studio 2023-01 Setup
Device Families
Cutor readies Prerequisites oftware ready to install Additional Software Initial: • Microsoft Visual C++ 2015-2022 Runtime (x86) • Microsoft Visual C++ 2015-2022 Runtime (x64) Licenses Change prerequisites to install. Shortcuts Internet connection available Change Proxy Settings. Installing Results Prerequisites oftware to install Vicrosoft Visual C++ 2015-2022 Runtime (x86) • Microsoft Visual C++ 201

Figure 2: Installation of e² studio – Welcome page



3. Device Families

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

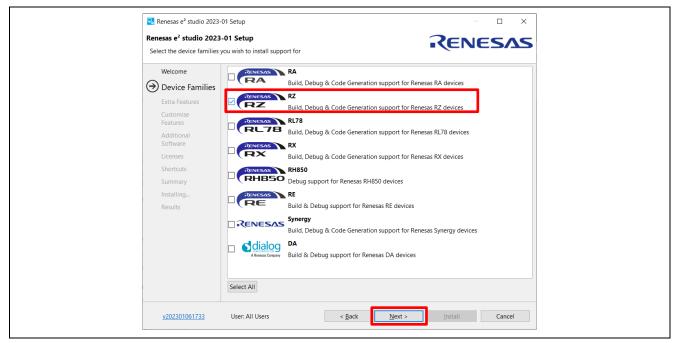


Figure 3: Installation of e² studio – Device Families

4. Extra Features

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

🔜 Renesas e² studio 2023	3-01 Setup	- D X
Renesas e ² studio 202 Select the extra features y	-	
Welcome Device Families		Japanese Language Support
Extra Features		Chinese (Simplified) Language Support
Customise Features Additional		Chinese (Traditional) Language Support
Software		Git Integration Git SCM Support
Shortcuts Summary		Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
Installing Results		
	Select All	
<u>v202301061733</u>	User: All Users	< <u>B</u> ack <u>Next</u> > Install Cancel

Figure 4: Installation of e² studio – Extra Features



5. Customize Features

Select the components to install and click the [Next] button to continue.

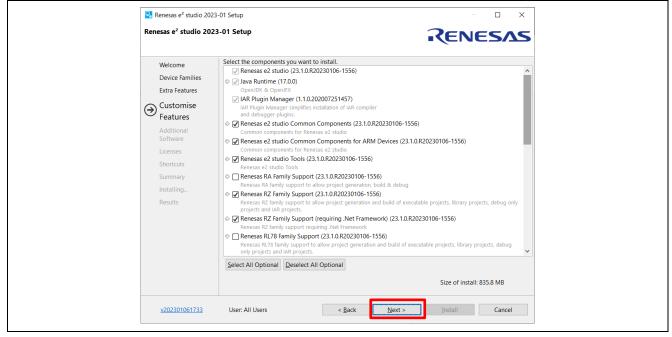


Figure 5: Installation of e² studio - Features

6. Additional Software

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

GNU ARM Embedded 9.2.1 2019q4 LibGen for GNU ARM Embedded

Renesas e ² studio 2	023-01 Setup		Renesas e² studio 202	23-01 Setup	
	oftware you wish to install	RENESAS	Select the additional sol		NESAS
Welcome	Renesas Toolchains & Utilities (GCC Toolchains & Utilities (1)		Welcome	Renesas Toolchains & Utilities (0) GCC Toolchains & Utilities (3)	
Device Families	👼 Renesas CC-RH Compilers	*	Device Families	😺 GNU ARM Embedded	* ^
Extra Features Customise Features	Renesas CCRH v2.04.01 v2.04.01 Renesas C/C++ Compiler Package for RH850 Family v2.04.01 Download size: 19.3 MB Reourises		Extra Features Customise Features	GNU ARM Embedded 9.3.1 2020q2 9.3.1.2020q2 GNU ARM Embedded 9.3.1 2020q2 Download size: 111.2 MB	
→ Additional	Renesas Tool License Manager - 2.05.00		Additional	GNU ARM Embedded 9.2.1 2019g4 9.2.1.2019g4	
Software	Renesas Utilities	*	→ Software	GNU ARM Embedded 9.2.1 2019q4 Download size: 94.0 MB	
Licenses	Renesas E2 Self-Checking Program V1.02.00 v1.02.00 Self-checking program for the E2 emulators V1.02.00.		Licenses	Download size: 94.0 MB	
	Download size: 5.0 MB		Shortcuts	GNU ARM Embedded 6 2017q2 6.0.0.2017q3	
Summary	Renesas E2 Lite Self-Checking Program V1.00.00 v1.00.00		Summary	GNU ARM Embedded 6 2017q2 Download size: 82.6 MB	
Installing	Self-checking program for the E2 Lite emulators V1.00.00.		Installing		
Results	Download size: 5.5 MB		Results	LibGen for GNU ARM Embedded 1.2020.9 LibGen for GNU ARM Embedded (requires GNU ARM Embedded toolchain)	
	Renesas E1, E20 Self-Checking Program V1.01.00 v1.01.00			Download size: 51.7 MB	
	Self-checking program for the E1 and E20 emulators V1.01.00. Download size: 1.5 MB			GCC ARM A-Profile	
	Download size: 1.3 Wb			GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 10.3.0.2021-07	
				GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 Download size: 97.8 MB	.
		97.8 MB download required		243.	4 MB download required
	User: All Users < Back Next >	Install Cancel	v202301061733	User: All Users < Back Next > Install	Cancel

Figure 6: Installation of e² studio – Additional Software

Be sure that LibGen for GNU ARM Embedded is essential for RZ/V2L FSP. If it's not installed, FSP won't be built successfully.

For more details on the installation of Additional Software, please see the section 2.1.4.



7. License Agreement

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

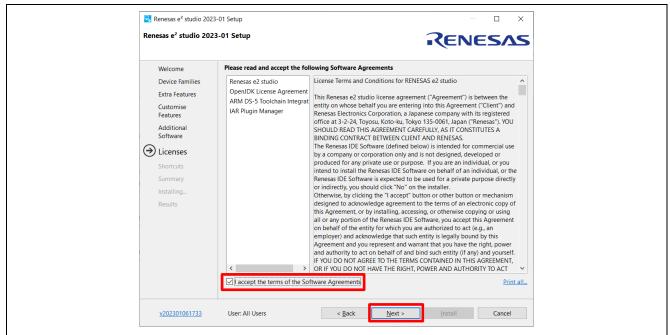


Figure 7: Installation of e² studio – Licenses

8. Shortcuts

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

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

Renesas e³ studio 2023-01 Setup Renesas e² studio 2023-01 Setup Renesas e² studio 2023-01 Setup Welcome Device Families Extra Features Customise Customise Reatures Additional Software Licenses
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing

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.

udio 2023-01 Setup				
		etup	* studio 2023-0	lenesas e ² st
milies Pre-equisite software to install: Microsoft Visual C++ 2015-2022 Runtime (x86) Microsoft Visual C++ 2015-2022 Runtime (x64) Software to install: Renesas e2 studio v23.1.0.R20230106-1556 Java Runtime v17.0.0 IAR Plugin Manager v1.1.0.202007251457 Renesas e2 studio Common Components v23.1.0.R20230106-1556 Renesas e2 studio Common Components for ARM Devices v23.1.0.R20230106-1556 Renesas e2 studio Common Components for ARM Devices v23.1.0.R20230106-1556 Renesas e2 studio Common Components for ARM Devices v23.1.0.R20230106-1556 Renesas R2 Family Support (requiring).Net Framework v23.1.0.R20230106-1556 Renesas R2 Family Support (requiring).Net Framework v23.1.0.R20230106-1556	C++ C++ C++ C++ C++ C++ C++ C++ C++ C++	Prerequi • Mici • Mici • Softwar • Ren • Java • IAR • Ren • Ren	ce Families Features mise rres tional ware ses tcuts hmary lling_	Welcome Device Fart Extra Feat Customise Features Additiona Software Licenses Shortcuts Shortcuts Shortcuts Shortcuts Shortcuts

Figure 9: Installation of e² studio – Summary

10.Installing...

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

11.Results

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

Renesas e² studio 2023-01 Setup Renesas e² studio 2023-01 Setup Welcome Installation of e2 studio is complete. Device Families Please click OK to close. Extra Features Launch e2 studio? Additional View Release Notes? Additional View What's New? Software Useful Links: GCC ARM A-Profile (AArch64 bare-metal): C\Renesas\e2 studio\toolchains\gcc arm aarch64\10 2021 07 Installing Assults Installing.
Device Families Please click OK to close. Extra Features Launch e2 studio? Features View Release Notes? Additional View What's New? Software Licenses Useful Links: GCC ARM A-Profile (AArch64 bare-metal); C\Renesas\e2 studio\toolchains\gcc arm aarch64\10 2021 07.

Figure 10: Summary Page



2.1.3.3 Installation of Additional Software

As mentioned in the section 2.1.3.2, the additional software listed below is essential for RZ/V2L FSP.

GNU ARM Embedded 9.2.1 2019q4 LibGen for GNU ARM Embedded

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

(1) GNU ARM Embedded Toolchain 9.2.1 2019q4

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.

💮 GNU Tools for Arm Embed	ded Processors 9-2019-q4-major — 🗌 🗙	🌍 GNU Tools for Arm Embedded Processors 9-2019-q4-major — 🗆 🗙
	Welcome to the GNU Tools for Arm Embedded Processors 9–2019–q4–major 9 2019 Setup Wizard	License Agreement Please review the license terms before installing GNU Tools for Arm Embedded Processors 9-2019-q4-major 9 2019. Press Page Down to see the rest of the agreement.
	This wizard will guide you through the installation of GNU Tools for Arm Embedded Processors 9-2019-q4-major 9 2019.	Contains code from project GNU Binutils (<u>https://www.anu.org/software/binutils</u>), GNU Debugger (<u>https://www.anu.org/software/adb/</u>) under the following license(s).
	It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.	GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007
	Click Next to continue.	Copyright (C) 2007 Free Software Foundation, Inc. < <u>http://fsf.org/</u> > Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
		If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install GNU Tools for Arm Embedded Processors 9-2019-q4-major 9 2019.
	Next > Cancel	Nullsoft Install System v2.46-7
🌍 GNU Tools for Arm Embedo	ded Processors 9-2019-q4-major — 🗆 🗙	🕤 GNU Tools for Arm Embedded Processors 9-2019-q4-major — 🗆 🗙
9-2019-q4-major 9 2019. Setup will install GNU Tools for	Arm Embedded Processors 9-2019-q4-major 9 2019 in the different folder, click Browse and select another folder. Click	Completing the GNU Tools for Arm Embedded Processors 9-2019-q4-major 9 2019 Setup Wizard GNU Tools for Arm Embedded Processors 9-2019-q4-major 9 2019 has been installed on your computer. Click Finish to dose this wizard.
Destination Folder Togram Files (x86)¥GNU T Space required: 498.9MB Space available: 1.2GB	Tools Arm Embedded¥9 2019-q4-major Browse	Show Readme Launch gccvar.bat Add path to environment variable Add registry information
Nullsoft Install System v2.46-7 —	3	< Back Finish Cancel

Figure 11: Installation of GNU ARM Embedded Toolchain



(2) Libgen Update for GNU ARM Embedded Toolchains

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

- 1. Click Click here to select your GNU ARM Embedded Toolchain installation folder.
- 2. Select the folder where GNU ARM Embedded Toolchain 9.2019-q4-major is installed.

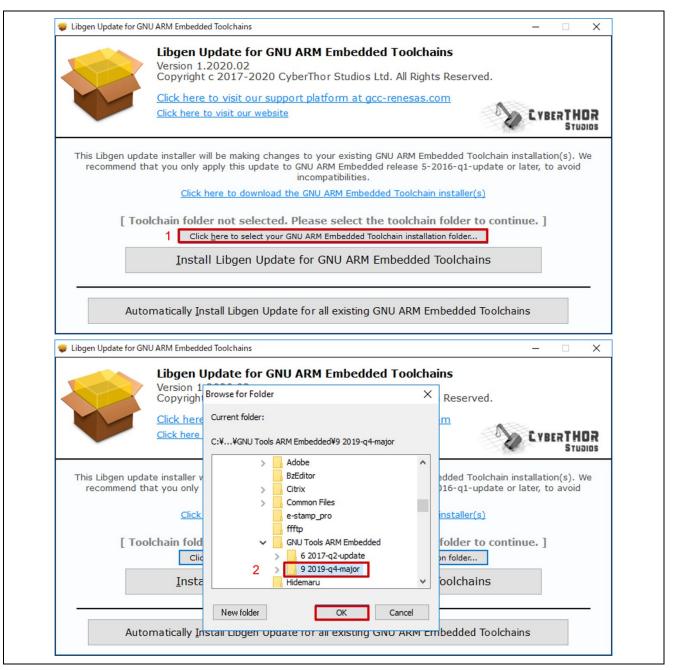


Figure 12: Libgen Update for GNU ARM Embedded: Select Toolchain Folder



3. Next, click Install Libgen Update for GNU ARM Embedded Toolchains.

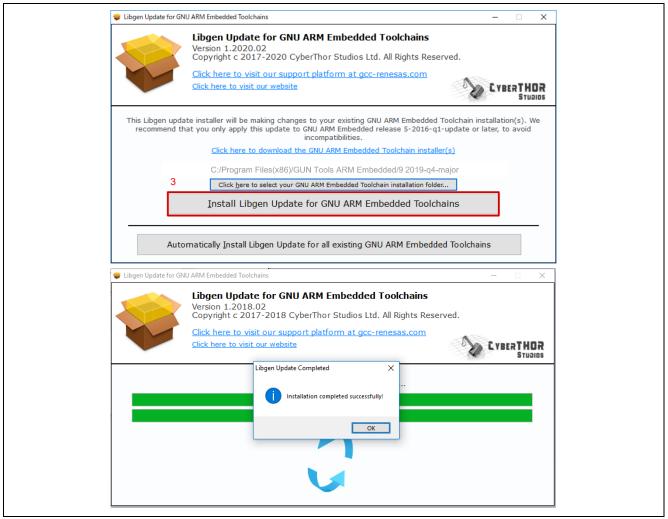


Figure 13: Libgen Update for GNU ARM Embedded: Update Toolchain



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 Download

Following files are required to download before installation.

SEGGER J-Link driver Please download the driver V7.84 or after from: <u>https://www.segger.com/downloads/jlink/JLink Linux x86 64.deb</u>

GNU ARM Embedded Toolchain

Please download gcc-arm-none-eabi-9-2019-q4-major-x86_64-linux.tar.bz2 in the Arm Developer site: https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads

Libgen Update for GNU ARM Embedded Toolchains

Please download the Libgen Update (Linux) for GCC ARM Embedded Toolchains v1.2020.02 or after from:

https://gcc-renesas.com/rz/rz-download-toolchains/

e² studio IDE installer

Please download the e² studio IDE installer package for Linux 2023-01 of after 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.

2.1.4.2 Installation

This section describes the procedure of each software installation.

Filename, version number and the file path is provided for example purpose only.

(1) SEGGER J-Link driver

1. Open a terminal window and enter below commands.

sudo dpkg -i JLink_Linux_V784_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_V784_x86_64.deb

(2) GNU ARM Embedded Toolchain

1. Enter below commands on terminal.

sudo mkdir -p /opt cd /opt sudo tar -xvf ~/Downloads/gcc-arm-none-eabi-9-2019-q4-major-x86_64-linux.tar.bz2



(3) Libgen Update for GNU ARM Embedded Toolchains

- 1. Run the Libgen Update Installer.
 - cd ~/Downloads

```
sudo chmod 755 LibgenUpdateInstall_v1.2020.02.run
sudo ./LibgenUpdateInstall_v1.2020.02.run
```

Enter the path where the GNU toolchain is installed.

```
Libgen Update for GNU ARM Embedded Toolchains Installer
Launch date: 21st of December, 2020
Copyright (c) 2015-2020 by CyberTHOR Studios Ltd. All Rights Reserved.
Get FREE worldwide support at https://gcc-renesas.com
Please specify a path where to install the Libgen update to: /opt/gcc-arm-none-e
abi-9-2019-q4-major
```

Figure 14: Installation – Libgen Update (1/2)

Ensure the input path and enter the 'y' to continue the installation. Then, the Libgen Update installation will complete.

```
The Libgen update will be installed in: /opt/gcc-arm-none-eabi-9-2019-q4-major

Please make sure that you have permissions to write to the "/opt/gcc-arm-none-ea

bi-9-2019-q4-major" folder

before continuing the installation process.

Are you sure you want to continue? [y/n] (default 'n') y

Attempting to install Libgen update in "/opt/gcc-arm-none-eabi-9-2019-q4-major".

..

Permissions have been verified.

Please wait, extracting files...

Detecting Libgen version... version 3.1.0 detected

Please wait, decompressing data...

Cleaning up leftover files...

Libgen Update for GNU ARM Embedded Toolchains has been installed successfully!

Installation complete!
```

Figure 15: Installation – Libgen Update (2/2)



(4) e² studio IDE

1. Run the e² studio IDE Installer. (Before running the installer, check the execution permission of the installer.)

— ./e2studio_installer-2023-01_linux_host.run

2. Welcome page

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

Note1: If you would like to have multiple versions of e^2 studio, please specify new folder here. **Note2**: Multi-byte characters cannot be used for e^2 studio installation folder name.

Renesas e² studio 2023-01 Setup	RENESAS
⊖ Welcome	Install directory ready Install localion: /home/v-koanagi/.local/share/renesas/e2 studio 2023 01
Device Families	[Change]
Customise Features	Prerequisite software already installed
Additional Software Licenses	Internet connection available Change Proxy Settings
Shortcuts Summary	Ready to install
Installing Results	Software to install: • Renesas e2 studio v23.10.R20230106-1556 · Java Runtime v17.0.0 • IAR Plugim Manager v1.1.0.202007251457 • JustJ Adoptium OpenJDK Hotspot JEC Complete v17.0.4.v20220903-1038 • openjfx.standard.feature v17.0.2.202205100908 • openjfx.swing.feature v17.0.2.202205100908 • openjfx.swing.feature v17.0.2.202205100908 • openjfx.swi.feature v17.0.2.202205100908 • openjfx.web.feature v17.0.2.202205100908 Sopenjfx.swi.feature v17.0.2.202205100908 Sopenjfx.swi.feature v17.0.2.202205100908 Additional components can be selected on the next page Size of install: S26.3 MB Temporary space required: 912 MB Available: 110 GB

Figure 16: Installation of e² studio – Welcome page



3. Device Families

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

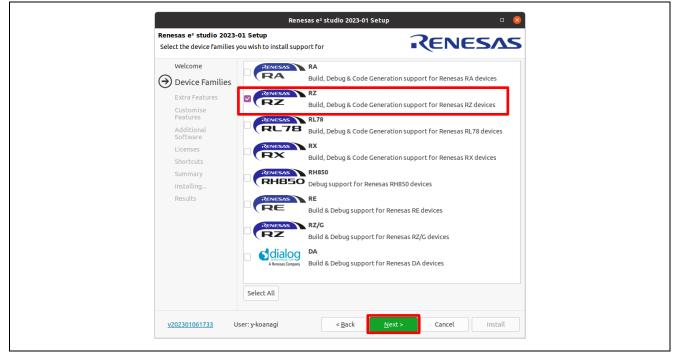


Figure 17: Installation of e² studio – Device Families

4. Extra Features

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

RENESAS	-	Renesas e ² studio 2023-0 Select the extra features yo
Japanese Language Support		Welcome Device Families
Chinese (Simplified) Language Support		Extra Features
Chinese (Traditional) Language Support		Customise Features Additional Software
Git Integration Git SCM Support		Licenses Shortcuts
Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet		Summary Installing
		Results
	Select All	

Figure 18: Installation of e² studio – Extra Features



5. Customize Features

Select the components to install and click the [Next] button to continue.

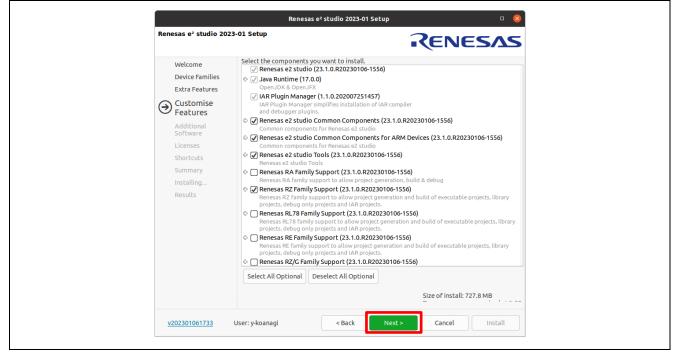


Figure 19: Installation of e² studio – Features

6. Additional Software

Select additional software (i.e., compilers, utilities, QE...) to be installed. Click [Next] to continue.

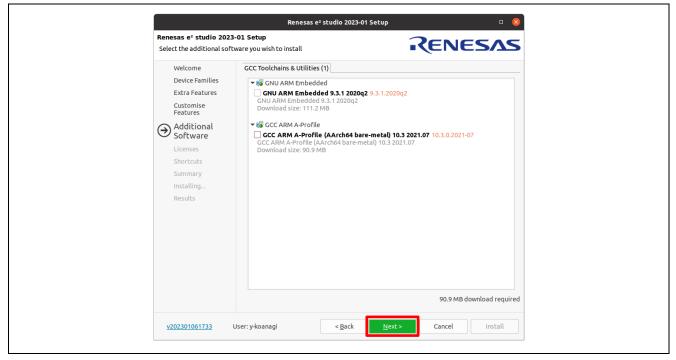


Figure 20: Installation of e² studio – Additional Software



7. License Agreement

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

Welcome	Please read and accept t	he following Software Agreements
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results	Renesas e2 studio OpenJDK License Agreen ARM DS-5 Toolchain Integ IAR Plugin Manager	License Terms and Conditions for RENESAS e2 studio This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2:4, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, ASIT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS. The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly you should click "No" on the installer. Otherwise, by clicking the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of this Agreement, or by Installing, accessing, or otherwise copying or using all or any portion of the Renesas IDE Software, ou accept this Agreement, or behalf of the entity for which you are authorized to act (e.g., an employer) and acknowledge that such entity is legally bound by this Agreement ad you represent and warrant that you have the right, power and authority to act on behalf of an dind such entity (if any) and yourself. IF YOU DO NOT AGREE TO THE TERMS CONTAINED IN THIS AGREEMENT, OR IF YOU DO NOT HAVE THE RICHT, POWER AND
	I accept the terms of t	he Software Agreements Print all

Figure 21: Installation of e² studio – Licenses

8. Shortcuts

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

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

Renesase * studio 2023-01 Setup Welcome Device Families Extra Features Customise Features Additional Software Licenses Summary Installing Results
Device Families Extra Features Customise Features Additional Software Licenses Summary Installing

Figure 22: Installation of e² studio – Shortcuts



RZ/V2L

9. Summary

Components list to be installed is shown. Please confirm the contents and click the [Install] button to install the Renesas e^2 studio IDE.

Renesas e² studio 202	3-01 Setup	CENESAS
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts	Ready to Install Software to install: • Renesas e2 studio (v23.1.0.R20230106-1556 Java Runtime v17.0.0 • IAR Plugin Manager v1.1.0.202007251457 Renesas e2 studio Common Components V23. • Renesas e2 studio Common Components V23. • Renesas e2 studio Tools v23.1.0.R20230106-155 • Renesas e2 studio Tools v23.1.0.R20230106 • Renesas E75 PSmart Configurator ARV 99.0.0V • Renesas F5P Smart Configurator ARV 99.0.0V • Renesas F5P Smart Configurator ARV 99.0.0V • Renesas F5P Smart Configurator ARV 99.0.0V • ARM D5-5 Toolchain Integration v1.0.4.V20220 • Eclipse CDT Linker Script Ediltor and 05L v1.0. • GCC for Renesas RZ Support V3.1.0.R20221031 • Renesas BUId Support Plies v23.1.0.V202 • Just J Adoptium OpenJDK Hotspot JLRE Compl • Renesas BUId Support Files v3.1.0.V20221031 • Renesas RTOS Debug Views v9.0.0.V20221025-1029 • Renesas RTOS Debug Views v9.0.0.V20221031-10.V202 • Renesas Smart Kelp v9.0.0.V0202104062 • Renesas Smart Help v9.0.0.V020210400 • Renesas e2 studio ARM Common Debug v9.0.0. • Renesas e2 studio ARM Common Debug v9.0. • Renesas e2 studio ARM Common Build v9.0.0V2022 • Renesas e2 studio Common Build v9.0.0V2022 • Renesas e2 studio Common Build for RCC & LL v9.0.0.V20221206-1637 • Renesas e2 studio Common Build for Renesas	ARM Devices 556 555 555 520230106-1254 1/20221202-2037 1/20221202-2037 1/20221203-1359 2/21212-1830 2/20922-1407 1/122 1/20221027-1512 -0612 -0612 -0612 -0828 2/229-2148 0/221214-1046 0.v20221202-1157 0.0.v20221102-1427 2103-1106 LIVM Toolchains

Figure 23: Installation of e² studio – Summary

10.Installing...

The installation is performed. Depending on selected items of additional software, new dialog prompts may appear during the installation process.

11.Results

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

R	studio 2023-01 Setup	• 😣
enesas e² studio 2023-01 Setup	RENE	SVZ
Welcome Installation of e2 Device Families Please click OK to Extra Features Launch e2 st Customise View Releas Features View What's Software Useful Links: Licenses GCCARM A4 Shortcuts GCCARM A4 Summary Installing → Results Results	complete. .rch64 bare-metal): /home/y-koanagi/.local/share/ren olchains/gcc arm aarch64/10 2021 07	nesas/-

Figure 24: Summary Page



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.

Select a directory as workspace e ² studio uses the workspace directory to store its preferences and development artifacts. Workspace: //home/epsd1=ubuntu=nucs3/e2_studio/workspace Use this as the default and do not ask again Cancel Launch
Use this as the default and do not ask again

2. Preferences

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

type filter text Renesas Toolchain Management My Renesas Renesas OF Toolchain Type Installation Path Renesas Toolchain Smart Browser GNU ARM Embedded Installation Path Smart Browser GCC ARM A-Profile (AArch64 bare-metal)
Renesas Colchain Smart Browser
Smart Demo Smart Demo Smart Manual Support Folders Tracealyzer Trace Fan/Debug Scripting Terminal Toolchains Tracing Validation Version Control (Tean Scan Add Remove

3. Add New Toolchain

Specify the path of the GNU ARM Embedded Toolchain. Click [OK]

	Add New Toolchain	8
_	e a new toolchain which is not already registered. NU ARM Embedded - 9.2.1.20191025	
Location:	/opt/gcc-arm-none-eabi-9-2019-q4-major	Browse
	_	
?	Cancel	ОК



4. Apply

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

type filter text	Renesas Toolchain Management	↓ ↓ ↓ 8
My Renesas	Toolchain Type	Installation Path
Renesas QE Renesas Toolchain Smart Browser > Smart Configuratol Smart Demo > Smart Manual Support Folders Tracealyzer TraceX > Run/Debug > Scripting Terminal Toolchains > Tracing Validation	 GNU ARM Embedded 9.2.1.20191025 KPIT GNUARM-NONE-EABI Toolchain GCC ARM A-Profile (AArch64 bare-metal) 	/opt/gcc-arm-none-ea
 Version Control (Tean XML 	Scan Add	Remove

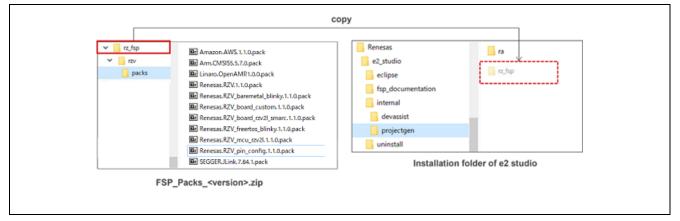


2.2 FSP setup

2.2.1 Installation of FSP Packs

FSP Packs can be downloaded from <u>here.</u> You can integrate FSP to e² studio by the procedure described below:

- 1. Exit e² studio.
- 2. Extract FSP Packs.
- 3. Copy the extracted FSP Packs (folders under \internal\projectgen\rz_fsp) to the following folder:
 - <e² studio installation folder>\internal\projectgen



- 4. Launch e² studio.
- 5. Click Help > About e² studio.

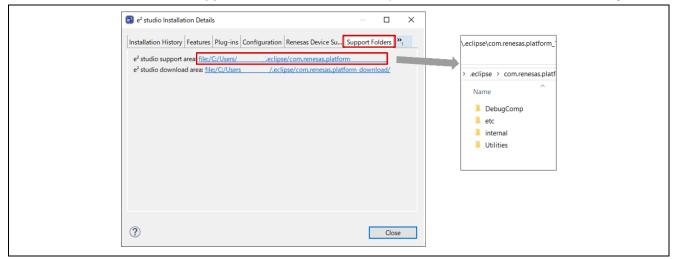
🖉 🍪 Welcome 🗙		Help Welcome			1
RENESAS	Welcome to e ² studio	 Help Conten Search Show Conten 			
		Show Active Cheat Sheets	Keybindings	Ctrl+Shift+L	
0	Create a new C/C++ project Create a new e ² studio C/C++ project	RA Helpdesk RenesasRulz Add Renesas	Community Forum Toolchains	>	rview in over
*	Import existing projects Import existing e ² studio projects from the filesys	 Perform Setu Check for Up Install New S 	dates		orials prough
±	Import sample projects Download and import sample projects from Renes		udio feedback ed Workbench plugin manager dio		uples at the s
8	Review IDE configuration settings			🔿 Wh	at's No

6. Click Installation Details.

About e ² studio	lio		×
	Renesas e ^z studio		
- 2	Version: 2022-01 (22.1.0) Build Id: R20211118-2110		
e-	Parts Copyright (C) 2010-2021 Renesas Electronics Corp. All rights reserved.		
	e ² studio IDE is an extension of software developed for eclipse.org.		
	e ² studio IDE is based on Eclipse Platform 4.21 (2021-09) and CDT version 10.4.1.		
	Source code for the Eclipse Foundation plug-ins is availa from www.eclipse.org, under the Eclipse Public License " see https://www.eclipse.org/org/documents/epi-z0/EPI and https://www.eclipse.org/org/documents/epi-v10.htt	-2.0.html	
6	nz e² 🔹 🚯 🚥 🖋 🤤		
(?) Installation	tion Details	⊊lose	:



7. Click the link to e² studio support area. At that time, the explorer should be launched automatically.



- 8. Check if the FSP Packs is copied to the following folder.
 - <e² studio support area>\internal\projectgen\rz_fsp\rzv\packs

 convenesas.platform DebugComp etc internal cmsis projectgen ra rzy rzy rzy packs packs Manzon.AWS.1.1.0.pack Amazon.AWS.1.1.0.pack Amazon.AWS.1.1.0.pack Amazon.AWS.1.1.0.pack Amazon.AWS.1.1.0.pack Amazon.AWS.1.1.0.pack Reness.RZV_board_rzv2[smarc.1.1.0.pack Reness.RZV_freetros_linky.1.1.0.pack

- 9. Check xml files is extracted to the following folder.
 - <e² studio support area>\internal\projectgen\rz_fsp\rzv\modules

com.renesas.pla 会約 DebugComp etc com.renal com.renal com.renesas.pla 会約 1.0 1.0 1.1 modules projectgen ra
etc 1.1 internal modules crusis packs
etc 1.1 internal modules crusis packs
crisis packs projectgen
cmis packs projectgen
n_fsp



3. Set up an RZ/V2L SMARC EVK

Below is an example of a typical system configuration.

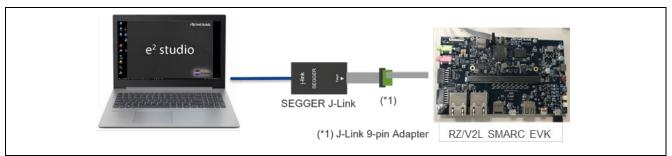


Figure 25: System Configuration Example – RZ/V2L SMARC EVK

3.1 Supported Emulator

• SEGGER J-Link

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

3.2 Board Setup

3.2.1 Boot MODE

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

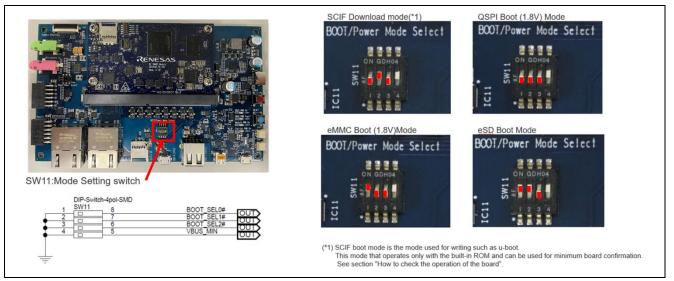


Figure 26: Boot MODE



3.2.2 JTAG connection

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

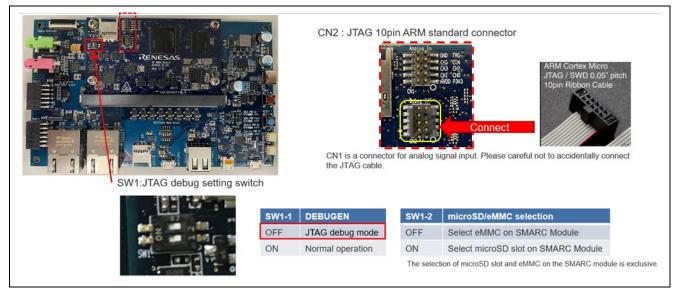


Figure 27: JTAG connection

Please note that RZ/V2L SMARC EVK has CoreSight 10 connector and therefore, the following adapter must be needed to connect Segger J-Link.

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

3.2.3 Debug Serial (console output)

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

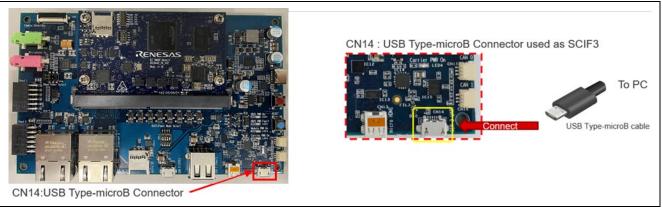


Figure 28: Debug Serial(console output)



RZ/V2L

3.2.4 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)

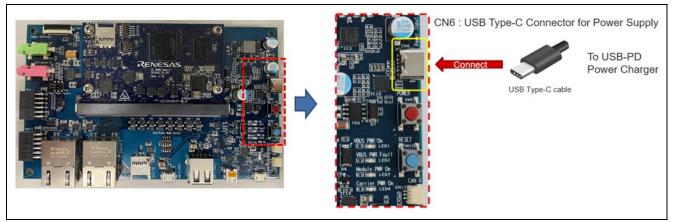


Figure 29: Power Supply

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.

Note: When turn on the power, press and hold the power button for 1 second. When turn off the power, press and hold the power button for 2 seconds

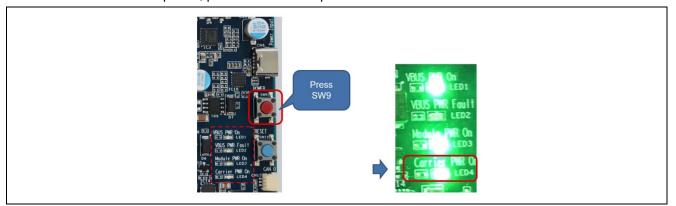


Figure 30: LED Status after Turning on EVK

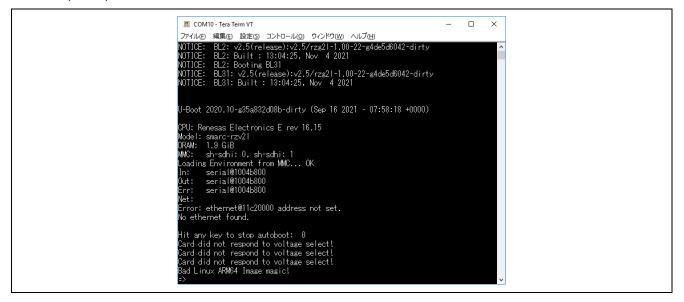


3.2.5 How to check the operation of the board

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

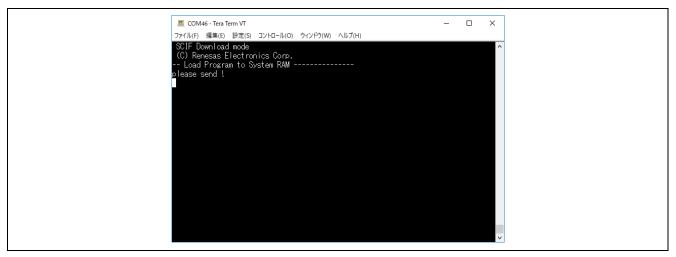
BOOT MODE: QSPI Boot(1.8V) Mode

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



BOOT MODE: SCIF Download Mode

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





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.

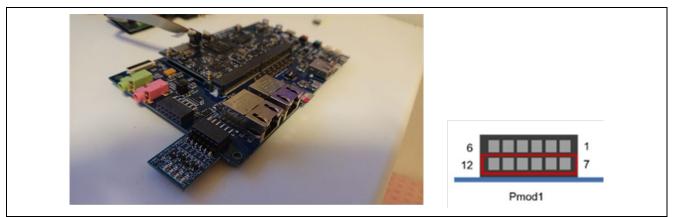


Figure 31: Connection Pmod LED module (410-076)

4.3 Create a New Project for Blinky

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

The base RZ/V2L 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.

6	work - e² studio				
File	Edit Source Refactor Navigate Search Project	ct	Renesas Views Run Window Help		
	New Alt+Shift+N >		Renesas C/C++ Project	>	- 01
	Open File	C	Makefile Project with Existing Code		
	Open Projects from File System	C	ງ C/C++ Project		(÷ -
	Recent Files >	C	9 Project		

Figure 32: New C/C++ Project



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

Figure 33: Renesas RZ/V 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.

🐻 Reness RZ/V C/C++ FSP Project — 🗆 🗙	
Renesas RZ/V C/C++ FSP Project	
Project Name and Location	
Driject name	
givjet ame Bioky	
Use gefault location	
Location Chuedek Billinky Bipmens. Cheose file system default -	
Choose his spirem: demain	
You zen deentiezd meer Rentza gaeka heen	
⑦ < ĝack <u>Bjest</u> ≻ Emilih Cancel	

Figure 34 : 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 **9.2.1.20191025** and Click **Next**.

Renesas RZ/V C/C++ FSP Project Renesas RZ/V C/C++ FSP Project Device and Tools Selection		
Device Selection FSP Version 10.0 Board: EZZV23. Evaluation Kit (2) Device R8A070554.23066 Cone Exercise Language: @ C ○ C++	Device Details TrutZone No Pins 455	
Teolchains CRU ABA Embedded	Processor Conter-M33 Debugger J-Link ARM	~
■2.1.20191025 ✓	< Back Boot > Emails	Cencel

Figure 35 : e² studio Project Configuration window (part 2)



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

Renesas RZ/V C/C++ FSP Project		- 0 X]
Renesas RZ/V C/C++ FSP Project Build Artifact, RTOS Selection and Sub-Core Selection		\rightarrow	
Build Artifact Selection • Executable • Project builds to an executable file	RTOS Selection 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			
			-
۲	< <u>Back</u> <u>Next</u> > <u>Finis</u>	ish Cancel	

Figure 36 : e² studio Project Configuration window (part 3)

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

Renesas RZ/V C/C++ FSP Project	
Renesas RZ/V C/C-+ 55P Project Project Template Selection	
Project Template Selection Bare Metal - Blinky Bare metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C nutrine environment. (RenesseR2V2L1.0.0.prxk)	
Bare Metal - Minimal Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RZV21.1.0.0 prck]	
Code Generation Settings Use Renesus Code Formatter	
() Cancel	

Figure 37 : 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.

 Blinky] FSP Configuratio Summary 	on X	Generate Proje	ct Content
Project Summary		RENESAS	^
	RZ/V2L Evaluation Kit (SMARC) R9A07G054L23GBG		
Toolchain: 0	GCC for Renesas RZ		
Toolchain Version: 9			
	1.0.0 Flat		
Selected software con	nponents		~

Figure 38 : 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 >/rzv_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.



RZ/V2L

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.

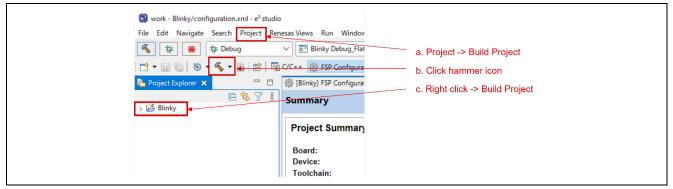


Figure 39 : e² studio Project Explorer window

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 📮 Console 🗙
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)
L. Contraction of the second se

Figure 40 : e² studio Project Build console



4.5 Debug the Blinky Project

4.5.1 Debug prerequisites

To debug the project on a board, you need

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

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

4.5.2 Debug steps

To debug the Blinky application, follow these steps:

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

Rur	Windo	indow Help	
	Renesas	esas Device Partition Manag	ger
1	TraceX	ceX	>
Ð	Tracealy	cealyzer	>
Q	Run		Ctrl+F11
*&	Debug	bug	F11
	Run Hist	History	>
0			>
	Run Cor	Configurations	
	Debug I	oug History	>
柃	Debug /	-	>
	Debug	oug Configurations	
9	External	ernal Tools	>

Figure 41 : e² studio Debug icon

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

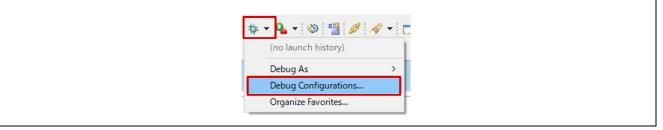


Figure 42 : e² studio Debugger Configurations selection option

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



Debug Configurations Create, manage, and run configurations				
Image: Second	Name: Blinky Debug	rce Common Variables		Browse Browse
Filter matched 13 of 15 items			Revert	Apply

Figure 43 : e² studio Debugger Configurations window with Blinky project (1)

- 3. Select the debug configuration for the generated project and select the **Debugger** tab.
- 4. Set the Secure and Non-secure Vector Address in the Connection Settings tab.

Click the "..." button in **Secure Vector Address** field and select "__Secure_Vectors" from the list in the **Select Symbol** window.

Also, click the "..." button in **Non-secure Vector Address** field and select "___Vectors" from the list in the **Select Symbol** window.

These commands 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			– 🗆 X		
eate, manage, and run config	irations		Ť.	Filter: Symbol SystemInit_S	Address 0x72eff549
Perfilter text C /C++ Application C /C++ Remote Application C /C++ Remote Application E ASE Script C OBB Hardware Debugging (D Bb Ardware Debugging () D Bb Application D Java Application	Name: Blinky Debug, Flat	evice: R9A07G044L23GBG_(×	Systemint, S UsageFault, Handler, NS UsageFault, Handler, S UsageFault, Handler, S Worm, Pasel, S Excerte, Vectors	0x/2eff549 0x6001084d 0x72eff535 0x73eff581 0x1001ff80 0x70010000 0x72eff5b9 0x600107b1 0x60010835 0x600108d9
A Launch Group	✓ Connection			ОК	Cancel
	Connection Register initialization Reset on connection Reset before run ID Code (8ytes) Hold reset during connect Reset before download Prevent Releasing the Reget of the CM3 Secure Vector Address Non-secure Vector Address SWV Core clock (MHz) V TrustZone	No No No FFFFFFFFFFFFFFFFFFFFFFFFFF No No		OK	Address 0x6001084d 0x72eff585 0x72eff585

Figure 44: e² studio Debugger Configurations window with Blinky project(2)



- 5. Click **Debug** to begin debugging the application.
- 6. Extracting **RZ Debug**.

Progress Information			×
Preparing launch delegate			
Configuring GDB			
	Cancel	Details	>>

7. When get the warning message, click Yes.

6

4.5.3 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.

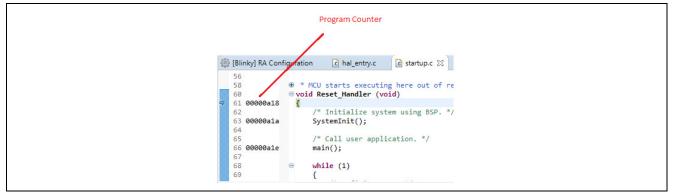


Figure 45 : e² studio Debugger memory window

4.5.4 Run the Blinky Project

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

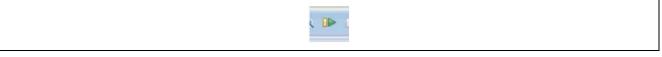


Figure 46 : e² studio Debugger Play icon

The LEDs on the Pmod LED marked LD0, LD1, LD2 and LD3 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.

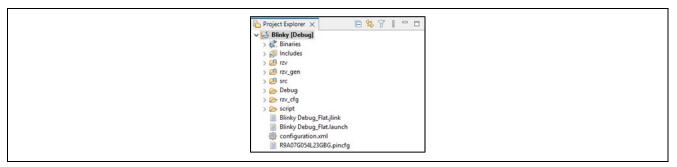


Figure 47 : e² studio Project Configuration file

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.

😰 🔤 C/C++ 🏟 FSP Configuration

Figure 48 : e² studio FSP Configuration Perspective

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

	□ 🖹 rzv_cfg.txt ×		- 1
Blinky [Debug]	1	FSP Configuration	^
> 🖏 Binaries	2	Board "RZ/V2L Evaluation Kit (SMARC)"	
> 🔊 Includes	3	R9A07G054L23GBG	
> 🥝 rzv	4	part_number: R9A07G054L23GBG	
	5	rom_size_bytes: 0	
> 🥴 rzv_gen	6	ram_size_bytes: 131072	
> 😂 src	7	package_style: LFBGA	
> 🦢 Debug	8	package_pins: 456	
> 🗁 rzv_cfg	9		
> 😂 script	10	RZV2L	
Blinky Debug_Flat.launch	11	series: 2	
	12		
configuration.xml	13	RZV2L Family	
R9A07G054L23GBG.pincfg	N 14	RZ/V2L Common	
rzv_cfg.txt	15	Secure stack size (bytes): 0x200	
	16	Main stack size (bytes): 0x200	
	17	Heap size (bytes): 0	
	18	MCU Vcc (mV): 3300	
	19	Parameter checking: Disabled	
	20	Assert Failures: Return FSP_ERR_ASSERTION	
	21	Error Log: No Error Log	
	22	PFS Protect: Enabled	
	23	C Runtime Initialization : Enabled	~

Figure 49 : 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.

		- 0
Summary		Generate Project Content
Project Summary Board: RZ/V2L Evaluation Kit (SMARC) Device: R3A07G054L23GBG Toolchain: GCC for Renesas RZ Toolchain Version: 9.2.120191025 FSP Version: 1.0.0	Renesa	• ^
Project Type: Flat		
Selected software components Simple application that blinks an LED. No RTOS included. Arm CMSIS Version 5 - Core (M) Board Support Package Common Files	v1.0.0 v5.7.0 v1.0.0	
General Timer I/O Port Board support package for R9A07G054L23GBG Board support package for RZV2L Board support package for RZV2L (RZV2L) - FSP Data	v1.0.0 v1.0.0 v1.0.0 v1.0.0 v1.0.0 v1.0.0	
Evaluation Kit RZ/V2L Support Files (RZ/V2L) - FSP Data	v1.0.0	~
Summary BSP Clocks Pins Interrupts Event Links Stacks Compor	hents	

Figure 50 : 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**.

6	work - e² studio					
File	Edit Source Refactor Navigate	Search Project	t Re	enesas Views Run Window Help		
	New	Alt+Shift+N >		Renesas C/C++ Project	>	- 61
	Open File		C+	Makefile Project with Existing Code		I –
	Open Projects from File System		C	C/C++ Project		(÷ •
	Recent Files	>		Project		

Figure 51 : New RZ MPU Project

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

New C/C++ P	C/C++ Project	0 X	
All CMake Make Renesas Debug Renesas RA Renesas R2	RA Renesas RZ/T C/C++ FSP Project	, ,	
?	< Back Next > Einish	Cancel	

Figure 52 : New Project Templates

3. Select a project name and location.

Renesse RZ/Y C/C - + FSP Project Project Name and Location Project Name Image: Churche RZ/Network Mindy Churche RZ/Network Mindy Churche Rie System: Oncoder Rie System:	Г	Renesas RZ/V C/C++ FSP Project	×
Project Name and Location			
Project name Use gefault location Location Channels, RZVwardt Slinky Channel RZvardt Slinky Van can download more Renease packs here			
Binky Urs getaut location Location Choose for system default You can download more Remeas pack here		Project Name and Location	<u></u>
Binky Use datable location Location: Chooses & R2/NeweRX Binky Chooses file system: default		Project name	
Lotation Chaotic Register Sense parks herp			
Choose File system industries and the second system in the second system is a second system in the second system is a second sy		Use gefault location	
Choose file system indexed more Renease packs here		Location: Chworks,RZV/work/Blinky Brow	136
		Vicu can disunfood more Reneral narks here	
(?) <back net=""> Emith Cancel</back>		() < Back Next > Emith Can	scel

Figure 53 : 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. To add threads, select **RTOS**, or **No RTOS** if an RTOS is not being used.
- 5. The Toolchain selection defaults to GNU Arm Embedded.
- 6. Select the Toolchain version. This should default to the installed toolchain version.
- 7. Select the **Debugger**. The J-Link Arm Debugger is preselected.
- 8. Click Next.

	///C/C++ FSP Project / C/C++ FSP Project Jols Selection				
Device: Core:		Board Description Device Details TrustZone Pins Processor	No 436 Cote-M3		
Toolchains		Debugger			
GNU ARM E	mbedded	J-Link ARM		~	
9.2.1.2019102	\$ v				
?		< Back	yest > Emish		

Figure 54 : RZ MPU Project Generator (Screen 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/V C/C++ FSP Project	- 🗆 X	
Renesas RZ/V C/C++ FSP Project		
Build Artifact, RTOS Selection and Sub-Core Selection		
Build Artifact Selection	RTOS Selection	
Executable Project builds to an executable file	No RTOS V	
Project ounds to an executable file 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		
		-
0	< Back Next > Einish Cancel	

Figure 55 : 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/V 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 - Blinky**.

Renesas RZ/V C/C++ FSP Project	Renesas RZ/V C/C++ FSP Project	
Project Template Selection Project Template Selection Image:	Project Template Selection FreeRTOS - Blinky - Dynamic Allocation FreeRTOS - Blinky - Static Alloc	8
Code Generation Settings	Code Generation Settings ⊡Use Renesas Code Formatter	

Figure 56 : 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.

[Blinky] FSP Configuration ×		- 0
Summary	G	ienerate Project Content
Project Summary Board: RZ/V2L Evaluation Kit (SMARC) Device: R9A076054123686 Toolchain: GCC for Renesas RZ Toolchain: Version: 9.2.1.2.0191025 FSP Version: 1.0.0 Project Type: Flat Selected software components Simple application that blinks an LED. No RTOS included. Arm CMSIS Version 5 - Cre (M) Board Support Package Common Files General Timer	V10.0 V5.7.0 V10.0	^
General Timer I/O Port Board support package for R9A070054L23GBG Board support package for R2V2L Board support package for R2/V2L (R2/V2L) - FSP Data Evaluation Kit R2/V2L Support Files (R2/V2L)	V1.00 V1.00 V1.00 V1.00 V1.00	v
Support		

Figure 57 : RZ MPU Project Editor and available editor tabs

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

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

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



5.2 Configuring a Project

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

5.2.1 Summary Tab

[Blinky] FSP Configuration ×		- 0
Summary		Generate Project Content
Project Summary Board: R2/V2L Evaluation Kit (SMARC) Device: R3A070654L236BG Tootchain: GCC for Renesas R2 Tootchain Version: 9.2.1.2011025	Renesas	^
FSP Version: 1.0.0 Project Type: Flat Selected software components		
Simple application that blinks an LED. No RTOS included. Arm CMSIS Version 5 - Core (M) Board Support Package Common Files General Time I/O Port Board support package for R9A07G054L23GBG Board support package for R2V2L Board support package for R2V2L Board support package for R2V2L (R2VV2L) - FSP Data Evaluation KR R2VV2L Support Files (R2VV2L)	100 570 100 100 100 100 100 100 100	Ţ
Support	9	

Figure 58 : 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.

Summary	Summary BSP Clocks Pins Interrupts Event Links Stacks Components				
Properties × Problems @ Smart Browser Console					
RZ/V2L	Evaluation Kit (SMARC)				
Settings	Property	Value			
	✓ R9A07G054L23GBG				
	part_number	R9A07G054L23GBG			
	rom_size_bytes	0			
	ram_size_bytes	131072			
	package_style	LFBGA			
	package_pins	456			
	 RZ/V2L 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			
	✓ RZV2L				
	series	2			

Figure 59 : 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 rzv_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. Thus, clocks configuration here must align with the settings on CA55.

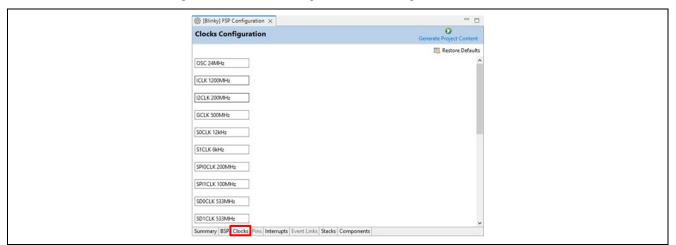


Figure 60 : Configuration Clocks tab

When you click the **Generate Project Content** button, the clock configuration contents are written to: rzv_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

RZ/V2L FSP doesn't support **Pins** tab, and it is grayed out.

If you would lite to use I/O port, I/O Port setting should be modified "pin_data.c" manually. For details on I/O Port setting, please refer to <u>RZ/V2L Setting GPIO with Flexible 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.

	FSP Configuration ×		Ga	Concerts Reside Content		
Threads	Movember of the second se	HAL/Common Stacks	🔊 New Stack > 🚊 Entend	nerate Project Content		
Properti	New Object > Remove SP_Clocks Pins Interrupts Event Links St ss X Problems @ Smart Brower @ Timer Driver on r.gtm Property v Commo		Value			
	Parameter Checking V Module g_timer2 Timer Driver on r_gtm		Default (BSP)			
	> General > Interrupts > Extra Features					

Figure 61 : Configuring Interrupts in the Stacks tab

*[Blinky] FSP Configu	uration ×		Q、三部一覧 CyC++ ® FSP Configuratio
itacks Configura	New Thread Remove	HAL/Common Stacks	Generate Project Content Silinary Set Binaries Set Bina
✓ Statute of the second se	Port Driver on r_joport	♥ g_joport I/O Port Driver on r_joport	
Objects	New Object + Remove		





5.2.6 Creating Interrupts from the Interrupts Tab

On the **Interrupts** tab, the interrupt of the driver selected in the **Stacks** tab is registered.

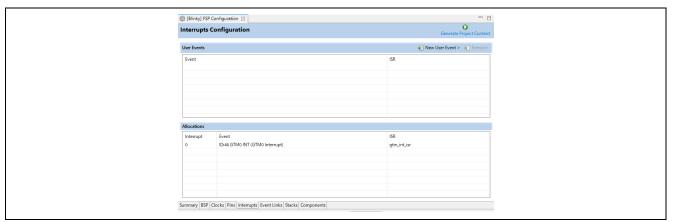


Figure 63 : Configuring interrupt in Interrupt Tab

And on the Interrupts tab, the user can add a peripheral interrupt create by user. This can be done by adding a new event via the **New User Event** button.

5.2.7 Viewing Event Links

RZ/V2L FSP doesn't support **Event Links** tab, and it is grayed out.



RZ/V2L

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**.

*(Blinky) FSP Config	uration ×		Q、 (部) 職 C/C++ 優 FSP Configurati 一 D
Stacks Configuration			Generate Project Content
Threads New Thread: Remove HAL/Common Stacks		@ g_ioport I/O Port	
Objects	ᡚ New Object > 🔊 Remove		

Figure 64 : 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 rzv/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?
rzv_gen/main.c	Contains main() calling generated and user code. When called, the BSP already has Initialized the MPU.	Yes
rzv_gen/hal_data.c	Configuration structures for HAL Driver only modules.	Yes
rzv_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: rzv_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/V 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.

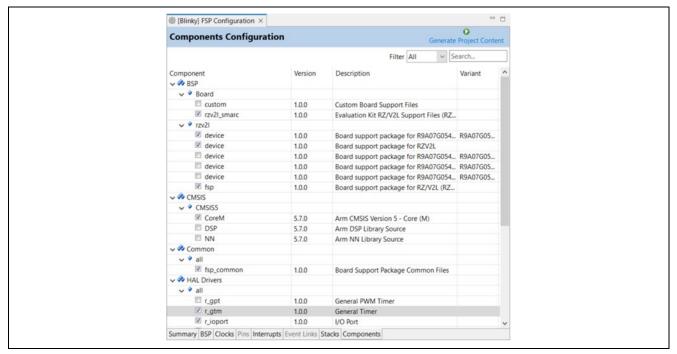


Figure 65 : Components Tab

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

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

 e^2 studio also creates configuration files in the rzv_cfg/fsp_cfg folder with configuration options set in the **Stacks** tab.

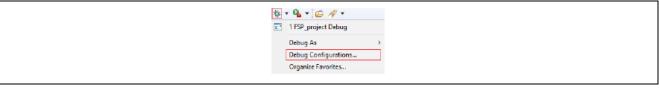


5.4 Debugging the Project

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.

	quations			15
Create, manage, and run con	ignations)
🗈 🖹 🖻 🛪 🗉 🗍	Name: MyProject Debug			
type filter text	Main 🎋 Debugger 🕨 Startup 🔲 🖸	ommon by Source		
C/C++ Application	Project:			
C/C++ Remote Applicati	MyProject			Browse
GDB Hardware Debuggin	C/C++ Application:			
GDB OpenOCD Debuggin GDB Simulator Debuggin	In a man to a			
Java Applet		Variables	Search Project	Browse
Java Application	Build (if required) before launching recate Build Configuration: Use Active tation			
Launch Group (Deprecation				
Remote Java Application				
 Renesas GDB Hardware D MyProject Debug 	Use workspace settings	Configure Works		
Renesas Simulator Debug				
< >				
Filter matched 14 of 16 items			Reyert	Apply
				Close
			Reyert Dobug	

 Set the Secure and Non-secure Vector Address in the Connection Settings tab of the Debugger tab. Click the "..." button in Secure Vector Address field and select "__Secure_Vectors" from the list in the Select Symbol window.

Also, click the "..." button in **Non-secure Vector Address** field and select "___Vectors" from the list in the **Select Symbol** window.

Debug Configurations			- 0 ×	Filter:	
Create, manage, and run configur	rations			Symbol	Address ^
			,	SystemInit_S	0x72eff549
🗋 🖻 🗫 🗎 🗶 🗖 🗸 🗸	Name: Blinky Debug_Flat			 UsageFault_Handler_NS 	0x6001084d
	-			UsageFault_Handler_S	0x72eff535
type filter text	📄 Main 🎋 Debugger 🔛 Startup 🔲 Comr	non 🤤 Source		Mann Poret S	0+72+#591
C C/C++ Application				Secure_Vectors	0x1001ff80
C/C++ Remote Application	Debug hardware: J-Link ARM V Target D	evice: R9A07G044L23GBG_1		Vectors	0x00010000
EASE Script				 _bsp_clock_init_veneer 	0x72eff5b9
C GDB Hardware Debugging	GDB Settings Connection Settings Debug	Tool Settings		bsp_clock_freq_var_init	0x600107b1
GDB OpenOCD Debugging	✓ JTAG Scan Chain		^	bsp_clock_init	0x60010835
GDB Simulator Debugging (I	Multiple Devices	No	*	bsp_init	0x600108d9 🗸
🜌 Java Applet	IRPre	0			
Java Application	DRPre	0			
🙀 Launch Group	✓ Connection			OK	Cancel
Remote Java Application	Register initialization	No	~		
✓ C Renesas GDB Hardware Deb	Reset on connection	No	×		
C [®] Blinky Debug_Flat	Reset before run	No	~		
En Renesas Simulator Debuggir	ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF		Select Symbol	×
	Hold reset during connect	No	~	- ,	
	Reset before download	No	~	Filter	
	Prevent Releasing the Reset of the CM	3 Core Yes	<u> </u>		
	Secure Vector Address			Symbol	Address ^
	Non-secure Vector Address			UsageFault_Handler_NS	0x6001084d
	Core clock (MHz)	0		UsageFault_Handler_S	0x72eff535
	v TrustZone	0		Warm_Reset_S	0x72eff581
	Set TrustZone secure/non-secure bour	idation No.	×	Course Masters	0-1001#80
	Set musizione secure/non-secure bour	luaries no	· •	Vectors	0x00010000
				bsp_clock_init_veneer	0x72e#5b9
< >				 bsp_clock_freq_var_init 	0x600107b1
Filter matched 13 of 15 items		Revert	Apply	 bsp_clock_init 	0x60010835
				 bsp_crock_init bsp_init 	0x600108d9
				bsp_init bsp_irg_cfg	0x60010849
		<u>D</u> ebug	Close	- usp_ird_crg	0x00010849
?					

These commands 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.



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

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



5.5 Modifying Toolchain Settings

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

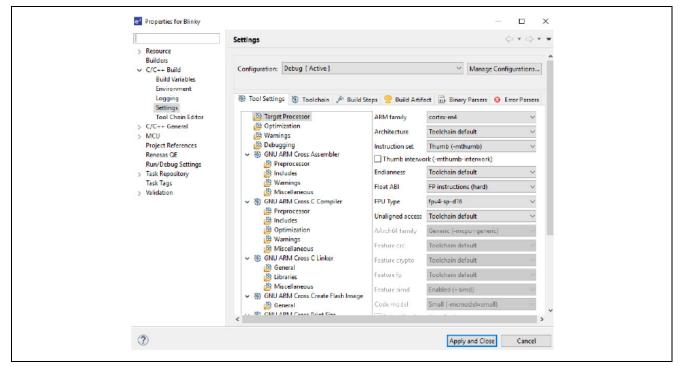


Figure 66 : e² studio Project toolchain settings

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 script folder (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.

e Eclipse Launcher		×
Select a directory as workspace		
\ensuremath{e}^2 studio uses the workspace directory to store its preferences and d	evelopment artifacts.	
Workspace: C:\Users\ <user_name\e2studio\workspace< td=""><td>~</td><td>Browse</td></user_name\e2studio\workspace<>	~	Browse
Use this as the default and do not ask again		
<u>R</u> ecent Workspaces		
	Launch	Cancel

Figure 67 : 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.

el 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</username>
Use this as the default and do not ask again
▶ <u>R</u> ecent Workspaces
Launch Cancel

Figure 68 : 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.

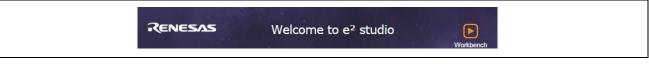


Figure 69 : Workbench arrow button

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.

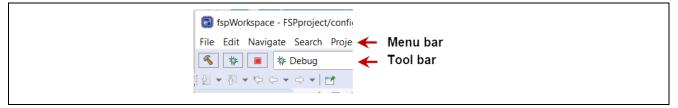


Figure 70 : Menu and tool bar



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

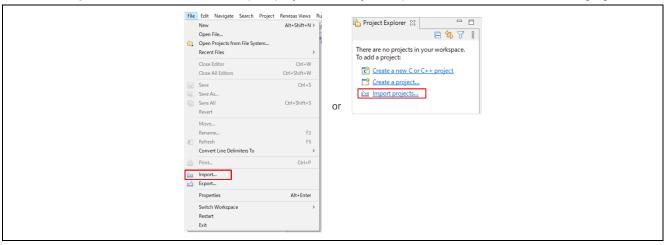


Figure 71 : File drop-down menu

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	– 🗆 X	
Select Create new projects from an archive file or directory.	25	
Select an import wizard: type filter text		
type filter text General CMSIS Pack CMSIS Pack File System File		
(Cancel	

Figure 72 : 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		-		\times
Import Projects				5
	rch for existing Eclipse projects.			7
	· · · · · ·		<u> </u>	-4
O1	1,			_
Select root directory:	J	~	Browse	
O Select archive file:		\sim	Browse	
Projects:				
			Select /	AII
			Deselect	All
			Refres	h
Options				
Search for nested pro				
Copy projects into we				
	ed projects upon completion			
Hide projects that alr	Iready exist in the workspace			
Working sets				
Add project to worki	king sets		New	
	-		Select	
Working sets:			Select	
?	< Back Next > Finish		Cance	
•			201111	

Figure 73 : Import Existing Project dialog 1 - Select root directory

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

Import				×
Import Projects Select a directory to sea				
 Select root directory: Select archive file: Projects: 	C:\works_RZV\work\Blinky	~		wse
Blinky (C:\works_RZV\work\Blinky)		[Desel	ct All ect All resh

Figure 74 : Import Existing Project dialog 2

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



Revision History

		Descriptio	n
Rev.	Date	Page	Summary
1.00	Jan.14.22	-	First Edition issued
1.10	1.10 Jan,31.23		Updated e2sutio version to install for Windows PC and changed images of e2studio installation
		16 to 20	Updated e2studio version to install for Linux PC and changed images of e2studio installation
		23 to 24	Updated pack version to install
		34 to 36	Changed Chapter 4.6 to 4.9 to a sub-chapter of Chapter 4.5 (These have been changed to Chapters 4.5.1 to 4.5.4)
		35, 49	Updated the method of specifying Secure Vector Address and Non-secure Vector Address



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 reaches the level at which resetting is specified.

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

- 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/.