

RZ/T2, **RZ/N2**

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/T2, RZ/N2 microprocessor series.

Target Device

RZ/T series: RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H RZ/N series: RZ/N2L

About the video contents

We provide videos about the development tools using the RZ/T and RZ/N FSP. Access the following links:

• How to install the development tools

- <u>RZ/T RZ/N FSP Quick Start Guide FSP Installation and Generating Your First Project for e2 studio</u> (English, Japanese, Chinese)
- <u>RZ/T RZ/N FSP Quick Start Guide FSP Installation & Generating Your First Project for EWARM & FSP SC</u>

(English, Japanese, Chinese)

- Instructions and usage of each tab in FSP Configuration
 - <u>RZ/T RZ/N FSP Tutorial Pin Configuration Function</u> (English, Japanese, Chinese)
 - <u>RZ/T RZ/N FSP Tutorial for FSP Configuration (1/2)</u> Introduction of Tabs (English, Japanese, Chinese)
 - <u>RZ/T RZ/N FSP Tutorial for FSP Configuration (2/2) How to Use</u> (English, Japanese, Chinese)



Contents

1.	Introduction	5
1.1	Overview	5
1.2	Introduction to FSP	5
1.2.1	Purpose	5
1.2.2	e ² studio IDE	5
1.2.3	FSP Smart Configurator	5
1.2.4	FSP Documentation	5
1.3	Related Documentation Files	6
1.3.1	Evaluation Board User's Manual	6
1.3.2	FSP Documentation	6
1.4	Starting Development Introduction	7
2.	Set up Evaluation Board	8
2.1	Obtaining an Evaluation Board	8
2.2	System Configuration	8
2.3	Supported Emulator	9
2.3.1	SEGGER J-Link	9
2.3.2	IAR I-Jet	9
2.4	RZ/T Series Board Setup	10
2.4.1	RSK+RZT2M	10
2.4.2	RSK+RZT2L	13
2.4.3	RSK+RZT2ME	15
2.4.4	RZ/T2H Evaluation Board	16
2.5	RZ/N Series Board Setup	19
2.5.1	RSK+RZN2L	19
3.	e ² studio Setup	22
3.1	What is e ² studio?	22
3.2	e ² studio Prerequisites	22
3.2.1	Windows PC Requirements	22
3.2.2	Installing e ² studio, Platform Installer and FSP Package	22
3.2.3	Choosing a Toolchain	22
3.2.4	Licensing	
4.	Tutorial: Your First RZ/T2, RZ/N2 MPU Project – Blinky	23
4.1	Tutorial Blinky	23
4.2	What Does Blinky Do?	23
4.3	Create a New Project for Blinky	24
4.3.1	Details about the Blinky Configuration	30
4.3.2	Configuring the Blinky Clocks	30



<u>RZ/T2, RZ/N</u>2

4.3.3	Configuring the Blinky Pins	
4.3.4	Configuring the Parameters for Blinky Components	
4.3.5	Where is main()?	
4.3.6	Blinky Example Code	
4.4	Build the Blinky Project	
4.4.1	Build	
4.4.2	Build for Multiprocessing	
4.5	Debug the Blinky Project	
4.5.1	Debug Prerequisites	
4.5.2	Debug Steps	
4.5.3	Details about the Debug Process	
4.6	Run the Blinky Project	
4.7	Debug and Run for Multiprocessing	
4.8	Import the Project	
5.	FSP Smart Configurator User Guide	42
5. 5.1	What is FSP Smart Configurator?	
5.2	Tutorial Blinky	
5.3	Using Smart Configurator with IAR EWARM	
5.3.1	Prerequisites	
5.3.2	Create a New Project	
5.3.3	Build the Project	
5.3.4	Download & Debug the Project	
5.3.5	Debug for Multiprocessing	
5.4	Re-configuring Project with FSP SC	
5.4.1	Launch FSP Smart Configurator from IAR EWARM	
5.4.2	Launch from the Command Prompt	
5.5	Note when debugging in different workspaces	
	FSP Configuration Users Guide	
6.1	What is a Project?	
6.2	Create a Project	
6.2.1	Creating a New Project	
6.2.2	Selecting a Board and Toolchain	
6.2.3	Selecting a Project Template	
6.2.4	Duplication of Resources	71
6.3	Configuring a Project	72
6.3.1	Summary Tab	72
6.3.2	Configuring the BSP	73
6.3.3	Configuring Clocks	74
6.3.4	Configuring Pins	
6.4	Configuring Interrupts from the Stacks Tab	



6.4.1	Creating Interrupts from the Interrupts Tab	78
6.4.2	Viewing Event Links	79
6.5	Adding and Configuring HAL Drivers	80
6.6	Reviewing and Adding Components	81
Appe	ndix. Known Issues	82
Appe	ndix. Tool Software Limitations	. 106
Appe	endix. How to Debug FSP Project with Flash Boot Mode	. 117
Appe	ndix. How to Erase Flash Memory	. 119
Appe	endix. How to Change Boot Mode of FSP Project	. 124
Appe	endix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e ² studio	. 127
Appe	endix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM .	. 132
Revis	sion History	. 137



1. Introduction

1.1 Overview

This application note describes how to use the Renesas Flexible Software Package (FSP) running on the Cortex®-R52 and Cortex®-A55 (hereinafter referred to as CR52 and CA55) incorporated on RZ/T2 and RZ/N2.

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 the hardware abstraction layer (HAL) drivers and the board support package (BSP) 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/T2, RZ/N2 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.

1.2.3 FSP Smart Configurator

The Renesas FSP Smart Configurator (FSP SC) is a desktop application designed to configure device hardware such as clock set up and pin assignment as well as initialization of FSP software components when using a 3rd-party IDE and toolchain.

For creating RZ/T2, RZ/N2 project, the FSP SC can currently be used with

• IAR Systems Embedded Workbench for Arm (IAR EWARM) with IAR toolchain for Arm

1.2.4 FSP Documentation

The related file "FSP Documentation" contains HTML documentations describing the features, APIs and usage notes regarding the BSP and HAL drivers implemented as FSP modules and interfaces. After clicking the "index.html" in "FSP Documentation" to open the introduction page on your html browser, the reference documents for utilizing each FSP module and interface can be read from "API Reference" menu.



1.3 Related Documentation Files

The related documentation files are shown in the following.

1.3.1 Evaluation Board User's Manual

This Getting Started Guide refers to the following "Evaluation Board User's Manual".

- RZ/T series
 - RZ/T2M Group Renesas Starter Kit+ for RZ/T2M User's Manual (RZ/T2M and RZ/T2ME)
 Document No. R20UT4939
 - RZ/T2L Group Renesas Starter Kit+ for RZ/T2L User's Manual
 Document No. R20UT5164
 - ► RZ/T2H Group RZ/T2H Evaluation Board User's Manual
 - Document No. **R20UT5405**
- RZ/N series
 - > RZ/N2L Group Renesas Starter Kit+ for RZ/N2L User's Manual
 - Document No. R20UT4984

These documents can be found on Renesas web site by inputting their **Document No.** into a search box.

• URL: <u>https://www.renesas.com/</u>



Figure 1: Search Box in Renesas Web Page

1.3.2 FSP Documentation

This Getting Started Guide refers to the following "FSP Documentation". It contains notes on the use of the software modules packaged with FSP.

These documents are available in Renesas Git repository in GitHub.

- RZ/T series
 - ▶ RZ/T2 Flexible Software Package Documentation
 - URL: https://github.com/renesas/rzt-fsp/releases
 - File name: fsp_documentation_vx.x.x.zip
- RZ/N series

٠

- > RZ/N2 Flexible Software Package Documentation
 - URL: https://github.com/renesas/rzn-fsp/releases
 - File name: fsp_documentation_vx.x.x.zip

Note:

The "vx.x.x" is the FSP version number such as "v1.0.0".



1.4 Starting Development Introduction

FSP application project can be created by e² studio or FSP SC (for IAR EWARM), and this Getting Started includes tutorial for both tools; the chapters you should read changes.

e² studio users should read the following chapters:

- Chapter 2 "Set up Evaluation Board"
- Chapter 3 "e² studio Setup"
- Chapter 4 "Tutorial: Your First RZ/T2, RZ/N2 MPU Project Blinky"
- Chapter 6 "FSP Configuration Users Guide"

FSP SC users (for IAR EWARM users) should read the following chapters:

- Chapter 2 "Set up Evaluation Board"
- Chapter 5 "FSP Smart Configurator User Guide"
- Chapter 6 "FSP Configuration Users Guide"

The summary of each chapter is shown below.

- Chapter 2 "Set up Evaluation Board"
 - Explains how to setup Evaluation Board to proceed the tutorials in Chapter 4 and 5.
- Chapter 3 "e² studio Setup"
 - > Explains the setup of e^2 studio for utilizing FSP.
- Chapter 4 "Tutorial: Your First RZ/T2, RZ/N2 MPU Project Blinky"
- \blacktriangleright Explains the tutorial with minimal steps to create, run, and debug a FSP project by using e² studio.
- Chapter 5 "FSP Smart Configurator User Guide"
 - Explains the tutorial with minimal steps to create an FSP project as IAR EWARM project by using the FSP Smart Configurator and to run and debug the created IAR EWARM project.
- Chapter 6 "FSP Configuration Users Guide"
 - > Explains how to create and configure an FSP project in detail.
 - The explanation is described based on e² studio, but most of the explanations are applied to the FSP smart configurator.



2. Set up Evaluation Board

2.1 Obtaining an Evaluation Board

To develop applications with RZ/T2 FSP and RZ/N2 FSP, start with Evaluation Board and Renesas Starter Kit+ (RSK+).

The Evaluation Board and Renesas Starter Kit+ for RZ/T2 and RZ/N2 CPU Board are designed to seamlessly integrate with the e^2 studio.

Ordering information, User's Manuals, and other related documents for boards are available. Please contact Renesas to get them.

2.2 System Configuration

Below is an example of a typical system configuration of evaluation board.

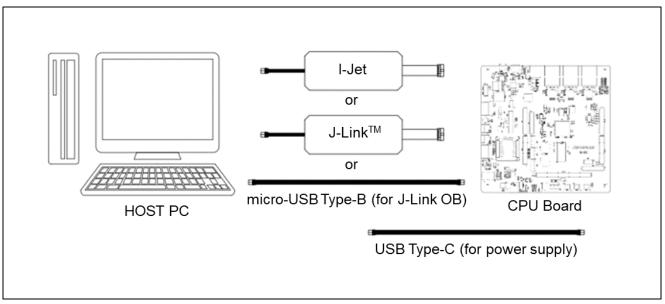


Figure 2: System Configuration Example – with Evaluation Board

For the details, please refer to the related document "1.3.1 Evaluation Board User's Manual".



2.3 Supported Emulator

2.3.1 SEGGER J-Link

SEGGER J-Link can be used on Renesas e² studio only for debugging on RZ/T2 and RZ/N2 devices. Renesas e² studio supports the following emulators.

- J-Link EDU V11 and later
- J-Link BASE V11 and later
- J-Link PLUS V11 and later
- J-Link WiFi V1 and later
- J-Link ULTRA+ V5 and later
- J-Link PRO V5 and later
- J-Link OB-S124 V1.00

Renesas has tested debugging RZ/T2 and RZ/N2 devices with J-Link BASE V11 and J-Link OB-S124. For the details on SEGGER J-Link, please see SEGGER website.

Debugging FSP Project was verified with the following software environment.

Table 1 Verified Operating Environment

Series	Device	FSP version	e ² studio version	J-Link Software version
RZ/T	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H	RZ/T2 FSP v2.2.0	2024-10	V7.98c
RZ/N	RZ/N2L	RZ/N2L FSP v2.0.0	2024-01.1	V7.94h

Regarding how to update J-Link firmware, please confirm the procedure described in the following link into Renesas Knowledge Base web site.

https://en-support.renesas.com/knowledgeBase/20736714

2.3.2 IAR I-Jet

IAR I-jet can be used on IAR EWARM only for debugging on RZ/T2 and RZ/N2 devices. For the details on IAR I-jet, please see IAR Systems website.



2.4 RZ/T Series Board Setup

2.4.1 RSK+RZT2M

2.4.1.1 Boot Mode

The operation mode settings for the RSK+RZT2M board are as follows.

Note:

This section shows the settings for running on RAM without external flash memory. For settings to run in other boot modes, please refer to the manual of the RSK boards listed in chapter 1.3.1. For <u>the sample codes</u> <u>available on Renesas web site</u>, please refer to the documentation included with each code and implement the appropriate board settings respectively.

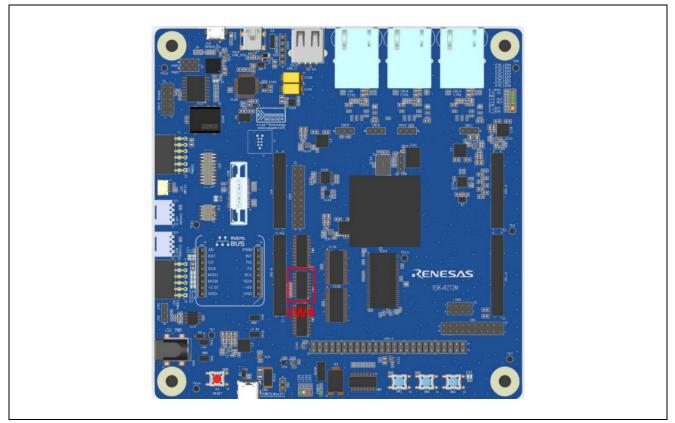


Figure 3: Switch Position of Operation Mode Settings for RSK+RZT2M

Table 2	Operation Mode Switch Settings for RSK+RZT2M

Switch	Setting	Description
SW4.1	ON	16-bit bus boot mode (NOR Flash)
SW4.2	OFF	
SW4.3	ON	
SW4.4	ON	JTAG Authentication by Hash is disabled.
SW4.5	ON	ATCM 0 wait
		Valid for CPU operating frequency equal to or less than 400MHz.



2.4.1.2 Debugger Connection

If you use JTAG connection with I-Jet or J-Link,

- 1. Short the jumper pin (J9) for switching the debug connection so that RSK+RZT2M board can use the emulator connected to JTAG connector (J20).
- 2. Connect the emulator (J-Link or I-jet) to a free USB port on your computer.
- 3. Connect the IAR I-Jet to the RSK+RZT2M board ensuring that it is plugged in to the header "J20".

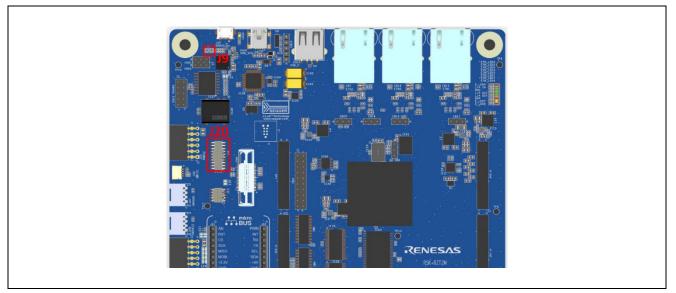


Figure 4: Jumper Position of JTAG Connection for RSK+RZT2M

If you use J-Link OB on RSK+RZT2M board,

- 1. Open the jumper pin (J9) for switching the debug connection so that RSK+RZT2M can use J-Link OB on the board.
- 2. Connect the micro-USB type-B to J-Link OB USB connector (J10), and then the LED4 is lighted.

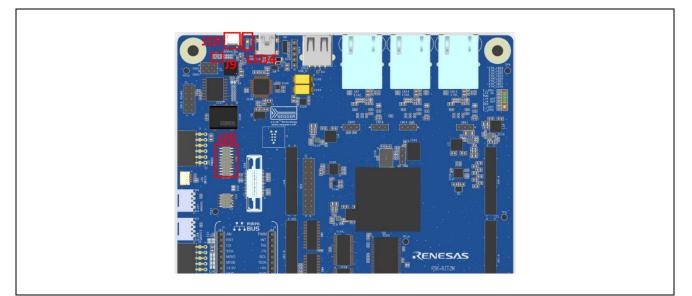


Figure 5: J-Link OB Connection Settings for RSK+RZT2M



2.4.1.3 Power Supply

Power is supplied using a USB cable (Type-C) or an AC / DC adapter.

- When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RSK+RZT2M board.
- When connecting the AC / DC adapter, connect it to the USB connector "CN6" of the RSK+RZT2M board.

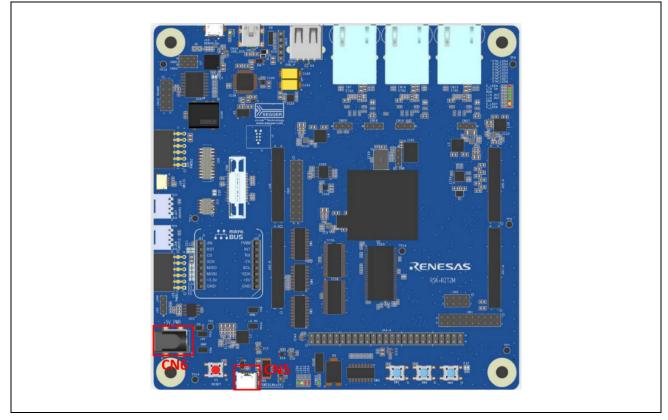


Figure 6: How to Power Supply for RSK+RZT2M



2.4.2 RSK+RZT2L

2.4.2.1 Boot Mode

The operation mode settings for the RSK+RZT2L board are as follows.

Note:

This section shows the settings for running on RAM without external flash memory. For settings to run in other boot modes, please refer to the manual of the RSK boards listed in chapter 1.3.1. For <u>the sample codes</u> <u>available on Renesas web site</u>, please refer to the documentation included with each code and implement the appropriate board settings respectively.

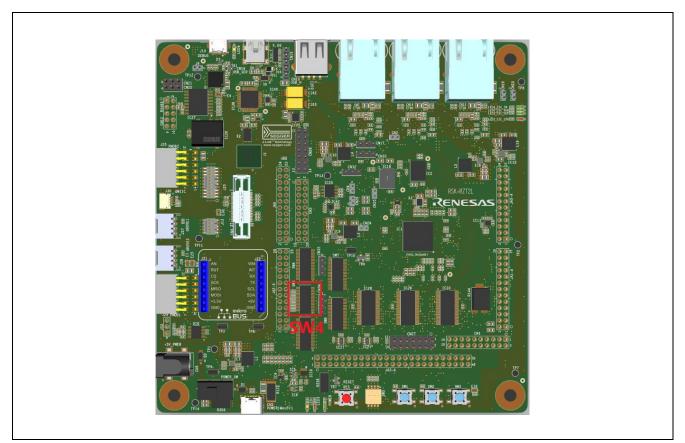


Figure 7: Switch Position of Operation Mode Settings for RSK+RZT2L

Table 3 Operation Mode Swi	tch Settings for RSK+RZT2L
----------------------------	----------------------------

Switch	Setting	Description
SW4.1	ON	xSPI0 boot mode (x1 boot serial flash)
SW4.2	ON	
SW4.3	ON	
SW4.4	ON	ATCM wait cycle = 0 wait
SW4.5	ON	JTAG mode = Normal mode



2.4.2.2 Debugger Connection

If you use JTAG connection with I-Jet or J-Link,

- 1. Short the jumper pin (J9) for switching the debug connection so that RSK+RZT2L board can use the emulator connected to JTAG connector (J20).
- 2. Connect the emulator (J-Link or I-jet) to a free USB port on your computer.
- 3. Connect the IAR I-Jet to the RSK+RZT2L board ensuring that it is plugged in to the header "J20".

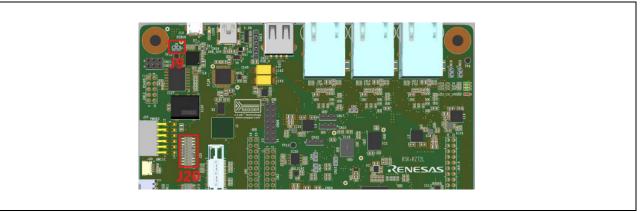


Figure 8: Jumper Position of JTAG Connection for RSK+RZT2L

If you use J-Link OB on RSK+RZT2L board,

- 1. Open the jumper pin (J9) for switching the debug connection so that RSK+RZT2L can use J-Link OB on the board.
- 2. Connect the micro-USB type-B to J-Link OB USB connector (J10), and then the LED6 is lighted.

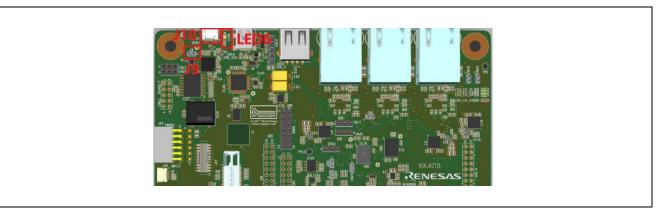


Figure 9: J-Link OB Connection Settings for RSK+RZT2L



2.4.2.3 Power Supply

Power is supplied using a USB cable (Type-C) or an AC / DC adapter.

- When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RSK+RZT2L board.
- When connecting the AC / DC adapter, connect it to the USB connector "CN6" of the RSK+RZT2L board.
- After connecting to the power (CN5 or CN6), turn on the POWER_SW slide switch to start power supply.

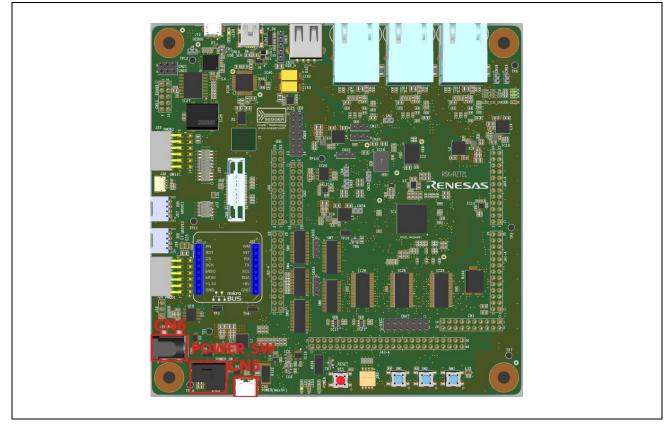


Figure 10: How to Power Supply for RSK+RZT2L

2.4.3 RSK+RZT2ME

For each setting, see 2.4.1 RSK+RZT2M.



2.4.4 RZ/T2H Evaluation Board

2.4.4.1 Boot Mode

The operation mode settings for the RZ/T2H evaluation board are as follows.

Note:

This section shows the settings for running on RAM without external flash memory. For settings to run in other boot modes, please refer to the manual of the evaluation board listed in chapter 1.3.1. For <u>the sample</u> <u>codes available on Renessa web site</u>, please refer to the documentation included with each code and implement the appropriate board settings respectively.



Figure 11: Switch Position of Operation Mode Settings for RZ/T2H Evaluation Board

Switch	Setting	Description
SW14.1	ON	xSPI1 boot mode (x1 boot serial flash)
SW14.2	OFF	
SW14.3	ON	
SW14.4	ON	CPU0 ATCM 0 wait
SW14.7	ON	JTAG Authentication by Hash is disabled.
SW2.3	OFF	This is necessary to light up LED3 (corresponding to CA55 Core1 blinky operation). Note: This switch is not present on the provisional version of the board. Due to this setting, P17_4, P08_5, and P08_6 cannot be used as SD1 control terminals.

Table 4 Operation Mode Switch Settings for RZ/T2H Eva	lustion Board
Table + Operation Mode Switch Settings for RE/ 1211 Eva	Iuanon Doaru



2.4.4.2 Debugger Connection

If you use JTAG connection with I-Jet or J-Link,

- 1) Short the jumper block (CN62) for switching the debug connection so that RZ/T2H evaluation board can use the emulator connected to JTAG connector (CN61).
- 2) Connect the emulator (J-Link or I-jet) to a free USB port on your computer.
- 3) Connect the emulator to the RZ/T2H evaluation board ensuring that it is plugged in to the header "CN61".

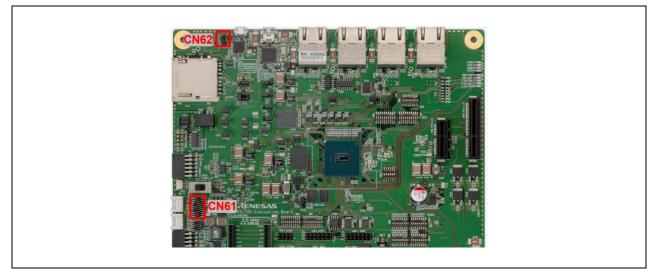


Figure 12: Jumper Position of JTAG Connection for RZ/T2H Evaluation Board

If you use J-Link OB on RZ/T2H evaluation board,

- 1) Open the jumper block (CN62) for switching the debug connection so that RZ/T2H evaluation board can use J-Link OB on the board.
- 2) Connect the micro-USB type-B to J-Link OB USB connector (CN14), and then the LED10 is lighted.

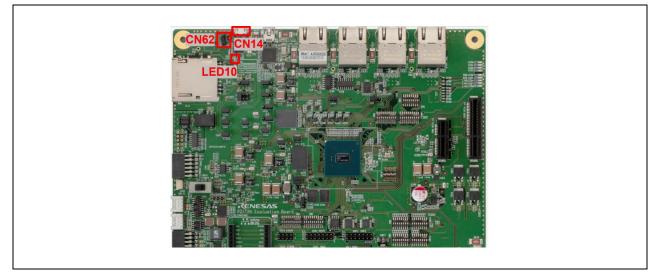


Figure 13: J-Link OB Connection Settings for RZ/T2H Evaluation Board



2.4.4.3 Power Supply

Power is supplied using a USB cable (Type-C) or an AC / DC adapter.

- When using a USB cable (Type-C), connect it to the USB connector "CN46" of the RZ/T2H evaluation board.
- When connecting the AC / DC adapter, connect it to the USB connector "CN47" of the RZ/T2H evaluation board.
- After connecting to the power (CN46 or CN47), turn on the POWER_SW slide switch to start power supply. Note:

Some Renesas boards, such as the Renesas Starter Kit, require a 12-V or 5-V power supply, the supply of this board is 15-V/3 A. Be careful not to accidentally connect a 12-V or 5-V power supply. When supplying power through CN47, use a stabilized power source that is capable of supplying at least 15-V/3 A.

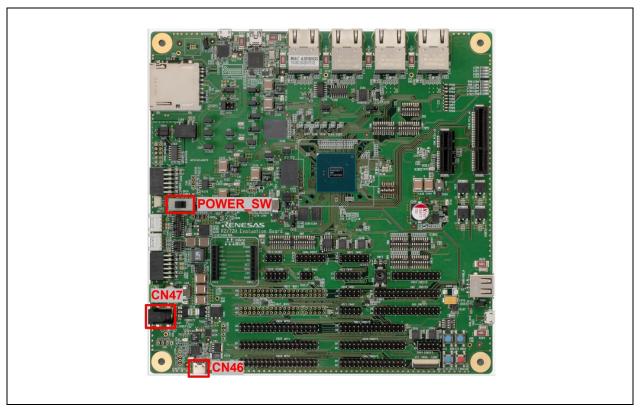


Figure 14: How to Power Supply for RZ/T2H Evaluation Board



2.5 RZ/N Series Board Setup

2.5.1 RSK+RZN2L

2.5.1.1 Boot Mode

The operation mode settings for the RSK+RZN2L board are as follows.

Note:

This section shows the settings for running on RAM without external flash memory. For settings to run in other boot modes, please refer to the manual of the RSK boards listed in chapter 1.3.1. For <u>the sample codes</u> available on Renesas web site, please refer to the documentation included with each code and implement the appropriate board settings respectively.

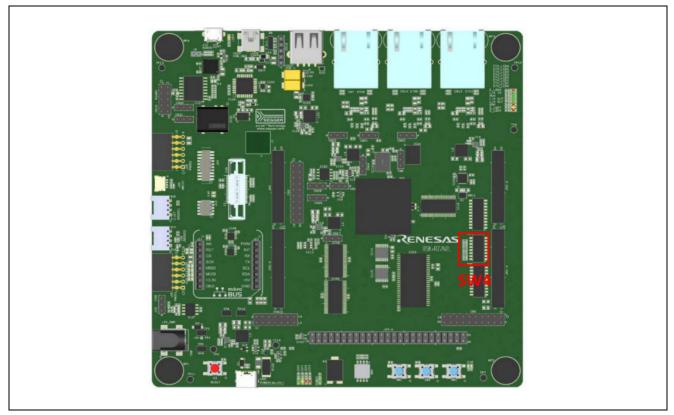


Figure 15: Switch Position of Operation Mode Settings for RSK+RZN2L

Switch	Setting	Description
SW4.1	ON	16-bit bus boot mode (NOR flash)
SW4.2	OFF	
SW4.3	ON	
SW4.4	ON	JTAG Authentication by Hash is disabled.



2.5.1.2 Debugger Connection

If you use JTAG connection with I-Jet or J-Link,

- 1. Short the jumper pin (J9) for switching the debug connection so that RSK+RZN2L board can use the emulator connected to JTAG connector (J20).
- 2. Connect the Emulator (J-Link or I-jet) to a free USB port on your computer.
- 3. Connect the IAR I-Jet to the RSK+RZN2L board ensuring that it is plugged in to the header "J20".

The figure below is when a I-jet is used as Emulator.

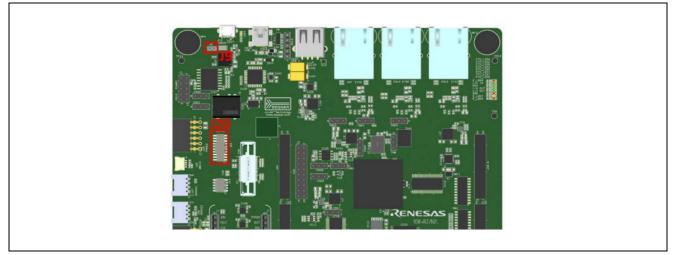


Figure 16: Jumper Position of JTAG Connection for RSK+RZN2L

If you use J-Link OB on RSK+RZN2L board,

- 1. Open the jumper pin (J9) for switching the debug connection so that RSK+RZN2L can use J-Link OB on the board.
- 2. Connect the micro-USB type-B to J-Link OB USB connector (J10), and then the LED4 is lighted.

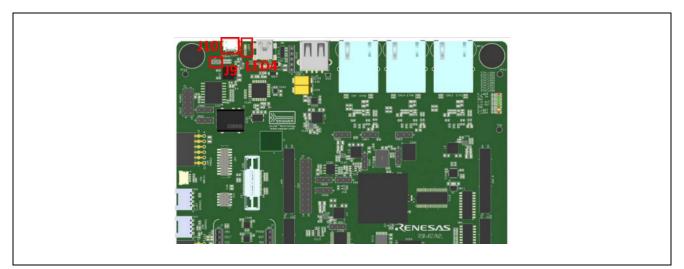


Figure 17: J-Link OB Connection Settings for RSK+RZN2L



2.5.1.3 Power Supply

Power is supplied using a USB cable (Type-C) or an AC / DC adapter.

- When using a USB cable (Type-C), connect it to the USB connector "CN5" of the RSK+RZN2L board.
- When connecting the AC / DC adapter, connect it to the USB connector "CN6" of the RSK+RZN2L board.

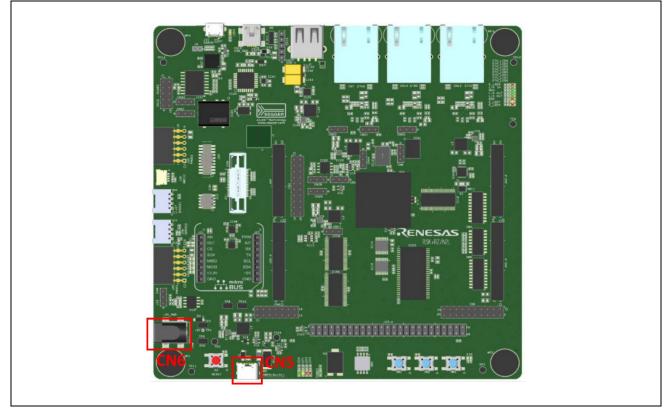


Figure 18: How to Power Supply for RSK+RZN2L



3. e² studio Setup

3.1 What is e² studio?

Renesas e^2 studio is a development tool encompassing code development, build, and debug. e^2 studio is based on the open-source Eclipse IDE and the associated C/C++ Development Tooling (CDT).

When developing for RZ/T2, RZ/N2 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.

3.2 e² studio Prerequisites

3.2.1 Windows PC Requirements

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

For Windows 64-bit version

- Windows® 11 (64-bit version)
- Windows® 10 (64-bit version)
- Windows® 8.1 (64-bit version)
- Memory capacity: We recommend 8 GB or more. At least 4 GB.
- Capacity of hard disk: At least 2 GB of free space.
- Display: Graphics resolution should be at least 1024 x 768, and the mode should display at least 65,536 colors.
- Interface: USB 2.0
- Microsoft Visual C++ 2010 SP1 runtime library *1
- Microsoft Visual C++ 2015-2019 runtime library *1
- *1. This software will be installed at the same time as the e² studio.

3.2.2 Installing e² studio, Platform Installer and FSP Package

Detailed installation instructions for the e^2 studio and the FSP are available on the Renesas website. Review the release notes for e^2 studio to ensure that the e^2 studio version supports the selected FSP version. The starting version of the installer includes all features of the RZ/T2, RZ/N2 MPUs.

3.2.3 Choosing a Toolchain

The GNU ARM Embedded Toolchain (version 12.2.1.arm-12-24 for CR52) and GNU ARM A-Profile (AArch64 baremetal) (version 10.3.1.20210621 for CA55) are required.

If the version of the toolchain has not been installed, please download the toolchain from ARM Developer website, and install it.

3.2.4 Licensing

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



4. Tutorial: Your First RZ/T2, RZ/N2 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/T2, RZ/N2 evaluation board. This chapter guides you through creating projects for a single-core processing and a multiprocessing with RAM execution without flash memory. In this chapter, the multiprocessing refers to a process in which CR52 CPU0 core is activated first and second core (CR52 CPU1 or CA55 Core0) operates after CR52 CPU0 core sets up for second core.

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.



4.3 Create a New Project for Blinky

The creation and configuration of an RZ/T and RZ/N C/C++ FSP Project is the first step in the creation of an application. The base RZ/T2 pack and RZ/N2 packs include a pre-written Blinky example application. In the case of multiprocessing, two projects with different settings must be created. A project that starts first is called the primary project and the secondary project that runs after releasing reset by the primary project is called the secondary project.

Note for multiprocessing projects:

The primary project and the secondary project should be created in the same workspace.

The secondary project should be created after the primary project is created in 4.3 section and built the primary project in 4.4 section.

Follow these steps to create an RZ/T2, RZ/N2 MPU project:

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

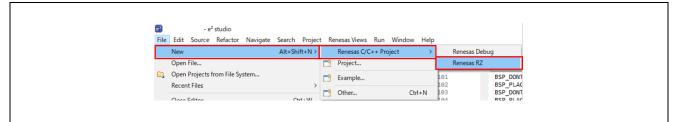


Figure 19: New C/C++ Project

- 2. Select either one depending on your RZ/T2, RZ/N2 MPU.
 - RZ/T series: All > Renesas RZ/T C/C++ FSP Project
 - RZ/N series: All > Renesas RZ/N C/C++ FSP Project
- 3. Click Next.

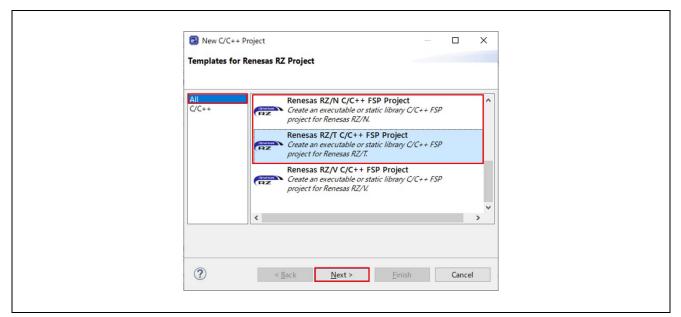


Figure 20: Renesas RZ C/C++ FSP Project



4. Assign a name to this new project. An example of naming is shown below.

	14		Ty Created Project	(I)	
	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
Project name	Blinky	Blinky_primary	Blinky_secondary	Blinky_primary	Blinky_secondary

 Table 6 e² studio Newly Created Project Settings (1)

5. Click Next. The Project Configuration window shows your selection.

				_		
Pr	oject Name and Locat				Ď	
	Project name Blinky Use default locati Location:	¥	¥Blinky here		Browse	
C	D [< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel	

Figure 21 : e² studio Project Configuration Window (Part 1)



- 6. Select the board support package by selecting the name of your board from the drop-down list. In this tutorial, please select either one depending on your device and board.
- 7. (Multicore device ONLY) Select the Core from the drop-down list.
- 8. Select toolchains and version, then click Next.
 - If there is NOT the target toolchain, please download the version of the toolchain from ARM Developer website and install it.

	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
Board			ithout flash memory) M execution without	flash memory)	
Core	CR52_0 or CR52 CPU0	CR52_0 or CR52 CPU0	CR52_1 or CR52 CPU1	CR52 CPU0	CA55 Core0
Toolchains	GNU ARM Er	nbedded and 12.2.1.a	arm-12-24		GNU ARM A- Profile (AArch64 bare-metal) and 10.3.1.20210621

 Table 7 e² studio Newly Created Project Settings (2)

Renesas RZ/T C/C++ FSP Project	- 0
Renesas RZ/T C/C++ FSP Project	
Device and Tools Selection	
Device Selection	
FSP Version: <fsp version=""></fsp>	Board Description
Board: RZT2H Evaluation Board (RAM execution with	RZT2H Evaluation Board (RAM execution without flash memory)
Device: R9A09G077M44GBG	
Core: CR52 CPU0	Device Details
Language: OC OC++	TrustZone No Pins 729
	Processor Cortex-R52
Toolchains	Debugger
GNU ARM Embedded	J-Link ARM
<toolchain version=""> V Manage To</toolchain>	atabaina
< routinain version>	orchams
?	< <u>Back Next > Einish</u> Cancel

Figure 22 : e² studio Project Configuration Window (Part 2)

9. (The secondary project of multiprocessing ONLY) Select a bundle file of the primary project as Proceeding Project. Built the primary project names in the same workspace appear as an option in the drop-down list.



Note:

Warnings occur if the FSP version or Board (boot mode) used is different between the primary project and the secondary project. Use the same FSP version and Board (boot mode).

Warnings occur when cores of the primary project and the secondary project are different in multiprocessing, because Toolchain and Toolchain version do not match. Ignore the warning and proceed to the next step.

Preceding Project or Smart Bundle Selection Preceding Project Blinky_primary Choose this option if you have access to the project source code of the preceding processor core or security context. O Smart Bundle: Resolved location: Workspace File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details FSP version GNU ARM Embedded Toolchain version Board A Soard (boot mode)> Device Resolved (boot mode)> Device CR52_0 Zones CR52_0	Renesas RZ/T C/C++	FSP Project				
Choose this option if you have access to the project source code of the preceding processor core or security context. O Smart Bundle Resolved location: Workspace File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Variables Variables Preceding Project/Smart Bundle Details FSP version > StSP Version > Toolchain version Toolchain version Board Device R9A096077M44G8G Core CR52_0						\square
Smart Bundle: Resolved location: Workspace File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details FSP version GNU ARM Embedded Toolchain version GNU ARM Embedded Toolchain version Goard	Preceding Project:	Blinky_primary				v
Besolved location: Workspace File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. File System Variables Preceding Project/Smart Bundle Details FSP version > Toolchain GNU ARM Embedded Toolchain version <toolchain version=""> Board <6bard (boot mode) > Device R8A09G077M44G8G</toolchain>		Choose this option if y	ou have access to the project s	ource code of the preceding p	processor core or se	ecurity context.
Workspace File System Variables Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context. Preceding Project/Smart Bundle Details Preceding Project/Smart Bundle Details FSP version > Colohain GRU ARM Embedded FSP version > FSP version > <td< td=""><td>O Smart Bundle:</td><td>Resolved Incation:</td><td></td><td></td><td></td><td></td></td<>	O Smart Bundle:	Resolved Incation:				
core or security context. Preceding Project/Smart Bundle Details FSP version <fsp version=""> Toolchain GNU ARM Embedded Toolchain version <toolchain version=""> Board <toolchain version=""> Device R9A09G077M44G8G Core CR52_0</toolchain></toolchain></fsp>				Workspace	File System	Variables
FSP version <fsp version=""> Toolchain GNU ARM Embedded Toolchain version <toolchain version=""> Board <board (boot="" mode)=""> Device RA090077M44GBG Core CR52_0</board></toolchain></fsp>				Bundle describing the config	uration of the prec	eding processor
Toolchain GNU ARM Embedded Toolchain version <toolchain version=""> Board <board (boot="" mode)=""> Device RBA09G077M4468G Core CR52_0</board></toolchain>	Preceding Project/Sm	art Bundle Details				
	Toolchain Toolchain version Board Device Core		GNU ARM Embedded «Toolchain Version» «Board (boot mode)» R9A09G077M44GBG CR52_0			

Figure 23 e² studio Project Configuration Window (Part 3)

10. Select the **Build artifact** and RTOS.

Renesas RZ/T C/C++ FSP Project	— — X	
Renesas RZ/T C/C++ FSP Project Build Artifact and RTOS Selection		
Build Artifact Selection	RTOS Selection No RTOS	
0	< Back Next > Finish Cancel	
U	Sourk Next > Times Cancel	

Figure 24 : e² studio Project Configuration Window (Part 4)



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

Renesas RZ/T C/C++ FSP Project Renesas RZ/T C/C++ FSP Project	- □ X
Project Template Selection	2
Project Template Selection	
Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This Cruntime environment. [Renesss.RZIT. <fsp version="">.pack]</fsp>	project will initialize clocks, pins, stacks, and the
Bare Metal - Minimal Bare metal FSP project that includes BSR This project will initialize clocks, pin [Renesas.RZT, <fsp version="">.pack]</fsp>	s, stacks, and the C runtime environment.
Code Generation Settings	
(?) < <u>B</u> ack	Next > Einish Cancel

Figure 25 : e² studio Project Configuration Window (Part 5)

Once the project has been created, the name of the project will show up in the Project Explorer window of e² studio.

Note for the primary project using CR52 CPU0:

If the primary project selects CR52 CPU0 as **Core** and the secondary or later project uses a CA55 core, you need to set "PLL0 is released from standby state" and enable PLL0 in the Clocks tab of FSP Configuration.

	,/configuration.xml -e²studio Edit Navigate Search Project Renessa Views Run Window Help છે ≪ ଐ → () () () () → √ () →	- ロ × Q. 11部1 配 C/C++ (図 FSP Configuration 体 Debug
8 10 10	I 1FSP Configuration × Clocks Configuration Main Clock: 25MHz >PLL0 1200MHz	CASSOCILK ISSOMHZ
• •	PLO is standby state LLO is shallow state CLMAD error not mask MMF: 11.7/Hz MMR: 1 Modulation rate Offic	
	⇒ PLL1 1900/Hz CLMA1 Enabled ~ ↓ CLMA1 error not mask ~	CASSCLIK SOUMH2 CRE2COLIK SOUMH2 CRE2CILIK SOUMH2 CRE2CILIK SOUMH2 CRE2CILIKA 200MH2
	c Summary BSP Clocks Pins Interrupts Event Links Stacks Components	

Figure 26 : Enable PLL0 in the Primary Project using CR52 CPU0



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 Configurat	ion X		- <i>a</i>
Summary			Generate Project Content
Project Summary	,	2000000	^
Board:	RZT2H Evaluation Board (RAM execution without flash memory)	RENESAS	`
Device:	R9A09G077M44GBG		
Core:	CR52 CPU0		
Toolchain:	GCC for Renesas RZ		
Toolchain Version:			
FSP Version:	<esp varsion=""></esp>		
Project Ty			
Location:	/Blinky 🚭		

Figure 27 : 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 6.3.3 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 some of LEDs are configured by e^2 studio for the Blinky application.

The pin configuration tab shows the pin configuration for the Blinky application (see 6.3.4 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_ioport

To see the configuration parameters for any of the components, check the Properties tab in the HAL window for the respective drivers (see 6.5 Adding and Configuring HAL Drivers).

4.3.5 Where is main()?

The main function is located in:

- < RZT2 FSP project >/rzt_gen/main.c.
- < RZN2 FSP project >/rzn_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 6.5 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 e^2 studio when you select the Blinky Project template and is located in the project's folder < project >/src/ folder.

The application performs the following steps:

- 1. Get the LED information for the selected board by **bsp_leds_t** structure.
- 2. Initialize output level for LED pin to LOW using R_BSP_PinClear((bsp_io_region_t) leds.p_leds[i][1], (bsp_io_port_pin_t) leds.p_leds[i][0]).
- 3. Use **R_BSP_PinToggle ((bsp_io_region_t) leds.p_leds[i][1], (bsp_io_port_pin_t) leds.p_leds[i][0])** to set the output level to the LED pin.
- 4. **R_BSP_SoftwareDelay(delay, bsp_delay_units)** waits for a certain period of time. Then run #3 again.



4.4 Build the Blinky Project

Highlight the new project in the Project Explorer window by clicking on it and build it. When multiprocessing, please refer to Section 4.4.2 Build for Multiprocessing.

4.4.1 Build

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**.

🕲 workspace - Blinky/configuration.xml - e² studio		a. Project -> Build Project
File Edit Navigate Search Project Renesas Views Run Window Help		
🔦 🏠 🔳 🎋 Debug 🗸 🖻 Blinky Debug_Flat	🗸 🌞 i 🖻 🕶 🔚 🐚 i 🛞 🔹 🦠 🕶	b. Click hammer icon
🔁 Project Explorer 🗙 🕞 🛱 🖓 🖇 🖳 🗖	德 *[Blinky] FSP Configuration 🛛	c. Right click -> Build project
> 🧾 Blinky 🔸	Summary	

Figure 28 : 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 🗙	X ↓ ↑ <
T Build Console [Blinky]	
<pre>rm-none-eabi-gcc -mcpu=cortex-r52 -mthumb -mfloat-abi=hard -mfpu=neon-fp-armv8 rm-none-eabi-gcc -mcpu=cortex-r52 -mthumb -mfloat-abi=hard -mfpu=neon-fp-armv8 m-none-eabi-gcc @Blinky.elf.in" m-none-eabi-gcc @Blinky.elf.in" m-none-eabi-sizeformat=berkeley "Blinky.elf" text data bss dec hex filename 3004 0 65560 68564 10bd4 Blinky.elf</pre>	-fdiagnostics-parseable-fixits -Og -fmessage-length=0 -fdiagnostics-parseable-fixits -Og -fmessage-length=0 -fdiagnostics-parseable-fixits -Og -fmessage-length=0 -fdiagnostics-parseable-fixits -Og -fmessage-length=0
:46:51 Build Finished. 0 errors, 0 warnings. (took 12s.789ms)	

Figure 29 : e² studio Project Build Console

4.4.2 Build for Multiprocessing

Build the projects for multiprocessing with the following steps.

- 1. Create and build the primary project. (1st build of the primary project) Refer to 4.3 Create a New Project for Blinky and 4.4.1 Build.
- 2. Create the secondary project and build it.
- 3. Build the primary project again. (2nd build of the primary project)



4.5 Debug the Blinky Project

4.5.1 Debug Prerequisites

To debug the project on a board, you need the following:

- The board to be connected to e^2 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 of your microprocessor. To run or debug the application, the application must first be programmed to ram by JTAG debugger.

Evaluation board has a JTAG header and requires an external JTAG debugger to the header.

4.5.2 Debug Steps

When multiprocessing, please refer to Section 4.7 Debug and Run for Multiprocessing.

Note:

The main chapter of this documentation describes a RAM execution without flash memory project. When debugging a project with flash boot mode, please also refer to Appendix. How to Debug FSP Project with Flash Boot Mode.

To debug the Blinky application, follow these steps. If the step is preceded by (XXX), it is executed only if the condition is met.

(RAM exec): The boot mode used in the project is RAM execution without flash memory.

(CR52): The core used in the project is CR52.

(CA55): The core used in the project is CA55.

1. Configure the debugger for your project by clicking **Run** > **Debugger Configurations** ... or by selecting the dropdown menu next to the bug icon and selecting **Debugger Configurations** ...

Run	Window Help	
	Renesas Device Partition Manage	r
T	TraceX	>
Ð	Tracealyzer	>
0	Run	Ctrl+F11
杨	Debug	F11
	Run History	>
-	Run As	>
	Run Configurations	
	Debug History	>
*	Debug As	>
	Debug Configurations	
9	External Tools	>

Figure 30 : e² studio Debugger Configurations Selection Option

1	祢 、	• 🤷 • 🗄 0₀ • 🗞 🕪 💷 🖡	
		(no launch history)	
		Debug As	>
		Debug Configurations	
		Organize Favorites	

Figure 31 : e² studio Debug Icon



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.

Create, manage, and run configu	rations
	, and the second se
🕻 🖻 🍖 🗊 🗙 🖻 🔬 🗸	Name: Blinky Debug_Flat
type filter text	Main 🕸 Debugger 🔛 Startup 🤤 Source 🔲 Common
 C/C++ Application C/C++ Remote Application 	Project
EASE Script	Blinky Browse
C GDB Hardware Debugging	C/C++ Application:
GDB OpenOCD Debugging GDB Simulator Debugging (I	Debug/Blinky.elf
Java Applet Java Application Lunck Group Remote Java Application Renesas GDB Hardware Deb Binky Debug Flat Renesas Simulator Debuggir	Build (if required) before launching Browse Build Configuration: Use Active © Enable auto build O Disable auto build @ Use workspace settings Configure Workspace Settings
Filter matched 13 of 15 items	Revert Apply Debug Close

Figure 32 : e² studio Debugger Configurations Window with Blinky Project

- 3. If you use RAM execution without flash memory boot mode, it needs following configuration.
 - Debugger > Connection Settings > Connection
 - (RAM exec) Set No to Reset after download to avoid resetting MPU after program download
 - (CR52 CPU0) Set Yes to Set CPSR(5bit) after download to set the CPSR register value of CR52 general register before running the application.

	Single-core processing	Multiprocessing (CR52 CPU0, CR52 CPU1)		Multiprocessing (CR52 CPU0, CA55 Core0)	
		Primary	Secondary	Primary	Secondary
Reset after download	No (default)				
Set CPSR(5bit) after download	Yes	Yes	No (default)	Yes	No (default)

 Table 8 e² studio Newly Created Project Debug Settings (1)



Create, manage, and run configur	stions		
	nions		Ś
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Name: Blinky Debug_Flat		
type filter text	Main 🗱 Debugger 🍉 Startup 🔤 Source	Common	
C C/C++ Application C C/C++ Remote Application EASE Script GDB Hardware Debugging GDB Simulator Debugging (RHi	Debug hardware: J-Link ARM V Target Devi GDB Settings Connection Settings Debug To		
Launch Group	* connection		^
✓ C [™] Renesas GDB Hardware Debugo	Register initialization	No	~
C Blinky Debug_Flat	reset at the beginning of connection	Yes	~
C Renesas Simulator Debugging (Reset at the end of connection	No	~
Nenesas sinulator beougging (Neset before download	No	× .
	Reset after download	No	~
	ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
	Hold reset during connect	No	~
	Set CPSR(5bit) after download	Yes	~
	Prevent Releasing the Reset of the CM3 C	fore Yes	~
	Secure Vector Address		
	Non-secure Vector Address		
	Hot Plug	No	~
	Disconnection Mode	Stop	~
	~ SWV		
	Core clock (MHz)	0	
	✓ TrustZone		
	Set TrustZone secure/non-secure bounda	aries No	× •
< >			Revert Apply
Filter matched 9 of 11 items			

Figure 33 : e² studio Debugger Configurations Window with Blinky Project (CR52 CPU0)

4. (CA55) Check and modify the target device and GDB common settings of the CA55 core project to connect to debugging.

	Single-core processing	Multiprocessing (CR52 CPU0, CR52 CPU1)		Multiprocessing (CR52 CPU0, CA55 Core0)		
		Primary	Secondary	Primary	Secondary	
Debugger > Target Device	(default)	(default)	(default)	(default)	target device	
Debugger > GDB settings > GDB > GDB Command	(default)	(default)	(default)	(default)	aarch64-elf- gdb	

Table 9 e² studio Newly Created Project Debug Settings (2)



Debug Configurations Create, manage, and run configurations	Name Binky,cdS_secondary Debug, Flat
Type Filter tot C/C Application C/C Application C/C Particle Application EASS Strept COB Hardwere Debugging COB Hardwere Debugging	Main So Debugar (a) Surface Debug handware: Fill MARK (a) Unget Debug handware: Fill MARK (a) OBE connection Settings: ODE point number: Connection timese: ODE Connect to remote COB server: ODE Connect (b) Step Mode
	Revert App

Figure 34 : e² studio Debugger Configurations Window with Blinky Project (CA55)

5. Click **Debug** to begin debugging the application.

Preparing launch delegate		
Configuring GDB		
	Cancel Details >>	

Figure 35: Start Debugging



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 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 system_init().
- 5. Displaying the startup code where the program counter points to.

💮 [Blinky] FSP Configu	ration 🚺 startup.c 🗙	- 8
384	"RFEIA sp! \n" /* Return from system mode tack using	g RFE. */ 🔺
385	::: "memory");	
386	}	
387		
	BSP_TARGET_ARM BSP_ATTRIBUTE_STACKLESS void system_init (void)	
389	(
> 390 00010000	asm volatile (
391	"set_hactlr: \n"	
392	<pre>" MOVW r0, %[hactlr_bit_1] \n" /* Set HACTLR bits(L) */</pre>	
393	" MOVT r0, #0 \n"	
394	" MCR p15, #4, r0, c1, c0, #1 \n" /* Write r0 to HACTLR */	
395	<pre>::[hactlr_bit_l] "i" (HACTLR_BIT_L) : "memory");</pre>	
396		
397 0001000c	asm volatile (
398	"set_hcn: \n"	
399	" MRC p15, #4, r1, c1, c1, #0 \n" /* Read Hyp Configuration Register */	
400	<pre>" ORR r1, r1, %[hcr_hcd_dis] \n" /* HVC instruction disable */ " MCP p15 #4 p1 p1</pre>	
401	new pip, #4, ii, ci, ci, #0 (ii / write ()) configuration Register /	
402	<pre>::[hcr_hcd_dis] "i" (HCR_HCD_DIS) : "memory");</pre>	
403		
404 00010018 405	asm volatile (set vbar:\n"	
405		
405	" LDR r0, =vector_table \n" " MCR p15, #0, r0, c12, c0, #0 \n" /* Write r0 to VBAR */	
408	::: "memory");	

Figure 36 : e² studio Debugger Memory Window

4.6 Run the Blinky Project

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

Figure 37 : e² studio Debugger Play Icon

The following LEDs on the board should now be blinking.

- RZ/T series
 - ▶ RSK+RZ/T2M: LED0-1 (CPU0), LED2-3 (CPU1)
 - RSK+RZ/T2L: LED0-6 (including LEDx_ESC_xxx)
 - ► RSK+RZ/T2ME: LED0-1 (CPU0), LED2-3 (CPU1)
 - > RZ/T2H Evaluation Board: LED0 (CR52 CPU0), LED1 (CR52 CPU1), LED2 (CA55 Core0)
- RZ/N series
 - ➢ RSK+RZ/N2L: LED0-3



To suspend program execution, click **Run** > **Suspend** or click on the **Pause** icon.

Figure 38 : e² studio Debugger Pause Icon

To exit Debug mode and disconnect from the debugger, click **Run** > **Terminate** or click on the **Stop** icon.

-	

Figure 39 : e² studio Debugger Stop Icon

4.7 Debug and Run for Multiprocessing

To debug the Blinky application, follow these steps:

- 1. Connect the debugger with the primary project using the procedure in 4.5.2 Debug Steps .
- 2. The primary project stays connected, connect the debugger with the secondary project using the procedure in 4.5.2 Debug Steps.
- 3. When the following dialog box is shown, please click No to start debugging.

C Launcher	×
A Renesas GDB debug session is already active. Do you want to terminate all currently active debug sessions before starting session? (Note: Selecting No may result in unstable debug functionality)	the new
Remember my decision	
Yes No Ca	incel

Figure 40 Warning Window of Starting Debug Session

4. When Figure 41 is shown, please click **Yes** to proceed the launch.

Proceed with launch?	×
The device (R9A07G075M24_CR52_1) set in the launch configuration does not match the target device (R9A07G075M24GBG) set in the project. The launch may not function correctly. Do you wish to proceed with the launch?	h

Figure 41 Warning Window of Device Name



- 5. Run the primary project with procedure 4.6 Run the Blinky Project to copy the binaries of the secondary and subsequent projects to the internal RAM in the primary project. After the primary project reaches **hal_entry** in **main.c**, other cores are executed. If the LEDs are blinking, proceed to the next step.
- 6. Run the secondary project with procedure 4.6 Run the Blinky Project.
- 7. When exiting Debug mode and disconnecting from the debugger, terminate both projects, the primary and the secondary.

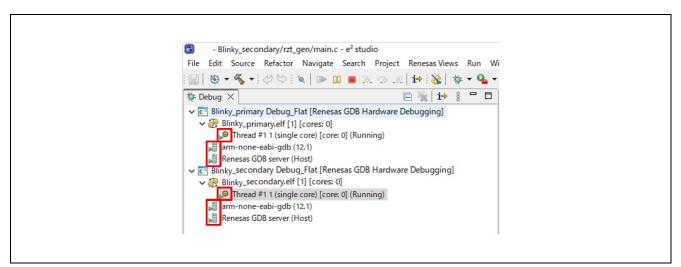


Figure 42 During Program Execution

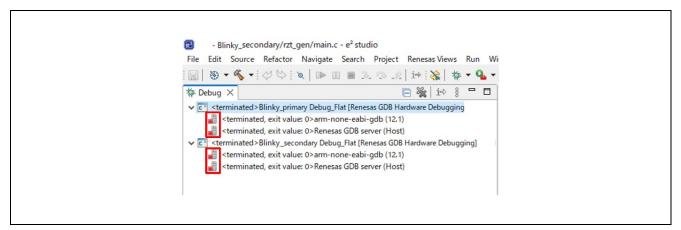


Figure 43 Terminated Program

4.8 Import the Project

The project created, built, and debugged in chapters 4.3 through 4.7 can be imported and run in other workspaces.

Note:

Apply the same version of FSP package used for the project to the other workspace.



To import the projects, follow these steps:

1. Click **File** > **Import**.

File	Edit Source Refactor Na	wigate Search Project	Renesas Views	Run	Windo	w
	New	Alt+Shift+N >	1			
	Open File		E 🕏 🍸 🕴	- [5	
۵,	Open Projects from File System	.				
	Recent Files	>				
	Close Editor	Ctrl+W	existing code			
	Close All Editors	Ctrl+Shift+W				
	Save	Ctrl+S				
	Save All	Ctrl+Shift+S				
	Revert					
	Move					
ľ	Rename	F2				
8	Refresh	F5				
	Convert Line Delimiters To	>				
۵	Print	Ctrl+P				
\geq	Import					
	Export					
	Properties	Alt+Enter				
	Switch Workspace	>				
	Restart					
	Exit					

Figure 44 e² studio Import

2. Click General > Existing Projects into Workspace.

Select Create new projects from an archive file or directory.	
Select an import wizard:	
Mext > Einish Cancel	_

Figure 45 e² studio Select Import Type



- 3. Select root directory or Select archive file where the project you would like to import into the other workspace resides.
- 4. Select projects to import in **Projects**. When using **Select root directory**, it is recommended to set **Copy projects into workspace** in **Options** to avoid updating the same project from multiple workspaces.

Figure 46 e² studio Select Root Directory to Import Project



Import	— 🗆 X
Import Projects	
Select a directory to search for existing Eclipse projects.	
○ Select roo <u>t</u> directory:	→ B <u>r</u> owse
Select archive file: D:¥ws¥ Blinky.zip	 ✓ Browse
Projects:	
Blinky_cpu0_primary (Blinky_cpu0_primary/)	Select All
 Blinky_cpu1_secondary (Blinky_cpu1_secondary/) Blinky_nor_cpu0_primary (Blinky_nor_cpu0_primary/) 	Deselect All
Blinky_nor_cpu1_secondary (Blinky_nor_cpu1_secondary/)	Refresh
Options Search for nested projects Copy projects into workspace Close newly imported projects upon completion Hide projects that already exist in the workspace	
Working sets	
Add projec <u>t</u> to working sets	Ne <u>w</u>
Working sets:	✓ S <u>e</u> lect
? < <u>B</u> ack <u>N</u> ext > <u>Finish</u>	n Cancel

Figure 47 e² studio Select Archive File to Import Project

5. The projects have been imported into the other workspace.

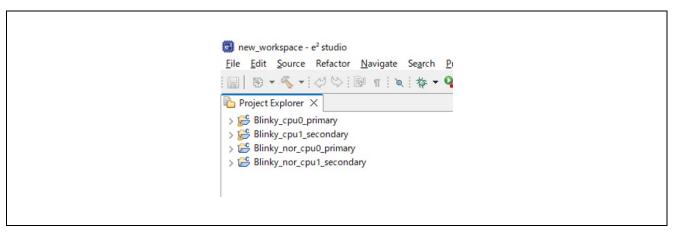


Figure 48 e² studio New Workspace

Note:

The imported project must be clicked the Generate Project Content button and built before debugging.



5. FSP Smart Configurator User Guide

5.1 What is FSP Smart Configurator?

The Renesas FSP Smart Configurator (FSP SC) is a desktop application designed to configure device hardware such as clock set up and pin assignment as well as initialization of FSP software components when using a 3rd-party IDE and toolchain.

For creating RZ/T2 and RZ/N2 project, the FSP SC can currently be used with

• IAR EWARM with IAR toolchain for Arm

Projects can be configured, and the project content generated in the same way as in e^2 studio. Please refer to 5.2 Configuring a Project section for more details.

5.2 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 FSP SC and 3rd-party IDE and running that application on an RZ/T2, RZ/N2 MPU board. This chapter guides you through creating projects for a single-core processing and a multiprocessing with RAM execution without flash memory. In this chapter, the multiprocessing refers to a process in which CR52 CPU0 core is activated first and second core(CR52 CPU1 or CA55 Core0) operates after CR52 CPU0 core sets up for second core.

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.



5.3 Using Smart Configurator with IAR EWARM

IAR EWARM includes support for Renesas RZ/T2, RZ/N2 devices. These can be set up as bare metal designs within IAR EWARM. However, most RZ/T2, RZ/N2 developers will want to integrate RZ/T2, RZ/N2 FSP drivers and middleware into their designs. SC will facilitate this.

FSP SC generates a "Project Connection" file that can be loaded directly into IAR EWARM to update project files.

5.3.1 Prerequisites

- IAR EWARM installed and licensed.
 - > Please refer to IAR systems website regarding IAR EWARM.
- FSP SC and FSP Pack installed.
 - > Please refer to Renesas website regarding to FSP SC and FSP Pack.

Note for RZ/T2ME:

If you use the IAR EWARM 9.60.1 or 9.60.2 to debug RZ/T2ME FSP project, please apply the following patch file.

- EWARM_Patch_for_RZT2ME (EWARM_Patch_for_RZT2ME_rev1.0.zip)
 - This patch file is available in <u>http://www.renesas.com/rzt2me</u>.

Regarding how to apply the patch, please read the readme file in patch file.

Note for RZ/T2H:

If you use the IAR EWARM to debug RZ/T2H FSP project, please apply the following patch file.

- EWARM_Patch_for_RZT2H_RZN2H (EWARM_Patch_for_RZT2H_RZN2H_rev1.0.zip)
 - This patch file is available in <u>http://www.renesas.com/rzt2h</u>.

Regarding how to apply the patch, please read the readme file in patch file.



5.3.2 Create a New Project

The following steps are required to create a project using IAR EWARM, FSP SC and FSP. In the case of multiprocessing, two projects with different settings must be created. A project that starts first is called the primary project and the second project that runs after releasing reset by the primary project is called the secondary project.

Note for multiprocessing projects:

The primary project and the secondary project should be created in the same workspace.

The secondary project should be created after the primary project is created in 5.3.2 section and done 1st build of the primary project in 5.3.3 section.

- 1. Start the FSP Smart Configurator.
 - FSP Smart Configurator is installed in the following path as default.
 - For RZ/T series, it is installed in C:\Renesas\rzt\sc_vYYYY-MM_fsp_vX.X.X\eclipse\rasc.exe
 - For RZ/N series, it is installed in C:\Renesas\rzn\sc_vYYYY-MM_fsp_vX.X.X\eclipse\rasc.exe
- 2. Select the File > New > FSP Project...
 - This step may be unnecessary depending on old FSP SC version.

Import <
Import Open Close Crt-W Close Crt-Strift-W Save All Crt-Strift-S Exit Exit Properties Properties Properties Properties Properties The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view The active editor element does not use this view<
Open Close Ctri-W Close All Ctri-Shift-W Save All Ctri-Shift-S Evit Evit Propertis € Problems € Propertis € Problems

Figure 49 : FSP SC New Project



3. Enter a project folder and project name. An example of naming is shown below.

	Table 10 FSF SC Rewly created Hojeet Settings (1)					
	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)	
		Primary	Secondary	Primary	Secondary	
Project name	Blinky	Blinky_primary	Blinky_secondary	Blinky_primary	Blinky_secondary	

Table 10 FSP SC Newly Created Project Settings (1)

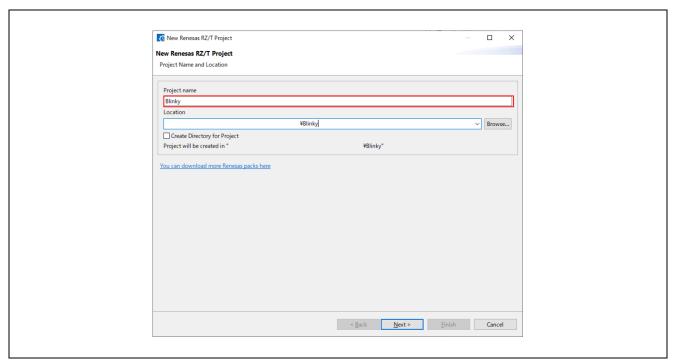


Figure 50 : FSP SC Project Settings



- Select the FSP version. 4.
- 5. Select the Board for your application.
 - You can select an existing RZ/T2, RZ/N2 MPU Evaluation Board or select Custom User Board for any of the • RZ/T2, RZ/N2 MPU devices with your own BSP definition.
 - Here, select either of following boards to create a FSP project for Evaluation board. •
 - (Multicore device ONLY) Select the Core from the drop-down list.

7. Select IDE Project Type.

- As the Toolchain, IAR Toolchain for ARM is preselected. •
- 8. Click Next.

6.

	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5)	5 Core0)		
		Primary	Secondary	Primary	Secondary		
Board	RSK+RZXXX (RAM execution without flash memory) or RZXXX Evaluation Board (RAM execution without flash memory)						
Core	CR52_0 or CR52_CPU0CR52_0 or CR52CR52_1 or CR52CR52_CPU0CA55 Core0CR52_CPU0CPU0CPU1CPU1CA55 Core0						
IDE Project Type	IAR EWARM (FSP SC is older than 2024-04 version)or IAR EWARM [v9.40+] (FSP SC is 2024-04 version)or IAR EWARM [v9.60+] (FSP SC is equal to or newer than 2024-10 version)						

Table 11	FSP SC Newly Created Project Settings (2)	

Device Select FSP Version: Board: Device:	tion		Board (RAM execution without flash memory)
Core:	CR52 CPU0	Device Details	
Language:	●C ○C++	TrustZone Pins Processor	No 729 Cortex-R52
IDE Project T		7	
Toolchains			
IAR Toolcha	ain for ARM		
	\checkmark		

Figure 51 : Target Device and IDE Selections



9. (The secondary project of multiprocessing ONLY) Select a bundle file of the primary project. Built the primary project names in the same workspace appear as an option in the drop-down list.

Note:

Warnings occur if the FSP version or Board (boot mode) used is different between the primary project and the secondary project. Use the FSP same version and Board (boot mode).

💰 New Renesas RZ/T Project			
New Renesas RZ/T Project			
Existing Smart Bundle Selection			
Smart Bundle:	¥Blinky_	primary¥Debug¥Exe¥Blinky_primary.sl	bd Browse
Select a Smart Bund	lle (*.sbd) file describing the configuration (of the preceding processor core.	
Smart Bundle Details			
FSP version Toolchain Toolchain version Board Device Core Zones	 IAR Toolchain for ARM <80ard (boot mode)> R9A09G077M44GBG CR52_0 CR52_0		
		: <u>B</u> ack <u>N</u> ext > E	nish Cancel

Figure 52 Bundle File Selection



10. Select RTOS.

- Here, select **No RTOS** for proceeding the following tutorial.
- 11. Click Next.

New Renesas RZ/T Project RTOS Selection RTOS Selection TOS Selection TOS Selection TOS Selection TOS Selection TOS Selection TOS Selectio	Kew Renesas RZ/T Project		— D	×
RTOS Selection No RTOS	New Renesas RZ/T Project			
Vo RTOS				
	RTOS Selection			
< Back Next> Excide Carrel	No RTOS ~			
< Back Next > Endeh Cancel		•		
< Back Next > Einteh Cancel				
< Back Next > Finish Carrel				
< Back Next > Enrich Concel				
< Back Next > Enrich Concel				
< Back Next > Enrich Cancel				
< Back Next > Enrich Cancel				
< Back Next > Enrich Cancel				
< Back Next > Einich Cancel				
< Back Next > Sinisk Cancel				
< Back Next > Sinisk Cancel				
< Back Next > Sinisk Cancel				
< Back Next > Sinisk Cancel				
< Back Next > Sinisk Cancel				
< Back Next > Finish Cancel				
- geven the second seco		< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel	

Figure 53 : RTOS Selection



- 12. Select a **project template** from the list of available templates.
 - By default, this screen shows the templates that are included in your current RZ/T MPU Pack.
 - Here, select Bare Metal Blinky for proceeding the following tutorial.
 - If you want to develop your own application, select the basic template for your board, Bare Metal Minimal.
- 13. Click Finish.

Kew Renesas RZ/T Project			×
New Renesas RZ/T Project			
Project Template Selection			
Project Template Selection			
 Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if avaithe C runtime environment. [Renesas.RZTpack] Bare Metal - Minimal Bare Metal FSP project that includes BSP. This project will initialize (Renesas.RZTpack) 			
Sack	<u>N</u> ext > <u>F</u> inish	Cancel	

Figure 54 : Template Selection

- 14. Configure the FSP configuration by referring to Chapter 6.3 "Configuring a Project".
 - Here, skips this configuration step for proceeding the following tutorial.



Note for the primary project using CR52 CPU0:

If the primary project selects CR52 CPU0 as **Core** and the secondary or later project uses a CA55 core, you need to set "PLL0 is released from standby state" and enable PLL0 in the Clocks tab of FSP Configuration.

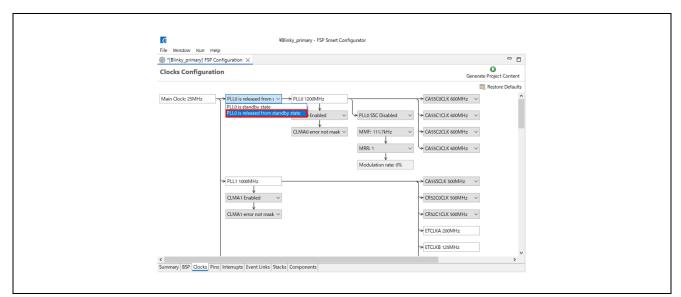


Figure 55 : Enable PLL0 in the Primary Project using CR52 CPU0

15. On completion of the FSP configuration, click Generate Project Content.

(Blinky] FSP Cor	nfiguration $ imes$				- 8
Summary					Generate Project Content
Project Sum	nmary			RE	NESAS
Board:	RSK+RZT2N	I (RAM execution wit	hout flash memory)		
Device:	R9A07G075	M24GBG			
FSP Version:	<fsp td="" version<=""><td>></td><td></td><td></td><td></td></fsp>	>			
Project Type:	Flat				
Location:	C:/Users/		'/Blinky 🐟		

Figure 56 : FSP Project Configuration and Generation

A new IAR EWARM project file will be generated in the project path.



16. Double click IAR EWARM Workspace file (.eww) to open IAR EWARM with workspace.



Figure 57 : FSP Project Workspace

5.3.2.1 NOTE: Configure IAR EWARM Project [Only RZ/T2H]

1. Change the device tag name of buildinfo.ipcf in the project and save it.

	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
Device name	R9A09G077 M44_R52_0	R9A09G077M44 _R52_0	R9A09G077M44_ R52_1	R9A09G077M44 _R52_0	R9A09G077M44 _A55

Table 12 FSP SC Newly Created Project Debug Settings



Figure 58 : IAR EWARM Project File (CR52)

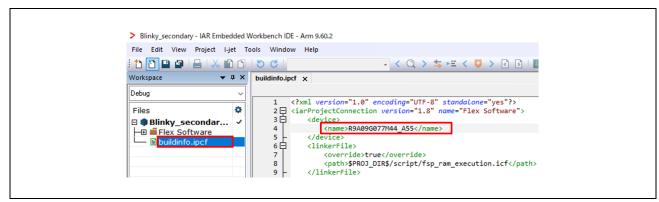


Figure 59 : IAR EWARM Project File (CA55)



- 2. Click on **Project** and then click on **Option...** to open project option window.
- 3. Select General Options category and Target tab.
- 4. Confirm that the name changed in step 1 appears in the **device** of **Processor variant**.

C/C++ Compiler Assembler Assembler Cutput Converter Cutput Con	Options for node "Blinky_prim Category: General Options Static Analysis Runtime Checking	ary" Library Configuration Library Options 1 Library Options 2	×
	Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CADI CMSIS DAP E.2/E2 Lite GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris NU-Link PE micro	Target 32-bit 64-bit Output Processor variant Cgre Cortex-R52 Image: Cortex-R52 Ima	

Figure 60 : Project Options – Device (CR52 CPU0)

Calegoy: General Options State Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Catom Build Unker Build Actions Debugger Simulator CADI CADI CMSIS DAP E2/22 Lite GDB Server G+LINK T Jstellaris Nu-Link PE micro V

Figure 61 : Project Options – Device (CA55)



Note:

After changing the FSP configuration and clicking "Generate Project Content" in FSP SC, the contents of buildinfo.ipcf will be overwritten and the device name reverts to its pre-modified name. This will result in an error message when the project is re-opened in IAR EWARM, but the setting of project options in step 4 is maintained and do not need to be modified again in buildinfo.ipcf. Please build the project as is.

Build
Messages S Unknown device name: 'R9A09G077M44_CR52_0' S Unknown device name: 'R9A09G077M44_CR52_0'
Build DebugLog

Figure 62 : Error Message after Configuration Change

5. Select Build Actions category and then Remove "Pre-compile" row

Options for node "Blinky_pri	mary" ×
Category: General Options ^ Static Analysis Runtime Checking C/C++ Compiler Assembler	Build Actions Configuration
Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/F2 Lite GDB Server G+LINK I-jet	Build actions: Command line Build order Output file(s) Cmd /c "start"Renesas" /w cmd /c ""\$P Pre-compile \$BUILD_FIL cmd /c "start "Renesas" /w cmd /c ""\$P Post-link \$BUILD_FIL
J-jek J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET V	New Edit Remove

Figure 63 : Project Options – Build Actions



5.3.3 Build the Project

When multiprocessing, please refer to Section 5.3.3.2 Build for Multiprocessing.

5.3.3.1 Build

- Single-core processing Click on **Project** -> **Make** from menu bar or **Make** button on tool bar to build.
- Multiprocessing Build both the primary and secondary projects. Click on Project -> Rebuild All from menu bar.

	-	iect I-jet Tools Window		
1 🗅 🔛 🕋 📗		Add Files		- < Q > 歩柱 < Q > 2 1 0 0 1
Workspace		Add Group		
Debug	(1	Import File List		
Files		Add Project Connection		
🗆 🌒 Blinky - Debu		Edit Configurations		
- Flex Softw	×	Remove		
buildinfo.				
L 🖬 🖬 Output	0	Create New Project		
	0	Add Existing Project		
	٥	Options	Alt+F7	
		Version Control System	,	
	0	Make	F7	1
		Compile	Ctrl+F7	
	0	Rebuild All		
	₫	Clean		
	P	Batch build	F8	
		C-STAT Static Analysis	,	
Blinky	٥	Stop Build	Ctrl+Break	

Figure 64 : Make Button

										^
Messages File									Line	
5	0% Generating	Secure Bun	dle100% Genera	ating Secure Bun	dle					
Te	tal number of	arrans: 0								
	tal number of									
	dal number of	warnings. u								
B	uild succeeded									~
1 00	ind succeeded									~
	Debug Log									

Figure 65 : Build Message Console

Once the build is completed, the build message is displayed in the Build Console window that displays compilation target files and the number of error/warnings.



5.3.3.2 Build for Multiprocessing

For multiprocessing, note the build order and build settings. If the step is preceded by (XXX), it is executed only if the condition is met.

(CR52): The core used in the project is CR52. (CA55): The core used in the project is CA55.

- 1. Create and build the primary project. (1st build of the primary project) Set the following before building:
 - i. Click **Project** > **Options...**

File Edit View	Proj	ter - IAR Embedded Workbench IDE - Arm 9.6 ject I-jet Tools Window Help Add Files	1
Image: Constraint of the second s	1 1 1 1 1 1	Add Group Import File List Add Project Connection Edit Configurations Add CMakeList.stx to Project Configure Project Force Reconfiguration	- < Q > ± -"1.0" encoding="UTF-8" nection version="1.8" / >R9A09G077M44_R52_0le> -ide>true >\$PR0J_DIR\$/script/fsp_ le>
		Remove Create New Project Add Existing Project	lons> -f "\$PROJ_DIR\$/xcl/rasc_ tions> traOptions> -f "\$PROJ_DIR\$/xcl/rasc_ traOptions>
	٥	Options Alt+F7 Version Control System	<pre>tryPoint> l>system_init EntryPoint></pre>

Figure 66 IAR EWARM Project Options

ii. Click **Debugger** > **Setup** and uncheck "Run to".

Options for node * Calegory: General Options Static Analysis Runtime Checking C(C++ Compiler Assembler Output Converter Custom Build Unker Build Actions Debuoger Simulator CAD1 OMSIS DAP E2/E2 Lite GDB Server G+L1WK I-jet J-link/J-Trace T1 Stellaris Nu-Link PEmico V		×
- (T), NV	OK Cancel	

Figure 67 IAR EWARM Project Options for the Primary Project (Run to)



iii. (CA55) Click I-jet > Interface, select From file in Probe config and select core in CPU of Probe Configuration file.

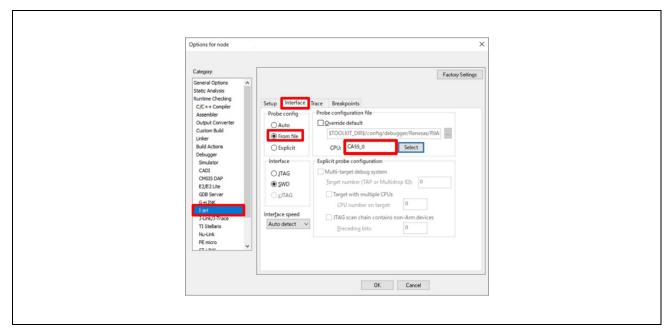


Figure 68 IAR EWARM Project Options for the Primary Project (I-jet Interface)

iv. (CA55) Select General Options > 64-bit and select LP64 of Data model.

Options for node * Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server I jet Julnk/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Verbind Rusch, Debug	Library Configuration Library Options 1 Library Options 2 Target 32-bit 64-bit Output Data model ULD32 (32-bit int, long, pointer) Image Library Configuration 1 Library Options 2 Target 32-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Target 32-bit 0-bit Output Image Library Configuration 1 Library Options 2 Image Library Configuration 2 Library Options
3-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK	

Figure 69 : Project Options – Data Model

v. Proceed to 5.3.3.1 Build.



- 2. Create the secondary project. Change the project options setting and build it.
 - i. Click **Project** > **Options...**.
 - ii. Click **Debugger** > **Setup** and uncheck "**Run to**".
 - iii. Click Debugger > Extra Options and add "--macro_param cpu1_enable=1" to Command line options: (one per line).

Options for nod Category: General Option State: Analysis Runtime Check C (C++ Compi Assembler Output Comm Custom Build Linker Build Actions Ibdiagoet Simulator C ADI OMISIS DAP E2)E2 Lite GDB Server G 4LINK Light Julink2-Tare T II Stellaran N-Li-Ink PE mico	Setup Download Images Multicore Authentication Extra Options Plugins	
(7) (No.	OK Cancel	

Figure 70 IAR EWARM Project Options for the Secondary Project (Debugger Extra Options)

iv. Click I-jet > Setup and select Software as Reset.

Options for node "Blinky_sev Calegooy: General Options A State Analysis Runtime Checking C/C++C compiler Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CADI CHSIS DAP E2/E2 Lite GDB Server G 4LINK Ident J 4/rk/J-Trace T 15 telleris Nu-Link	Setup Interface Trace Breakpoints Beset Software Override timing Quration: 300 ms Dglay after. 200 ms Target power Emulator Always prompt for probe Selection Seriet debugging Serial no: Synitch off after debugging Serial no: SPR01_DIRSVespycomm.log and	
- CT 1 112	OK Cancel	

Figure 71 IAR EWARM Project Options for the Secondary Project (Reset)



- v. (CA55) Click I-jet > Interface, select from file in Probe config and select core in CPU of Probe Configuration file.
- vi. (CA55) Select General Options > 64-bit and select LP64 of Data model.
- vii. Proceed to 5.3.3.1 Build.
- viii. Close the secondary project.
- 3. Build the primary project. (2nd build of the primary project) No setting is required, proceed to 5.3.3.1 Build.

5.3.4 Download & Debug the Project

When multiprocessing, please refer to Section.5.3.5 Debug for Multiprocessing

Note:

The main chapter of this documentation describes a RAM execution without flash memory project. When debugging a project with flash boot mode, please also refer to Appendix. How to Debug FSP Project with Flash Boot Mode.

Click on **Project** -> **Download and debug** from menu bar or **Download and Debug** button on tool bar to download and debug.

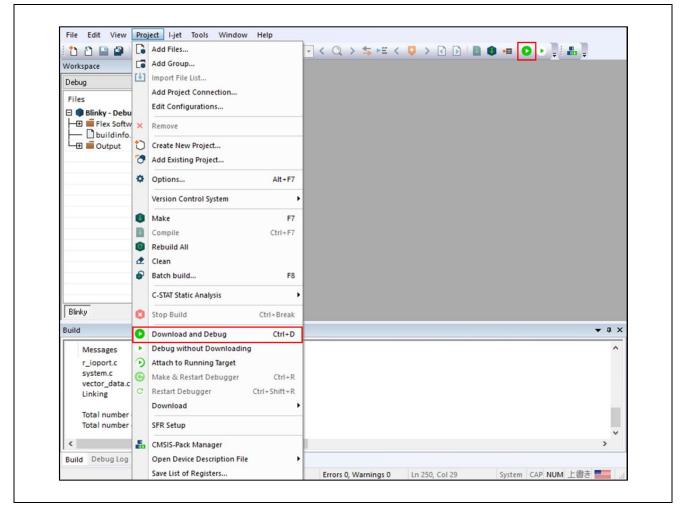


Figure 72 : Download and Debug Button



Once the download is completed and the debug is started, the program breaks at the beginning of **main** in **main.c**.

1 🗅 🖸 🖬 🖨 🛯 🗶 🛍 🖸 1 つ C	- < Q > ⇆ म < 🛛 > d 🦻 📓 🌒	📼 🕒 C 🔕 📜 📭	국 IT 위 위 🕨 🔹	ETM
Workspace 👻 🗭 🗙	main.c 🗙	*	Disassembly	▼ ₽>
Debug ~	main()	fo	Go to:	
Files	<pre>1 /* generated main source file - do not edit */ 2 #include "hal_data.h"</pre>		0x6ec: 0x0004 ?7Dat 0x6f8: 0x8228 ?7Dat 0x6f4: 0x8128 ?7Dat 0x6f4: 0x8128 ?7Dat 0x6f5: 0x0000 ?7Dat 0x706: 0x8040 0x706: 0xe810 0x706: 0xe810 0x714: 0xe87	arable4_6 la10 D arable4_7 arable4_7 la00 D arable4_8 0874 D arable4_8 0874 D arable4_9 0866 D arable4_1 0866 D arable4_1 0866 O b arable4_1 0866 O b arable4_1 0866 O b arable4_1 0866 O b arable4_1 0866 O b arable4_1 0866 O b arable4_1 0866 O b arable4_1 0 cost 0 cost cost 0 cost cost cost cost cost cost cost cost
Binku		>	hal_entr	y();

Figure 73 : Starting Debug

Click on **Debug->Go** from menu bar or **Go** button on tool bar to run this program.

File Edit View Project	4.		F5	- < Q > \$= E < Q > R 0 = 0 C Q 0		
Workspace	0	Break				- 4 ×
Debug	5	Reset		fo		
Files	0	Stop Debugging Step Over	F10	<pre>neroted main source file - do not edit */ ude "hal_data.h" int main(void) { hal_entry(); hal_entry(); </pre>	Disassembly ??DataTabl 0x6ec: 0x0004'0000	D
buildinfo.ipcf	14 14 14 00	Next Statement Run to Cursor Autostep Set Next Statement C++ Exceptions	F11 Shift+F11	return 0; }	??DataTabl 0x6f0: 0x8028'1a10 ??DataTabl 0x6f4: 0x8128'1a00 ??DataTabl 0x6f5: 0x0000'0874 ??DataTabl 0x6fc: 0x0000'0860 ??DataTabl 0x700: 0x0004'0004 init	D 24_7 D 24_8 D 24_9 D 24_9 D 24_1 D
		Memory Refresh Logging			0x704: 0xcel1 0f50 0x708: 0xc80 0f50 0x708: 0xc80 0f50 0x710: 0xc80 0f50 0x714: 0xc80 0f50 0x714: 0xc80 0f50 0x714: 0xc80 0f50 0x714: 0xc80 0f50 0x728: 0xc80 0f50 0x724: 0xc80 0f50 0x724: 0xc81 0f10 0x728: 0xc12f ff10 0x728: 0xc12f ff10 0x728: 0xc216 0f50 0x726: 0x250	M M M M M M Vfp M V M N B
Blinky					hal_entry();	

Figure 74 : Go Button



The blinky application is stored in the **hal_entry.c** file. This file is generated by FSP SC when you select the Blinky Project template and is located in the project's src/ folder. In IAR EWARM workspace view, the **hal_entry.c** is registered **Flex Software > Program Entry**.

The application performs the following steps:

- 1. Get the LED information for the selected board by **bsp_leds_t** structure.
- 2. Initialize output level for LED pin to LOW using **R_BSP_PinClear((bsp_io_region_t) leds.p_leds[i][1]**, **(bsp_io_port_pin_t) leds.p_leds[i][0]**).
- 3. Use **R_BSP_PinToggle ((bsp_io_region_t) leds.p_leds[i][1], (bsp_io_port_pin_t) leds.p_leds[i][0])** to set the output level to the LED pin.
- 4. **R_BSP_SoftwareDelay(delay, bsp_delay_units)** waits for a certain period of time. Then run #3 again.

On debugging on IAR EWARM, the break point can be set by click the left space next to line number.

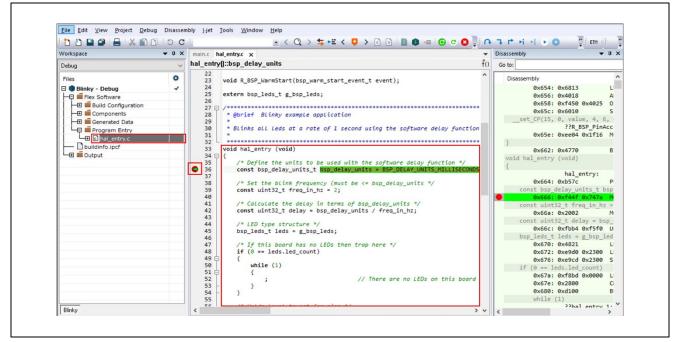


Figure 75 : hal_entry.c and Setting Breakpoint



By using the break point and the **Debug** menu or **Debug** tool bar, you can check the behavior of the Blinky application step by step.

	Go F5		: 0		ETM
	Break				▼ 4 ×
Workspace		try.c X		Disassembly	• 4 >
Debug	Reset	sp_delay_units	fo	Go to:	
Files	Stop Debugging	<pre>d R_BSP_WarmStart(bsp_warm_start_event_t event);</pre>	^	Disassembly	1
🗄 🌒 Blinky - Debug	Step Over F10			0x654: 0x6813	U
E Flex Software		ern bsp_reds_t g_bsp_reds,		0x656: 0x4018	A
- Build Configurati	Step Into F11			0x658: 0xf450	
E Components	Step Out Shift+F11	Obrief Blinky example application		0x65c: 0x6010	S
	Next Statement	en zer bernky example apperedetan		set_CP(15, 0, valu	e, 4, 6,
		Blinks all leds at a rate of 1 second using the software delay function	n		SP_PinAcc
H lal_entry.c	I Run to Cursor			0x65e: 0xee04	0x1f16 M
	Autostep	*****	*	}	
		d hal_entry (void)		0x662: 0x4770	B.
-tel Cutput	Set Next Statement	/* Define the units to be used with the software delay function */		void hal_entry (void)	
		const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS		{	
	C++ Exceptions	Const bsp_deray_dnits_c bsp_deray_dnits = bsp_deck1_onits_niterseconds		hal_e	ntry:
	11	/* Set the blink frequency (must be <= bsp_delay_units */		0x664: 0xb57c	P
	Memory	const uint32 t freq in hz = 2;		const bsp delay un	its t bsp
	Refresh			0x666: 0xf44f	0x747a M
	Logging	/* Calculate the delay in terms of bsp_delay_units */		const uint32 t fre	q in hz =
		<pre>const uint32_t delay = bsp_delay_units / freq_in_hz;</pre>		0x66a: 0x2002	M
	43			const uint32 t del	ay = bsp
	44	<pre>/* LED type structure */ bsp_leds_t leds = g_bsp_leds;</pre>		0x66c: 0xfbb4	0xf5f0 U
	45	oshTrongTr roos = RToshTrons!		bsp leds t leds =	g bsp led
	47	/* If this board has no LEDs then trap here */		0x670: 0x4821	U
	48	<pre>if (0 == leds.led_count)</pre>		0x672: 0xe9d0	0x2300 L
	49 🖨	{		0x676: 0xe9cd	
	50	while (1)		if (0 == leds.led	
	51 🛱	{		0x67a: 0xf8bd	
	52	; // There are no LEDs on this board		0x67e: 0x2800	0,0000 1
	53 -	1		0x680: 0xd100	В
	54 -	1		while (1)	
	55	18 Wolds loval to sat for pins \$/			entry 1.

Figure 76 : Debug Menu

When clinking Go button, the following LEDs on the board should now be blinking.

- RZ/T series
 - ► RSK+RZ/T2M: LED0-1 (CPU0), LED2-3 (CPU1)
 - RSK+RZ/T2L: LED0-6 (including LEDx_ESC_xxx)
 - ► RSK+RZ/T2ME: LED0-1 (CPU0), LED2-3 (CPU1)
 - ▶ RZ/T2H Evaluation Board: LED0 (CR52 CPU0), LED1 (CR52 CPU1), LED2 (CA55 Core0)
- RZ/N series
 - ► RSK+RZ/N2L: LED0-3

To suspend program execution, click **Debug** > **Break** or click on the **Pause** icon.



Figure 77 IAR EWARM Debugger Pause Icon



To exit Debug and disconnect from the debugger, click **Debug > Stop Debugging** or click on the **Stop** icon.



Figure 78 IAR EWARM Debugger Stop Icon

5.3.5 Debug for Multiprocessing

To debug the Blinky application of multiprocessing, follow these steps:

- 1. Open the primary project and close the secondary project on IAR EWARM.
- 2. Set the following in the primary project before debugging:
 - i. Click **Project** > **Options...**
 - ii. Click **Debugger** > **Multicore** and check the setting value of **Symmetric multicore** and set the following contents in **Asymmetric multicore**.
 - Symmetric multicore
 - ➢ Number of cores: 1
 - Asymmetric multicore
 - Simple
 - Partner workspace: \$PROJ_DIR\$\..\[the secondary project name]\[the secondary project name].eww
 - ♦ Partner project: [the secondary project name]
 - ♦ Partner configuration: Debug

Options for node "Blinky_prid Calegory: General Options A Static Analysis Runtme Checking C/C++ Compiler Assembler Output Converter Output Converter CADI CMSIS DAP E2/E2 Lite GOB Server G+LINK I-jet J-Link/J-Trace TI Stellaris	Factory Settings Setup Download Images Multicore Authentication Extra Options Plugins Symmetric multicore Number of cores: Opisabled Opisabled Opisabled Partner groject: Blinky_secondary Partner configuration: Debug Attach partner to punning target Partner debugger location
Nu-Link PEmicro	Partner debugger: OAdyanced Session configuration: DK

Figure 79 IAR EWARM Project Options for the Primary Project (Multicore)

iii. Click **OK** and close Options window.



- 3. Download of the primary project with procedure 5.3.4 Download & Debug the Project as shown in Figure 72.
- 4. The secondary project is automatically launched. Once the download is completed and the debug is started, the program breaks at the beginning of **system_init** in **startup_core.c**.
- 5. Run the program of primary project as shown in Figure 74 to copy the binaries of the secondary and subsequent projects to the internal RAM in the primary project. After the primary project reaches **hal_entry** in **main.c**, another core is executed. If the LEDs are blinking, proceed to the next step.
- 6. The primary project in operation, run the program of secondary project.
- 7. When exiting Debug and disconnect from the debugger, if debugging is stopped in one of the projects, either the primary or the secondary, the other will automatically stop as well.

When changing the project and debugging it again, refer No. 13 in the Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM.

5.4 Re-configuring Project with FSP SC

For proceeding the tutorial with Blinky project, the FSP configuration steps of the Blinky project was skipped in this chapter. The FSP SC can be launched from IAR EWARM or command prompt, and the FSP project configuration can be re-configured by FSP SC.

There are two ways to launch FSP Smart Configurator with an exciting project.

5.4.1 Launch FSP Smart Configurator from IAR EWARM

- 1. Select "Tools -> Configure Tools..."
- 2. Select "New" and fill in the fields as follows:
 - Menu Text FSP Smart Configurator
 - Command \$RASC_EXE_PATH\$
 - Argument --compiler IAR configuration.xml
 - Initial Directory \$PROJ_DIR\$

Configure Tools		
Menu Content:		
FSP Smart Configurator	Ok	
	Cancel	
	New	
	Delete	
Menu <u>I</u> ext:		
FSP Smart Configurator		
Command:		
\$RASC_EXE_PATH\$	Browse	
Argument:		
compiler IAR configuration.xml		
Initial Directory:		
\$PR0J_DIR\$		
Redirect to Output Window		
Prompt for Command Line		
Tool Available:		
Always 🗸 🗸		

Figure 80 : Settings to Launch FSP SC from IAR EWARM



5.4.2 Launch from the Command Prompt

- 1. Open command prompt window.
- 2. Move to the folder where the created project is located.
- 3. Execute the following command.
 - {FSP Smart Configurator installation folder} \ eclipse \ rasc.exe -compiler IAR configuration.xml

5.5 Note when debugging in different workspaces

The project created, built, and debugged in chapters 5.3.2 through 5.3.5 can be run in other workspaces. When debugging in the other workspace, please note the following two points:

- Apply the same version of FSP package used for the project to the FSP SC.
- The project must be clicked the Generate Project Content button and built before debugging.



6. FSP Configuration Users Guide

6.1 What is a Project?

In e² studio, all FSP applications are organized in RZ/T2, RZ/N2 MPU projects. Setting up an RZ/T2, RZ/N2 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 e^2 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.

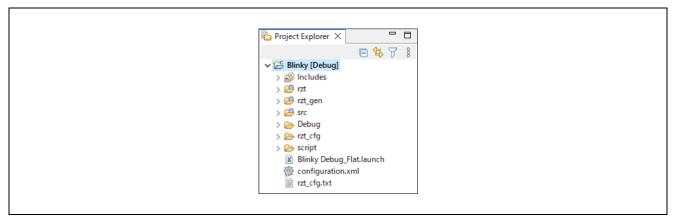


Figure 81 : e² studio Project Configuration File

Double-click on the configuration.xml file to open the RZ/T2, RZ/N2 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





Note:

Whenever the RZ/T2, RZ/N2 project configuration (that is, the configuration.xml file) is saved after configuring the project, a verbose RZ/T2, RZ/N2 Project Report file (rzt_cfg.txt, or rzn_cfg.txt) with all the project settings is generated. The format allows differences to be easily viewed using a text comparison tool. The generated file is located in the project root directory.

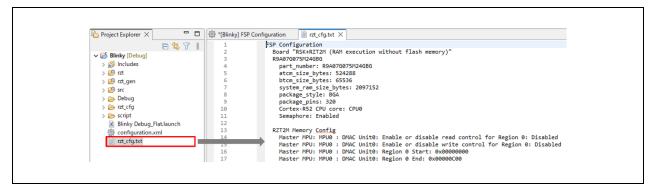


Figure 83 : RZ/T2, RZ/N2 Project Report

The RZ/T2, RZ/N2 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/T2, RZ/N2 Project Editor depend on the e² studio version and the layout may vary slightly, however the functionality should be easy to follow.

Project Summary Image: Constant State	Summary	Generate Project Content
Toolchain: <project information="" summary=""> FSP Version: Project summary information> Project Type: Location: Selected software components Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></board></project>		RENESAS
Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></board>	Toolchain: Toolchain Version: FSP Version: Project Type:	
Board Support Package Common Files Memory Config Checking I/O Port <board> Components version information> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></board>	Selected software components	
<board> <components information="" version=""> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></components></board>	Board Support Package Common Files Memory Config Checking	
	<board> Arm CMSIS Version 5 - Core Board support package for <device></device></board>	<components information="" version=""></components>

Figure 84 : RZ/T2, RZ/N2 Project Editor Tabs



6.2 Create a Project

6.2.1 Creating a New Project

For RZ/T2, RZ/N2 MPU applications, generate a new project using the following steps:

1. Click on File > New > Renesas C/C++ Project > Renesas RZ.

•	- e² studio						
File	Edit Source Refactor Navigate	Search Projec	t Re	enesas Views Run Window He	lp		
	New	Alt+Shift+N >		Renesas C/C++ Project	>	Renesas Debug	
	Open File		Ľ	Project	Т	Renesas RZ	
۵,	Open Projects from File System		-9	Example	Ī	101 BS	SP_DONT
	Recent Files	>		· · · · · · · · · · · · · · · · · · ·	- F	102 BS	SP_PLAC
	Close Editor	Ctel - M		Other Ctrl+N		103 BS	SP_DONT

Figure 85 : New RZ/T2, RZ/N2 MPU Project

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

	📴 New C/C	t —		\times
/C++ Create an executable or static library C/C++ FSP project for Renesas RZ/N. Renesas RZ/N Create an executable or static library C/C++ FSP project Create an executable or static library C/C++ FSP project for Renesas RZ/V. Renesas RZ/V C/C++ FSP Project Create an executable or static library C/C++ FSP project for Renesas RZ/V. V	Templates	sas RZ Project		
	All C/C++	Create an executable or static library C/C++ FS project for Renesas RZ/N. Renesas RZ/T C/C++ FSP Project Create an executable or static library C/C++ FS project for Renesas RZ/T. Renesas RZ/V C/C++ FSP Project Create an executable or static library C/C++ FS	2	~

Figure 86 : New Project Templates



- 3. Select a project name and location.
- 4. Click Next.

Renesas RZ/T C/C++ FSP Project – C	x c
Renesas RZ/T C/C++ FSP Project Project Name and Location	Ĵ
Project name Blinky	
Use default location	owse
You can download more Renesas packs here	
	Cancel

Figure 87 : RZ/T2, RZ/N2 MPU Project Generator (Part 1)

6.2.2 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** and **Device** for your application.

Note:

You can select an existing RZ/T2, RZ/N2 MPU Evaluation Kit (Such as RSK) or can select **Custom User Board** for any of the RZ/T2, RZ/N2 MPU devices with your own BSP definition. When you use the RZ/T2, RZ/N2 MPU Evaluation Kit,

- First, please set the **Board** to the Evaluation Kit and the boot mode which you use.
- In this case, please don't change the **Device** which is automatically set to the device which RSK board uses.

When you use Custom User Board,

- First, please set the **Device** to your device on your board.
- Second, please set the Board to Custom User Board with the boot mode which you use.



- 3. Select the **Core**. You could select if you selected multicore device for **Device**.
- 4. Select the **Toolchains**.
- 5. Select the Toolchain version.
- 6. Select the **Debugger**. The J-Link Arm Debugger is preselected.
- 7. Click Next.

Device and Tools Selection Board Description PSP Version: CSP Version: Board: E2T2H Evaluation Board (RAM execution with out flach memory) Device: RRA05077M44GBG Device: RRA05077M44GBG Core: CR52 CPU0 Language: © C C ++ Processor Cortex-R52 ToutZone No Pins 729 Processor Cortex-R52 Toolchains Debugger Codchain Version: Manage Toolchains Debugger	Renesas RZ/T C/C++ FSP Project Renesas RZ/T C/C++ FSP Project	– – ×
F5P Version: -F5P Version: Board: PZT2H Evaluation Board (RAM execution with v i i i i i i i i i i i i i i i i i i		
GNU ARM Embedded GCC ARM A-Profile (AArch64 bare-metal)	FSP Version:	RZT2H Evaluation Board (RAM execution without flash memory) Pervice Details TrustZone No Pins 729
	GNU ARM Embedded GCC ARM A-Profile (AArch64 bare-metal)	

Figure 88 : RZ/T2, RZ/N2 MPU Project Generator (Part 2)

If CR52 CPU0 is not selected for the secondary project of multiprocessing in procedure 3, you need to select the preceding project. To select the preceding project when creating the secondary project for multiprocessing, it is required to prepare CR52 CPU0 as the primary project before the secondary project creation.

Renesas RZ/T C/C+	+ FSP Project					
Renesas RZ/T C/C++	FSP Project					
Preceding Project or S	mart Bundle Selection					
Preceding Project:	Blinky_primary					
	Choose this option if y	ou have access to the proje	ct source code of	the preceding pr	ocessor core or se	curity context
O Smart Bundle:						
	Resolved location:					
					File System	
	Choose this option if y core or security contex	ou only have access to a Sn t.	art Bundle descril	bing the configu	ration of the prece	ding processo
Preceding Project/Sm	art Bundle Details					
FSP version		<fsp version=""></fsp>				
Toolchain Toolchain version		GNU ARM Embedded <toolchain version=""></toolchain>				
Board		<board (boot="" mode)=""></board>				
Device		R9A09G077M44GBG				
Core Zones		CR52_0 CR52_0				
Zones		CK32_0				
?			< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel
			- Mark	Text >	Funan	Concer

Figure 89 RZ/T2, RZ/N2 MPU Project Generator (Part 3)



6.2.3 Selecting a Project Template

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

Renesas RZ/T C/C++ FSP Project Build Artifact and RTOS Selection Build Artifact Selection Image: Selection <t< th=""><th>Build Artifact and RTOS Selection Build Artifact Selection Executable Project builds to an executable file Static Library</th><th>Build Artifact and RTOS Selection Build Artifact Selection Build Artifact Selection Project builds to an executable file Static Library</th><th>Renesas RZ/T C/C++ FSP Project</th><th></th><th>- X</th><th></th></t<>	Build Artifact and RTOS Selection Build Artifact Selection Executable Project builds to an executable file Static Library	Build Artifact and RTOS Selection Build Artifact Selection Build Artifact Selection Project builds to an executable file Static Library	Renesas RZ/T C/C++ FSP Project		- X	
Build Artifact Selection	Build Artifact Selection	Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file 	Renesas RZ/T C/C++ FSP Project			
Executable Project builds to an executable file Static Library	Executable Project builds to an executable file Static Library	 Executable Project builds to an executable file Static Library Project builds to a static library file 	Build Artifact and RTOS Selection		<u> </u>	
Project builds to an executable file Static Library	Project builds to an executable file Static Library	Project builds to an executable file Static Library Project builds to a static library file	Build Artifact Selection	RTOS Selection		
		Project builds to a static library file		No RTOS	~	

Figure 90 : RZ/T2, RZ/N2 MPU Project Generator (Part 4)

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/T2, RZ/N2 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.

Renesss RZT, C/C++ FSP Project Project Template Selection Project Template Selection Image: Constraint of the Construction of the Con	Project Template Selection Project Template Selection Image:
Project Template Selection Project Template Selection Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C nuntime environment. [Renesse.RZT	Project Template Selection Image: Selection
Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C nuntime environment. [Renesss.RZTp.ok.] Bare Metal - Minimal Bare metal FSP project that includes BSP This project will initialize clocks, pins, stacks, and the C nuntime environment.	Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment. Reness.RZT
Bare Metal - Minimal Bare Metal - Minimal Bare Metal - Sproject that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment.	Bare metal For project that includes BoF and wind blink LEDS if available. It is project will initialize clocks, pins, stacks, and the C number environment. [Reneas.RZTpack] Bare metal For project that includes BSR This project will initialize clocks, pins, stacks, and the C number environment. [Reneas.RZTpack] Code Generation Settings
Bare metal PSP project that includes BSR This project will initialize clocks, pins, stacks, and the Cruntime environment.	Image: In address and the project with initialize cloces, prog. stacks, and the C number environment. [Reness RZT

Figure 91 : RZ/T2, RZ/N2 MPU Project Generator (Part 5)



When the project is created, e² studio displays a summary of the current project configuration in the RZ/T2, RZ/N2 MPU Project Editor.

Project Summary Board:	
Baards	RENESAS
Device:	
Toolchain: Toolchain Version:	
Toolchain Version: <pre><project information="" summary=""></project></pre>	
Project Type:	
Location:	
Selected software components	
Simple application that blinks an LED. No RTOS included.	
Board Support Package Common Files	
Memory Config Checking	
I/O Port <board></board>	<components information="" version=""></components>
Arm CMSIS Version 5 - Core	<components mornation="" version=""></components>
Board support package for <device></device>	
Board support package for RZT2M	· · · · · · · · · · · · · · · · · · ·
Board support package for RZT2M - FSP Data	i! v

Figure 92 : RZ/T2, RZ/N2 MPU Project Editor and Available Editor Tabs

On the bottom of the RZ/T2, RZ/N2 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 **Pins** tab, you can configure the electrical characteristics and functions of each port pin.
- With the **Interrupts** tab, you can add new user events/interrupts.
- With the Event Links tab, you can configure events used by the Event Link Controller.
- 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.

6.2.4 Duplication of Resources

In the case of creating a project with a core other than CR52 CPU0 on a multicore device, duplicate resources will be grayed out or hidden in each tab of Configuration. For more details, see Configuration section of <u>RZT Flexible</u> <u>Software Package Documentation</u> > API Reference > BSP > MCU Board Support Package page.



6.3 Configuring a Project

Each of the configurable elements in an FSP project can be edited using the appropriate tab in the RZ/T2, RZ/N2 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, e^2 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.

6.3.1 Summary Tab

Project Summary Image: Constant of the second s	Board: Image: Component State St	Summary		Generate Projec	t Content
Device: Toolchain: Toolchain: <project information="" summary=""> FSP Version: Project Type: Location: Selected software components Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for <zt2m< td=""></zt2m<></device></board></project>	Device: Toolchain: Toolchain Version: <project information="" summary=""> FSP Version: Project Type: Location: Selected software components Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board <device="" for="" package="" support=""> Board support package for <device> Board support package for RZT2M Board support package for RZT2M - FSP Data</device></board></project>	Project Summary		RENESAS	^
Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board> VO Port Sboard> Arm CMSIS Version 5 - Core Board support package for Board support package for RZT2M</board>	Simple application that blinks an LED. No RTOS included. Board Support Package Common Files Memory Config Checking I/O Port <board> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M Board support package for RZT2M - FSP Data</device></board>	Device: Toolchain: Toolchain Version: FSP Version: Project Type:	<project information="" summary=""></project>		
Board Support Package Common Files Memory Config Checking I/O Port <board> <components information="" version=""> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></components></board>	Board Support Package Common Files Memory Config Checking I/O Port <board> Arm CMSIS Version 5 - Core Board support package for RZT2M Board support package for RZT2M - FSP Data Point Content of the content of</board>	Selected software comp	oonents		
		Board Support Packag Memory Config Check I/O Port <board> Arm CMSIS Version 5 Board support packag Board support packag</board>	je Common Files ing - Core e for <device> e for RZT2M</device>	<components information="" version=""></components>	· · · · · · · · · · · · · · · · · · ·

Figure 93 : 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.



6.3.2 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.

when formed	FSP Configuration ×		
Board	Support Package Configuratio	n	Generate Project Content
			Restore Defaults
Device	Selection		
FSP ve	rsion: <fsp version=""></fsp>	Board Details	for RZ/T2M CPU Board (RAM execution
Board	RSK+RZT2M (RAM execution witho		
Device	e: R9A07G075M24GBG		
Core:	CR52_0	~	
RTOS:	No RTOS	\sim	
			~
Summary	BSP Clocks Pins Interrupts Event Links	Stacks Components	
	bor clocks rins interrupts event clinks		
	ties × 🔝 Problems 🏟 Smart Browser		📑 🖬 🏹 🖾 🛷 🕴 🗖 🗖
Proper			📑 🖬 🏹 🗔 🛷 🕴 🧮 🗖
Proper	ties 🗙 🔝 Problems 🏟 Smart Browser		C 🔚 7 🗔 🖋 🕴 🗖
Proper RSK+RZ	ies × R Problems Smart Browser	sh memory)	
Proper RSK+RZ	Problems Smart Browser T2M (RAM execution without flat Property	sh memory)	
Proper RSK+RZ	Image: Second state Image: Second state	sh memory) Value	
Proper RSK+RZ	Problems Smart Browser T2M (RAM execution without flas Property R9A07G075M24GBG part_number	sh memory) Value R9A07G075M24GBG	
Proper RSK+RZ	 Problems Smart Browser Property RAA07G075M24GBG part_number atcm_size_bytes 	Sh memory) Value R9A07G075M24GBG 524288	
Proper RSK+RZ	 K Problems Smart Browser Property R9A07G075M24GBG part_number atcm_size_bytes btcm_size_bytes 	sh memory) Value R9A07G075M24GBG 524288 65536	
Proper RSK+RZ	 Problems Smart Browser Property R9A07G075M24GBG part_number atcm_size_bytes bytes bytes_bytes 	sh memory) Value R9A07G075M24GBG 524288 65536 2097152	
Proper RSK+RZ	 K Problems Smart Browser T2M (RAM execution without flas Property R9A076075M24GBG part_number atcm_size_bytes btcm_size_bytes system_ram_size_bytes package_style 	Sh memory) Value R9A07G075M24GBG 524288 65536 2097152 BGA	

Figure 94 : 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:

- rzt_cfg/fsp_cfg/bsp/bsp_cfg.h, or
- rzn_cfg/fsp_cfg/bsp/bsp_cfg.h

This file is created if it does not already exist.

Warning:

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



6.3.3 Configuring Clocks

The Clocks tab presents a graphical view of the MPU's clock tree, allowing the various clock dividers and sources to be modified.

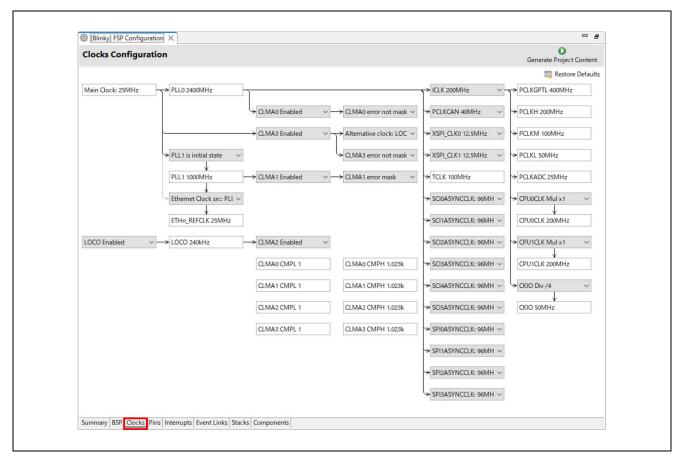


Figure 95 : Configuration Clocks Tab

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

- rzt_gen/bsp_clock_cfg.h, or
- rzn_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.



6.3.4 Configuring Pins

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

Note:

If the **Package** view window is not open in e^2 studio, select **Window > Show View > Pin Configurator > Package** from the top menu bar to open it.

The **Pins** tab simplifies the configuration of large packages with highly multiplexed pins by highlighting errors and presenting the options for each pin or for each peripheral. If you selected a project template for a specific board such as RSK+RZT2M, some peripherals connected on the board are preselected.

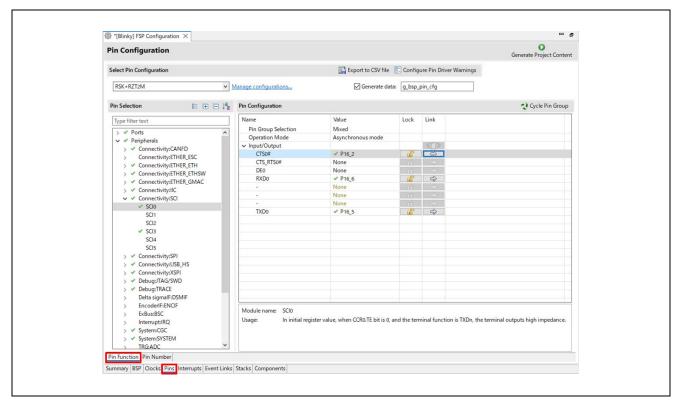


Figure 96: Pin Configuration



The pin configurator includes a built-in conflict checker, so if the same pin is allocated to another peripheral or I/O function the pin will be shown as red in the package view and also with white cross in a red square in the **Pin** Selection pane and **Pin Configuration** pane in the main **Pins** tab. The **Pin Conflicts** view provides a list of conflicts, so conflicts can be quickly identified and fixed.

In the example shown below, port P162 is already used by the GPIO, and the attempt to connect this port to the Serial Communications Interface (SCI) results in a dangling connection error. To fix this error, select another port from the pin drop-down list or disable the GPIO in the **Pin Selection** pane on the left side of the tab.

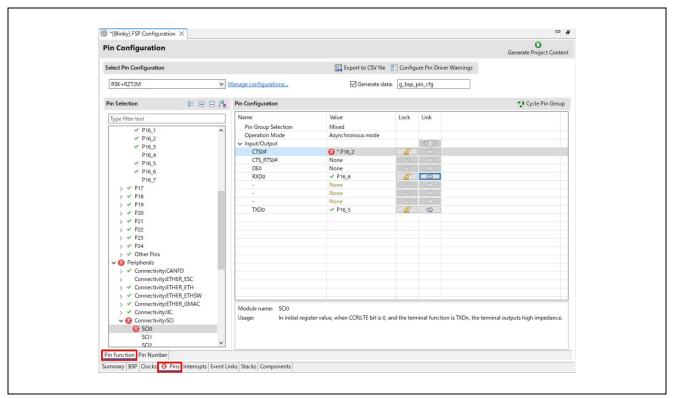


Figure 97: Conflict Checker in Pin Configuration



The pin configurator also shows a package view and the selected electrical or functional characteristics of each pin.

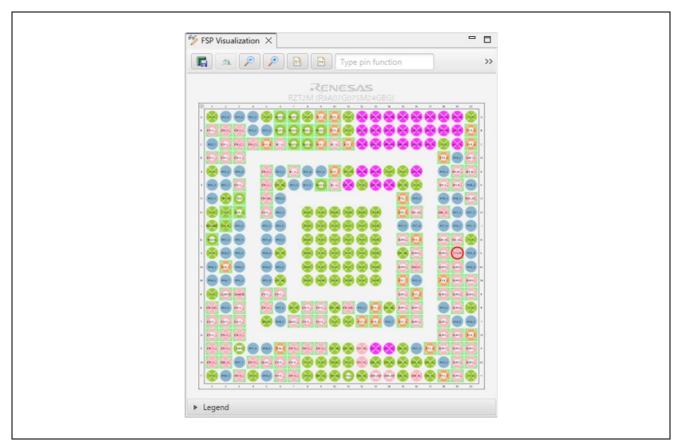


Figure 98: Pin Configurator Package View

When you click the Generate Project Content button, the pin configuration contents are written to:

- rzt_gen\bsp_pin_cfg.h, or
- rzn_gen\bsp_pin_cfg.h

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

Warning:

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



6.4 Configuring Interrupts from the Stacks Tab

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

☆ *[Blinky] FSP Configuration ×	rzt_cfg.txt			- 8
Stacks Configuration				Generate Project Content
Threads 🕢 New Threa	d 🔊 Remove 📄	HAL/Common Stack	🕢 New Stack > 🔮	Extend Stack > 🙀 Remove
 ✓ Mat/Common <i>G</i>_ioport I/O Port (r_ioport) <i>G</i> Memory config check <i>G</i>_timer0 Timer, Compare M 		g_ioport I/O P (r_ioport)	ort	imer, Alatch W
Objects 🐑 New Ob	ject > 🔟 Remove			
Summary BSP Clocks Pins Interrupts	Event Links Stacks	omponents		
			📑 🖬 🎖 🖾 🔗 🖩 🎖 🖾 /	
🔲 Properties 🗙 🔝 Problems 🏶 Si	1		🔽 🔤 J BG 🔗 🔤 J BG /	🔗 💷 JT EG 🔗 8 🗖 🗖
g_timer0 Timer, Compare Mate	ch W (r. cmtw)			
2				
			Value	^
			Value	^
Settings Property			Value Default (BSP)	^
Settings Property V Common				^
Settings Property Common Parameter Checking		mtw)	Default (BSP)	Î
Settings Property Common Parameter Checking Multiplex Interrupt		mtw)	Default (BSP)	Î
Settings Y Common Parameter Checking Multiplex Interrupt V Module g_timer0 Timer,		mtw)	Default (BSP)	
Settings Property Common Parameter Checking Multiplex Interrupt V Module g Linero Timer S General D Output Input		mtw)	Default (BSP)	
Settings Property		mtw)	Default (BSP) Disabled	
Settings Property Common Parameter Checking Multiplex Interrupt V Module g_timer0 Timer, S General Output Interrupts Callback	Compare Match W (r_c	mtw)	Default (8SP) Disabled NULL	
Settings Property < Common Parameter Checking Multiplex Interrupt < Module g_timer0 Timer > General > Output > Input < Interrupts Callback Compare Match I	Compare Match W (r_c nterrupt Priority	mtw)	Default (BSP) Disabled	
Settings Property	Compare Match W (r_c nterrupt Priority terrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled	
Settings Property	Compare Match W (r_c nterrupt Priority nterrupt Priority terrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled Disabled	
Settings Property < Common Parameter Checking Multiplex Interrupt < Module g_timer0 Timer, > General > Output > Input 	Compare Match W (r_c nterrupt Priority terrupt Priority nterrupt Priority 0 Interrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled Disabled Disabled	
Settings Property < Common Parameter Checking Multiplex Interrupt < Module g_timero Timer > General > Output > Input < Interrupts Callback Compare Match Input Capture 10 Input Capture 10 Input Capture 20 Output Compare Output Compare	Compare Match W (r_c nterrupt Priority nterrupt Priority terrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled Disabled	
Settings Property	Compare Match W (r_c nterrupt Priority terrupt Priority nterrupt Priority 0 Interrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled Disabled Disabled Disabled	
Settings Property < Common Parameter Checking Multiplex Interrupt < Module g_timero Timer > General > Output > Input < Interrupts Callback Compare Match Input Capture 10 Input Capture 10 Input Capture 20 Output Compare Output Compare	Compare Match W (r_c nterrupt Priority terrupt Priority nterrupt Priority 0 Interrupt Priority	mtw)	Default (BSP) Disabled NULL Priority 11 Disabled Disabled Disabled	

Figure 99 : Configuring Interrupts in the Stacks Tab

6.4.1 Creating Interrupts from the Interrupts Tab

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

Interrupts (Configuration	Generate Project	Content
User Events		🐑 New User Event > 🎪 R	emove
Event		ISR	
Allocations			
Interrupt	Event	ISR	
59	CMTW0_CMWI (CMTW0 Compare match) Clocks Pins Interrupts Event Links Stacks Components	cmtw_cm_int_isr	

Figure 100 : Configuring Interrupt in Interrupt Tab

And the user can add a peripheral interrupt created by the user's own. This can be done by adding a new event via the **New User Event** button.



6.4.2 Viewing Event Links

The Event Links tab can be used to view the Event Link Controller events. The events are sorted by peripheral to make it easy to find and verify them.

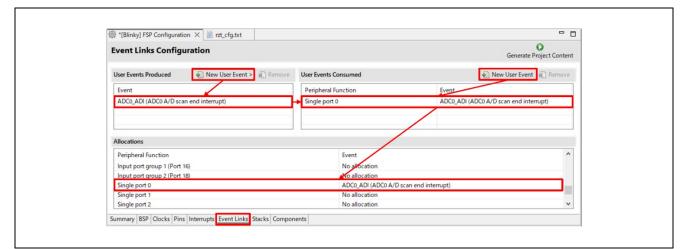


Figure 101 : Viewing Event Links

Like the Interrupts tab, user-defined event sources and destinations (producers and consumers) can be defined by clicking the relevant **New User Event** button. Once a consumer is linked to a producer the link will appear in the **Allocations** section at the bottom.

Note:

When selecting an ELC event to receive for a module (or when manually defining an event link), only the events that are made available by the modules configured in the project will be shown.



RZ/T2, RZ/N2

6.5 Adding and Configuring HAL Drivers

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

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

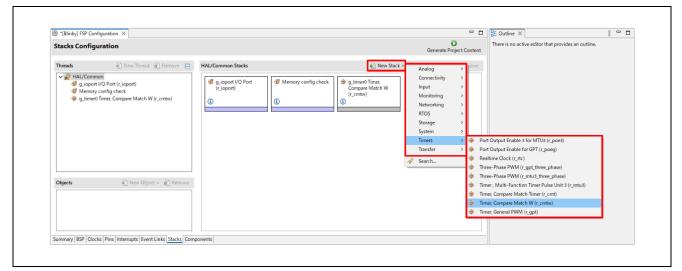


Figure 102 : e² studio Project Configurator – Adding Drivers

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 rzt/fsp or rzn/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?
rzt_gen/main.c or rzn_gen/main.c	Contains main() calling generated and user code. When called, the BSP already has Initialized the MPU.	Yes
rzt_gen/hal_data.c or rzn_gen/had_data.c	Configuration structures for HAL Driver only modules.	Yes
rzt_gen/hal_data.h or rzn_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

Table 13 Generate Contents on Smart Configurator

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

- rzt_cfg/fsp_cfg or
- rzn_cfg/fsp_cfg



6.6 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 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.

Image: Section and Section	t_cfg.txt		- 6	
Components Configuration			Generate Project Content	
		Group by: Vendor V Filter All	V Search	
Component	Version	Description	Variant	
v ⊕ Arm				
V 🐼 CMSIS				
V V CMSIS5				
Core	5.7.0+renesas.2.fsp.2	Arm CMSIS Version 5 - Core		
> 🖶 AWS				
✓ ♣ Renesas				
🗸 💸 BSP				
🗸 🥥 all				
Memory	2.0.0	Memory Config Checking		
🗸 🤗 Board				
rzt2l_rsk	2.0.0	RSK+RZT2L Board Support Files (xSPI0 x1 boot mode)	xspi0_x1_boot	
rzt2l_rsk	2.0.0	RSK+RZT2L Board Support Files (xSPI1 x1 boot mode)	xspi1_x1_boot	
rzt2l_rsk	2.0.0	RSK+RZT2L Board Support Files (RAM execution without flash memory)	ram_execution	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (xSPI0 x1 boot mode)	xspi0_x1_boot	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (xSPI0 x8 boot mode)	xspi0_x8_boot	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (xSPI1 x1 boot mode)	xspi1_x1_boot	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (16-bit bus NOR flash boot mode)	16bit_nor_boot	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (32-bit bus NOR flash boot mode)	32bit_nor_boot	
rzt2m_custom	2.0.0	RZT Custom Board Support Files (RAM execution without flash memory)	ram_execution	
v rzt2m_rsk	2.0.0	RSK+RZT2M Board Support Files (RAM execution without flash memory)	ram_execution	
rzt2m_rsk	2.0.0	RSK+RZT2M Board Support Files (xSPI0 x1 boot mode)	xspi0_x1_boot	
rzt2m_rsk	2.0.0	RSK+RZT2M Board Support Files (16-bit bus NOR flash boot mode)	16bit_nor_boot	
> 🔗 rzt2l				
✓ ♀ rzt2m				
V device	2.0.0	Board support package for R9A07G075M24GBG	R9A07G075M24GBG	
V device	2.0.0	Board support package for RZT2M		
device	2.0.0	Board support package for R9A07G075M28GBG	R9A07G075M28GBG	
Summary BSP Clocks Pins Interrupts	200	Poard support package for POA07G07EM2EGPG	PoAo7Go7EM3eGPG	

Figure 103 : Components Tab

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

- rzt/fsp/inc/api
- rzt/fsp/inc/instances
- rzt/fsp/src/bsp
- rzt/fsp/src/<Driver_Name>
- or
- rzn/fsp/inc/api
- rzn/fsp/inc/instances
- rzn/fsp/src/bsp
- rzn/fsp/src/<Driver_Name>

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

- rzt_cfg/fsp_cfg
- rzn_cfg/fsp_cfg



Appendix. Known Issues

This chapter describes the known issues regarding the current version of FSP and related platform software.

Most of the issues may require users to follow some manual operations to resolve the issues or to avoid the problems caused by the issues. Please follow the operations in the description of the issues if you use the features related to the issues. The grayed-out items have been resolved.

The known issues are categorized into two main groups, FSP Configuration and FSP Modules.

• FSP Configuration

e² studio and FSP SC have various configuration features worked on GUI with FSP. Regarding the overview of each configuration feature (GUI tab) provided as a part of FSP configuration in e² studio and FSP SC, please see the chapter 6. "FSP Configuration Users Guide".

• FSP Modules

The FSP provides HAL drivers and BSP configured by FSP Configuration on e² studio and FSP SC. Regarding their features, usage notes and API references, please see the related file "FSP Documentation".

No.	o. Title		t Devic	e			Category
		T2M	T2L	T2ME	T2H	N2L	
1	"r_gmac" may be showed as "r_ether" incorrectly.	1				1	FSP Configuration, Stacks
2	"Edge" can be selected as Transfer End Interrupt Detect Type in "r_dmac", but it cannot be used.	1	1			1	FSP Configuration, Stacks
3	When the "Device" or "Board" selection in BSP tab is changed, the BSP properties are sometimes configured to incorrect configuration.	1	1	1	~	1	FSP Configuration, BSP
4	(FSP SC ONLY) Device name is not output correctly depending on the selected device.	1		1			FSP Configuration, BSP
5	Errors occur when changing board settings.		1			1	FSP Configuration, BSP
6	Pin configuration error occurs in MPX-IO 16bit operating mode of "r_bsc".	1	1			1	FSP Configuration, Pins
7	Build error when using definition name of input/output external pins for module.	1	1			1	FSP Configuration, Pins
8	"R_SCI_UART_BaudCalculate()" of "r_sci_uart" module properly works ONLY when its clock source is SCInASYNCCLK and its frequency is 96MHz.	1	1			1	FSP Modules, SCI UART
9	"R_SPI_CalculateBitrate()" of "r_spi" module properly works ONLY when its clock source is SPInASYNCCLK and its frequency is 96MHz.	1	1			1	FSP Modules, SPI
10	A warning occurs when building "r_gmac" module with the gcc compiler.	1	1				FSP Modules, Ethernet

Table 14 List of Known Issues



No.	Title	Targe	t Devic	e			Category
		T2M	T2L	T2ME	T2H	N2L	
11	In FSP Documentation, there is incorrect description in. "API Reference > Modules > Ethernet PHY" page.	1				1	FSP Modules, Ethernet PHY
12	The interrupt number cannot be successfully acquired by the R_FSP_CurrentIrqGet() when multiple interrupt occurs.					1	FSP Modules, FreeRTOS
13	Block Media Custom Implementation can be selected as Memory Implementation for "rm_freertos_plus_fat" module, but it cannot be used.	1	1	1		1	FSP Configuration, Stacks
14	The second argument of "r_mtu3" APIs do not match with common API.	1	1	1	1	1	FSP Modules, MTU3
15	In multiprocessing, a configuration error occurs when "r_gpt" module is used for both projects for CPU0 and CPU1.	1		1			FSP Configuration, Stacks
16	Project build error occur when 32-bit bus NOR flash and xSPI0 x8 boot modes are selected on RZT Custom User Board.	1					FSP Configuration, BSP
17	The secondary project for multiprocessing cannot be created when xSPI1 x1 boot modes are selected on RZT Custom User Board.	1					FSP Configuration, BSP
18	An incorrect value is set to a pin select value for MTU0-B/MTU6/MTU7 as MTU3 output pin.		1				FSP Modules, POE3
19	Build error when using DSMIFn_ERR as an additional trigger for "r_poe3" module.		1				FSP Modules, POE3
20	Control setting values for MTU3 output pins in Stacks tab of FSP Configuration are set to the incorrect pin.	1	1	1			FSP Configuration, Stacks
21	A bug that prevented the setup of PLL1.					1	FSP Configuration, Clocks
22	A section cannot be copied successfully when its size is not a multiple of the alignment size.					1	FSP Modules, BSP
23	Initial values of data placed in some sections were overwritten with 0.					1	FSP Modules, BSP
24	Some sections were not initialized in the flash boot project.					1	FSP Modules, BSP
25	DSMIF 0/1 error 1 trigger macros are not defined.		1				FSP Modules, POEG
26	DSMIF 0/1 error 1 status macros are not defined.		1				FSP Modules, POEG
27	Missing constraint for DSMIF error trigger in channel 1 and channel 2.	1	1	1	1		FSP Modules, POEG
28	FreeRTOS+FAT format process is not executed correctly.	1	1	1	1		FSP Modules, FreeRTOS+FAT
29	Caution when specifying program placement in linker scripts.				1		Others, Linker script



No.	Title	Target Device				Category	
		T2M	T2L	T2ME	T2H	N2L	
30	In the secondary project for multiprocessing, no error occurs when there is a conflict in a resource used with the preceding project.				1		FSP Configuration, Stacks
31	Errors occur when setting ELC in r_gpt module.				1		FSP Configuration, Stacks
32	CR52 CPU1 of RZ/T2H is implemented to start programs from System SRAM instead of CPU1 ATCM.				1		Others, Linker script
33	No Error Occurs when entering out-of-range values for window parameters in r_pcie_ep and r_pcie_rc module configurations.				1		FSP Configuration, Stacks
34	Address space of DDR and PCIE cannot be used in the secondary (or later) projects with flash boot mode.				1		Others, Address space
35	r_gmac_b module cannot use zero-copy mode.				1		FSP Modules, GMAC
36	r_ade module does not support the calibration function.				1		FSP Modules, ADC
37	The USB driver for CA55 project does not work.				1		FSP Modules, USB
38	No error returns when entering the virtual addresses that cannot be translated to physical addresses as arguments.				1		FSP Modules, xSPI_OSPI, xSPI_QSPI, DMAC

No. 1 Resolved

Title	"r_gmac" may be showed as "r_ether" incorrectly.
Target	RZ/T2M, RZ/N2L
Category	FSP Configuration, Stacks
Description	In Stacks tab, "r_gmac" may be showed as "r_ether" incorrectly.
Workaround	Please read the "r_ether" as "r_gmac".

No. 2 Resolved

Title	"Edge" can be selected as Transfer End Interrupt Detect Type in "r_dmac", but it cannot be used.
Target	RZ/T2M, RZ/T2L, RZ/N2L
Category	FSP Configuration, Stacks
Description	"Edge" of interrupt detect type is not available due to a change in hardware specifications.
Workaround	Please don't set Edge to Transfer End Interrupt Detect Type



Title	When the "Device" or "Board" selection in BSP tab is changed, the BSP properties are sometimes configured to incorrect configuration.
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L
Category	FSP Configuration, BSP
Description	When the "Device" or "Board" selection in BSP tab is changed, the BSP properties are sometimes configured for incorrect configuration. Once this issue occurs, the project cannot be fixed to correct configuration.
Workaround	If changing the "Device" or "Board", please reselect "FSP Version" from the drop-down list. If you want to change only the boot mode on the same board, please refer to Appendix. How to Change Boot Mode of FSP Project

No. 4 Resolved

Title	(FSP SC ONLY) Device name is not output correctly depending on the selected device.							
Target	RZ/T2M, RZ/T2ME							
Category	FSP Configuration, BSP							
Description	If you create a project by selecting a single core device (R9A07G075M01xxx, R9A07G075M05xxx), the device setting will be "None" when you open the project in IAR EWARM.							
Workaround	Please reselect device name from the device list in IAR EWARM project options. Options > General Options > Target > Processor variant > Device Options for node "Blinky" Calegopy: Cale							
	OK Cancel							



<pre>memory) to RZN2L Custom User Board (RAM execution without flash memory) and from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (xSP10 x) boot mode).</pre> 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom U Board (RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. Preprint Product real to get the real to get to get the real to get the real to get the real to get the real to get to get the real to get to get the real to get the real to get the real to get to get to get to get the real to get to	Category FSP Configuration, BSP Description Errors occur when changing board settings from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (SSPI0 x1 boot mode). 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. Understand User State (RAM execution RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. Understand Control (RAM execution RAM	Title	Errors occur when changing board	settings.				
Description Errors occur when changing board settings from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (xSPI0 xi boot mode). 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom U Board (RAM execution without flash memory) to RZN2L Custom U Board (RAM execution without flash memory) to RZN2L Custom U Board (RAM execution without flash memory) In these case, the build is successful. But the following sereen is displayed after the build. Image: State of the state of t	Description Errors occur when changing board settings from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) In Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) In this case, the build is successful. But the following sereen is displayed after the build. Image: Image from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (RAM execution without flash memory) In this case, the build is successful. But the following sereen is displayed after the build. Image from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Image flash is interventional set in the set intervention of the s	Target	RZ/T2L, RZ/N2L					
<pre>memory) to RZN2L Custom User Board (RAM execution without flash memory) and from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (xSPI0 x boot mode).</pre> 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom U Board (RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. The first screece for the following screen is displayed after the build. <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is successful. But the following screen is displayed after the build. </pre> <pre> int this case, the build is the following screen is displayed after the build. </pre> <pre> int this case, the build is the following screen is displayed after the build. </pre> <pre> int this case, the following screen is displayed after the build. </pre> <pre> int this case, the following screen is displayed after the build. </pre> <pre> int this case, the following screen is displayed after the build. </pre> <pre> int this case, the following screen is displayed after the build. </pre> <pre> int this case, the following screen is displayed after the build. </pre> <pr< th=""><th><pre>memory) to RZN2L Custom User Board (RAM execution without flash memory) and from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (xSP10 x1 boot mode).</pre> 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom Use Board (RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. This case, the build is successful. But the following screen is displayed after the build. </th><th>Category</th><th>FSP Configuration, BSP</th><th></th><th></th><th></th><th></th><th></th></pr<>	<pre>memory) to RZN2L Custom User Board (RAM execution without flash memory) and from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom User Board (xSP10 x1 boot mode).</pre> 1. Changed from RSK+RZN2L (RAM execution without flash memory) to RZN2L Custom Use Board (RAM execution without flash memory) In this case, the build is successful. But the following screen is displayed after the build. This case, the build is successful. But the following screen is displayed after the build.	Category	FSP Configuration, BSP					
Board (xSPI0 x1 boot mode) bsp_mcu_device_pn_cfg.h is not generated and builds error occurs. Properties Problems × Smart Browser Console 6 errors, 18 warnings, 227 others (Filter matched 124 of 251 items) Description Path Location Type • Errors (6 items) • Errors (6 items) • Beg.mcu_danice_pr.dg.h: No such file or directory bsp.mcu_danily_cfg.h /rmal_minimal C/C++ Probl • Make **** [ran_gen/ubdir.mk42: ran_gen/main.ol Error 1 rmal_minimal C/C++ Probl C/C++ Probl • make **** [ran_gen/ubdir.mk42: ran_gen/main.ol Error 1 rmal_minimal C/C++ Probl C/C++ Probl • make **** [ran_gen/ubdir.mk42: ran_gen/main.ol Error 1 rmal_minimal C/C++ Probl	Board (xSPI0 x1 boot mode) bsp_mcu_device_pn_cfg.h is not generated and builds error occurs. Properties Problems X Smart Browser Console 6 errors, 18 warnings, 227 others (Filter matched 124 of 251 tems) Description Petrors (6 items) 6 fatal error bsp.mcu_device_pn_cfg.h No such file or directory bsp.mcu_family_cfg.h 7 fatal error bsp.mcu_device_nn_cfg.h No such file or directory bsp.mcu_family_cfg.h 7 make: *** [rn.gen/subdir.mk42: rn.gen/subdir.mk42: rn.gen/subdir.mk42	Description	<pre>memory) to RZN2L Custom User Boz RSK+RZN2L (RAM execution withor boot mode).</pre> 1. Changed from RSK+RZN2L (RA Board (RAM execution without find In this case, the build is successful. Bu Propertie Problems & Smart Browser Console X Memory COT build Console [rm2]_minmail Building file:/ran/fsp/src/hsp/mcu/arn2l/bsp_cache.c Building file:/ran/fsp/src/hsp/mcu/arn2l/bsp_cache.c Building file:/ran/fsp/src/hsp/mcu/arn2l/bsp_cache.c Building file:/ran/fsp/src/hsp/mcu/arn2l/bsp_cache.c Building file:/ran/fsp/src/hsp/mcu/arn2l/bsp_cache.c /ran/fsp/src/bsp/mcu/arn2l/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cachell/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_cache.c /ran/fsp/src/bsp/mcu/all/bsp_cachell/bsp_ster_ Building file:/ran/fsp/src/bsp/mcu/all/bsp_ster_cachell/bsp_	tion.c (startup.c (system.c (system.c (system.c) (startup.c) (system.c)	cution with ory) to RZN without fla: og screen is og screen is og screen is over the second during the build. I during the bui	iout fla: 12L Cu: sh menr display red a problem. uild. er on project ug/rzn/board/rzn/ (rzn/board/rzn/	sh memory) an stom User Boa nory) to RZN21 ved after the bu	d from rd (xSPI0 x1 L Custom Us 1ild.
□ Properties Image: Problems X image: Smart Browser □ Console 6 6 errors, 18 warnings, 222 others (Filter matched 124 of 251 items) Description Resource Path Location Type ✓ ④ Errors (6 items) Path Location Type ④ fatal error: bsp.mcu.device.pn_cfg.h: No such file or directory bsp.mcu.family_cfg.h /rzn2Lminimal/rzn line 4 C/C++ Probl ④ make: **** [rzn_gen/subdir.mb42: rzn_gen/cal.data.o] Error 1 rzn2Lminimal C/C++ Probl ④ make: *** [rzn_gen/subdir.mb42: rzn_gen/main.o] Error 1 rzn2Lminimal C/C++ Probl ④ make: *** [rzn_gen/subdir.mb42: rzn_gen/main.o] Error 1 rzn2Lminimal C/C++ Probl ④ make: *** [rzn_gen/subdir.mb42: rzn_gen/rzn	Image: Second		Board (xSPI0 x1 boot mode)				nory) to RZN2	L Custom Us
6 errors, 18 warnings, 227 others (Filter matched 124 of 251 items) Description Resource Path Location Type ♥ Errors (6 items) (7n22_minimal/rzn line 4 C/C++ Probl ● make: *** [rzn.gen/subdir.mk42: rzn.gen/common_data.o] Error 1 rzn2_minimal C/C++ Probl ● make: *** [rzn.gen/subdir.mk42: rzn.gen/nah_d data.o] Error 1 rzn2_minimal C/C++ Probl ● make: *** [rzn.gen/subdir.mk42: rzn.gen/nah_ol Error 1 rzn2_minimal C/C++ Probl ● make: *** [rzn.gen/subdir.mk42: rzn.gen/nah_ol Error 1 rzn2_minimal C/C++ Probl ● make: *** [rzn.gen/subdir.mk26: rzn.gen/nah.ol Error 1 rzn2_minimal C/C++ Probl	6 errors, 18 warnings, 227 others (Filter matched 124 of 251 items) Description Resource Path Location Type • © Errors (6 items) • • • • • • • • • • • • • • • • • • •			erated and buil	ds error oc	curs.		
V © Errors (6 items) Sp_mcu_device_pn_cfg.h: No such file or directory More that error bsp_mcu_device_pn_cfg.h: No such file or directory More that error bsp_mcu_device_pn_cfg.h: No such file or directory make ***: [rar, gen/subdir.msk-2: rar, gen/chal_data.o] Error 1 ran2_minimal C/C++ Probl make ***: [rar, gen/subdir.msk-2: rar, gen/hal_data.o] Error 1 ran2_minimal make ***: [rar, gen/subdir.msk-2: rar, gen/hal_data.o] Error 1 ran2_minimal C/C++ Probl make ***: [rar, gen/subdir.msk-2: rar, gen/hal_data.o] Error 1 ran2_minimal C/C++ Probl C/C++ Probl	♥ Ø Errors (6 items) Image: Section (2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2							
Image: Text gen/subdit.mik-22: zng.gen/subdit.mik-24: zng.ge	Image: Text grage: Second S			Resource	Path	Location	Туре	
Imake: *** [rzn_gen/subdir.mls42: rzn_gen/common_data.o] Error 1 rzn2l_minimal C/C++ Probl Imake: *** [rzn_gen/subdir.mls42: rzn_gen/main.o] Error 1 rzn2l_minimal C/C++ Probl Imake: *** [rzn_gen/subdir.mls42: rzn_gen/main.o] Error 1 rzn2l_minimal C/C++ Probl Imake: *** [rzn_gen/subdir.mls42: rzn_gen/main.o] Error 1 rzn2l_minimal C/C++ Probl Imake: *** [rzn_gen/subdir.mls26: src/hal_entry.o] Error 1 rzn2l_minimal C/C++ Probl	Image: **** [rzn_gen/subdir.mk:42: rzn_gen/common_data.o] Error 1 rzn2Lminimal C/C++ Probl Image: **** [rzn_gen/subdir.mk:42: rzn_gen/main.o] Error 1 rzn2Lminimal C/C++ Probl Image: **** [rzn_gen/subdir.mk:42: rzn_gen/main.o] Error 1 rzn2Lminimal C/C++ Probl Image: **** [rzn_gen/subdir.mk:42: rzn_gen/main.o] Error 1 rzn2Lminimal C/C++ Probl Image: **** [rzn_gen/subdir.mk:26: rzn/subdir.mk:26: r			bsp_mcu_family_cfg.h	/rzn2l_minimal/rzn.	. line 4	C/C++ Probl	
Image: **** [zn_gen/subdir.mk:42: zn_gen/main.o] Error 1 zn2l_minimal C/C++ Probl Image: **** [src/subdir.mk:26: src/hal_entry.o] Error 1 zn2l_minimal C/C++ Probl	Image: Text geng/subdir.mb42: rg.gen/main.0] Error 1 rzn21_minimal C/C++ ProbL Image: Text geng/subdir.mb42: rg.gen/main.0] Error 1 rzn21_minimal C/C++ ProbL Image: Text geng/subdir.mb42: rg.gen/main.0] Error 1 rzn21_minimal C/C++ ProbL Image: Text geng/subdir.mb42: rg.gen/main.0] Error 1 rzn21_minimal C/C++ ProbL Image: Text geng/subdir.mb42: rg.geng/subdir.mb42: rg.geng/sub42: rg.g		Make: *** [rzn_gen/subdir.mk:42: rzn_gen/common_data.o] Error 1					
make: *** [src/subdir.mk:26: src/hal_entry.o] Error 1 rzn2l_minimal C/C++ Probl	Image: Text State							
	make: *** Waiting for unfinished jobs rzn2L_minimal C/C++ Probl							
				rzn2l minimal			C/C++ Probl	

No. 5 Deleted due to one of the issues with Known Issues No. 3



Title	Pin configuration er	ror occurs in M	PX-IO 16b	it ope	erating	g mode of "	r_bsc'	
Target	RZ/T2M, RZ/T2L, I	RZ/N2L						
Category	FSP Configuration,	Pins						
Description	Pin assignment is an error will occur if Inp				16bit o	operation m	ode of	"r_bsc", but an
	Pin Configuration					O Generate Project	Content	
	Select Pin Configuration	🖬 Ex	port to CSV file	Config	gure Pin D	Priver Warnings		
	RSK+RZN2L	 Manage configurations. 	🗹 Gen	erate da	ta: g_bsp	_pin_cfg		
	Pin Selection $\models \oplus = \downarrow^{a}_{Z}$	Pin Configuration				Cycle Pin	n Group	
	bsc X	Name	Value	Lock	Link		^	
	✓ ✓ Ports	Pin Group Selection	Mixed				_	
	👻 🖌 Other Pins	Operation Mode	OMPX-IO 16bit		1000			
	✓ BSCANP	✓ Input/Output			<□>		_	
	✓ ❷ Peripherals	A0 A1	None 8 * None	(T) (T)				
	✓ [©] ExBus:BSC	A1 A2	None None				_	
	😣 BSC	A3	✓ P05 1	ď	4			
		A4	* None		⇒		~	
		<					>	
	Module name: BSC							
	Pin Function Pin Number							
	Summary BSP Clocks ⁰ Pins In	terrupts Event Links Stack	s Components					
Workaround	Please set "Custom" to operation mode of "r		de" of "r_b	sc" in	Pins t	tab when yo	ou use]	MPX-IO 16bit

No. 6 Resolved



Title	Build error when using definition name of input/output external pins for module.	
Target	RZ/T2M, RZ/T2L, RZ/N2L	
Category	FSP Configuration, Pins	
Description	After code generation, the definition of input/output external pins for the module is generated in fsp_cfg/bsp/bsp_pin_cfg.h, but the defined values are not defined in FSP. When using the defined name in a user application, a build error occurs.	
	Image: Description of the state of the	
	h bsp_pin_cfg.h c hal_entry.c × h bsp_io.h 48 & unused variable 'tmp' [-Wunused-variable] 49 uint16_t tmp ETH0_RXD0: 50	
	Properties Problems Console × Search Search Search Search Search Properties OTBuild Console [rzt2m_ether_test] /src/hal_entry.c: In function 'hal_entry': D:\ws\rz\t2120\rzt2m_ether_test\rzt_cfg\fsp_cfg\bsp/bsp_pin_cfg.h:51:20: error: 'IOPORT_PORT_10_PIN_1' undeclared S1 #define ETH0_RXD0 (IOPORT_PORT_10_PIN_1)	
Workaround	Please add definition to read IOPORT_PORT_mm_PIN_n as BSP_IO_PORT_mm_PIN_n in hal_entry. Do NOT edit file fsp_cfg/bsp/bsp_pin_cfg.h because its contents will be overwritten.	
	An example of a setting: When using ETH0_RXD0 (IOPORT_PORT_10_PIN_1), add definition of #define IOPORT_PORT_10_PIN_1 (BSP_IO_PORT_10_PIN_1) in hal_entry.c.	
	Image: bsp_pin_cfg.h Image: hal_entryc Image: hal_entryc Image: hal_entryc 2 Image: mail of the second secon	

No. 7 Resolved

No. 8 Resolved

Issue	"R_SCI_UART_BaudCalculate()" of "r_sci_uart" module properly works ONLY when its clock source is SCInASYNCCLK and its frequency is 96MHz.
Target	RZ/T2M, RZ/T2L, RZ/N2L
Category	FSP Modules, Serial Communication Interface (SCI) UART
Description	The "R_SCI_UART_BaudCalculate()" of "r_sci_uart" module works ONLY when its clock source is "SCInASYNCCLK" and its frequency is "96MHz"; therefore, when the module uses "PCLKM" as its clock source or the frequency is not 96MHz, the API function will be not work properly.
Workaround	The clock source and frequency are limited in Clocks and Stacks tab; therefore, you can NOT use the PCLKM clock and can NOT change the clock frequency.



Issue	"R_SPI_CalculateBitrate()" of "r_spi" module properly works ONLY when its clock source is SPInASYNCCLK and its frequency is 96MHz.	
Target	RZ/T2M, RZ/T2L, RZ/N2L	
Category	FSP Modules, Serial Peripheral Interface	
Description	The "R_SPI_BaudCalculate()" of "r_spi" module works ONLY when its clock source is "SPInASYNCCLK" and its frequency is "96MHz"; therefore, when the module uses "PCLKM" as its clock source or the frequency is not 96MHz, the API function will be not work properly.	
Workaround	The clock source and frequency are limited in Clocks and Stacks tab; therefore, you can NOT use the PCLKM clock and can NOT change the clock frequency.	

No. 9 Resolved

No. 10 Resolved

Issue	A warning occurs when building "r_gmac" module with the gcc compiler.
Target	RZ/T2M, RZ/T2L
Category	FSP Modules, Ethernet
Description	The following warning occurs when building "r_gmac" module with the gcc compiler. /rzt/fsp/src/r_gmac/r_gmac.c:2173:14: warning: the comparison will always evaluate as 'false' for the pointer operand in 'pp_phy_instance + (sizetype)(port * 12)' must not be NULL [-Waddress] 2173 if (NULL == pp_phy_instance[port]) ^~
Workaround	Please ignore this warning.

No. 11 Resolved

Issue	In FSP Documentation, there is incorrect description in. "API Reference > Modules > Ethernet PHY" page.
Target	RZ/T2M, RZ/N2L
Category	FSP Modules, Ethernet PHY
Description	In the "API Reference > Modules > Ethernet PHY" page in FSP Documentation, the default column description of "Select PHYs to use" configuration is incorrect.
Workaround	When reading the incorrect description, please replace the reading of it with follows. [Error]
	config.driver.ether_phy.phy_lsi.default,config.driver.ether_phy.phy_lsi.0,config.driver.ether_ phy.phy_lsi.1,config.driver.ether_phy.phy_lsi.2,config.driver.ether_phy.phy_lsi.3,config.drive r.ether_phy.phy_lsi.
	 All check boxes are enabled.



Issue	The interrupt number cannot be successfully acquired by the R_FSP_CurrentIrqGet() when multiple interrupt occurs.	
Target	RZ/N2L	
Category	FSP Modules, FreeRTOS	
Description	The interrupt number cannot be successfully acquired by the R_FSP_CurrentIrqGet() when using multiple interrupt handlers with different priority levels in FreeRTOS.	
Workaround	<pre>Please modify the followings for the countermeasure against nested interrupts. Target File: port.c void vAplicationIRQHendler (uint32_t ulICCIAR) { /* Re-mable interrupt. */ ase("GoSie 1"); endif bsp_commo_interrupt_handler(ulICCIAR); bsp_commo_interrupt_handler(ulICCIAR); /* Get interrupt_handler (uint32_t id) (</pre>	

No. 12 Resolved



Issue	Block Media Custom Implementation can be selected as Memory Implementation for "rm_freertos_plus_fat" module, but it cannot be used.	
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/N2L	
Category	FSP Configuration, Stacks	
Description	In Stacks tab of Configuration, Block Media Custom Implementation can be selected as Memory Implementation for "rm_freertos_plus_fat" module, but it is unsupported and causes build errors.	
Workaround	Please select Block Media USB as Memory Implementation for "rm_freertos_plus_fat" module.	

No. 13 Resolved

No. 14

Issue	The second argument of "r_mtu3" APIs do not match with common API.
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L
Category	FSP Modules, MTU3
Description	The second argument of these three APIs "R_MTU3_PeriodSet()", "R_MTU3_InfoGet()", and "R_MTU3_StatusGet()" of the "r_mtu3" module, do not match with the API in "r_timer api.h" header file
Workaround	You cannot call these API by using function pointer g_timer0.p_api->periodSet() g_timer0.p_api-> InfoGet() g_timer0.p_api-> StatusGet() Please use API by calling them directly R_MTU3_PeriodSet() R_MTU3_InfoGet() R_MTU3_StatusGet() For reference how to use these APIs, please refer to MTU3 Examples in <u>FSP documentation</u> .



No. '	15
-------	----

Issue	In multiprocessing, a configuration error occurs when "r_gpt" module is used for both projects for CPU0 and CPU1.
Target	RZ/T2M, RZ/T2ME
Category	FSP Configuration, Stacks
Description	When using "r_gpt" module in Stacks tab of both projects for CPU0 and CPU1, a configuration error occurs. "r_gpt" module can only be used with either CPU0 or CPU1 in multiprocessing, regardless of the Unit or Channel number used.
Workaround	Please use "r_gpt" module ONLY with either CPU0 or CPU1 in multiprocessing.

No. 16 Resolved

Issue	Project build error occur when 32-bit bus NOR flash and xSPI0 x8 boot modes are selected on RZT Custom User Board.	
Target	RZ/T2M	
Category	FSP Configuration, BSP	
Description	When the following boards (boot mode) are selected, the required definitions are not generated a build error occurs.	
	• RZT Custom User Board (32-bit bus NOR flash boot mode)	
	• RZT Custom User Board (xSPI0 x8 boot mode)	
Workaround	Please don't select 32-bit bus NOR flash and xSPI0 x8 boot modes on RZT Custom User Board.	

No. 17 Resolved

Issue	The secondary project for multiprocessing cannot be created when xSPI1 x1 boot modes are selected on RZT Custom User Board.	
Target	RZ/T2M	
Category	FSP Configuration, BSP	
Description	When the following boards (boot mode) are selected for the primary project of multiprocessing, a variable required for multiprocessing is not defined and the secondary project cannot be created.	
	• RZT Custom User Board (xSPI1 x1 boot mode)	
Workaround	Please don't select xSPI1 x1 boot modes on RZT Custom User Board when multiprocessing.	



-			
Issue	An incorrect value is set to a pin select value for MTU0-B/MTU6/MTU7 as MTU3 output pin.		
Target	RZ/T2L		
Category	FSP Modules, POE3		
Description	Pin select value of MTU3 output pins used in FSP do not match with User's Manual: Hardware. Therefore, what you want to set up is not correctly described in the generated file of FSP Configuration.		
Workaroun d	 When using "r_poe3" and MTU0-B/MTU6/MTU7 as MTU3 output pin, please follow these four steps: 1. Add "Port Output Enable 3 for MTU3 (r_poe3)" on Stacks tab of FSP Configuration. 2. Click Generate Project Content button and "r_poe3" code is generated. 3. Disable code generating function. After this setting, the code cannot be generated. [For e² studio Smart Configurator] Use the following settings to suppress the code generating operation. If this setting is missed, code generating operation is automatically executed at clean build, and the changes made in step 4 revert to the original. a. Uncheck "Project Properties > Builders > DDSC Builder" 		
	 Properties for Builders Configure the builders for the project: Configure the builders for the project: Configure the builders for the project: DDSC Builder MCU Project Natures Project References Renease QE Run/Debug Settings Task Tags Validation 		
	Image: Cancel		
	 [For FSP Smart Configurator (IAR EWARM)] No need setting since code generation is not executed automatically. 4. Modify definitions in rzt_gen/hal_data.c Change the value of [module name]_pwm_pin_setting[] or [module name]_complementary_pwm_setting[1].pin_setting[X](X=0,1,2) according to the MTU3 output pins used. The tables below show the replacements required for each MTU3 output pins. 		
	<pre>File to be modified: rzt_gen/hal_data.c /* Setting structure for pwm pin. */ static const poe3_pwm_pin_setting_t g_poe30_pwm_pin_setting[] = { .pwm_pin_select = POE3_PIN_SELECT_0, .hiz_output_enable = false }, .pwm_select =</pre>		

No. 18 Resolved



RZ/T2, RZ/N2

<pre>.negative .positive .negative</pre>	pwm_pin_se pwm_pin_se pwm_pin_ac	<pre>lect = POE3_PIN_SELECT_0, lect = POE3_PIN_SELECT_0, tive_level = POE3_ACTIVE_LEVEL_SETTING_NOM tive_level = POE3_ACTIVE_LEVEL_SETTING_NOM false },</pre>	
For MTU0-B	,		
MTU3	Port	Location (Struct [module name]	Replace with
output pin	1011	pwm pin setting[])	
mtioc0b	P14 4	the second pwm pin select	.pwm pin select =
mnocoo	114_4	the second pwin_pin_select	POE3_PIN_SELECT_1
mtioc0b	P24 0	the second pwm pin select	.pwm_pin_select =
	_	1 _1 _	POE3_PIN_SELECT_2
mtioc0b	P13_3	the second pwm_pin_select	<pre>.pwm_pin_select =</pre>
			POE3_PIN_SELECT_3
For MTU6/N	1TU7		
MTU3	Port	Location (Struct [module	Replace with
output pin		name]_complementary_pwm_setting[])	
mtioc6b	P21_2	the second pin_setting[0]	.positive_pwm_pin_sele
	D 00 F		= POE3_PIN_SELECT_1,
mtioc6b	P08_5	the second pin_setting[0]	.positive_pwm_pin_sele
<i>i</i> : (1	D21 4		= POE3_PIN_SELECT_2,
mtioc6d	P21_4	the second pin_setting[0]	<pre>.negative_pwm_pin_sele</pre>
mtioc6d	P08 7	the second pin setting[0]	.negative_pwm_pin_sele
muocod	100_/	the second pin_setting[0]	= POE3_PIN_SELECT_2,
mtioc7a	P21 5	the second pin setting[1]	.positive_pwm_pin_sele
		···· ••••••• •••••••••••••••••••••••••	= POE3_PIN_SELECT_1,
mtioc7a	P09_0	the second pin_setting[1]	.positive_pwm_pin_sele
			= POE3_PIN_SELECT_2,
mtioc7c	P21_7	the second pin_setting[1]	.negative_pwm_pin_sele
			= POE3_PIN_SELECT_1,
mtioc7c	P09_2	the second pin_setting[1]	.negative_pwm_pin_sele
			= POE3_PIN_SELECT_2,
mtioc7b	P21_6	the second pin_setting[2]	.positive_pwm_pin_sele
	D00_1	the second min settin [2]	= POE3_PIN_SELECT_1,
mtioc7b	P09_1	the second pin_setting[2]	.positive_pwm_pin_sele
mtioc7d	P22 0	the second pin setting[2]	<pre>= POE3_PIN_SELECT_2, .negative pwm pin sele</pre>
muoc/u	122_0	the second pin_setting[2]	= POE3_PIN_SELECT_1,
	P09 3	the second pin setting[2]	.negative_pwm_pin_sele
mtioc7d			



Issue	Build error when using DSMIFn_ERR as an additional trigger for "r_poe3" module.				
Target	RZ/T2L				
Category	FSP Modules, POE3				
Description	When using DSMIFn_ERR as an additional trigger in FSP Configuration, after code generatives of DSMIFn_ERR additional trigger for the module are generated in rzt_gen/hal_date the defined values are not defined in FSP, so a build error occurs. Property of r_poe3 stack		le are generated in rzt_gen/hal_data.c. But		
	g_poe30 Port Output Enable 3 for MTU3 (r_poe3)				
	Settings Property	Value			
	Common Parameter Checking	Default (BSP)			
	Multiplex Interrupt Module g_poe30 Port Output Enable 3 for MTU3 (r_poe3)	Disabled			
	> General				
	> Input > Output				
	V MTU0 Pin Control				
	> MTU0-A (MTIOCOA) > MTU0-B (MTIOCOB)				
	> MTU0-C (MTIOCOC) > MTU0-D (MTIOCOD)				
	 Additional MTU0 pin control request condition (Always enable POE8# Input) 				
	POE0# Input POE4# Input				
	POE10# Input POE11# Input				
	DSMIF0 Error				
	DSMIF1 Error > MTU3 and MTU4 Pin Control				
	> MTU6 and MTU7 Pin Control				
	Build error log				
	DT Properties 🔐 Problems 🏶 Smart Browser 🖳 Console X 🕺 🗘 😨 🐨 💷 🖝 🗇 🖓 🐨 🛄 🖉 🐨 🐨 🐨 🖓 🐨				
	CDT Build Console [rtz1]/rzt gen/hal_data.c:70:40: error: 'R_POE3_POECR5_DE0ADDMT0ZE_Pos' undeclared here (not in a function); did you mean 'R_POE3_POECR5_D0E0ADDMT0ZE_Pos'?				
	Building file:/rzt_gen/hal_data.c 70 .mtu0_control_channel_mask = ((1U << R_POE3_POECR5_DE0ADDMT0ZE_Pos) (1U << R_POE3_POECR5_DE1ADDMT0ZE_Pos) 0U),				
	R POES POECR5 December Voer Pos				
	<pre>fix-it:"/rzt_gen/hal_data.c":{70:40-70:60}:rCRS_DECADDMT02E_POS"/rzt_gen/hal_data.c":{70:40-70:60}:R_POE3_POECRS_DECADDMT02E_POS"/rzt_gen/hal_data.c:70:80: error: 'R_POE3_POECRS_DELADDMT02E_POS' undeclared here (not in a function); did you mean 'R_POE3_POECRS_D0E1ADDMT02E_POS' 70mtu@_control_channel_mask = ((1U << R_POE3_POECRS_DECADDMT02E_POS) (1U << R_POE3_POECRS_DELADDMT02E_POS) (0U), 70mtu@_control_channel_mask = ((1U << R_POE3_POECRS_DECADDMT02E_POS) (1U << R_POE3_POECRS_D0E1ADDMT02E_POS) (0U), 70mtu@_control_channel_mask = (1U << R_POE3_POECRS_D0E0ADDMT02E_POS) (1</pre>				
	 fix-it:"/rzt_gen/hal_data.c":{70:80-70:109}:"R_POE3_POECR5_D0E1ADDMT0	ZE_Pos"	POE3_POECR5_DØE1ADDMTØZE_Pos		
Workaround	When using "r poe3" and DSMIFn ERR(n=	0,1) as an a	additional trigger, please follow these four		
W OI Kai Uuliu	steps:				
	1-3. Refer to the workaround of No. 18.				
	4. Modify definitions in rzt_gen/hal_data.c		MTHO - 14:4: 14:-		
	When using DSMIF 0 ERROR, DSMIF 1				
	R_POE3_POECR5_DE0ADDMT0ZE_Pos to R_POE3_POECR5_D0E0ADDMT0Z				
	• R_POE3_POECR5_DE1ADDMT0ZE_Pos to R_POE3_POECR5_D0E1ADDMT0ZE				
	When using DSMIF 0 ERROR, DSMIF 1	ERROR for	· MTU3/4 additional trigger, modify:		
	R POE3 POECR4 DE0ADDMT34Z				
	 K_IOES_IOECR4_DE0ADDMI34ZE_IOS to R_POE3_POECR4_D0E0ADDMT34ZE_Pos R_POE3_POECR4_DE1ADDMT34ZE_Pos to 				
	R_POE3_POECR4_D0E1ADDMT34ZE_Pos				
	When using DSMIF 0 ERROR, DSMIF 1	ERROR for	MTU6/7 additional trigger, modify:		
	R POE3 POECR4 DE0ADDMT67Z				
	R POE3 POECR4 D0E0ADDMT67				
		_			
	R_POE3_POECR4_DE1ADDMT67Z				
	R POE3 POECR4 D0E1ADDMT67	ZE Pos			

No. 19 Resolved



Issue	Control setting values for MTU3 output pins in Stacks tab of FSP Configuration are set to the incorrect pin.		
Target	RZ/T2M, RZ/T2L, RZ/T2ME		
Category	FSP Configuration, Stacks	1	
Description	Settings for MTU4-B(MTIOC4B) and MTU4-D(MTIOC4D) in the Stacks tab property would be treated as settings for MTU4-A(MTIOC4A) and MTU4-C(MTIOC4C) in the generated file by Smart Configurator. All MTU3 output pins for MTU4 and MTU7, are the same as above.		
		atput Enable 3 for MTU3 (r_poe3) of St	
	Second Category	Third Category	Used for
	MTU3 and MTU4 Pin Control	MTU4-B(MTIOC4B) and MTU4- D(MTIOC4D)	MTU4-A(MTIOC4A) and MTU4- C(MTIOC4C)
	MTU3 and MTU4 Pin Control	MTU4-A(MTIOC4A) and MTU4- C(MTIOC4C)	MTU4-B(MTIOC4B) and MTU4- D(MTIOC4D)
	MTU6 and MTU7 Pin Control	MTU7-B(MTIOC7B) and MTU7- D(MTIOC7D)	MTU7-A(MTIOC7A) and MTU7- C(MTIOC7C)
	MTU6 and MTU7 Pin Control	MTU7-A(MTIOC7A) and MTU7- C(MTIOC7C)	MTU7-B(MTIOC7B) and MTU7- D(MTIOC7D)
Workaround	In configuration of MTU4 and MTU7, please replace B/D and A/C.		
	 MTU3 and MTU4 Pin Control MTU3-B (MTIOC3B) and MTU3-D (MTIOC3D) MTU4-B (MTIOC4B) and MTU4-D (MTIOC4D) Use this field to configure for MTU4-A (MTIOC4A) and MTU4-C (MTIOC4C) Use this field to configure for Additional MTU3/4 pin control request condition (Always enable POE0# Input) MTU6-B (MTIOC6B) and MTU6-D (MTIOC6D) MTU7-B (MTIOC7B) and MTU7-D (MTIOC7D) Use this field to configure for MTU7-A (MTIOC7A) and MTU7-C (MTIOC7C) Use this field to configure for Additional MTU6/7 pin control request condition (Always enable POE4# Input) 		MTU4-D MTU7-C

No. 20 Resolved

No. 21

Issue	A bug that prevented the setup of PLL1.	
Target	RZ/N2L	
Category	FSP Configuration, Clocks	
Description	The PLL1 state setting in Clocks tab is invalid.	
Workaround	Please don't use PLL1 state setting.	



Issue	A section may not be copied correctly when it is not aligned and the section size is not a multiple of the alignment width.	
Target	RZ/N2L	
Category	FSP Modules, BSP	
Description	Depending on a combination of section size and placement address, when the section is copied, some data from the following section may also be copied with it.	
	A case that cannot be copied correctly: The address of .data_noncache section is 0x30190005 and its size is 0x23 bytes. (Alignment si is 4 bytes.) .data_noncache @x30190005 @x23 ./src/hal_entry.o	
Workaround	All section sizes must be aligned by 4 bytes and the data size should be a multiple of the number of bytes in the alignment. Section sizes can be found in the following files:	
	gcc: [project name]/Debug/[project name].map	
	• iccarm: [project name]/Debug/List/[project name].map	

No. 23

Issue	Initial values of data placed in some sections were overwritten with 0.	
Target	RZ/N2L	
Category	FSP Modules, BSP	
Description	 When selecting XXXXX (RAM execution without flash memory) as "Board" in BSP tab of FSP Configuration, variables placed in the following sections are always cleared to zero. .dmac_link_mode .shared_noncache_buffer .noncache_buffer 	
Workaround	Do NOT place data with initial values in the above sections.	



Issue	Some sections were not initialized in the flash boot project.	
Target	RZ/N2L	
Category	FSP Modules, BSP	
Description	When selecting a flash boot mode as "Board" in BSP tab of FSP Configuration, variables placed in the following sections are NOT initialized.	
	Boards for flash boot mode	
	• XXXXX (xSPI0 x1 boot mode)	
	• XXXXX (16-bit bus NOR flash boot mode)	
	RZN2L Custom User Board (xSPI0 x8 boot mode)	
	RZN2L Custom User Board (xSPI1 x1 boot mode)	
	Sections	
	• .dmac_link_mode	
	.shared_noncache_buffer	
	.noncache_buffer	
Workaround	Please initialize the variables placed in the above sections in the user application.	

Issue	DSMIF 0/1 error 1 trigger macros are not defined.			
Target	RZ/T2L			
Category	FSP Modules, POEG			
Description	In "bsp_override.h" of rzt2l device, enum e_poeg_trigger, the definition for DSMIF0 error 1 and DSMIF1 error 1 are missing.			
	6			
	When setting as below, build errors will occur.			
	Property of r poeg stack			
	✓ Module g_poeg0 Port Output Enable for GPT (r_poeg)			
	✓ General			
	✓ Trigger GTETRG Pin			
	GPT Output Level			
	Oscillation Stop			
	DSMIF0 error			
	DSMIF1 error			
	DSMIF0 error 1			
	DSMIF1 error 1			
	Name g_D64g0 Channel 0			
	Group A			
	Generated code by configurator const poeg_cfg_t g_poeg0_cfg = { .trigger = (poeg_trigger_t) (POEG_TRIGGER_DERR0E_1 POEG_TRIGGER_DERR1E_1 POEG_TRIGGER_SOFTWARE), .polarity = POEG_GTETRG_POLARITY_ACTIVE_HIGH, Build error log make -routput-sync -j8 all /rat gen/hal data.ci7:36: error: 'POEG_TRIGGER_DERR0E 1' undeclared here (not in a function); did you mean 'POEG_TRIGGER_DERR0E'?			
	Building file: ./rzt_gen/hal_data.c 7 POEG_TRIGGER_PIN POEG_TRIGGER_DERR0E_1 POEG_TRIGGER_DERR1E_1 POEG_TRIGGER_SOFTWARE),			
	POEG_TRIGGER_DERRØE fix-it:"/rzt_gen/hal_data.c":{7:36-7:57}:"POEG_TRIGGER_DERRØE"			
	<pre>/rzt_gen/hal_data.c:7:60: error: 'POEG_TRIGGER_DERRIE 1' undeclared here (not in a function); did you mean 'POEG_TRIGGER_DERRIE ? 7 POEG_TRIGGER_DIRN POEG_TRIGGER_DERROE_1 POEG_TRIGGER_DERROE_1 POEG_TRIGGER_SOFTWARE),</pre>			
	POEG_TRIGGER_DERR1E fix-it:"/rzt_gen/hal_data.c":{7:60-7:81}:"POEG_TRIGGER_DERR1E"			
	make: *** [rzt_gen/subdir.mk:42: rzt_gen/hal_data.o] Error 1			
	make: *** Waiting for unfinished jobs			
	August 22 - 22 - I and and the states -			



Workaround	Add definition for DSMIF0 error 1 and DSMIF1 error 1 trigger in enum e_poeg_trigger. Location: rzt/fsp/src/bsp/mcu/rzt2l/bsp_override.h, enum e_poeg_trigger Add content:	
	<pre>POEG_TRIGGER_DERRØE_1 = 1U << 18,</pre>	///< Permit output disabled by DSMIF0 error 1 detection
	<pre>POEG_TRIGGER_DERR1E_1 = 1U << 19,</pre>	<pre>///< Permit output disabled by DSMIF1 error 1 detection</pre>

Issue	DSMIF 0/1 error 1 status macros are not defined.
Target	RZ/T2L
Category	FSP Modules, POEG
Description	In "bsp_override.h" of rzt2l device, enum e_poeg_state, the definition for DSMIF0 error 1 state and DSMIF1 error 1 state are missing.
	When using R_POEG_StatusGet,
	• If POEG module is in state GPT output disabled due to DSMIF0 error 1, the p_status will be 0x100000 instead of POEG_STATE_DSMIF0_1_DISABLE_REQUEST.
	· If POEG module is in state GPT output disabled due to DSMIF1 error 1, the p_status will be 0x200000 instead of POEG_STATE_DSMIF1_1_DISABLE_REQUEST.
Workaround	When the POEG module is in the 'GPT output disabled' state due to a DSMIF0 error 1, assume that $p_{status} = 0x100000$ corresponds to POEG_STATE_DSMIF0_1_DISABLE_REQUEST.
	When the POEG module is in the 'GPT output disabled' state due to a DSMIF1 error 1, assume that $p_status = 0x200000$ corresponds to POEG_STATE_DSMIF1_1_DISABLE_REQUEST.

No. 27

Issue	Missing constraint for DSMIF error trigger in channel 1 and channel 2.			
Target	RZT2M, RZT2ME, RZT2L, RZT2H			
Category	FSP Modules, POEG			
Description	POEG channel 1 and channel 2 do not support DSMIF error trigger in all RZT devices. But currently there are no constraints to prevent configuring DSMIF error trigger for channel 1 and channel 2.			
	g_poeg0	Port Output Enable for GPT (r_poeg)		
	Settings API Info	Property ✓ Module g_poeg0 Port Output Enable for GPT (r_poeg) ✓ General ✓ Trigger GTETRG Pin GPT Output Level Oscillation Stop DSMIF0 error DSMIF1 error 1 Name Channel Group > Input > Interrupts ✓ Pins	Value	
Workaround	Use th	e DSMIF error trigger for channel 0 only.		



Issue	FreeRTOS+FAT format process is not executed correctly.		
Target	RZT2M, RZT2ME, RZT2L, RZT2H		
Category	FSP Modules, FreeRTOS+FAT		
Description	Executing the FF_Format function causes a USB AHB bus error and the FreeRTOS+FAT format function processing is not executed correctly.		
Workaround	6 <u> </u>		

No. 29

Issue	Caution when specifying program placement in linker scripts		
Target	RZ/T2H		
Category	Others, Linker script		
Description	The reserved area in the device address space cannot be used for a program placement, but no error occurs when placed there by a linker script.		
	For example, CA55 core has reserved areas ATCM and BTCM, however the linker script is ready to place them there, and no error occurs when the following description exists.		
	<pre>.text TEXT_ADDRESS : AT (TEXT_ADDRESS) { abbreviation } > ATCM</pre>		
Workaround	Do not specify that any program is to be placed at the reserved area in a linker script.		



Issue	In the secondary project for multiprocessing, no error occurs when there is a conflict in a resource used with the preceding project.		
Target	RZ/T2H		
Category	FSP Configuration, Stacks		
Description	As noted in 6.2.4 Duplication of Resources, Smart Configurator has the feature to inform about resource duplication. However, it does not show an error when the secondary project for multiprocessing uses the same channel/unit of ADC, MTU3, GPT, and TSU_B as preceding projects.		
Workaround	Please don't use the same channel/unit of ADC, MTU3, GPT, and TSU_B between the preceding project and the secondary project of RZ/T2H.		

Issue	Errors occur when setting ELC in r_gpt module.
Target	RZ/T2H
Category	FSP Configuration, Stacks
Category Description	When setting ELC event trigger source in the stack of r_gpt module, errors will occur and cannot generate the value.
Workaround	Description Configuration ser. Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events only support unit 0 Guined Timer, General PWM (r.gpt): LLPP GPT events o
	 Configurator, follow these steps. Add the r_elc stack in the Stacks tab of FSP Configuration and include the r_elc header in your application. Call the following functions to initially set up the r_elc module. R_ELC_Open(); R_ELC_Enable(); Use the R_ELC_LinkSet(); function to configure the event triggers. Example: R_ELC_Open(&g_gpt_test_elc_ctrl, &g_elc_cfg); R_ELC_Enable(&g_gpt_test_elc_ctrl); R_ELC_LinkSet(&g_gpt_test_elc_ctrl, ELC_PERIPHERAL_GPT00_A, ELC_EVENT_INTCPU0);



Issue	CR52 CPU1 of RZ/T2H is implemented to run program from System SRAM instead of CPU1 ATCM.
Target	RZ/T2H(CR52)
Category	Others, Linker script
Description	CR52 CPU1 of RZ/T2H has CPU1 ATCM and CPU1 BTCM as hardware, and when reset is released, the program runs from CPU1 ATCM. On the other hand, this FSP is implemented to run from System SRAM for RZ/T2H CR52 CPU1 like the program for RZ/T2M, and does not use CPU1 ATCM or CPU1 BTCM of RZ/T2H as the start of the program.
Workaround	Please consider using this implementation that uses System SRAM.

No. 33

Issue	No Error Occurs when entering out-of r_pcie_rc module configurations.	f-range values for	window parameters in	r_pcie_ep and
Target	RZ/T2H			
Category	FSP Configuration, Stacks			
Description	There is a problem with the r_pcie_ep and r_pcie_rc module configuration screens where entering unsupported values does not generate an error and the code can be generated with unusable values. The configuration items for which the validity judgment of input value does not work are as follows.			nusable values.
	Configuration item	r_pcie_ep	r_pcie_rc	
	AXI Window Base	1	1	
	AXI Window Mask	1	1	
	AXI Window Destination	1	1	
	PCIe Window Base	1	1	
	PCIe Window Mask	1	1	
	PCIe Window Destination	1	1	
	MSI Receive Window Address	-	1	
	MSI Receive Window Mask	-	1	
Workaround	Please enter the values so that they meet AXI Window Base and PCIe Window B - "greater than or equal to 0" and "addre AXI Window Mask and PCIe Window N - "greater than or equal to 0", 'the lower AXI Window Destination and PCIe Win - "greater than or equal to 0" and "addre MSI Receive Window Mask - "greater than or equal to 0" and "the low MSI Receive Window Address - Align according to "MSI Receive Win	ase ess is 4Kbyte align Aask 12 bits are 1', and dow Destination ess is 4Kbyte align ower 2 bits are 1"	ed" 1 "the 63rd bit is 0"	



Issue	DDR and PCIE0/1 memory cannot be used in secondary (or later) projects with flash boot mode.
Target	RZ/T2H
Category	Others, Address space
Description	If you use DDR or PCIE0/1 memory in a secondary (or later) project with flash boot mode, the binary file will be huge size. Therefore multicore operation is not possible.
Workaround	Secondary (or later) projects with flash boot mode do not use DDR or PCIE0/1 memory.

No. 35

Issue	r_gmac_b module cannot use zero-copy mode.
Target	RZ/T2H
Category	FSP Modules, GMAC
Description	r_gmac_b module cannot use zero-copy mode. The "Zero-copy mode" setting in the r_gmac_b configuration cannot be changed from the default "Disable."
Workaround	Please use the standard buffers provided by the r_gmac_b module for transmit and receive buffers.

Issue	r_adc module does not support the calibration function.
Target	RZ/T2H
Category	FSP Modules, ADC
Description	The 12-bit A/D converter needs to be calibrated before A/D conversion after reset is released. However, since the FSP does not support the calibration function, the accuracy shown in the electrical characteristics chapter of device user's manual cannot be guaranteed.
Workaround	This will be implemented in the next version of FSP.
	As a temporary measure, the calibration process must be implemented in the user application before the R_ADC_Open function is executed.
	The following is an example of implementation. (In the case of ADC Unit2)
	#define ADC_ADCALCTL_SET_CAL 1U
	/* Release module stop for ADC12 */
	<pre>R_BSP_RegisterProtectDisable(BSP_REG_PROTECT_LPC_RESET);</pre>
	R_BSP_MODULE_START(FSP_IP_ADC12, 2);
	R_BSP_RegisterProtectEnable(BSP_REG_PROTECT_LPC_RESET);
	R_BSP_SoftwareDelay(1, BSP_DELAY_UNITS_MICROSECONDS);
	/* Write ADCCALCTL.CAL bit to 1 to start calibration. */
	R_ADC12 <mark>2</mark> ->ADCALCTL_b.CAL = ADC_ADCALCTL_SET_CAL;
	/* Poll ADCCALCTL.CAL_RDY bit until it is changed to 1. */
	FSP_HARDWARE_REGISTER_WAIT(R_ADC12 <mark>2</mark> ->ADCALCTL_b.CAL_RDY, 1U);
	/* Confirm ADCCALCTL.CAL_ERR bit is 0.*/
	<pre>FSP_HARDWARE_REGISTER_WAIT(R_ADC122->ADCALCTL_b.CAL_ERR, 0U);</pre>
	/* Write ADCCALCTL.CAL bit to 0 */
	R_ADC12 <mark>2</mark> ->ADCALCTL_b.CAL = 0U;
	/* Initializing the ADC module */
	R_ADC_Open(&g_adc0_ctrl,&g_adc0_cfg);



No. 37	
Issue	The USB driver for CA55 project does not work.
Target	RZ/T2H(CA55)
Category	FSP Modules, USB
Description	The R_BSP_MmuPatoVA function executed from USB driver (r_usb_hmsc, r_usb_hcdc, r_usb_hhid modules) in a CA55 project fails to perform the expected address translation, resulting in a USB transfer failure.
Workaround	Please modify \rzt\fsp\src\r_usb_basic\src\driver\r_usb_mmu_pa_to_va.c as follows.
	1. Modify the r_usb_pa_to_va function.
	uint64_t r_usb_pa_to_va (uint64_t paddr)
	{
	uint64_t vaddr = 0;
	#if defined(BSP_CFG_CORE_CA55)
	/* Converts a physical address to a virtual address. */
	<pre>if (FSP_SUCCESS != R_BSP_MmuPatoVa(paddr, &vaddr, BSP_MMU_CONVERSION_NON_CACHE))</pre>
	<pre>{ /* On error, returns the physical address without conversion. */</pre>
	vaddr = paddr;
	}
	<pre>#else /* #if defined(BSP_CFG_CORE_CA55) */</pre>
	vaddr = paddr;
	<pre>#endif /* #if defined(BSP_CFG_CORE_CA55) */</pre>
	return vaddr;
	<pre>} /* End of function r_usb_pa_to_va() */</pre>
	2. Modify the r_usb_va_to_pa function.
	uint64_t r_usb_va_to_pa (uint64_t vaddr)
	{
	uint64_t paddr = 0;
	<pre>#if defined(BSP_CFG_CORE_CA55)</pre>
	/* Converts a virtual address to a physical address. */
	if (FSP_SUCCESS != R_BSP_MmuVatoPa(vaddr, &paddr))
	{
	/* On error, returns the virtual address without conversion. */
	paddr = vaddr;
	<pre>} #else /* #if defined(BSP_CFG_CORE_CA55) */</pre>
	<pre>paddr = vaddr;</pre>
	<pre>#endif /* #if defined(BSP_CFG_CORE_CA55) */</pre>
	return paddr;
	}



Issue	No error returns when entering the virtual addresses that cannot be translated to physical addresses as arguments.				
Target	RZ/T2H(CA55)				
Category	FSP Modules, xSPI_OSPI, xSPI_QSPI, DMAC				
Description	In a CA55 project, there is a problem with the "r_xspi_qspi", "r_xspi_ospi", and "r_dmac" modules that no error returns when entering virtual addresses that cause translation error in MMU as arguments. The following functions have the problem. -R_XSPI_QSPI_Write() -R_XSPI_OSPI_Write() -R_DMAC_Open() -R_DMAC_Reconfigure() -R_DMAC_Reload() -R_DMAC_LinkDescriptorSet()				
Workaround	Do not enter the virtual addresses that cannot be translated to physical addresses as arguments.				



Appendix. Tool Software Limitations

This section describes the limitations regarding the tool software (e² studio, FSP SC) to create and debug FSP projects.

No.	Title	Target Device					Category
		T2M	2M T2L T2M	T2ME	T2H	N2L	
1	When installing, please install into the default installation folder specified by installer.	1	1	1		1	SC, FSP SC
2	Before pressing the reset button on the board, disconnect the e^2 studio connection first.	1	1			1	e ² studio
3	An error has occurred because the program download to the NOR flash area has failed. The download is successful on the second connection.	1				1	e ² studio
4	The user program cannot be stopped immediately after the device boot process.	1	1	1	1	1	e ² studio
5	When using e ² studio installer, if checking the multiple check boxes such as "View Release Notes" and so on to show information on browser, the ONLY head item of checked items is shown.	1	1			1	e ² studio
6	The Memory Region Usage of ATCM displayed in the Memory Usage window of e ² studio is smaller than the actual size by memory region usage of DUMMY.					1	e ² studio
7	When debugging RAM execution without flash memory project with program written to flash memory, erase flash memory before debugging.	1	1	1	1	1	e ² studio
8	Applying RZ/T2 FSP v.1.2.0 pack to a project that is already working with RZ/T2M FSP v.1.1.0 causes an error when connecting the debugger.	1					IAR EWARM
9	The Device Memory Usage of CPU1 in the Memory Usage window does not work properly.	1		1	1		e ² studio
10	When adding the CallbackSet function using the Developer Assistance feature, the second argument needs to be changed.	1	1	1	1	1	e ² studio, SC
11	In IAR EWARM 9.60.1, an error occurs when starting to debug multiprocessing projects of RAM execution without flash memory.	1		1			IAR EWARM
12	Build is failed when executed with different install path.	1	1	1	1	1	FSP SC
13	Unable to debug CA55 flash boot project with e ² studio.				1		e ² studio
14	Unable to restart debugging immediately after debugging ends of CA55 RAM execution without flash memory project in e^2 studio.				1		e ² studio
15	Unable to re-download CA55 binary file.				1		IAR EWARM
16	Unable to access the upper 32-bit address area in memory view.				1		e ² studio
17	Unable to create a CMake project using FSP Smart Configurator.	1	1	1	1		FSP SC

Table 15 List of Tool Software Limitations



RZ/T2, RZ/N2

No.	No. Title		Target Device				
		T2M	T2L	T2ME	T2H	N2L	
18	The secondary project aborts when debugging multicore with flash boot mode.	1		1	~		IAR EWARM
19	Build errors occur in CA55 projects when install e ² studio as the Current user.				1		e ² studio

No. 1 Resolved

Limitation	When installing, please install into the default installation folder specified by installer.			
Target Device	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/N2L			
Category	Smart Configurator, FSP Smart Configurator			
Description	When sharing a project between different PCs, build errors will occur if the installation folders are different.			

No. 2 Resolved

Limitation	Before pressing the reset button on the board, disconnect the e ² studio connection first.
Target	RZ/T2M, RZ/T2L, RZ/N2L
Category	e ² studio
Description	If the reset button is pressed on the board while connected with e ² studio, debugging will not be able to continue.

No. 3 Resolved

Limitation	An error has occurred because the program download to the NOR flash area has failed. The download is successful on the second connection.				
Target	RZ/T2M, RZ/N2L				
Category	e ² studio				
Description	If the following error is displayed when connecting the debugger or when downloading the program, click the [OK] button to close the dialog and try connecting again.				



No. 4	
-------	--

Limitation	The user program cannot be stopped immediately after the device boot process.
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L
Category	e ² studio
Description	Immediately after the device boot process (boot code), the program cannot be stopped at the beginning of the user program (loader program).
	When debugging, please follows the guide in
	Appendix. How to Debug FSP Project with Flash Boot Mode.

No. 5 Invalid

Limitation	When using e ² studio installer, if checking the multiple check boxes such as "View Release Notes" and so on to show information on browser, the ONLY head item of checked items is shown.				
Target	RZ/T2M, RZ/T2L, RZ/N2L				
Category	e ² studio				
Description	For example, if checking "View Release Notes" check box and other check boxes on the following window, the ONLY "Release Notes" is shown, and the other contents are NOT shown.				
	v202210250746 User: All Users < Back Next > OK Cancel				



Limitation	The Memory Region Usage of ATCM displayed in the Memory Usage window of e ² studio is smaller than the actual size by Memory Region Usage of DUMMY.				
Target	RZ/N2L				
Category	e ² studio				
Description	The Memory Region Usage by the system. The DUMMY NOT include its size. Therefore, please note that the actual size by the Memory R	' is placed in ATCM, howev ne Memory Region Usage of	ver Memory Region	Usage of ATCM do	
		ch 🔋 Memory Usage 🗡	ي الح 🛃		
	Memory Region Usag	e Device Memory Usage			
	Memory Region Usa	ge:		^	
	АТСМ	6 % 33KB used	512KB		
	втсм	28KB used	64KB		
	DUMMY	100 % 8B used	8B		
	SYSTEM_RAM	0 % OB used	2048KB		

No. 6 Resolved in RZ/T2 FSP v2.1.0 for RZ/T2M and RZ/T2L

No. 7

Limitation	When debugging RAM execution without flash memory project with program written to flash memory, erase flash memory before debugging.
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L
Category	e ² studio
Description	If you run an RAM execution without flash memory project with a program written in flash memory, it may be impossible to debug the project. When erasing flash memory, please follow the guide in
	Appendix. How to Erase Flash Memory



Limitation	Applying RZ/T2 FSP v.1.2.0 pack to a project that is already working with RZ/T2M FSP v.1.1.0 causes an error when connecting the debugger.
Target	RZ/T2M
Category	IAR EWARM
Description	An error occurs when connecting to the debugger because the function name of vector table was changed in RZ/T2 FSP v.1.2.0. Change the following command in the "command line options (one per line)" to • (Before change) -drv_vector_table_base=vector_table • (After change) -drv_vector_table_base=_Vector Options for node "Blinky" Runtime Chedding Output Converter Output Converter
	0K Cancel



Limitation	The Device Memory Usage of CPU1 in the Memory Usage window does not work properly.				
Target	RZ/T2M, RZ/T2ME, RZ/T2H				
Category	e ² studio				
Description	The Device Memory Usage in the Memory Usage window, it cannot distinguish between CPU0 and CPU1.				
	When debugging CPU1, the memory area available for CPU1 should be displayed, but the memory area available for CPU0 is incorrectly displayed.				
	iger Console 🔋 Memory Usage 🗡 🚺 Memory 🖉 🤣 🦃 🖓 🖏 🖏 🗖 🗖				
	Memory Region Usage Device Memory Usage				
	• 0x00000000 InternalRam				
	0x0007FFFF 34040/524288 byte(s) (6.49%)				
	> 0x00080000 Non-map area				
	0x000FFFFF				
	► 0x00100000 InternalRam				
	0x0010FFFF 28850/65536 byte(s) (44.02%)				
	> 0x00110000 Non-map area				
	0x0FFFFFF 🗸				

No. 10

Limitation	When adding the CallbackSet function using the Developer Assistance feature, the second argument needs to be changed.		
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L		
Category	e ² studio, Smart Configurator		
Description	<pre>When adding the R_xxx_CallbackSet() function (xxx means any module name) using the Developer Assistance feature, the second argument does not have the correct value. Please replace the second argument with "p_callback". An example of SCI_SPI module, adding CallbackSet() using the Developer Assistance results in the following. status = R_SCI_SPI_CallbackSet(&g_spi0_ctrl, spi_callback_args_t, p_context, p_callback_memory); It needs to replace the second argument with "p_callback, p_context, p_callback_memory); status = R_SCI_SPI_CallbackSet(&g_spi0_ctrl, p_callback, p_context, p_callback_memory);</pre>		



Limitation	In IAR EWARM 9.60.1, an error occurs when starting to debug multiprocessing projects of RAM execution without flash memory.				
Target	RZ/T2M, RZ/T2ME				
Category	IAR EWARM				
Description	When IAR EWARM 9.60.1 is used, there is no defined value required for debugging CPU1 project, and an error occurs when debugging begins. In EWARM 9.60.1, when debugging projects of multiprocessing, it is necessary to add " macro_param cpu1_enable=1" in the " command line options (one per line)" of CPU1 project. Options for node "Blinky_cpu1_seconday" Categoy: Categoy: Categoy: Command line options Cont Couton Bud Linker Budd Actors Budd Actors Budd Actors Simulator CADT CADT CADT State: Analysis Simulator CADT CADT State: Analysis Simulator CADT CADT CADT CADT CADT CADT CADT CADT				
	OK Cancel				

No. 11 Moved to 5.3.3.2 Build for Multiprocessing No. 2-iii

No. 12

Limitation	Build is failed when executed with different install path
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L
Category	FSP Smart Configurator
Description	If install path of FSP SC is different between creating project and executing project, build is failed.
	Please reselect the execution path by following the steps bellow.
	Start EWARM
	• Select "Tools" -> "Configure Tools"
	• Select "New" to create a new tool and specify the following:
	Menu Text: RA Smart Configurator
	Command: Select "Browse" and navigate to rasc.exe in the installed RASC
	Argument:compiler IAR configuration.xml
	Initial Directory: \$PROJ_DIR\$
	Tool Available: Always



	Configure Tools			
	Menu Content:			
	FSP Configurator	Ok		
	The smart configurator	Cancel		
	FSP Configurator RA Smart Configurator fenu I_ext: RA Smart Configuratod C:\Renesas\rzt\sc_v20\^Y-MM_fsp_v2.X> rgument: -compiler IAR configuration.xml nitial Directory: spROJ_DIR\$	New		
		Delete		
	Menu Text			
	RA Smart Configurator			
	Command:			
	C:\Renesas\rzt\sc_v20YY-MM_fsp_v2.X.>	Browse		
	Argument:			
	compiler IAR configuration.xml			
	Initial Directory:			
	\$PR0J_DIR\$			
	Redirect to Output Window			
	Prompt for Command Line			
	Tool Available:			
	Always ~			
• Sele	ct "Tools" -> "RA Smart Co	onfigurator"	" to launch RASC.	

Limitation	Unable to debug CA55 flash boot project with e ² studio.
Target Device	RZ/T2H(CA55)
Category	e ² studio
Description	When debugging with e ² studio, the CA55 Core0 system reset is not performed. Therefore, the program in the external flash is not copied to the internal RAM, and it cannot be operated correctly. Please check the operation with the RAM execution without flash memory project.

No. 14

Limitation	Unable to restart debugging immediately after debugging ends of CA55 RAM execution without flash memory project in e2 studio.	
Target Device	RZ/T2H(CA55)	
Category	e ² studio	
Description	When debugging with e ² studio, the CA55 Core0 system reset is not performed. Therefore, after debugging is complete, if you want to run the debug again, press the reset button (red) on the board.	

No. 15

Limitation	Unable to re-download CA55 binary file.	
Target Device	RZ/T2H(CA55)	
Category	IAR EWARM	
Description	If you download the CA55 binary file, then download it again, the process never finishes. To avoid this issue, you need to erase the flash from the CR52 flash boot project.	



No. 16					
Limitation	Unable to access the upper 32-bit address area in memory view.				
Target Device	RZ/T2H(CA55)				
Category	e ² studio				
Description	 In the Memory view of e² studio, data in the upper 32-bit address area cannot be displayed. When debugging the CA55 project, please make the following settings when accessing the upper 32-bit address. Open Debug Configurations of CA55 project. Select Renesas GDB Hardware Debugging > [project name] Debug_Flat. Select Startup and specify the command in Run Commands. Referring to the following 				
	command, specify the address area you want to access, and you will be able to access the up 32-bit address area. "mem 0x20000000 0x30000000 rw"	pper			
	Create, manage, and run configurations				
	🔯 🖻 🐌 🖹 🖻 🖓 \star 🛛 Name: Blinky Debug. Flat				
	type filter text Image: Application C /C++ Application Image: Application Image: Application Image: Application				
	C Uob Hardware Debuggin (RH) Runtime Options Q Sol Smulator Debuggin (RH) Step program counter at (hex): Q Renesas GDB Hardware Debug; Step trogram counter at (hex): Q Renesas Simulator Debugging (L) Step trogram counter at (hex): C Renesas Simulator Debugging (L) Step trogram counter at (hex):				
	Run Commands mem 0x20000000 0x30000000 nv				
	Keyert Apply				
	⑦ Debug Close				
	Before adding the command Memory × Monitors Monitors				
	After adding the command				
	Ox200000000 Address 0 - 3 4 - 7 8 - B C - F 000000000000 FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF				
	000000200000000 FFFFFFF FFFFFFF FFFFFFFF				
	0000000200000020 FFFFFFF FFFFFFF FFFFFFFF				
	0000000200000030 FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF				
	000000200000040 FFFFFFF FFFFFFFFFFFFFFFF				
	000000200000050 FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF				
	000000200000000 FFFFFFF FFFFFFF FFFFFFFF				



Limitation	Unable to create a CMake project using FSP Smart Configurator.
Target Device	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H
Category	FSP Smart Configurator
Description	Even if you specify CMake as IDE Project Type when creating a project, an error message "No toolchain selected" is displayed, and you cannot proceed to the next screen for project creation.

No. 18

Limitation	The secondary project aborts when debugging multicore with flash boot mode.
Target Device	RZ/T2M, RZ/T2ME, RZ/T2H
Category	IAR EWARM
Description	When performing multicore debugging of a flash boot project on IAR EWARM, after copying an application program from the primary core memory to the secondary one, executing the secondary project will abort.
	For the multicore debugging with flash boot mode in IAR EWARM, follow the steps below:
	1. Run the primary project up to main().
	2. Software reset the secondary project.
	Click on the downward triangle to the right of the reset icon and select Software.
	🗂 🗝 😮 📮 į ETM SWO į 📲
	Disabled (no reset)
	Software
	Hardware
	Custom
	3. Run the secondary project



Limitation	Build errors occur in CA55 projects when install e ² studio as the Current user.
Target Device	RZ/T2H(CA55)
Category	e ² studio
Description	If you install e ² studio as the Current user, an error will occur when building a CA55 project. As a workaround, install e ² studio as All Users.



Appendix. How to Debug FSP Project with Flash Boot Mode

When debugging FSP project with flash boot mode (xSPI boot, NOR flash boot), the program cannot be stopped at the beginning of the user program (loader program).

Please note the following point depending on your IDE (e² studio or IAR EWARM) to debug the user program from its beginning.

Note for multiprocessing projects:

Only the primary project requires the following step. No modification is required for the secondary or subsequent projects.

1. (Both e² studio and IAR EWARM) Insert the loop part in startup_core.c.

When debugging is started, the debugger stops the user program (loader program) about 100ms after the device boot process (boot code). If using e² studio 2022-10 or later, the PC (program counter) is replaced at the entry point (first line in **system_init()** function) after the debugger stops, otherwise, the PC points the address of somewhere in the user program.

When debugging the program immediately after the boot process (boot code), insert the loop part in

• /XXX/fsp/src/bsp/cmsis/Device/RENESAS/Source/YY/startup_core.c (XXX = rzt, rzn, YY = cr, ca)

The detailed position, at which the loop part should be inserted, depends on the IDE(Debugger) and Boot mode.

IDE	Core	Boot Mode	Position at which the loop part should be inserted.
e ² studio	CR52	xSPI boot	Line after static_constructor_init in mpu_cache_init() function.
2022-04 2022-07			<pre>#if BSP_CFG_C_RUNTIME_INIT /* Initialize static constructors */ asm volatile ("static_constructor_init: \n" " ldr r0, =bsp_static_constructor_init \n" " blx r0 \n"); #endif</pre>
			<pre>#if 1 // Software loops are only needed when debugging. asm volatile (" mov r0, #0 \n" " movw r1, #0xf07f \n" " movt r1, #0x2fa \n" "software_loop: \n" " adds r0, #1 \n" " cmp r0, r1 \n" " bne software_loop \n" ::: "memory"); #endif</pre>
		NOR flash boot	First line in system_init() function.
e ² studio 2022-10 or later IAR EWARM	CR52	xSPI boot NOR flash boot	<pre>BSP_TARGET_ARM BSP_ATTRIBUTE_STACKLESS void system_init (void) { #if 1 // Software loops are only needed when debugging. asm volatile (" mov r0, #0</pre>



IDE	Core	Boot Mode	Position at which the loop part should be inserted.
e ² studio	CA55	xSPI boot	First line in system_init() function.
IAR EWARM		NOR flash boot	<pre>BSP_ATTRIBUTE_STACKLESS void system_init (void) { #if 1 // Software loops are only needed when debugging. asm volatile (" mov x0, #0 \n" " movx x1, #0xf07f \n" " movk x1, #0x2fa, LSL #16 \n" " movk x1, #0x2fa, LSL #16 \n" " movk x1, #0x2fa, LSL #16 \n" " software_loop: \n" " adds x0, x0, #1 \n" " adds x0, x0, #1 \n" " b.ne software_loop \n" ::: "memory"); #endif /* Set Vector Base Address Register (VBAR) to point to initializer routine */ asm volatile ("LDR x0, =Vectors \n" "MSR VBAR_EL3, x0 \n" "MSR VBAR_EL1, x0 \n" ::: "memory"); </pre>



Appendix. How to Erase Flash Memory

If you run RAM execution without flash memory project with a program written in flash memory, it may be impossible to debug the project.

Please erase flash memory by following steps depending on your IDE (e² studio or IAR EWARM) before running the project.

- 1. e^2 studio
 - If you would like to erase the flash memory on the board using J-Link Commander, execute the following steps.
 - i) Set the switch for boot mode on RSK to correspond to the area to be erased.
 - ii) Open the J-Link Commander.

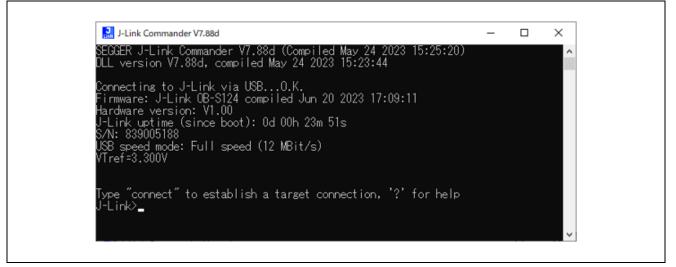


Figure 104 : Launch J-Link Commander

iii) First, type "connect" to establish a target connection and press enter.Next, specify the connection conditions as follows.

• Device> (Device type name)

Table 16 Device Type Name on Renesas Board

Board	Device type name
RSK + RZT2M	R9A07G075M24_CPU0
RSK + RZT2L	R9A07G074M04
RSK + RZT2ME	R9A07G075M29_CPU0
RZT2H Evaluation Board	R9A09G077M44_R52_0
RSK + RZ/N2L	R9A07G084M04

- TIF>S
- Speed> (Default: press enter without inputting any data)



🔜 J-Link Commander V7.88d —
ype "connect" to establish a target connection, '?' for help -Link>connect 'lease specify device / core. <default>: R9A07G075M24_CPU0 ype '?' for selection dialog levice>R9A07G075M24_CPU0 'lease specify target interface: J) JTAG (Default) S) SWD T) cJTAG 'IF>S pecify target interface speed [kHz]. <default>: 4000 kHz peed> levice "R9A07G075M24_CPU0" selected.</default></default>

Figure 105 : Initial Setup for Connecting to the Device

After that, confirm the message "Cortex-R52 identified." is displayed.

J-Link Commander V7.88d	_	×
EL2 support: AArch32 EL3 support: N/A		^
FPU support: Single + Double + Conversion Add. info (CPU temp. halted)		
Current exception level: EL2		
Exception level AArch usage: ELO: AArch32		
EL1: AArch32		
EL2: AArch32 EL3: AArch32		
Non-secure status: Non-secure		
Cache info: Inner cache boundary: none		
LoU Uniprocessor: 1		
LoC: 1 LoU Inner Shareable: 1		
I-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way D-Cache L1: 16 KB, 64 Sets, 64 Bytes/Line, 4-Way		
Memory zones:		
Zone: "Default" Description: Default access mode Zone: "APO" Description: MEM-AP (APB-AP)		
Zone: "AP1" Description: MEM-AP (APB-AP)		
Zone: "AP2" Description: MEM-AP (AXI-AP) Cortex-R52 identified.		
J-Link>_		
		~

Figure 106 : Message of Device Core Identification

- iv) Use the commands below to enable flash erase and erase the flash memory.
 - J-Link>exec EnableEraseAllFlashBanks
 - J-Link>erase (Start address), (Endaddress)



		adiess space to be ese		
Board	Boot mode	External address space to be used	Start address	End address
RSK + RZT2M,	xSPI0 x1	xSPI0 CS0	0x60000000	0x63FFFFFF
RSK + RZT2ME	16-bit bus	CS0	0x70000000	0x71FFFFFF
RSK + RZT2L	xSPI0 x1	xSPI0 CS0	0x60000000	0x63FFFFFF
	xSPI1 x1	xSPI1 CS0	0x68000000	0x68FFFFFF
RZT2H Evaluation Board	xSPI0 x1	xSPI0 CS0	0x40000000	0x47FFFFFF
	xSPI1 x1	xSPI1 CS0	0x50000000	0x57FFFFFF
RSK + RZN2L	xSPI0 x1	xSPI0 CS0	0x6000000	0x63FFFFFF
	16-bit bus	CS0	0x70000000	0x71FFFFFF

Table 17 External Address Space to Be Used in Each Boot Mode

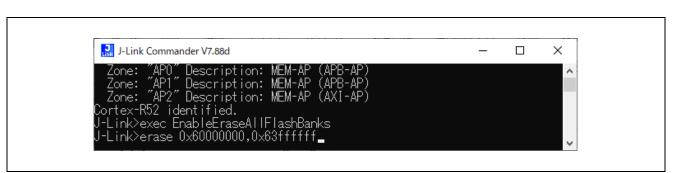


Figure 107 : Specify Erase Range

After that, confirm the message "Erasing done." is displayed.

🔜 J-Link Commander V7.88d 🦳 🗌 🗌
Reset: Halt core immediately after reset using reset catch. Authenticated device detected. Skipping authentication process. OCDREG_STATUS: 0x00000001 Disabled core power domain detected. Enabling debug mode ResetTarget() end - Took 240ms Erasing selected range J-Link: Flash download: Total time needed: 250.391s (Prepare: 0.603s, ompare: 0.000s, Erase: 249.551s, Program: 0.000s, Verify: 0.000s, Rest re: 0.237s) J-Link: Flash download: Flash sectors within Range [0x60000000 - 0x63FFFFFF] deleted. Erasing done.

Figure 108: Message of Flash Memory Erase Complete

v) Enter "q" to exit J-Link Commander.



2. IAR EWARM

If you want to erase the flash memory on the board using IAR EWARM, execute the following steps. If the asymmetric multicore setting is enabled, the erase function cannot be used; it must be disabled.

Disable asymmetric multicore setting:

- a. Click **Project** > **Options...**.
- b. Click **Debugger** > **Multicore** and select **Disable** in Asymmetric multicore.
- i) Set the switch for boot mode on the board to correspond to the area to be erased.
- ii) Open the workspace of a project.

xxx.eww

Blinky.ewd	
🖹 Blinky.ewp	
Blinky.eww	

Figure 109 : Open Workspace for IAR EWARM

iii) Select "Project" -> "Download" -> "Erase memory".

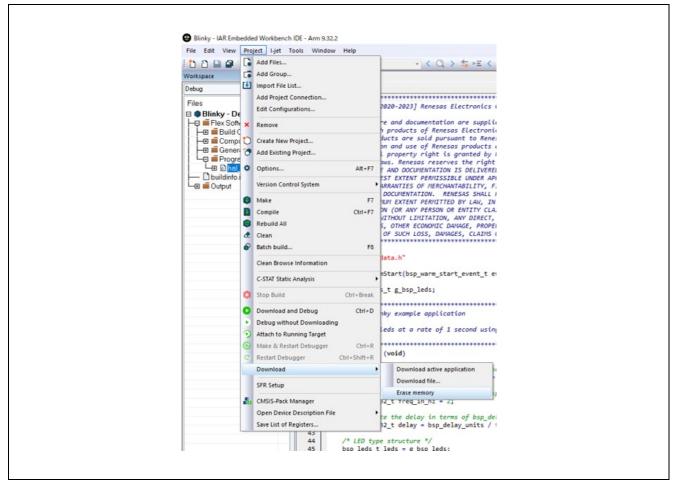


Figure 110 : Select Erase memory Command



iv) Select erase memory space.

1	Erase Memory			×
	Flash loader	Range		
	C:¥Program Files¥IAR Systems¥Embedded Workbench 9.32.2¥arm/config/flashloader/Renesas/FlashRSK_RZT	0x60000000	- 0x63ffffff	
	C: ¥Program Files ¥IAR Systems ¥Embedded Workbench 9.32.2 ¥arm/config/flashloader/Renesas/FlashRSK_RZT	0x70000000	- 0x71fffffff	
	Erase all	Erase	Cancel	

Figure 111 : Select Erase Memory Space

v) After the following dialog appears, erasing of the flash is complete if no error occurs.

Busy	
Starting debugger session: Flashing	
Programming flash memory:	
Cancel	

Figure 112 : Screen During Erasing

lebug Log	→ ₽ >
Log	^
Wed Sep 06, 2023 09:40:04: Target reset	
Wed Sep 06, 2023 09:46;23: Unloaded macro file: C\Program Files\IAR Systems\Embedded Workbench 9.32.2\	
am/config/flashloader/Renesas/FlashRSK_RZT2M_SerialFlash.mac	
Wed Sep 06, 2023 09:46:23: Flash memory has been erased.	
Wed Sep 06, 2023 09:46:23: Unloaded macro file: C\Program Files\VAR Systems\Embedded Workbench 9.32.2\arm/config/debugger/Renesas/RZT2M.dmac	
Wed Sep 06, 2023 09:46:24: IAR Embedded Workbench 9.32.2 (C:\Program Files\IAR Systems\Embedded Workbench 9.32.2\arm\bin\armPROC.dll)	
Wed Sep 06, 2023 09:46:24: Loading the l-jet driver	
	~
Build Debug Log	

Figure 113 : Message of Flash Memory Erase Complete



Appendix. How to Change Boot Mode of FSP Project

When the boot mode of the project is changed, the Pin Configuration needs to be recreated.

It also needs to rename and save the pin configuration to retain the original one before changing the boot mode.

For example, one of specific cases in which re-configure is necessary is when a RAM execution without flash memory project is changed to flash boot mode (xSPI0 x1 boot mode and others).

Please change the boot mode by following steps.

Note for FSP version earlier than v1.3.0:

If the FSP version of your project is earlier than FSP v1.3.0, change it to FSP v1.3.0 before doing the following steps.

- 1. Rename and save the current Pin Configuration in the **Pins** tab.
 - How to rename Pin Configuration: Click "Manage Configurations..."

Pin Configuration	Generate F	Project Content
Select Pin Configuration	🔛 Export to CSV file 🛛 Configure Pin Dr	Priver Warnings
✓ Generate data: g_bt ✓ Manage Pin Configurations Pin Selection Image Pin Configuration ✓ Multiple Pin Configuration Multiple Pin Configuration Modify pin configuration Image Pin Configuration Multiple Pin Configuration Modify pin configuration Image Pin Configuration Multiple Pin Configuration Modify pin configuration Image Pin Configuration Multiple Pin Configuration Image Pin Configuration <th>Management or import/export external file</th> <th>ycle Pin Group</th>	Management or import/export external file	ycle Pin Group
> ✓ P03 > ✓ P04 > ✓ P05 > ✓ P05 > ✓ P07 > ✓ P08 > ✓ P09 < <u>端子機能</u> 隨子書号 Summary BSP Clocks Pin		>

Figure 114: How to Rename Pin Configuration



2. Change the boot mode in the **BSP** tab. (The board must be the same as before the change.)

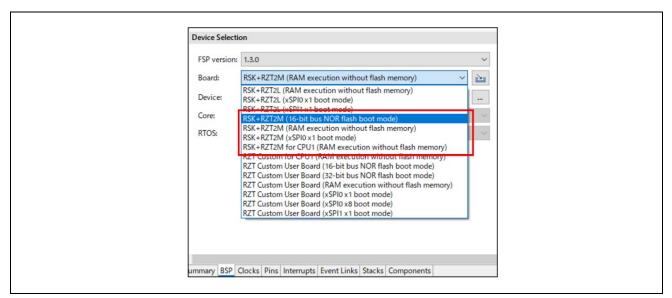


Figure 115: Change the Boot Mode in the BSP Tab

3. Reselect "FSP Version" from the drop-down list.

(This operation is necessary even if there is only one version in the list.)

Device Selecti	on
FSP version:	1.3.0 ×
poard.	
Device:	R9A07G075M24GBG
Core:	CR52_0 ~
RTOS:	No RTOS ~

Figure 116: Reselect "FSP Version" from the Drop-down List



4. Uncheck "Generate data" in the **Pins** tab.

# *[sample] FSP Configuration ×	
Pin Configuration	
Select Pin Configuration	
RSK+RZT2M_OLD Manage	
Generate data: g_bsp_pin_cfg	

Figure 117: Uncheck Generate data in the Pins Tab

5. Select the regenerated configuration for the board.

Pin Configuration Select Pin Configuration RSK+RZT2M_OLD R	∰ *[sample] FSP Configuration ×	
RSK+RZT2M_OLD V Manage RSK+RZT2M_OLD	Pin Configuration	
RSK+RZT2M_OLD	Select Pin Configuration	
_	RSK+RZT2M_OLD V Man	anage
	RSK+RZT2M_OLD R9A07G075M24GBG.pincfg	

Figure 118: Select the Regenerated Configuration for the Board

6. Check "Generate data" again and enter "g_bsp_pin_cfg" as the name.

፨ *[sample] FSP Configuration ×
Pin Configuration
Select Pin Configuration
RSK+RZT2M V Man
Generate data: g_bsp_pin_cfg

Figure 119: Check Generate data again



Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e² studio

4 Tutorial: Your First RZ/T2, RZ/N2 MPU Project – Blinky describes how to create and debug a project in e^2 studio for RAM execution of a single core process using CR52_0 and multiprocessing processes where CR52_0 is the primary core. This chapter describes project creation and debugging methods in e^2 studio applicable to other boot modes and core combinations.

If the procedure is preceded by (XXX), it is executed only if the condition is met.

(RAM exec): The boot mode used in the project is RAM execution without flash memory.

(Flash boot): The boot mode used in the project is NOR flash boot mode or xSPI flash boot mode.

(CR52): The core used in the project is CR52.

(CA55): The core used in the project is CA55.

(Homogeneous system): In multiprocessing, only ONE type of core is used. (e.g., only CR52, only CA55) (Heterogeneous system): In multiprocessing, both CR52 and CA55 cores are used in combination.

- 1. Create a primary project according to the procedures in 4.3 Create a New Project for Blinky.
 - (CA55) To blink the LED on the T2H CA55 Core1, you need to change the pin configuration of the primary project in the FSP Smart Configurator as follows.
 - i. After creating the primary project in 4.3 Create a New Project for Blinky No. 11, set the pins before clicking **Generate Project Content**.
 - ii. In Pins tab of FSP configuration, click **Pin Selection** -> **Peripherals** -> **Connectivity:SDHI** -> **SDHI1**
 - iii. Change the value of SD1_PWEN to None.
 - iv. In Pins tab of FSP configuration, click **Pin Selection** -> **Ports** -> **P08** -> **P08**_5.
 - v. Change the value of Symbolic Name to LED3.
 - vi. Change the value of Mode to Output mode (Low & Not Into Input)
- 2. (Flash boot) Insert the loop part in startup_core.c of the primary project with reference to Appendix. How to Debug FSP Project with Flash Boot Mode.
- 3. Build the primary project according to the procedures in 4.4.1 Build.
- 4. Create a secondary project using the bundle file (.sbd) of the primary project according to the procedures in 4.3 Create a New Project for Blinky.
- 5. (RAM exec) Build the secondary project according to the procedures in 4.4.1 Build.



- 6. (Flash boot) (Homogeneous system) Build the secondary project according to the procedures in 4.4.2 Build for Multiprocessing No. 2. The following additional properties must be set.
 - i. In the Properties window, click C/C++ Build > Settings > Build Steps.
 - ii. Add **Command(s)** at **Post-build steps**.
 - (CR52)

arm-none-eabi-objcopy -I elf32-littlearm -O binary \${ProjName}.elf secondary.bin && arm-noneeabi-objcopy -I binary -O elf32-littlearm -B arm --renamesection .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o

• (CA55)

aarch64-none-elf-objcopy -I elf64-littleaarch64 -O binary \${ProjName}.elf secondary.bin && aarch64-none-elf-objcopy -I binary -O elf64-littleaarch64 -B aarch64 --renamesection .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o

Properties for Blinky_cpu1_se	condary		
type filter text	Settings		⇔ - ⇔ * 8
 > Resource Builders C/C++ Build Build Variables Environment Logging Settingsi Tool Chain Editor > C/C++ General Project References Rencess QE Run/Debug Settings Task Tags > Validation 	Configuration: Debug [Active]		~
		Restore <u>D</u> efaults	дрру
?		Apply and Close	Cancel

Figure 120 e² studio Build Setting for the Secondary Project (Flash Boot) (Homogeneous System)

7. (Flash boot) (Heterogeneous system) Create an object file and set it to the secondary project.

- Build the secondary project according to the procedures in 4.4.2 Build for Multiprocessing No. 2. The following additional properties must be set.
 - a. In the Properties window of the secondary project, click C/C++ Build > Settings > Build Steps.
 - b. Add Command(s) at Post-build steps.
 - (The primary project uses CR52, and the secondary project uses CA55)
 - aarch64-none-elf-objcopy -I elf64-littleaarch64 -O binary \${ProjName}.elf secondary.bin
 (The primary project uses CA55, and the secondary project uses CR52)
 - arm-none-eabi-objcopy -I elf32-littlearm -O binary \${ProjName}.elf secondary.bin
- ii. Since secondary.bin is output to the Debug folder of the secondary project, move it to the Debug folder of the primary project.

(Continued on next page)

i.



- iii. Build the primary project with the following additional properties.
 - a. In the Properties window of the primary project, click C/C++ Build > Settings > Build Steps.
 - b. Add Command(s) at Post-build steps.
 - (The primary project uses CR52, and the secondary project uses CA55) arm-none-eabi-objcopy -I binary -O elf32-littlearm -B arm --renamesection .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o
 - (The primary project uses CA55, and the secondary project uses CR52) aarch64-none-elf-objcopy -I binary -O elf64-littleaarch64 -B aarch64 --rename-
- section .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o
 iv. Since secondary.o is output to the Debug folder of the primary project, move it to the Debug folder of the secondary project.
- 8. (Flash boot) Set the following additional properties to the primary project.
 - i. In the Properties window of the primary project, click C/C++ Build > Settings > Tool Settings > Cross ARM C Linker > Miscellaneous.
 - ii. Set a file pass of secondary.o in the secondary project to **Other objects**. e.g. \${workspace_loc:/Blinky_secondary/Debug/secondary.o}

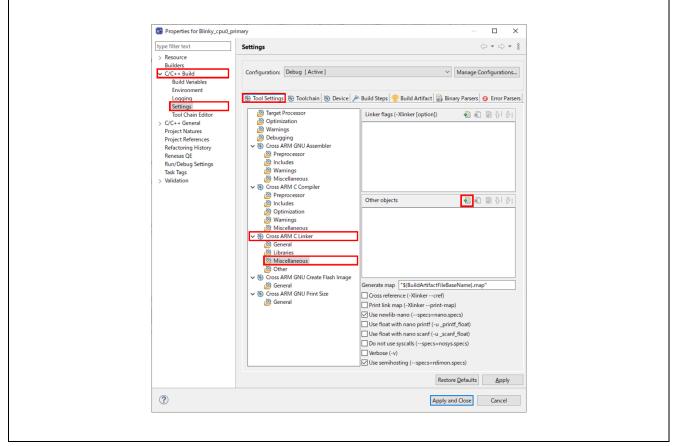


Figure 121 e² studio Build Setting for the Primary Project (Part 1)



🚱 Add file path	×
File: \${workspace_loc:/Blinky_cpu1_secondary/Debug/secondary.o}	
OK Cancel Workspace	File system

Figure 122 e² studio Build Setting for the Primary Project (Part 2)

- 9. Build the primary project.
- 10. Debug the projects according to the procedures in 4.7 Debug and Run for Multiprocessing.
 - (CR52 CPU0) When using the CR52 CPU0 core for the secondary or later project, set the additional debug connection settings:
 - Debugger > Connection Settings > Connection
 - Reset at the beginning of connection : No

Debug Configurations			
Create, manage, and run configu	ations		Ť
	Name: Secondary Debug_Flat		
type filter text	Main Statup Startup Source C	ammon	
C C/C++ Application C C/C++ Remote Application EASE Script C GDB Hardware Debugging	Debug hardware: J-Link ARM V Target Device: GDB Settings Connection Settings Debug Tool Set	R9A09G077M44_CR5	
C GDB OpenOCD Debugging	Connection Method	IP via LAN	~ ^
GDB Simulator Debugging (I	Host Name/IP Address[:port number]		
🖅 Java Applet	Identifier		
Java Application	Tunnel Server		
R Launch Group	Port Number		
Remote Java Application	Password		
✓ C Renesas GDB Hardware Deb	✓ Interface		
Primary Debug_Flat	Туре	SWD	~
C Secondary Debug_Flat	Speed (kHz)	4000	~
C Renesas Simulator Debuggir	✓ JTAG Scan Chain		
	Multiple Devices	No	~
	IRPre	0	
	DRPre	0	
	✓ Connection		
	Register initialization	No	~
	Reset at the beginning of connection Reset at the end of connection	No	~
	Reset before download	No	č
	Reset after download	No	~
	ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
	Hold reset during connect	No	~
	Set CPSR(Sbit) after download	Yes	
	Prevent Releasing the Reset of the CM3 Core	Yes	
	Secure Vector Address		
	Non-secure Vector Address		
	Hot Plug	No	~

Figure 123 e² studio Debug Configuration for the Secondary or later Project (CR52 CPU0)

- (Flash boot) The flash boot mode differs from RAM execution without flash memory in two points.
 - Both the primary and secondary project binaries are downloaded to the device when connecting debugger with the primary project.
 - Change Debug Configuration settings in the No. 3 procedure of 4.5.2 Debug Steps.
 - **Debugger** > **Connection Settings** > **Connection**
 - (Primary project) Reset after download: Yes
 - (Secondary project) Reset after download: No (default)
 - Set CPSR(5bit) after download: No (default)



- 11. When changing the project and debugging it again, follow these steps.
 - i. Build the primary project (No. 2).
 - ii. (RAM exec) Build the secondary project (No. 4).
 - iii. (Flash boot) (Homogeneous system) Build the secondary project (No. 5).
 - iv. (Flash boot) (Heterogeneous system) Create an object file and set it to the secondary project (No. 6).
 - v. Build the primary project again (No. 8).
 - vi. Debug the projects (No. 9).



Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM

5.3 Using Smart Configurator with IAR EWARM describes how to create in FSP SC and debug a project in IAR EWARM for RAM execution of a single core process using CR52_0 and multiprocessing processes where CR52_0 is the primary core. This chapter describes project creation in FSP SC and debugging methods in IAR EWARM applicable to other boot modes and core combinations.

If the procedure is preceded by (XXX), it is executed only if the condition is met.

(RAM exec): The boot mode used in the project is RAM execution without flash memory.

(Flash boot): The boot mode used in the project is NOR flash boot mode or xSPI flash boot mode.

(CR52): The core used in the project is CR52.

(CA55): The core used in the project is CA55.

- 1. Create projects according to the procedures in 5.3.2 Create a New Project.
 - (CA55) To blink the LED on the T2H CA55 Core1, you need to change the pin configuration of the primary project in the FSP Smart Configurator as follows.
 - i. In 5.3.2 Create a New Project No. 14, set the pins before clicking Generate Project Content.
 - ii. In Pins tab of FSP configuration, click **Pin Selection** -> **Peripherals** -> **Connectivity:SDHI** -> **SDHI1**
 - iii. Change the value of SD1_PWEN to None.
 - iv. In Pins tab of FSP configuration, click Pin Selection -> Ports -> P08 -> P08_5.
 - v. Change the value of Symbolic Name to LED3.
 - vi. Change the value of Mode to Output mode (Low & Not Into Input)
- 2. (Flash boot) Insert the loop part in startup_core.c of the primary project with reference to Appendix. How to Debug FSP Project with Flash Boot Mode.
- 3. Build the primary project according to the procedures in 5.3.3.2 Build for Multiprocessing No. 1.
- 4. Create a secondary project using the bundle file (.sbd) of the primary project according to the procedures in 5.3.2 Create a New Project.
- 5. (RAM exec) Build the secondary project according to the procedures in 5.3.3.2 Build for Multiprocessing No. 2.
- 6. (Flash boot) Build the secondary project according to the procedures in 5.3.3.2 Build for Multiprocessing No. 2. The following additional properties must be set.
 - i. Click **Project** > **Options...**.
 - ii. Click **Output Converter > Output** and set **Raw binary** to **Output format**.



Options for node "Blinky_no Categosy: General Options A Static Analysis Runtime Checking C/C++ Compiler Assembler Output, Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G+LINK	Cutput Cutput Qutput Qutput Qutput Qutput file Qutput file Durride default Blinky_nor_cpu1_secondary.bin	×
I-jet J-Link(J-Trace TI Stellaris Nu-Link PE micro	OK Cancel	

Figure 124 IAR EWARM project options for the secondary project in flash boot mode

- 7. (Flash boot) Set the following additional options to the primary project.
 - i. Click **Project** > **Options...**.

•

- ii. Click Linker > Input and set it as follows:
 - Keep symbols: (one per line)
 - SECONDARY
 - Raw binary image
 - File: the path of binary file in the secondary project. e.g. \$PROJ_DIR\$\..\Blinky_secondary\Debug\Exe\Blinky_secondary.bin
 - Symbol: SECONDARY
 - Section: SECONDARY
 - Align: 8



Options for node "Primary"	×
Category: General Options	Factory Settings
Static Analysis Runtime Checking C/C++ Compiler	#define Diagnostics Checksum Encodings Extra Options Config Library Input Optimizations Advanced Output List
Assembler Output Converter Custom Build	Keep symbols: (one per line)
Linker Build Actions Debugger	
Simulator CADI CMSIS DAP	
E2/E2 Lite GDB Server G+LINK	~
I-jet J-Link/J-Trace TI Stellaris Nu-Link	Baw binary image File: Symbol: \$PROJ_DIR\$#,#Secondary*Debug3 ONDARY ONDARY 8
PE micro v	File: Symbol: Section: Align:
	OK Cancel

Figure 125 IAR EWARM project options for the primary project in flash boot mode

8. Build the primary project according to the procedures in 5.3.3.2 Build for Multiprocessing No. 3.



9. (Flash boot) In the primary project, click **Project** > **Download** > **Download file...** and select out file of the primary project.

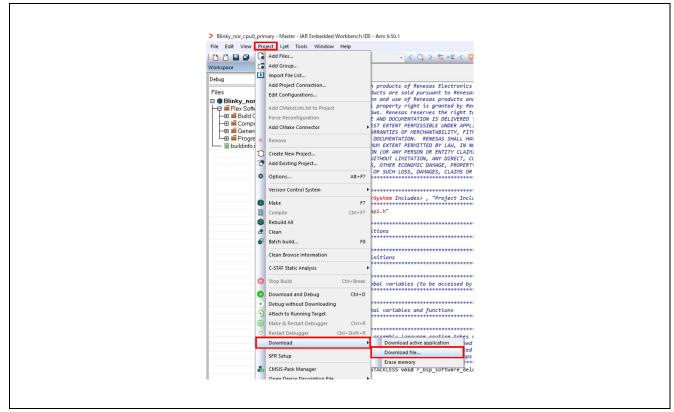


Figure 126 IAR EWARM Download File

- 10. (RAM exec) Debug the projects according to the procedures in 5.3.5 Debug for Multiprocessing.
- 11. (Flash boot) Enable settings for multicore debugging in 5.3.5 Debug for Multiprocessing No. 2.



12. (Flash boot) Click **Project** > **Debug without Downloading** of the primary project to debug. Debug the projects according to the procedures in 5.3.5 Debug for Multiprocessing No. 4 and after.

> Blinky_nor_cpu0	primary - Master - IAR Embedded Workbench IDE - Arm 9.50.1
File Edit View	Project Ljet Tools Window Help
1 h h 🖻 🔍	▲ Add Files < Q. > な ► =
Workspace	Add Group
	Import File List
Debug	Add Project Connection n products of Renesas Elect
Files	Edit Configurations ducts are sold pursuant to
🗉 🌑 Blinky_no	on and use of Renesas produ
- 🖓 🗰 Flex Soft	
Here Build 🛛	Force Reconfiguration E AND DOCUMENTATION IS DELI
— ⊕ ■ ⊂ Comp	Add CMake Connector
🖵 🗎 buildinfo.	NUM EXTENT PERMITTED BY LAW
	Create New Project DN (OR ANY PERSON OR ENTITY
	Add Existing Project S, OTHER ECONOMIC DAMAGE, P
	OF SUCH LOSS, DAMAGES, CLA
	Options Alt+F7
	Version Control System
	Make F7
	Rebuild All
	d Clean itions
	Batch build F8
	Clean Browse Information
	C-STAT Static Analysis
	Stop Build Ctri+Break obal variables (to be acces

	period without bownoading pal variables and functions
	Attach to Running larget
	Make & Restart Debugger Ctrl+R
	C Restart Debugger Ctrl+Shift+R assembly Language routine
	Download I when the Loop exits. The
	baue/epiloaue seauences aen

Figure 127 IAR EWARM Debug without Downloading

- 13. When changing the project and debugging it again, follow these steps.
 - i. (Flash boot) Disable asymmetric multicore setting.
 - a. Click **Project** > **Options...**.
 - b. Click **Debugger** > **Multicore** and select **Disable** in Asymmetric multicore.
 - ii. Build the primary project (No. 2).
 - iii. (RAM exec) Build the secondary project (No. 5).
 - iv. (Flash boot) Build the secondary project (No. 6).
 - v. Build the primary project again (No. 8).
 - vi. (Flash boot) Download the file (No. 9).
 - vii. (RAM exec) Debug the projects (No. 10).
 - viii. (Flash boot) Enable asymmetric multicore setting (No. 11).
 - ix. (Flash boot) Debug the projects (No. 12).



Revision History

		Description	
Rev.	Date	Page	Summary
1.00	Jun.7.22	-	First Edition issued
1.01	Aug.9.22	-	Added the RZ/N2L device as target device.
		All	Unified some terminologies.
		p.9	Updated "SEGGER J-Link" section.
			• Added the software environment on which FSP projects are verified.
		p.13	Added the "2.5.1 RSK+RZN2L" section.
		p.22	Updated "e ² studio Prerequisites"
			• Updated the Windows PC requirements.
		p.43	Update "Prerequisites" section
			• Added the note regarding the patch for debugging RZ/N2L FSP project on EWARM.
		p.44	Updated "Create a New Project" section.
			• Added installation path of FSP SC.
			• Added some steps for creating a EWARM project.
			• Added Note subsection for debugging RZ/N2L EWARM project.
		p.68	Updated "Selecting a Board and Toolchain" section.
			• Added the detailed explanation how to select Board and Device for creating a FSP project.
		p.82	Updated "Appendix. Known Issues" chapter.
		p.91	Updated "Appendix. Tool Software Limitations" section.
		-	Added "Appendix. How to update J-Link DLL files in e ² studio" chapter.
		p.110	Added "Appendix. How to Debug FSP Project with Flash Boot Mode"
1.02	Oct.31.22	-	Updated documentation for RZ/T2M FSP v1.1.0.
			• Removed contents for RZ/T2M FSP v1.0.0
		All	Updated minor issues.
			• Fixed minor typo.
			Adjusted page breaks.
		p.9	Updated "2.3.1 SEGGER J-Link" section.
			• Updated the FSP version and J-Link version for RZ/T2M
			• Added the notification that J-Link OB S124 requires the firmware update to debug RZ/T2M FSP project.
			• Added the link to Renesas Knowledge Base which explains how to update J-Link DLL in e ² studio.
		p.45	Added "5.3.2.2 NOTE: Configure IAR EWARM Project [RZ/T2M, RZ/T2L]" section.
		p.82	Updated "Appendix. Known Issues" chapter.
			Remove some limitations regarding RZ/T2M



		Descriptio	on	
Rev.	Date	Page Summary		
		p.91	Updated "Appendix. Tool Software Limitations" section.	
			• Added new limitation of e ² studio regarding J-Link OB S124 firmware	
			version.	
			Added the link to explain how to update J-Link DLL.	
		-	Removed "Appendix. How to update J-Link DLL files in e ² studio" chapter.	
1.03	Dec.23.22	-	Updated documentation for RZ/N2L FSP v1.1.0.	
			Removed contents for RZ/N2L FSP v1.0.0	
		All	Updated minor issues.	
			Fixed minor typo.	
		p.69	Updated "Appendix. Known Issues" chapter.	
			Removed some limitations regarding RZ/N2L	
		p.73	Updated "Appendix. Tool Software Limitations" section.	
			Removed some limitations regarding RZ/N2L	
		p.75	Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" chapter.	
			• Added new procedure for RZ/N2L FSP v1.1.0.	
1.04	Mar.23.23	All	Updated documentation for RZ/T2 FSP v1.2.0.	
			• Removed contents for RZ/T2M FSP v1.1.0.	
			• Added contents for RZ/T2L.	
		p.1	Added video contents website link.	
		p.9	Updated "2.3.1 SEGGER J-Link" section.	
			• Updated the FSP version and J-Link version for RZ/T2M and RZ/T2L.	
			• Removed the notification that J-Link OB S124 requires the firmware update to debug RZ/T2M FSP project.	
		p.10-12	Updated "2.4.1 RSK+RZT2M" section.	
			• Added explanation that other boot mode board settings refer to the RSK User's Manual.	
			• Modified figure names and added a table title.	
		p.13-15	Added "2.4.2 RSK+RZT2L" section	
		p.16-18	Updated "2.5.1 RSK+RZN2L" section.	
			Corrected board name.	
			• Added explanation that other boot mode board settings refer to the RSK User's Manual.	
			• Modified figure names and added a table title.	
		p.23	Updated "4.3 Create a New Project for Blinky" section.	
			Added RSK+RZT2L Board Setting.	
		p.26	Updated "4.3.5 Where is main()?" section.	
			Corrected product group name.	



		Description			
Rev.	Date	Page Summary			
		p.32	Updated "4.5.4 Change CPSR Register Value" section.		
		-	• Revised description.		
			• Added description of how to automatically change CPSR register value.		
		p.33	Updated "4.6 Run the Blinky Project" section.		
		-	• Removed LEDs working in CPU1 project.		
		p.34	Updated "5.3.1 Prerequisites" section.		
			• Added note for RZ/T2L patch file.		
		p.37	Updated "5.3.2 Create a New Project" section.		
		1	Added RSK+RZT2L Board Setting.		
		p.50	Updated "5.3.4 Download & Debug the Project" section.		
		1	Removed LEDs working in CPU1 project.		
		p.69	Updated "6.5 Adding and Configuring HAL Drivers" section.		
		1	• Added a table title.		
		p.71-73	Updated "Appendix. Known Issues" section.		
		1	• Added a table "List of Known Issues"		
			• Numbered each issue.		
			• Removed issue of adding "r_dsmif" alone		
			• Updated issue contents that the BSP properties are sometimes configured		
			to incorrect configuration		
			Removed Ethernet SELECTOR issue.		
		p.74-77	Updated "Appendix. Tool Software Limitations" section.		
			• Added a table "List of Tool Software Limitations"		
			• Numbered each limitation.		
			• Added new limitation of applying RZ/T2 FSP v.1.2.0 pack.		
		p.78	Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section		
			• 1. (Both e ² studio and EWARM) Insert the loop part in startup.c.		
			Added e^2 studio 2023-01 to the table.		
			• 3. (e2 studio ONLY) Apply a macro file for RZ/N2L FSP v1.1.0 xSPI0 x1 boot mode.		
			• Added direct download URL of RZ/N2L patch file.		
1.05	Jun.30.23	All	Updated documentation for RZ/N2L FSP v1.2.0.		
			• Removed contents for RZ/N2L FSP v1.1.0		
		p.9	Updated "2.3.1 SEGGER J-Link" section.		
			• Updated the FSP version and e ² studio version for RZ/N2L.		
		p.30	Updated "4.5.2 Debug Steps" section.		
			• Added description of how to automatically change CPSR register value for RZ/N2L and e ² studio 2023-04.		
		p.33	Updated "4.5.4 NOTE: Change CPSR Register Value [RZ/T2M, RZ/T2L]" section.		
			Changed section title to limit the target device		



	I		Description		
Rev.	Date	Page	Summary		
		p.35	Updated "5.3.1 Prerequisites" section.		
			Removed EWARM Patch for RZ/N2L		
		p.72-77	 Updated "Appendix. Known Issues" chapter. Updated table "List of Known Issues" to add new issues and add N2L as target device for No.2 Added new issue related to RSD configuration when changing board 		
			 Added new issue related to BSP configuration when changing board setting. Added new issue related to FSP module FreeRTOS issue. 		
		p.78	 Updated "Appendix. Tool Software Limitations" chapter. Removed some limitations regarding breakpoint 		
		p.82	 Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" Updated IDE version in the table for including e² studio 2023-04 		
1.06	Sep.8.23	All	 Updated documentation for RZ/T2 FSP v1.3.0. Removed contents for RZ/T2 FSP v1.2.0 Changed GNU ARM Embedded Toolchain to version 12.2.1.arm-12-24. 		
		p.6	 Updated "1.3.2 FSP Documentation" section. Added note for RZ/N2L FSP documentation. 		
		p.26	Updated "4.3.6 Blinky Example Code" section.Changed the processing of blinky template code.		
		p.28	 Updated "4.5.2 Debug Steps" section. Added reset setting of debug configuration for RAM execution without flash memory. 		
		p.43	Removed "5.3.2.2 NOTE: Configure IAR EWARM Project [RZ/T2M, RZ/T2L]" section.		
		p.44	Updated "5.3.4 Download & Debug the Project" section.Changed the processing of blinky template code.		
		p.68-73	 Updated "Appendix. Known Issues" section. Updated table "List of Known Issues" to add new issues. Added new issues related to Pins configuration. Added new issue of warning message when building "r_gmac" with gcc compiler. 		
		p.75	 Updated "Appendix. Tool Software Limitations" chapter. Added "Smart Configurator" section. Added new limitation of displaying memory region usage 		
		p.80	 Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section. Removed limitation related to reset when using e² studio 		
		p.82-86	Added "Appendix. How to Erase Flash Memory" section.		
1.07	Sep.29.23	All	 Updated documentation for RZ/N2L FSP v1.3.0. Removed contents for RZ/N2L FSP v1.2.0 		



	Descripti		on
Rev.	Date	Page	Summary
		p.9	Updated "2.3.1 SEGGER J-Link" section.
			Updated the FSP version and e ² studio version for RZ/N2L.
		p.80	Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section.
			• Change the number of items.
			• Removed limitation related to RZ/N2 FSP v1.2.0 and J-ink V7.80b only
1.08	Jan.22.24	p.6	Corrected document numbers of RSK+RZ/T2L and RSK+RZ/N2L User's Manual
		p.68-75	Updated "Appendix. Known Issues" section.
			• Moved the position of FSP Configurations and FSP Modules descriptions to the beginning of the chapter.
			Added column "Category" to the List
			• Removed category headings (FSP Configurations, Stacks Configuration, FSP Module, BSP Configuration,)
			• Added item "Category" to description of each Known Issues.
			• Grayed out items where issues have been resolved.
			• Added description to workaround of No. 3.
			• Added RZ/T2L as target device to No. 5
			• Added RZ/T2M and RZ/T2L as target device to No. 6
			Corrected instructions in the code of No. 14.
		p.76-80	Updated "Appendix. Tool Software Limitations" chapter.
			Added column "Category" to the List
			Removed category headings (Smart Configurator, FSP Smart Configurator, e ² studio,)
			• Added item "Category" to description of each Tool Software Limitations.
			Grayed out items where limitations have been resolved.
		p.87-89	Added "Appendix. How to Change Boot Mode of FSP Project" section.
1.09	Mar.29.24	All	Updated documentation for RZ/T2 FSP v2.0.0.
			• Removed contents for RZ/T2 FSP v1.3.0
		p.1	List Target Device separately for each series.
		p.6	Updated "1.3 Related Documentation Files" section.
			List Target Device separately for each series.
		p.9	Updated "2.3.1 SEGGER J-Link" section.
			List Target Device separately for each series in a table.
		p.10-18	Updated "2.4 RZ/T Series Board Setup" section and added "2.5 RZ/N Series Board Setup" section.
			• Each series was divided into separate explanatory chapters.
			Delete unnecessary figure descriptions
		p.20	Updated "4.1 Tutorial Blinky" section.
			Added description of a multiprocessing.
		p.21	Updated "4.3 Create a New Project for Blinky" section.
			Added description of a multiprocessing.
		p.26	Updated "4.3.6 Blinky Example Code" section.
			• Updated Blinky code.



	Description		ion	
Rev.	Date	Page	Summary	
		p.27	Updated "4.4 Build the Blinky Project" section.	
			• Added description of a multiprocessing.	
		p.28	Updated "4.5.2 Debug Steps" section.	
			Added description of a multiprocessing.	
		p.31	Updated "4.6 Run the Blinky Project" section.	
			• Added LED2-3 of RSK+RZ/T2M for CPU1 core.	
			• Added descriptions to suspend program execution and exit debug mode.	
		p.32	Added "4.7 Debug and Run for Multiprocessing" section.	
		p.33	Added "4.8 Import the Project" section.	
		p.37	Updated "5.2 Tutorial Blinky" section.	
			Added description of a multiprocessing.	
		p.38	Updated "5.3.2 Create a New Project" section.	
			Added description of a multiprocessing.	
		p.43	Added "5.3.2.1 NOTE: Configure IAR EWARM Project [Only RZ/N2L]" section.	
		p.43	Updated "5.3.3 Build the Project" section.	
			• Moved text to "5.3.3.2 Build" section.	
		p.43	Added "5.3.3.1 NOTE: Build settings [Only Multiprocessing]" section.	
		p.47	Added "5.3.3.2 Build" section.	
		p.48	Updated "5.3.4 Download & Debug the Project" section.	
			• Added description of a multiprocessing.	
			Added LED2-3 of RSK+RZ/T2M for CPU1 core.	
		p.52	Added "5.3.5 Debug for Multiprocessing" section.	
		p.53	Added "5.5 Note when debugging in different workspaces" section.	
		p.72	Updated "Appendix. Known Issues" chapter.	
			• Resolved issues No. 2, No. 6, No. 8, No. 9, No. 10 and No. 11.	
			• Removed RZ/T2M and RZ/T2L as target device from No. 12	
			• Added new issue No. 13, No. 14, and No. 15.	
		p.89	Updated "Appendix. Tool Software Limitations" chapter.	
			Added new limitation No. 9 and No. 10.	
		P.105	Added "Appendix. How to Debug FSP multiprocessing projects with Flash Boot Mode" chapter.	
1.10	May.30.24	All	Updated documentation for RZ/N2L FSP v2.0.0.	
			• Removed contents for RZ/N2L FSP v1.3.0	
		p.9	Updated "2.3.1 SEGGER J-Link" section.	
			• List Target Device separately for each series in a table	
		p.44	Removed "5.3.2.1 NOTE: Configure IAR EWARM Project [Only RZ/N2L]" section.	



	Descrip		ription	
Rev.	Date	Page	Summary	
		p.44-46	Updated "5.3.3.2 Build for Multiprocessing" section.	
			• Changed the program execution start position setting.	
			Corrected reset setting.	
			Added images for the settings screen.	
		P.73	Updated "Appendix. Known Issues" chapter.	
			Resolved issue No. 12	
			• Added RZ/N2L as target device from No.13 and No.14	
			• Added new issues No. 16 and No. 17.	
		p.84	Updated "Appendix. Tool Software Limitations" chapter.	
			• Added RZ/N2L as target device from No. 10.	
		p.97	Updated "Appendix. How to Change Boot Mode of FSP Project" chapter.	
			Added new step 4	
		p.99	Updated "Appendix. How to Debug FSP multiprocessing projects with Flash	
			Boot Mode" section.	
			• Changed the program execution start position setting.	
			Modified explanation of debugging sequence.	
1.11	Jun.28.24	All	Updated documentation for RZ/T2 FSP v2.1.0.	
			• Removed contents for RZ/T2 FSP v2.0.0	
		All	Added the RZ/T2ME device as target device.	
		p.1	Added video links of FSP Configuration.	
		p.6	Updated "1.3.2 FSP Documentation" section.	
			• Added explanation of notes when using FSP software modules.	
		p.10	Updated "2.4.1.1 Boot Mode" section.	
			Added note for board setting.	
		p.13	Updated "2.4.2.1 Boot Mode" section.	
			Added note for board setting.	
		p.15	Added "2.4.3 RSK+RZT2ME" section.	
		p.16	Updated "2.5.1.1 Boot Mode" section.	
		1	Added note for board setting.	
		p.38	Updated "5.3.2 Create a New Project" section.	
			• Added IDE Project Type setting for newer versions of FSP SC.	
		p.43	Updated "5.3.3.2 Build for Multiprocessing" section.	
		*	 Removed Build Actions setting. 	
			Added Make before debugging setting.	
		p.54	Updated "6 FSP Configuration Users Guide" section.	
			• Updated figures with new tool screens.	
		P.74	Updated "Appendix. Known Issues" chapter.	
			 Resolved issues No. 16 and No. 17 	
			• Added new issues No. 18 to No. 24.	
		p.85	Updated "Appendix. Tool Software Limitations" chapter.	
			• Removed RZ/T2M and RZ/T2L as target device from No. 6.	
			Added new issue No. 11.	



		Descript	ion
Rev.	Date	Page	Summary
		p.102	Updated "Appendix. How to Change Boot Mode of FSP Project" section.
			Added note on FSP version changes
1.12	Nov.26.24	All	Updated documentation for RZ/T2 FSP v2.2.0.
			• Removed contents for RZ/T2 FSP v2.1.0
		All	Added the RZ/T2H as target device and CA55 core support.
		6	Updated "1.3.2 FSP Documentation" section.
			• Removed Note for RZ/N2L about the documentation issue.
		8	Updated "2 Set up Evaluation Board" chapter.
			Changed the boards designations.
		9	Updated "2.3.1 SEGGER J-Link" section.
			Added how to update J-Link firmware.
		16	Added "2.4.4 RZ/T2H Evaluation Board" section.
		24	Updated "4.3 Create a New Project for Blinky" section.
			 Modified to a generic description that does not specify which cores are used in multiprocessing.
			 Updated tool screen images.
			 Added tables describing the settings for each project.
		31	Added "4.4.2 Build for Multiprocessing" section.
		32	Updated "4.5.2 Debug Steps" section.
	52	52	 Added tables describing the settings for each project.
			 Added a note for debugging flash boot project.
		37	Updated "4.7 Debug and Run for Multiprocessing" section.
		- /	 Clarified explanation of step 2.
			• Added supplemental information on behavior to step 5.
		44	Updated "5.3.2 Create a New Project" section.
			 Modified to a generic description that does not specify which cores are used in multiprocessing.
			Updated tool screen images.
			• Added tables describing the settings for each project.
	4	51	Added "5.3.2.1 NOTE: Configure IAR EWARM Project [Only RZ/T2H]" section.
		54	Updated "5.3.3 Build the Project" section.
			 Swapped the order of "5.3.3.1 NOTE: Build settings [Only Multiprocessing]" and "5.3.3.2 Build" chapters.
			• Added how to build for multiprocessing in "5.3.3.1 Build" section.
			• Renamed "5.3.3.2 Build for Multiprocessing" section.
		55	Updated "5.3.3.2 Build for Multiprocessing" section.
			• Added the running setting to step 1.
			• Added the extra options to step 2.
			• Removed the running setting and tools options from step 3.
			• Moved multicore debugging setting from step 3 to "5.3.5 Debug for Multiprocessing" section.



	Description		tion
Rev.	Date	Page	Summary
		58	Updated "5.3.4 Download & Debug the Project" section.
			• Added a note for debugging flash boot project.
		62	Updated "5.3.5 Debug for Multiprocessing" section.
			• Added multicore debugging setting as step 2.
			• Added supplemental information on behavior to step 5.
			• Added how to debug when changing the project.
		67	Updated "6.2 Create a Project" section.
			• Updated tool screen images and menu names.
		71	Added "6.2.4 Duplication of Resources" section.
		82	Updated "Appendix. Known Issues" section.
			• Added RZ/T2H as target device of No. 3, No. 14, No. 27 and No. 28.
			• Added new issues No. 29 to No. 38.
			• Resolved issues No. 4, No. 13, No. 18, No. 19 and No. 20.
		106	Updated "Appendix. Tool Software Limitations" chapter.
			• Added RZ/T2H as target device of No. 4, No. 7, No. 9 and No. 10.
			• Added new issues No. 12 to No. 19.
			Resolved issues No. 1 and No. 11.
		117	Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section.
			• Add Note for multiprocessing projects.
			• Corrected boot mode name from xSPI0 to xSPI.
			• Updated path description.
			• Add a column for cores to the table.
		-	Removed "Appendix. How to Debug FSP multiprocessing projects with Flash
			Boot Mode" section.
		127	Added "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e ² studio" section.
		132	Added "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM" section.



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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a 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.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

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

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

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

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