

# **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, RZ/N2H

### 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)</u> How to Use (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 <sup>2</sup> studio IDE	5
1.2.3	FSP SC	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
2.5.2	RZ/N2H Evaluation Board	22
3.	e <sup>2</sup> studio Setup	25
3.1	What is e <sup>2</sup> studio?	25
3.2	e <sup>2</sup> studio Prerequisites	25
3.2.1	Windows PC Requirements	25
3.2.2	Installing e <sup>2</sup> studio, Platform Installer and FSP Package	25
3.2.3	Choosing a Toolchain	25
3.2.4	Licensing	25
4.	Tutorial: Your First RZ/T2, RZ/N2 MPU Project – Blinky	26
4.1	Tutorial Blinky	26
4.2	What Does Blinky Do?	26
4.3	Create a New Project for Blinky	26
4.3.1	Details about the Blinky Configuration	33



# RZ/T2, RZ/N2

4.3.2	Configuring the Blinky Clocks	33
4.3.3	Configuring the Blinky Pins	33
4.3.4	Configuring the Parameters for Blinky Components	33
4.3.5	Where is main()?	33
4.3.6	Blinky Example Code	33
4.4	Build the Blinky Project	34
4.4.1	Build	34
4.4.2	Build for Multiprocessing	34
4.5	Debug the Blinky Project	35
4.5.1	Debug Prerequisites	35
4.5.2	Debug Steps	35
4.5.3	Details about the Debug Process	40
4.6	Run the Blinky Project	40
4.7	Debug and Run for Multiprocessing	41
4.8	Import the Project	42
5. 1	FSP SC User Guide	46
5.1	What is FSP SC?	
5.2	Tutorial Blinky	46
5.3	Using FSP SC with IAR EWARM	
5.3.1	Prerequisites	
5.3.2	Create a New Project	48
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	69
5.4.1	Launch FSP SC from IAR EWARM	69
5.4.2	Launch from the Command Prompt	70
5.5	Note when debugging in different workspaces	70
6. 1	FSP Configuration Users Guide	71
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	
6.3	Configuring a Project	
6.3.1	Summary Tab	
6.3.2	Configuring the BSP	
6.3.3	Configuring Clocks	80
6.3.4	Configuring Pins	81



6.4	Configuring Interrupts from the Stacks Tab	84
6.4.1	Creating Interrupts from the Interrupts Tab	84
6.4.2	Viewing Event Links	85
6.5	Adding and Configuring HAL Drivers	86
6.6	Reviewing and Adding Components	87
Appe	ndix. Known Issues	88
Appe	ndix. Tool Software Limitations	122
Appe	ndix. How to Debug FSP Project with Flash Boot Mode	135
Appe	ndix. How to Erase Flash Memory	137
Appe	ndix. How to Change Boot Mode of FSP Project	142
Appe	ndix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e <sup>2</sup> studio	145
For R	Z/T2 FSP v3.0.0	145
Multi	processing with 2 cores for RZ/T devices	145
Multi	processing with 3 or more cores for RZ/T devices	150
For R	Z/N2 FSP v2.2.0	153
Multi	processing with 2 cores for RZ/N devices	153
Multij	processing with 3 or more cores for RZ/N devices	156
Appe	ndix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM	158
For R	Z/T2 FSP v3.0.0	158
Multi	processing with 2 cores for RZ/T devices	158
Multi	processing with 3 or more cores for RZ/T devices	164
For R	Z/N2 FSP v2.2.0	168
Multi	processing with 2 cores for RZ/N devices	168
Multi	processing with 3 or more cores for RZ/N devices	170
Revis	sion History	173



### 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 e<sup>2</sup> 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 e<sup>2</sup> 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 SC

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 3<sup>rd</sup>-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
  - > RZ/N2H Group RZ/N2H Evaluation Board User's Manual
    - Document No. R20UT5522

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>

R20UT4939 Q X @ 0	

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
  - > 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 e<sup>2</sup> studio or FSP SC (for IAR EWARM), and this Getting Started includes tutorial for both tools; the chapters you should read changes.

#### e<sup>2</sup> studio users should read the following chapters:

- Chapter 2 "Set up Evaluation Board"
- Chapter 3 "e<sup>2</sup> 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 SC 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 "e<sup>2</sup> 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 e<sup>2</sup> studio.
- Chapter 5 "FSP SC User Guide"
  - Explains the tutorial with minimal steps to create an FSP project as IAR EWARM project by using the FSP SC 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.
  - $\blacktriangleright$  The explanation is described based on e<sup>2</sup> studio, but most of the explanations are applied to the FSP SC.



### 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 RSK+ 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.

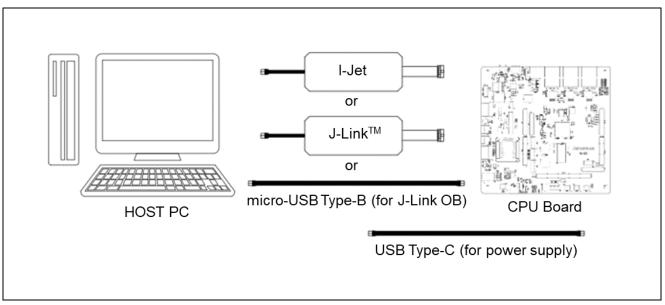


Figure 2: System Configuration Example – with Evaluation Board

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 e<sup>2</sup> studio only for debugging on RZ/T2 and RZ/N2 devices. Renesas e<sup>2</sup> 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 <sup>2</sup> studio version	J-Link Software version
RZ/T	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H	RZ/T2 FSP v3.0.0	2025-04.1	V8.30
RZ/N	RZ/N2L, RZ/N2H	RZ/N2 FSP v2.2.0	2025-01	V8.12e

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 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.

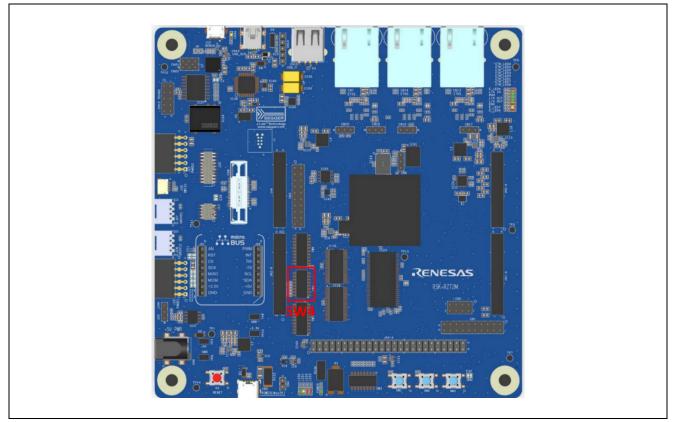


Figure 3: Switch Position of Operation Mode Settings for RSK+RZT2M

Table 2	<b>Operation Mode Switch Settings for RSK+RZT2M</b>

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 I-Jet to the RSK+RZT2M board ensuring that it is plugged in to the header "J20".

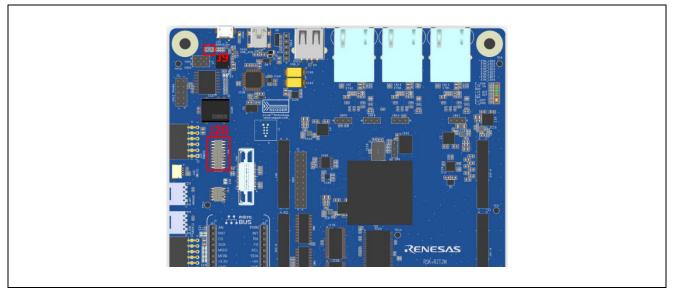


Figure 4: Jumper Position of JTAG Connection for RSK+RZT2M

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.

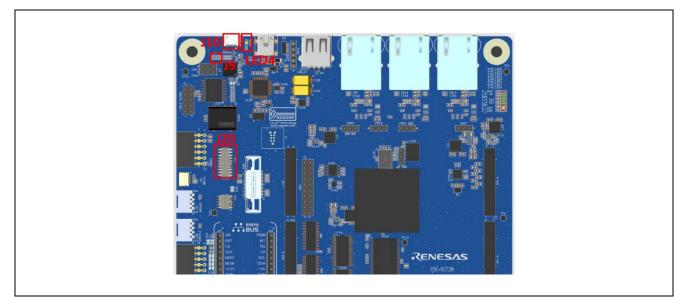


Figure 5: J-Link OB Connection Settings for RSK+RZT2M



### 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.

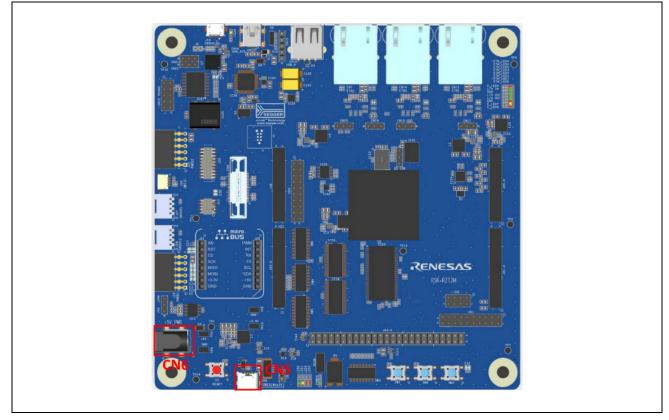


Figure 6: How to Power Supply for RSK+RZT2M



# 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.

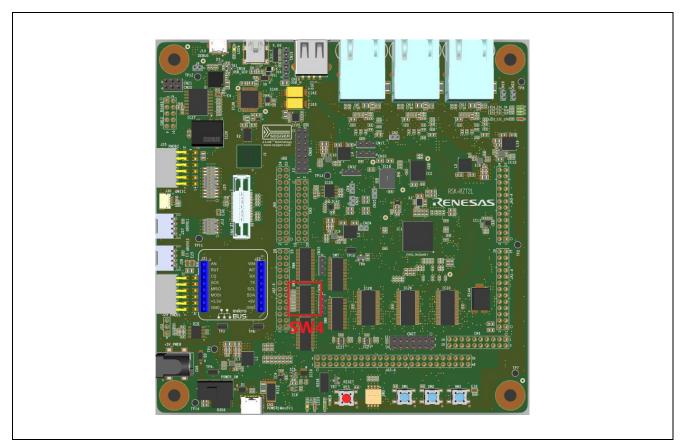


Figure 7: Switch Position of Operation Mode Settings for RSK+RZT2L

Table 3 Operation Mode Swite	h 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 I-Jet to the RSK+RZT2L board ensuring that it is plugged in to the header "J20".

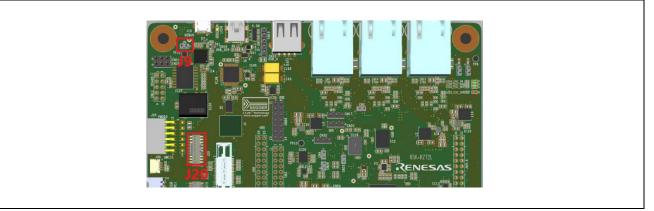


Figure 8: Jumper Position of JTAG Connection for RSK+RZT2L

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.

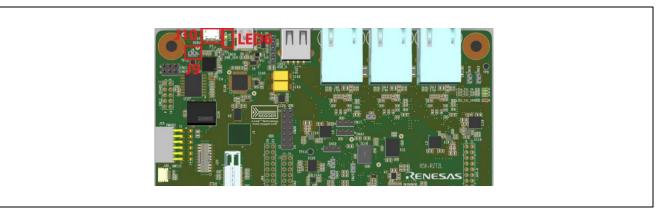


Figure 9: J-Link OB Connection Settings for RSK+RZT2L



### 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.

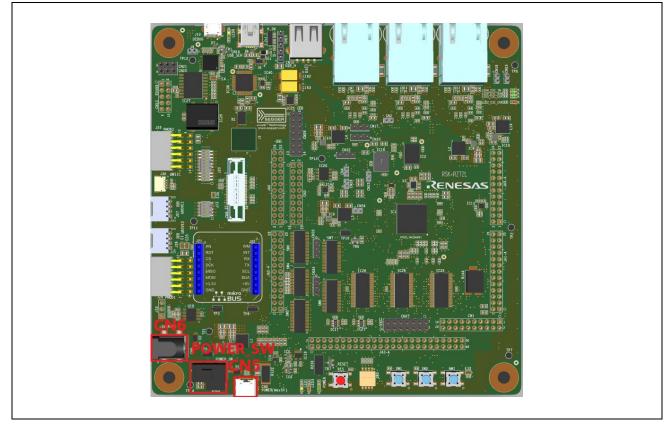


Figure 10: How to Power Supply for RSK+RZT2L

### 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 Renesas web site</u>, please refer to the documentation included with each code and implement the appropriate board settings respectively.

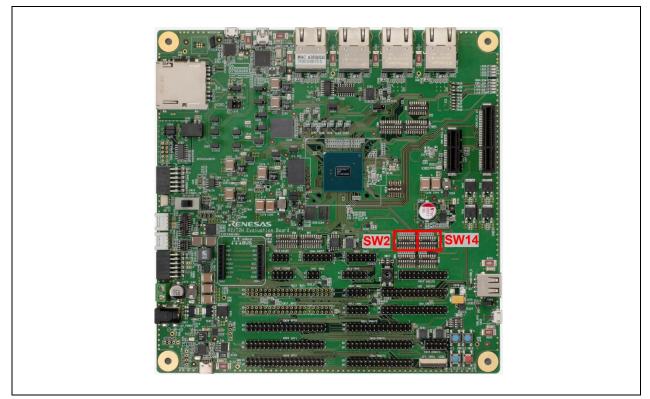


Figure 11: Switch Position of Operation Mode Settings for RZ/T2H Evaluation Board

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 Evaluatio	n Roard
Table 4 Operation Mode Switch Settings for KZ/12H Evaluatio	i Duaru



### 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".

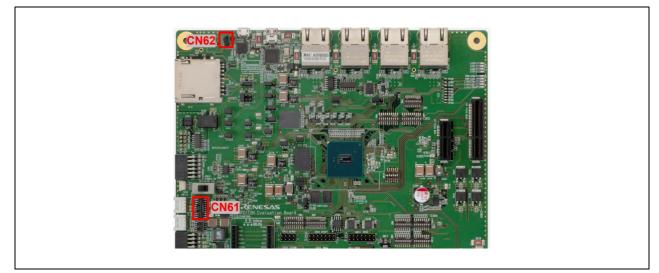


Figure 12: Jumper Position of JTAG Connection for RZ/T2H Evaluation Board

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.

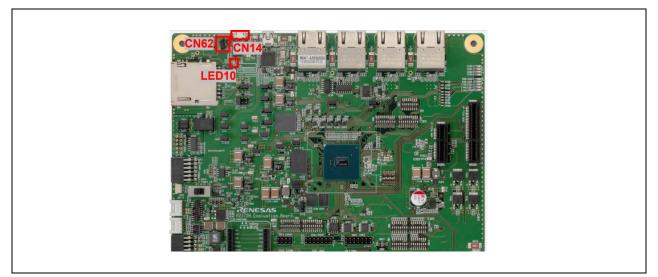


Figure 13: J-Link OB Connection Settings for RZ/T2H Evaluation Board



#### 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.

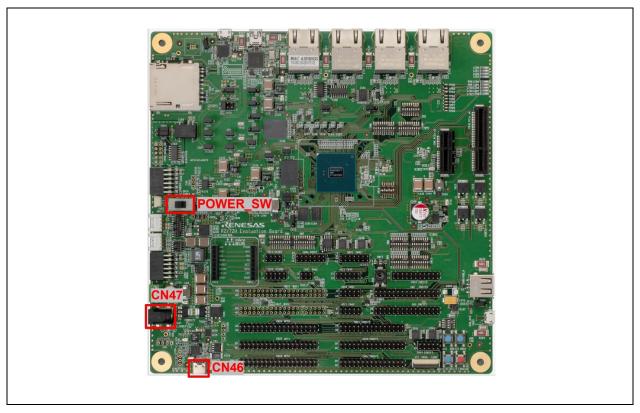


Figure 14: How to Power Supply for RZ/T2H Evaluation Board



### 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> <u>available on Renesas web site</u>, please refer to the documentation included with each code and implement the appropriate board settings respectively.

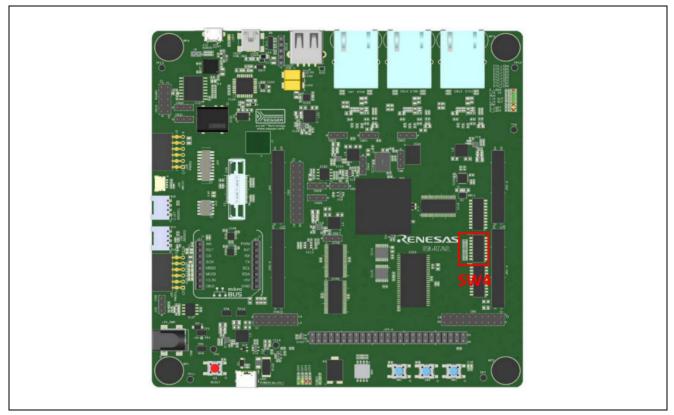


Figure 15: Switch Position of Operation Mode Settings for RSK+RZN2L

Table 5 Operation	Mode Switch	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 I-Jet to the RSK+RZN2L board ensuring that it is plugged in to the header "J20".

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

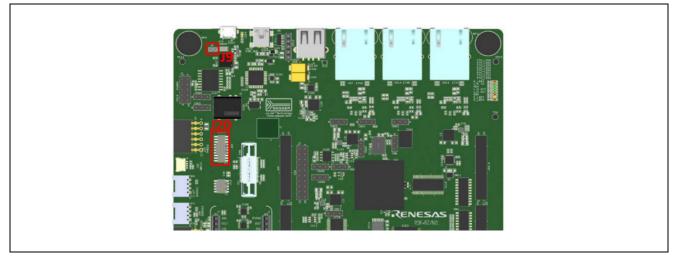


Figure 16: Jumper Position of JTAG Connection for RSK+RZN2L

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.

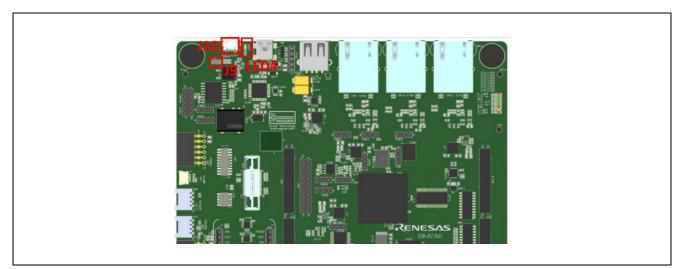


Figure 17: J-Link OB Connection Settings for RSK+RZN2L



### 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.

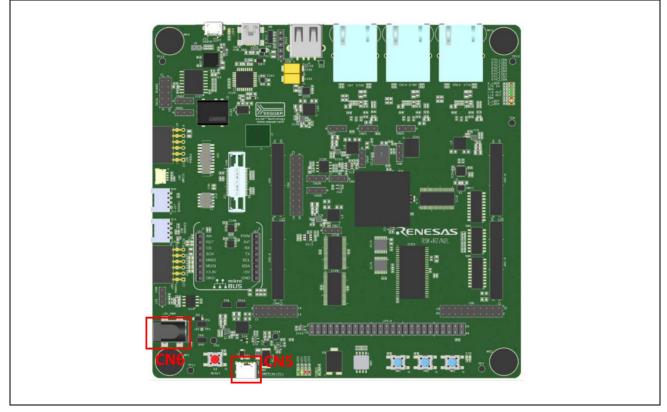


Figure 18: How to Power Supply for RSK+RZN2L



#### 2.5.2 RZ/N2H Evaluation Board

#### 2.5.2.1 Boot Mode

The operation mode settings for the RZ/N2H 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 the sample codes available on Renesas web site, please refer to the documentation included with each code and implement the appropriate board settings respectively.

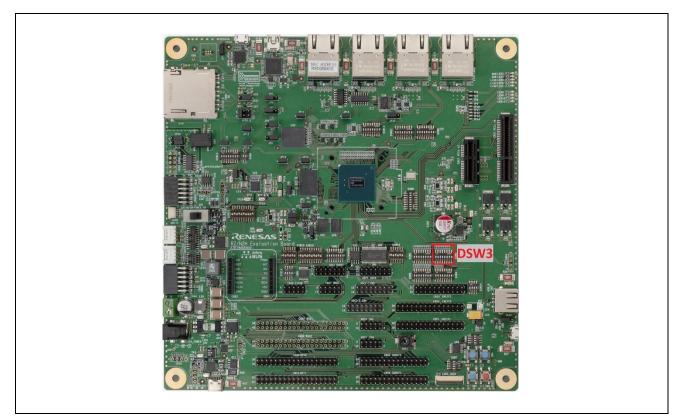


Figure 19: Switch Position of Operation Mode Settings for RZ/N2H Evaluation Board

Switch	Setting	Description
DSW3.1	ON	xSPI1 boot mode (x1 boot serial flash)
DSW3.2	OFF	
DSW3.3	ON	
DSW3.4	ON	CPU0 ATCM 0 wait
DSW3.7	ON	JTAG Authentication by Hash is disabled.

<b>Table 6 Operation Mode Switch</b>	Settings for RZ/N2H Evaluation Board
--------------------------------------	--------------------------------------



### 2.5.2.2 Debugger Connection

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

- 1. Short the jumper block (JP40) for switching the debug connection so that RZ/N2H evaluation board can use the emulator connected to JTAG connector (CN24).
- 2. Connect the emulator (J-Link or I-jet) to a free USB port on your computer.
- 3. Connect the emulator to the RZ/N2H evaluation board ensuring that it is plugged in to the header "CN24".

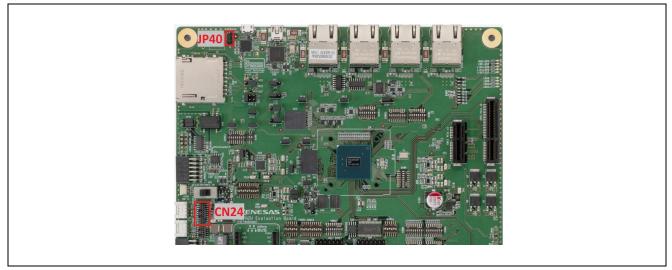


Figure 20: Jumper Position of JTAG connection for RZ/N2H Evaluation Board

If you use J-Link OB on RZ/N2H evaluation board,

- 1. Open the jumper block (JP40) for switching the debug connection so that RZ/N2H evaluation board can use J-Link OB on the board.
- 2. Connect the micro-USB type-B to J-Link OB USB connector (CN26), and then the LED12 is lighted.

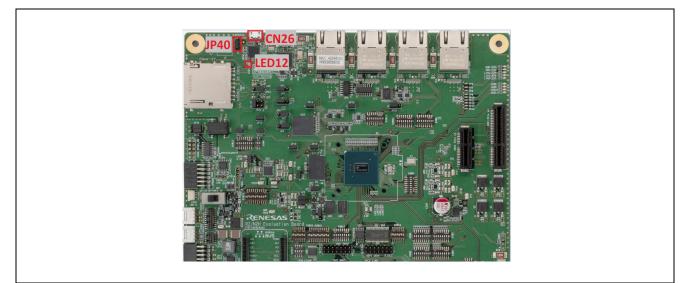


Figure 21: J-Link OB Connection Settings for RZ/N2H Evaluation Board



### 2.5.2.3 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 "CN13" of the RZ/N2H evaluation board.
- When connecting the AC / DC adapter, connect it to the USB connector "J1" of the RZ/N2H evaluation board.
- After connecting to the power (J1 or CN13), 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 J1, use a stabilized power source that is capable of supplying at least 15-V/3 A.

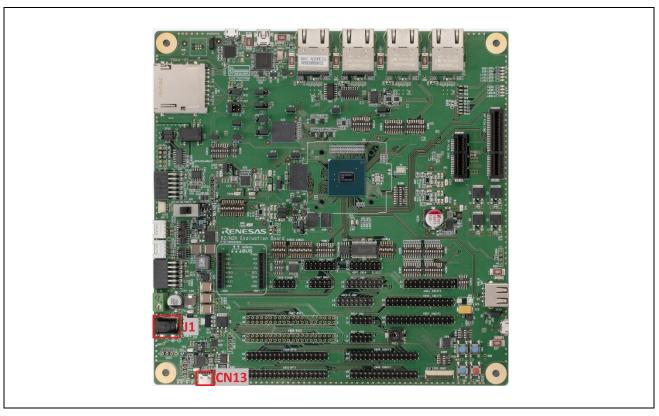


Figure 22: How to Power Supply for RZ/N2H Evaluation Board



# 3. e<sup>2</sup> studio Setup

### 3.1 What is e<sup>2</sup> 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, e<sup>2</sup> 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 e<sup>2</sup> studio Prerequisites

#### 3.2.1 Windows PC Requirements

The following are the Windows PC requirements to use e<sup>2</sup> studio:

For Windows 64-bit version

- System: x64 based processor, 2 GHz or faster, CPU has dual cores or more
  - Windows® 11 (64-bit version)
  - Windows® 10 (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 e<sup>2</sup> studio.

### 3.2.2 Installing e<sup>2</sup> 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 following toolchains are required.

FSP version	Core	Toolchain	Toolchain version
RZ/T2 FSP v3.0.0	CR52	GNU ARM Embedded Toolchain	<u>13.3.Rel1</u> (13.3.1.arm-13-24)
	CA55	GNU ARM A-Profile (AArch64 bare-metal)	<u>10.3-2021.07</u> (10.3.1.20210621)
RZ/N2 FSP v2.2.0	CR52	GNU ARM Embedded Toolchain	<u>13.3.Rel1</u> (13.3.1.arm-13-24)
	CA55	GNU ARM A-Profile (AArch64 bare-metal)	<u>10.3-2021.07</u> (10.3.1.20210621)

#### Table 7 Toolchain version for each FSP

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 e<sup>2</sup> 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. The procedure from creating a project to running it varies depending on the number of cores used and boot mode. This chapter shows only some of the cases where RAM execution is used. Refer to Table 8 Project Creation Procedure (e<sup>2</sup> studio) to find out which steps are required for your application.

Steps	Single-core proces	sing	Multiprocessing	
	RAM execution	Flash boot mode	RAM execution (Combination of (CR52 CPU0, CPU1) and (CR52 CPU0, CA55 Core0) only)	RAM execution (Other combinations) Flash boot mode
Check tool limitations		Appendix. Too	l Software Limitations	
Erase flash memory(if needed)		Appendix. How	to Erase Flash Memory	
Create a project	4.3 Create a New Project for Blinky	4.3 Create a New Project for Blinky Appendix. How to Debug FSP Project with Flash Boot Mode	4.3 Create a New Project for Blinky	Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e2 studio
Build the project	4.4.1	Build	4.4.2 Build for Multiprocessing	
Debug the project	4.5.2 De	bug Steps	4.7 Debug and Run for Multiprocessing	
Run the project	4.6 Run the	Blinky Project		

#### Table 8 Project Creation Procedure (e<sup>2</sup> studio)



#### Note for multiprocessing projects:

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.

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 e<sup>2</sup> studio, click File > New > Renesas C/C++ Project > Renesas RZ.

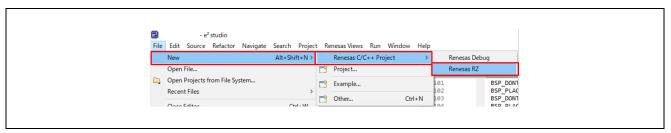


Figure 23: New C/C++ Project

- 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.

Renesas RZ Project         I/C++         I/
/C++       Create an executable or static library C/C++ FSP project for Renesas RZ/N.         Renesas RZ/T C/C++ FSP Project         Create an executable or static library C/C++ FSP project for Renesas RZ/T.         Renesas RZ/V C/C++ FSP Project         Create an executable or static library C/C++ FSP project for Renesas RZ/T.         Renesas RZ/V C/C++ FSP Project         Create an executable or static library C/C++ FSP         Create an executable or static library C/C++ FSP         Renesas RZ/V C/C++ FSP Project         Create an executable or static library C/C++ FSP
v

Figure 24: Renesas RZ C/C++ FSP Project



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

	14	ibic > c studio iven	Ty created roject	(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 9 e<sup>2</sup> studio Newly Created Project Settings (1)

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

🕲 Renesas RZ/T C/C++		- 0	×	
Renesas RZ/T C/C++ Project Name and Loca			)	
Project name Blinky Use gefault locat Location:	¥Blinky	Brows	se	
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish Canc	cel	

Figure 25 : e<sup>2</sup> 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			thout 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
IDE Project Type	e <sup>2</sup> studio mana	ged build			
Toolchains	GNU ARM Er	nbedded 13.3.1.arm-	13-24		GNU ARM A- Profile (AArch64 bare-metal) and 10.3.1.20210621
Debbuger	J-Link ARM				

 Table 10
 e<sup>2</sup> studio Newly Created Project Settings (2)

Renesas RZ/T C/C++ FSP Project Renesas RZ/T C/C++ FSP Project	- • ×
Device and Tools Selection	
	Board Description RZT2H Evaluation Board (RAM execution without flash memory)
Core: CR52 CPU0	Device Details
Language: ●C ○C++	TrustZone No Pins 729 Processor Cortex-R52
IDE Project Type	Debugger
	J-Link ARM
Toolchains <toolchain> <toolchain version="">       &lt;</toolchain></toolchain>	
(?)	< Back Next > Einish Cancel

**Figure 26 : e<sup>2</sup> studio Project Configuration Window (Part 2)** 



9. Select a bundle file. For the secondary project of multiprocessing, select the primary project as Preceding Project. Built the primary project names in the same workspace appear as an option in the drop-down list.

		Sie II e staalo Ite	ing electron in ogeee	2000.gs (0)	
	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
Preceding Project	None	None	The primary project	None	The primary project

 Table 11
 e<sup>2</sup> studio Newly Created Project Settings (3)

#### 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.

Resease RZ/T C/C++ FSP Project         Preceding Project or Smart Bundle Selection <ul> <li>None</li> <li>Choose this option when creating a project for the primary processor core (no preceding project or Smart Bundle).</li> <li>Preceding Project</li> <li>Blinky, primary</li> <li>Choose this option if you have access to the project source code of the preceding processor core or security context.</li> <li>Smart Bundle</li> <li>Resolved location:</li> <li>Workspace. File System. Variables.</li> <li>Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor core or security context.</li> </ul> Preceding Project/Smart Bundle Details           FSP version                Gold AMK Embedded               Toolchain version                Board                Drevice             Reador/MAK Embedded               Core                Cores                Cores                Cores                Zones                Cores                Cores                Cores                Cores	Renesas RZ/T C/C+	+ FSP Project - C X
One     Choose this option when creating a project for the primary processor core (no preceding project or Smart Bundle).     Preceding Project:     Binky_primary     Choose this option if you have access to the project source code of the preceding processor core or security context.     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	Renesas RZ/T C/C+	FSP Project
Preceding Project     Blinky_primary     Choose this option if you have access to the project source code of the preceding processor core or security context.     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         Colchain	Preceding Project or S	mart Bundle Selection
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 Solchain GNU ARM Embedded Toolchain version Board <solort mesion=""> Board <s< th=""><th>() None</th><th>Choose this option when creating a project for the primary processor core (no preceding project or Smart Bundle).</th></s<></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort></solort>	() None	Choose this option when creating a project for the primary processor core (no preceding project or Smart Bundle).
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              File System             Variables             Tookchain             GNU ARM Embedded             Tookchain version	Preceding Project:	Blinky_primary V
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     creer or security context.      Preceding Project/Smart Bundle Details      FSP version <free <="" <<="" context.="" free="" fsp="" security="" td="" version=""><td>O Smart Rundler</td><td>Choose this option if you have access to the project source code of the preceding processor core or security context.</td></free>	O Smart Rundler	Choose this option if you have access to the project source code of the preceding processor core or security 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         FSP version >           FSP version <fsp version="">         Toolchain         GNU ARM Embedded           Toolchain         <toolchain version="">          Baard         &lt;</toolchain></fsp>	O smart bundle.	Recolved Institute
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 <fsp version="">           Toolchain         GNU ARM Embedded           Toolchain         <foor (boot="" mode)="">           Board         <foor (boot="" mode)="">           Device         RBA0x9077M44G8G           Core         CR52_0</foor></foor></fsp>		
FSP version <fsp version="">           Toolchain         GNU ARM Embedded           Toolchain version         <toolchain version="">           Board         <board (boot="" mode)="">           Device         RBA090077M44GBG           Core         CR52_0</board></toolchain></fsp>		Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding processor
Toolchain         GNU ARM Embedded           Toolchain version <toolchain version="">           Board         <board (boot="" mode)="">           Device         RA09G077M44GBG           Core         CR32_0</board></toolchain>	Preceding Project/Sm	art Bundle Details
	Toolchain Toolchain version Board Device Core	GNU ARM Embedded <toolchain version=""> <board (boot="" mode)=""> RBA09G077M44GBG CR52_0</board></toolchain>

Figure 27 e<sup>2</sup> studio Project Configuration Window (Part 3)



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

Renesas RZ/T C/C++ FSP Project	– <b>–</b> ×
Renesas RZ/T C/C++ FSP Project	
Build Artifact and RTOS Selection	
Build Artifact Selection	RTOS Selection
Executable     Project builds to an executable file	No RTOS
O Static Library	
<ul> <li>Project builds to a static library file</li> </ul>	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel

Figure 28 : e<sup>2</sup> 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		
Project Template Selection		
Project Template Selection		
Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink C runtime environment. [Renesas.RZT. <fsp version="">.pack]</fsp>	c LEDs if available. This project will initialize clocks, p	ins, stacks, and the
Bare Metal - Minimal Bare metal FSP project that includes BSR This project w [Renesas.RZT. <fsp version=""> .pack]</fsp>	vill initialize clocks, pins, stacks, and the C runtime e	nvironment.
Code Generation Settings		
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

**Figure 29 : e<sup>2</sup> 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 e<sup>2</sup> 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.

File Edit Navigate Search Project	n.xml - e <sup>2</sup> studio tt Renesas Views Run Window Help		- 0	
📓 🖲 🕶 🗞 🕶 🚳 🕸 🕶 💁	•	Q 🗄	取 C/C++ 微 FSP Configuration 本	
₽ 🕸 [ ] FSP Configuration ×			0	
Clocks Configuration			Generate Project Cont	tent
			🐯 Restore Defa	aults
PLI	Like istanding state. Standing state. Standing state. CLMA0 error not mask. CLMA0 error not mask.	→ PLLo SSC Disabled  → CA55C1CLI MMF: 111.7kHz  → CA55C2CLI ↓	LK GOOMHiz v LK GOOMHiz v LK GOOMHiz v	
cu	LI 1000MH2 ↓ MAI Enabled ~ ↓ MAI enor not mask ~		K S00MHz K S00MHz 00MHz	
Clocks Pins Inter	upts Event Links Stacks Components			>

Figure 30 : 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	tion X		
Summary		Generate Project	Content
Project Summar	у	RENESAS	^
Board:	RZT2H Evaluation Board (RAM execution without flash memory)	RENESAS	
Device:	R9A09G077M44GBG		
Core:	CR52 CPU0		
	GCC for Renesas RZ		
Toolchain Version:			
FSP Version:	<esp varsion=""></esp>		
Project Ty			
Location:	/Blinky 🚭		

**Figure 31 : e<sup>2</sup> 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 e<sup>2</sup> 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 <del>Renesas Views Run Window Help</del>		a. Project -> Build Project
🔨 🐞 🔳 🎄 Debug 🗸 🖻 Blinky Debug_Flat		b. Click hammer icon
陷 Project Explorer 🗙 🕞 🛱 🏷 🖇 🖳 🗖	∰ *[Blinky] FSP Configuration ⊠	c. Right click -> Build project
> 🔄 Blinkyj	Summary	

Figure 32 : e<sup>2</sup> 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 E Console ×	🗶 🕹 🖓 😒 📰 🗉 = 🗟 🗐 🛃 🖬 ▾ ⊓ 🗖
CDT Build Console (Blinky)	
'arm-none-eabi-gcc -mcpu=cortex-r52 -mthumb -mfloat-abi=hard -mfpu=neon-fp-armv8 'arm-none-eabi-gcc @Blinky.elf.in" arm-none-eabi-gcc @Blinky.elf.in" arm-none-eabi-objcopy -0 srec "Blinky.elf" "Blinky.srec" arm-none-eabi-objcopy -0 srec "Blinky.elf" text data bss dec hex filename 3004 0 65560 68564 10bd4 Blinky.elf	-fdiagnostics-parseable-fixits -Og -fmessage-length=( -fdiagnostics-parseable-fixits -Og -fmessage-length=( -fdiagnostics-parseable-fixits -Og -fmessage-length=( -fdiagnostics-parseable-fixits -Og -fmessage-length=(
11:46:51 Build Finished. 0 errors, 0 warnings. (took 12s.789ms)	~

Figure 33 : e<sup>2</sup> 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 Manager	
		>
Ð	Tracealyzer	>
	Run	Ctrl+F11
拖	Debug	F11
	Run History	>
-	Run As	>
	Run Configurations	
	Debug History	>
	Debug As	>
	Debug Configurations	
9	External Tools	>

**Figure 34 : e<sup>2</sup> studio Debugger Configurations Selection Option** 

* -	Q : 0₀ - ₺; 0⊨ 00 1;	
(1	(no launch history)	
D	Debug As	>
D	Debug Configurations	
C	Organize Favorites	

**Figure 35 : e<sup>2</sup> 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
<ul> <li>C/C++ Application</li> <li>C/C++ Remote Application</li> </ul>	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 36 : e<sup>2</sup> 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 12 e<sup>2</sup> studio Newly Created Project Debug Settings (1)



Debug Configurations				
Create, manage, and run configura	tions			Ť
	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 <sup>®</sup> Renesas GDB Hardware Debugc	Register initialization	No Yes		~
C Blinky Debug_Flat	Reset at the beginning of connection Reset at the end of connection	Yes No		~
C Renesas Simulator Debugging (		No		~
increase simulator bebugging (	Reset before download Reset after download	No		-
		NO FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF		
	ID Code (Bytes)	No		***
	Hold reset during connect	Yes		~
	Set CPSR(5bit) after download			~
	Prevent Releasing the Reset of the CM3 C Secure Vector Address	lore tes		~
	Non-secure Vector Address			100
	Hot Plug	No		100
	Disconnection Mode	Stop		v
	✓ SWV	Stop		~
	Core clock (MHz)	0		
	✓ TrustZone	0		
	Set TrustZone secure/non-secure bounda	No.		~
	Set indstabile secure non-secure bounda	1105 110		*
< >			-	
Filter matched 9 of 11 items			Revert /	Apply

Figure 37 : e<sup>2</sup> 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, C	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 13 e<sup>2</sup> studio Newly Created Project Debug Settings (2)



<ul> <li>Debug Configurations</li> <li>Create, manage, and run configurations</li> </ul>		
Port Inst     Construct Application     Construct Application	Name         Binky_cx35_secondary Debug Fit           Main         \$> Debug bardware           Debug bardware         Action Settings           CORE         Settings           Connect to remote COB server         Host name or P address           Connect to remote COB server         Connect to remote COB server           Connect to remote COB server         Connect to remote COB server           Connect to remote COB server         Connect to remote COB server           Connect to remote COB server         Connect to remote COB server           Connect to remote COB server         Set to remote COB server Action to remote COB server           Connect to remote COB server Actions         Additional COB server Actions	Browse
Filter matched 10 of 12 items		Revert Apply
0		Debug Close

Figure 38 : e<sup>2</sup> studio Debugger Configurations Window with Blinky Project (CA55)

- 5. (RZ/T2H CR52 CPU1) Set the script file for TCM initialization.
  - Debugger > Connection Settings > J-Link
    - Script File : \${workspace\_loc:/\${ProjName}}/script/initialization\_TCM.JLinkScript

Debug Configurations			:
Create, manage, and run configurations			Ś
🖺 🖻 🐌 🗎 🗮 🖻 🔬 -	Name: Blinky_cpu1_secondary Debug_Flat		
type filter text	📄 Main 🕸 Debugger 🕨 Startup 🔲 Common	lie Source	
E C/C++ Application E C/C++ Remote Application E EASE Script	Debug hardware: J-Link ARM V Target Devic	e: R9A09G077M44_CR5	
C GDB Hardware Debugging	GDB Settings Connection Settings Debug Too	l Settings	
GDB OpenOCD Debugging	✓ J-Link		^
<ul> <li>GDB Simulator Debugging (RH850)</li> <li>Java Applet</li> </ul>	Туре	USB	~
Java Applet	J-Link Serial	(Auto)	 k
Launch Group	Settings File	\${workspace_loc:/\${ProjName}}/\${LaunchConfigName}.jlin	k
Remote Java Application	Script File	\${workspace_loc:/\${ProjName}}/script/initialization_TCM.JL	inkScript
Renesas GDB Hardware Debugging	Log File	\${workspace_loc:/\${ProjName}}/JLinkLog.log	
C Blinky_cpu1_secondary Debug_Flat	Low Power Handling	No	~
C* Renesas Simulator Debugging (RX, RL78	✓ IP Connection		
En Renesas Simulator Debugging (RA, RL/6	Connection Method	IP via LAN	~
	Host Name/IP Address[:port number]		
	Identifier		
	Tunnel Server		
	Port Number		
	Password		
	✓ Interface		×
Filter matched 13 of 15 items		Revert	Apply

Figure 39 e<sup>2</sup> studio Debugger Configurations Window with Blinky Project (RZ/T2H CR52 CPU1)



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

Progress Information	- D X
Preparing launch delegate	
Configuring GDB	
	Cancel Details >>

Figure 40: Start Debugging



### 4.5.3 Details about the Debug Process

In debug mode, e<sup>2</sup> 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 41 : e<sup>2</sup> 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 42 : e<sup>2</sup> 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
  - RZ/N2H Evaluation Board: LED3 (CR52 CPU0), LED4 (CR52 CPU1), LED8 (CA55 Core0)



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

· · · · · ·	

Figure 43 : e<sup>2</sup> studio Debugger Pause Icon

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

**Figure 44 : e<sup>2</sup> 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 45 Warning Window of Starting Debug Session

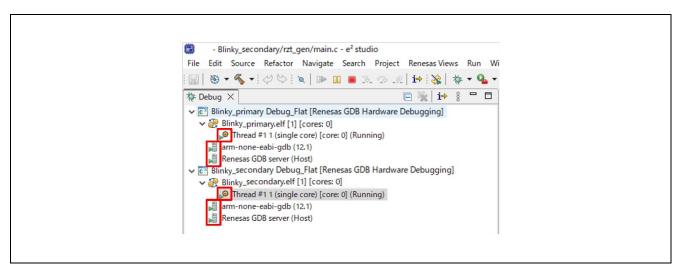
4. When Figure 45 is shown, please click **Yes** to proceed the launch.

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

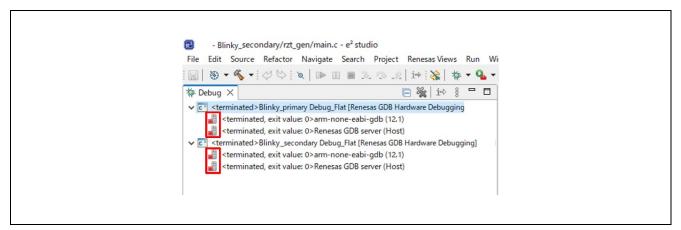
Figure 46 Warning Window of Device Name



- 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 47 During Program Execution**



#### Figure 48 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	<b>.</b>				
	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 49 e<sup>2</sup> studio Import

2. Click General > Existing Projects into Workspace.

<ul> <li>Import</li></ul>	

Figure 50 e<sup>2</sup> 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 51 e<sup>2</sup> 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	<ul> <li>✓ Browse</li> </ul>
Projects:	
Blinky_cpu0_primary (Blinky_cpu0_primary/)	Select All
<ul> <li>Blinky_cpu1_secondary (Blinky_cpu1_secondary/)</li> <li>Blinky_nor_cpu0_primary (Blinky_nor_cpu0_primary/)</li> </ul>	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 52 e<sup>2</sup> studio Select Archive File to Import Project

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

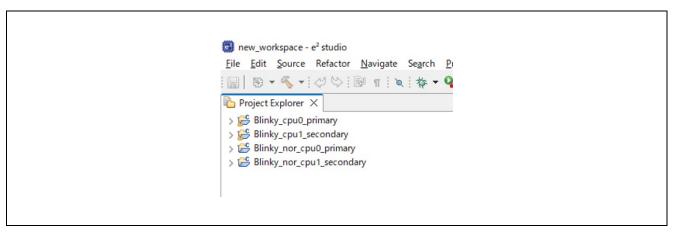


Figure 53 e<sup>2</sup> studio New Workspace

Note:

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



# 5. FSP SC User Guide

# 5.1 What is FSP SC?

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 3<sup>rd</sup>-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 3<sup>rd</sup>-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 FSP SC 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 and RZ/N2H:

If you use the IAR EWARM to debug RZ/T2H and RZ/N2H 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> and <u>http://www.renesas.com/rzn2h</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. The procedure from creating a project to running it varies depending on the number of cores used and boot mode. This chapter shows only some of the cases where RAM execution is used. Refer to Table 14 Project Creation Procedure (IAR EWARM, FSP SC) to find out which steps are required for your application.

		et creation riocedure (	, ,	/
Step	Single-core proces	sing	Multiprocessing	
	RAM execution	Flash boot mode	RAM execution (Combination of (CR52 CPU0, CPU1) and (CR52 CPU0, CA55 Core0) only)	RAM execution (Other combinations) Flash boot mode
Check tool limitations		Appendix. Tool	Software Limitations	
Erase flash memory(if needed)		Appendix. How t	o Erase Flash Memory	
Create a project	5.3.2 Create a New Project	5.3.2 Create a New Project Appendix. How to Debug FSP Project with Flash Boot Mode	5.3.2 Create a New Project	Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM
Build the project	5.3.3	3.1 Build	5.3.3.2 Build for Multiprocessing	
Debug the project	5.3.4 Download	& Debug the Project	5.3.5 Debug for	
Run the project			Multiprocessing	

## Table 14 Project Creation Procedure (IAR EWARM, FSP SC)

#### Note for multiprocessing projects:

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.

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 1<sup>st</sup> build of the primary project in 5.3.3 section.

#### 1. Start the FSP SC.

- FSP SC 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\_vYYY-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.

File Window Help Run		_			
New Import	FSP Project		- 0	SP Visualization	
	New FSP Project	ct		The active editor element does not	use this view
Open Close Ctrl+					
Close All Ctrl+Shift+					
Save Ctri-					
Save All Ctrl+Shift					
Exit					
🔲 Properties 😰 Problems					
Properties Problems Property	Value	년 1월 장 1월 1월 1월 18월 18월 18월 18월 18월 18월 18월 18	Console No consoles to display at this		: • · ° • • •
	Value	C E 7 C 8 - D			1 <b>□ • 1 • □</b>
	Value	· · · · · · · · · · · · · · · · · · ·			19×11ו•
	Value	1 <b>E</b> 7 8 - 0			: • · · • •
	Value	1 1 1 7 III 8 - D			: • · • • •
	Value	C E 7 8 - D			9 - 13
	Value	다. (1997) 1997 1997 1997 1997 1997 1997 1997			1 0 • 1 <b>1 • • •</b> •

Figure 54 : FSP SC New Project



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

	1 a	bit 15 FBI SC III	iy Created Project	settings (1)	
	Ŭ . I		Multiprocessing (CR52 CPU0, CR52 CPU1)		5 Core0)
		Primary	Secondary	Primary	Secondary
Project name	Blinky	Blinky_primary	Blinky_secondary	Blinky_primary	Blinky_secondary

Table 15 FSP SC Newly Created Project Settings (1)

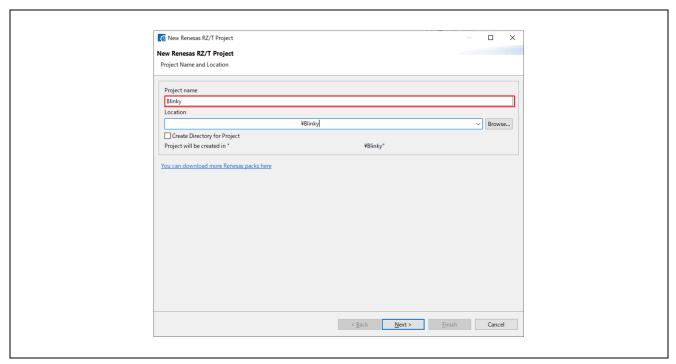


Figure 55 : FSP SC Project Settings



- 4. Select the **FSP** version.
- 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	(CR52 CPU0, CR52 CPU1)		Multiprocessing (CR52 CPU0, CA55 Core0)	
		Primary	Secondary	Primary	Secondary
Board		(RAM execution wite raluation Board (RAM	thout flash memory) A execution without f	lash memory)	
Core	CR52_0 or CR52 CPU0	CR52_0 or CR52 CPU0	CR52_1 or CR52 CPU1	CR52 CPU0	CA55 Core0
IDE Project Type	IAR EWARM	[v9.60+]			

#### Table 16 FSP SC Newly Created Project Settings (2)

e Selection	ery)
KZT2H Evaluation Board (RAM execution wit v	vry)
CR52 CPU0 V Device Details	
<b>T</b> 1 <b>T</b>	
uage:   C  C++  TrustZone No  Pins 729  Processor Cortex-R52	
oject Type	
WARM [v9.40+]	
hains	
Foolchain for ARM	

### Figure 56 : Target Device and IDE Selections



9. Select a bundle file. For the secondary project of multiprocessing, select the file of primary project. The file is generated in the Debug/Exe directory of the project after building the project.

			J J	8 ( )	
	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
Use Smart Bundle	Uncheck	Uncheck	-	Uncheck	Check
Smart Bundle	-	-	.sbd file of the primary project	-	.sbd file of the primary project

 Table 17 FSP SC Newly Created Project Settings (3)

#### 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           Existing Smart Bundle Selection           Smart Bundle Selection           Select a Smart Bundle (*.sbd) file describing the configuration of the preceding processor core.           Smart Bundle Details           FSP version            Toolchain         IAR Toolchain for ARM           Toolchain            Board            Device         RBA096077M446BG           Core         CR52_0           Zones         CR52_0	孩 New Renesas RZ/T Project		– <b>– ×</b>
Smart Bundle:     ¥Blinky, primary¥Debuq¥Exe¥Blinky, primary±Debug¥Exe¥Blinky, primary±Bd       FSP version  <			
Select a Smart Bundle (*sbd) file describing the configuration of the preceding processor core.           Smart Bundle Details           FSP version <fsp version="">           Toolchain         IAR Toolchain for ARM           Toolchain version         Board           Board         &lt;60ard (boot mode)&gt;           Device         R9A096077M44GBG           Core         CR52_0</fsp>	Existing Smart Bundle Selection		
Smart Bundle Details           FSP version           Toolchain           IAR Toolchain for ARM           Toolchain version           Board           Version           R8A0950771M4468G           Core         CR52_0	Smart Bundle:	¥Blinky_primary¥Debug¥Exe¥Blir	nky_primary.sbd Browse
FSP version <fsp version="">           Toolchain         IAR Toolchain for ARM           Toolchain version            Board            Device         RsA09G077M44GBG           Core         CR52_0</fsp>	Select a Smart Bund	le (*.sbd) file describing the configuration of the preceding process	sor core.
Toolchain         IAR Toolchain for ARM           Toolchain version	Smart Bundle Details		
	Toolchain Toolchain version Board Device Core	IAR Toolchain for ARM «Board (boot mode)» RSA09G077M44GBG CR52_0	

**Figure 57 Bundle File Selection** 



### 10. Select RTOS.

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

🕼 New Renesas RZ/T Project		— 🗆 X
New Renesas RZ/T Project		
RTOS Selection		
RTOS Selection		
No RTOS	~	
	< <u>B</u> ack <u>N</u> ext	Einish Cancel
	- Zack Henry	

Figure 58 : 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.

🐼 New Renesas RZ/T Project	- <b>D</b> X	
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 available the C runtime environment. [Renesas.RZTpack]     Bare Metal - Minimal Bare metal FSP project that includes BSR This project will initialize clock [Renesas.RZTpack]		
< <u>B</u> ack	Next > Einish Cancel	

**Figure 59 : 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.

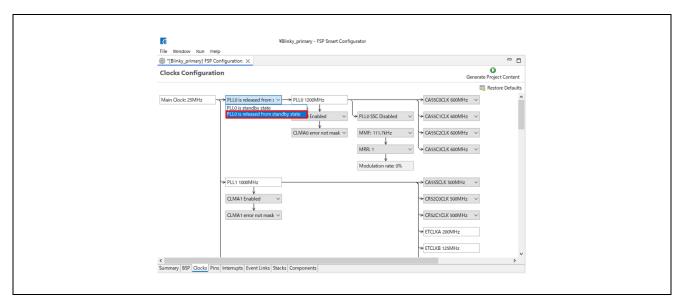


Figure 60 : Enable PLL0 in the Primary Project using CR52 CPU0

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

(Blinky] FSP Cor	figuration $ imes$				- 8
Summary					Generate Project Content
Project Sum	imary			ર	NESAS
Board:	RSK+RZT2N	I (RAM execution wi	thout flash memory)		
Device:	R9A07G075	M24GBG			
FSP Version:	<fsp td="" version<=""><td>&gt;</td><td></td><td></td><td></td></fsp>	>			
Project Type:	Flat				
Location:	C:/Users/		'/Blinky 🐳		

Figure 61 : 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.

script
src
🗋 .api_xml
.secure_azone
.secure_xml
Blinky.ewd
Blinky.ewp
🔮 Blinky.eww

Figure 62 : FSP Project Workspace

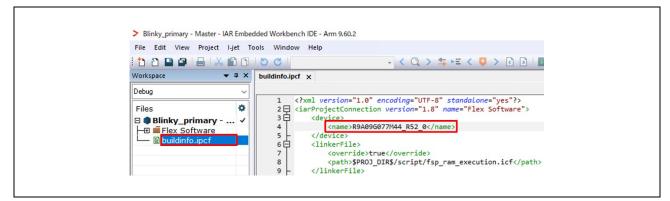


## 5.3.2.1 NOTE: Configure IAR EWARM Project [RZ/T2H and RZ/N2H]

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

Device name	Single-core processing	Multiprocessing (CR52 CPU0, CR5	2 CPU1)	Multiprocessing (CR52 CPU0, CA5	5 Core0)
		Primary	Secondary	Primary	Secondary
RZ/T2H	R9A09G077 M44_R52_0	R9A09G077M44 _R52_0	R9A09G077M44_ R52_1	R9A09G077M44 _R52_0	R9A09G077M44 _A55
RZ/N2H	R9A09G087 M44_R52_0	R9A09G087M44 _R52_0	R9A09G087M44_ R52_1	R9A09G087M44 _R52_0	R9A09G087M44 _A55





### Figure 63 : IAR EWARM Project File (CR52)

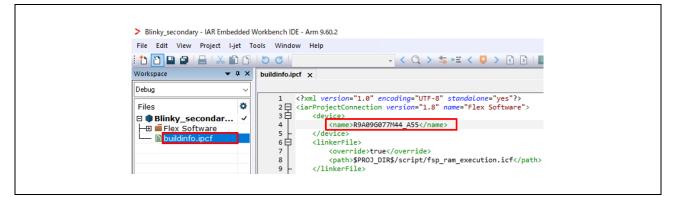


Figure 64 : IAR EWARM Project File (CA55)

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.

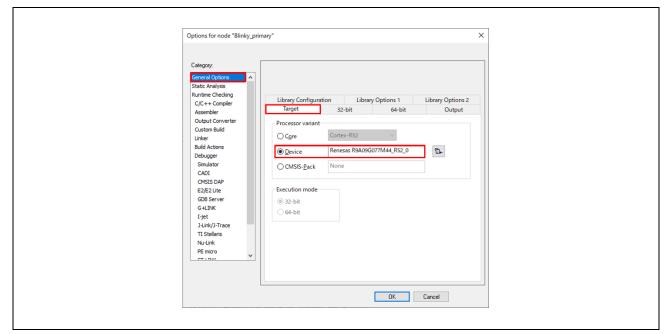


Figure 65 : Project Options – Device (CR52 CPU0)

Options for node "Blinky_see Category:	iondary" X
General Options A Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custon Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris NV-Link PE micro	Library Configuration Library Options 1 Library Options 2 Target 32-bit 64-bit Output  Processor variant  C Cgre Cortex-A55  Decrice Renesas R9A09G077M44_A55 CMSIS-Pack None Execution mode 32-bit 64-bit

Figure 66 : Project Options – Device (CA55)



#### Note:

If any of the following applies, the contents of buildinfo.ipcf will be overwritten and the device name reverts to its pre-modified name.

- A project is built for the first time after creating the project.
- A project is re-opened in IAR EWARM after changing the FSP configuration of it and clicking "Generate Project Content" in FSP SC.

This will result in an error message, 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.

luild
Messages Unknown device name: 'R9A09G077M44_CR52_0' Unknown device name: 'R9A09G077M44_CR52_0'
Build Debug Log

Figure 67 : Error Message after Configuration Change



## 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.

1000		Add Files		- < Q > \$ HE < Q > R >   <b>D</b> =   <b>D</b> • ]   <b>B</b> ]
and the second		Add Group		
Workspace				
Debug	<b>(</b> 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	]
		Compile	Ctrl+F7	
	0	Rebuild All		
	₫	Clean		
	P	Batch build	F8	
		C-STAT Static Analysis	,	
Blinky	٥	Stop Build	Ctrl+Break	

Figure 68 : Make Button

				<b>5</b> 11-	1.1	^
	lessages			File	Line	
5	0% Generating S	ecure Bundle100% Ge	enerating Secure Bundle			
	otal number of er					
To	otal number of w	arnings: 0				
	uild succeeded					~

Figure 69 : 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.

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

 Proj	ter - IAR Embedded Workbench IDE - Arm 9.6 ject I.jet Tools Window Help Add Files Add Group Import File List Add Project Connection Edit Configurations Add CMakeLists.tbt to Project Configure Project Force Reconfiguration Add CMake Connector	<pre>&gt;&gt; **********************************</pre>
	Remove Create New Project	ions> f "\$PROJ_DIR\$/xcl/rasc_ tions> traOptions> f "\$PROJ_DIR\$/xcl/rasc_ traOptions>
٥	Options Alt+F7	tryPoint> l>system_init
		EntryPoint> EVars>

Figure 70 IAR EWARM Project Options

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

Options for node * Category: General Options State: Analysis Runtime Checking C/C++ Compler Assembler Output Converter Cutom Build Linker Build Actions Compose Simulator CADI CHSS DAP E2/E2 Lite GDB Server G 4LDK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro	Factory Settings      Factory Settings	
	OK Cancel	

Figure 71 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.

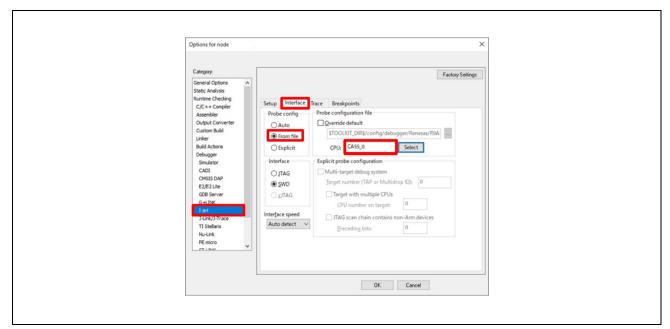


Figure 72 IAR EWARM Project Options for the Primary Project (I-jet Interface)

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

Output Converter Custom Buld Lirker Buld Actons Debugger Simulator CADI CADI CADI CADI CASIS DAP E2/E2 Lite GDB Server I-jet J-Link/J-Trace TI Stellarias Nv-Link PE micro ST-LINK
--

Figure 73 : 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. (RZ/T2H CR52 CPU1) Click **Debugger** > **Setup**, check **Use macro file(s)** and add "\$PROJ\_DIR\$\script\initialization\_TCM.mac".

General Options ∧ Static Analysis Runtime Checking C/C++ Compler Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G+LINK Light Device description file Override default
J-Jink/J-Trace TI Stellaris Nu-Link PE micro

Figure 74 IAR EWARM Project Options for the Secondary Project (Setup Macros)

iv. Click Debugger > Extra Options and add "--macro\_param cpu1\_enable=1" to Command line options: (one per line).



Options for node "Blinky_set Calegory: General Options * Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Output Converter Output Converter Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Litle GDB Server G+LINK 1-jet J-Link/J-Trace TI Stellaris	ary" ietup Download Images Multicore Authentication Extra Opt Diffee command line options? Command line options: (one per line) drv reset to c.pu start macro_param cpu1_enable=1	Factory Settings
Nu-Link PE micro	OK Cancel	· ·

Figure 75 IAR EWARM Project Options for the Secondary Project (Debugger Extra Options)

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

Celegony: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Cuttom Build Linker Build Actions Debugger Simulator CAD1 CMSIS DAP E2/E2 Lite GD8 Server G-LIDK J-Link/J-Trace TI Stellaris Nu-Link	Setup       Interface Trace       Breakpoints         Beset       Software       Image: Software         Override timing       200 ms       Delay after:       200 ms         Quration:       300 ms       Delay after:       200 ms         Target power       Emulator       Akwys prompt for probe selection         Switch off after debugging       Serial no:       Serial no:         Log communication       [\$PROJ_DIRSKcsprcomm.log	Factory Settings
PE micro v	OK Cancel	

Figure 76 IAR EWARM Project Options for the Secondary Project (Reset)



- vi. (CA55) Click I-jet > Interface, select From file in Probe config and select core in CPU of Probe Configuration file.
- vii. (CA55) Select General Options > 64-bit and select LP64 of Data model.
- viii. Proceed to 5.3.3.1 Build.
- ix. 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.

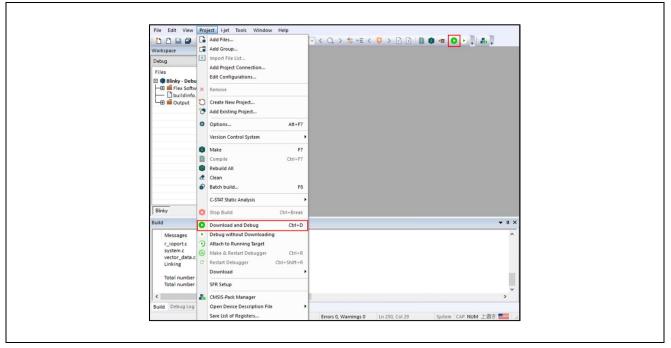


Figure 77 : Download and Debug Button



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

100 🖬 📾 🖴 🛝 🛍 Ö 🕸				
Workspace 👻	× main.c ×		Disassembly	▼ 4 >
Debug	v main()	fo	Go to:	
Files    Bilinky - Debug	<pre>1 /* generated moin source file - do not edit */ 2 #include "hal_data.h" 4 3 6</pre>		Disassembly ??DataTa @xGer: 0x0004'000 ??DataTa 0xGf8: 0x0028'1al ??DataTa 0xGf8: 0x0120'1aC ??DataTa 0xGf8: 0x0000'005 ??DataTa 0xGf8: 0x0000'005 ??DataTa 0x700: 0x0004'000 	30         D           able4_6         10           10         D           able4_7         30           able4_8         74           74         D           able4_9         50           50         D           able4_1         34           50         D           able4_1         56           56         M           56         M           56         M           56         M           57         M           58         M           59         M           50         M           50
Blinky			0x72c: 0xb580 hal_entry() 0x72e: 0xf7ff 0xf	

**Figure 78 : 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		• O Ein i T
Workspace	0	Break			Disassembly	
Debug	5	Reset		fo		
Files  Files Files Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  Files  File	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	??DataTable4_5 6ec: 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; }	Øxi Øxi Øxi	<pre>??DataTable4_6 6f0: 0x8028'1a10 D ??DataTable4_7 6f4: 0x8128'1a00 D ??DataTable4_8 6f8: 0x0000'0874 D ??DataTable4_9 6fc: 0x0000'0856 D ??DataTable4_1 700: 0x0004'0804 D in_init_vf0</pre>
		Memory Refresh Logging			0x 0x 0x 0x 0x 0x 0x 0x 0x 0x	704: 0xxe11'0f50 M 708: 0xe300'06f0 O 706: 0xe01'0f50 M 710: 0xe3a0'0601 M 714: 0xe07'0f95 M 
Blinky				>		72c: 0xb580 P hal_entry(); 72e: 0xf7ff 0xffqq R

Figure 79 : 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.

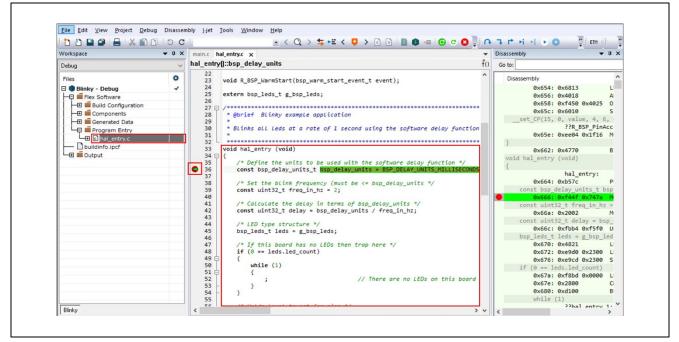


Figure 80 : hal\_entry.c and Setting Breakpoint



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.

inn 🖻 🖬 🚔 🗐 🕨	Go F5	💽 < Q, > ⇆ += < 📮 > d 🍺 📓 🖷 🕒 c 😫 🍹	10	3 2 3 3 3 5 0	ETH :
Workspace 0	Break	tryc x		Disassembly	- 
1	Reset	sp delay units	fo	Go to:	
Debug		op_deray_dinto	10	60 10:	
Files	Stop Debugging	<pre>d R_BSP_WarmStart(bsp_warm_start_event_t event);</pre>	^	Disassembly	
🗆 🌒 Blinky - Debug	Step Over F10			0x654: 0x6813	3 U
E Flex Software		ern bsp_leds_t g_bsp_leds;		0x656: 0x4018	
🕂 🖽 🖬 Build Configurati	Step Into F11			0x658: 0xf450	
E Components	Step Out Shift+F11	Obrief Blinky example application		0x65c: 0x6010	
	Next Statement			set_CP(15, 0, val	
Company Entry		Blinks all leds at a rate of 1 second using the software delay function	n		BSP_PinAcc
He hal entry.c	Run to Cursor			0x65e: 0xee04	0x1f16 M
	Autostep		*	}	
	Set Next Statement	d hal_entry (void)		0x662: 0x4776	
	Set Next Statement	/* Define the units to be used with the software delay function */		void hal_entry (void)	1
	C++ Exceptions	const bsp_delay_units_t bsp_delay_units = BSP_DELAY_UNITS_MILLISECONDS	s	{	
	C++ Exceptions				entry:
	Memory	/* Set the blink frequency (must be <= bsp_delay_units */		0x664: 0xb570	
		<pre>const uint32_t freq_in_hz = 2;</pre>		const bsp_delay_u	
	Refresh	/* Calculate the delay in terms of bsp_delay_units */		0x666: 0xf441	
	Logging	const uint32 t delay = bsp_delay_units / freq in hz;		const uint32_t fr	
	43	const dintsz_t delay - bsp_delay_dnits / ned_in_nz,		0x66a: 0x2002	
	44	/* LED type structure */		const uint32_t de	
	45	<pre>bsp_leds_t leds = g_bsp_leds;</pre>		0x66c: 0xfbb4	
	46			<pre>bsp_leds_t leds =</pre>	
	47	/* If this board has no LEDs then trap here */		0x670: 0x4821	Contract of the second second second
	48	<pre>if (0 == leds.led_count)</pre>		0x672: 0xe9d	
	50	while (1)		0x676: 0xe9co	
	51 🖻	{		if (0 == leds.led	
	52	; // There are no LEDs on this board		0x67a: 0xf8bo	
	53 -	}		0x67e: 0x2800	
	54 -	}		0x680: 0xd100	ð B
	55			while (1)	

Figure 81 : 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
  - RZ/N2H Evaluation Board: LED3 (CR52 CPU0), LED4 (CR52 CPU1), LED8 (CA55 Core0)

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



### Figure 82 IAR EWARM Debugger Pause Icon



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



#### Figure 83 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_pri Category: Ceneral Options A Static Analysis Runtme Checking C/C+L+Compiler Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CADI CADI CMSIS DAP E2/E2 Lite G05 Server G+LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link	Factory Settings         Setup Download Images Multicore         Authentication Extra Options Plugins         Symmetric multicore         Number of cores:         Disabled         Simple         Partner groject:         Blinky_secondary         Partner groject:         Blinky_secondary         Partner debugger location         Partner debugger:
PE micro v	O Adyanced Session configuration:

Figure 84 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 75.
- 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 77 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 SC with an existing project.

## 5.4.1 Launch FSP SC 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\$

Canflining Table		
Configure Tools		
Menu Content:		
FSP Smart Configurator	Ok	1
	Cancel	1
	New	1
	<u>D</u> elete	
Menu <u>T</u> ext:		
FSP Smart Configurator		
<u>C</u> ommand:		
\$RASC_EXE_PATH\$	Browse	
Argument:		
compiler IAR configuration.xml		
Initial Directory:		
\$PROJ_DIR\$		
Redirect to Output Window		
Prompt for Command Line		
Tool Available:		
Always 🗸		

Figure 85 : 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 SC 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

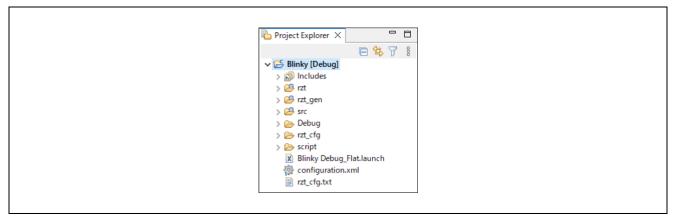
# 6.1 What is a Project?

In e<sup>2</sup> 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 86 : e<sup>2</sup> 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 e<sup>2</sup> 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.

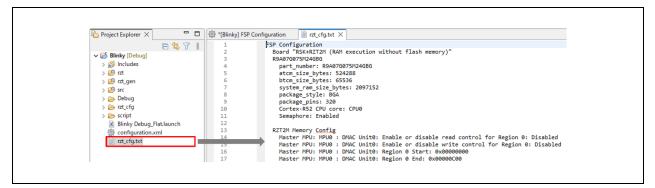


Figure 88 : RZ/T2, RZ/N2 Project Report

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 e<sup>2</sup> 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 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           Board support package for           Board support package for RZT2M</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 information="" version=""> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></components></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> Board support package for RZT2M</device></board>	<components information="" version=""></components>

Figure 89 : 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 90 : 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	Project			×
Templates f	r Renesas RZ Project			
All C/C++	Renesas RZ/N C/C++ FSP Project Create an executable or static library C/C- project for Renesas RZ/N. Renesas RZ/T C/C++ FSP Project Create an executable or static library C/C- project for Renesas RZ/N. Renesas RZ/V C/C++ FSP Project Create an executable or static library C/C- project for Renesas RZ/V. <	+ FSP		^ >
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cancel	

**Figure 91 : New Project Templates** 



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

Renesas RZ/T C/C++ FSP Project	×
Renesas RZ/T C/C++ FSP Project Project Name and Location	Ź
Project name Blinky	
Use default location	vse
You can download more Renesas packs here	
< Back         Next >         Finish         Can	ncel

Figure 92 : 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.

Renesas RZ/T C/C++ FSP Project	— 🗆 X
Renesas RZ/T C/C++ FSP Project	
Device and Tools Selection	
Device Selection	
	Board Description RZT2H Evaluation Board (RAM execution without flash memory)
Board: RZT2H Evaluation Board (RAM execution witl ~	K212H Evaluation Board (KAW execution without hash memory)
Device: R9A09G077M44GBG	
Core: CR52 CPU0	Device Details TrustZone No
Language: O C O C++	Pins 729
	Processor Cortex-R52
L	
	Debugger
GNU ARM Embedded GCC ARM A-Profile (AArch64 bare-metal)	J-Link ARM 🗸 🗸
<toolchain version=""> V Manage Toolchains</toolchain>	
?	< Back Next > Einish Cancel

Figure 93 : 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	imart Bundle Selection	2
() None	Choose this option when creating a project for the primary processor core (no preceding project or Smart Bundle)	
Preceding Project:	Blinky_primary	~
	Choose this option if you have access to the project source code of the preceding processor core or security con	text.
O Smart Bundle:		
	Resolved location:	
	Workspace File System Variabl	
	Choose this option if you only have access to a Smart Bundle describing the configuration of the preceding proc core or security context.	essor
Preceding Project/Sn	art Bundle Details	
FSP version	<fsp version=""></fsp>	
Toolchain	GNU ARM Embedded	
Toolchain version Board	<toolchain version=""> <board (boot="" mode)=""></board></toolchain>	
Device	R9A09G077M44GBG	
Core	CR52_0	
Zones	CR52_0	
?	< <u>Back</u> <u>Next</u> > Einish Can	cel
•	- Text Text - Turn Con	

Figure 94 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	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 and RTOS Selection       Build Artifact Selection	Build Artifact and RTOS Selection Build Artifact Selection Build Artifact Selection Build Artifact Selection Build Artifact Selection No RTOS Selection No RTOS	Build Artifact and RTOS Selection       Build Artifact Selection       Image: Constraint of the selection       Image: Constrese of the	🕲 Renesas RZ/T C/C++ FSP Project	—	×
Build Artifact Selection       Build Artifact Selection	Build Artifact Selection       Build Artifact Selection	Build Artifact Selection       Build Artifact Selection	Build Artifact Selection       Build Artifact Selection	Build Artifact Selection       Build Artifact Selection	Build Artifact Selection	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	Executable     Project builds to an executable file     Static Library	Executable     Project builds to an executable file     Static Library	<ul> <li>Executable</li> <li>Project builds to an executable file</li> <li>Static Library</li> <li>Project builds to a static library file</li> </ul>	Build Artifact and RTOS Selection		
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 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 95 : 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: Selection	Project Template Selection  Project Template Selection   Remeal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C nutrime environment.  [Remeass.RZT
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 number of the clock and the c	Project Template Selection
Bare Metal - Blinky Bare metal FSP project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment. Renessas RZTpack]      Bare Metal - Minimal Bare metal FSP project that includes BSR This project will initialize clocks, pins, stacks, and the C runtime 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. Remess.RZT
Bare Metal - Minimal Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment.	Image: Second

Figure 96 : RZ/T2, RZ/N2 MPU Project Generator (Part 5)



When the project is created, e<sup>2</sup> studio displays a summary of the current project configuration in the RZ/T2, RZ/N2 MPU Project Editor.

Project Summary       Renessa         Board:       Device:         Toolchain:          Project Type:          Location:          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       FSP Data</device></board>	Summary	Generate Project Content
Board:       Device:         Device:       Toolchain:         Toolchain Version: <project information="" summary="">         FSP Version:       Project summary information&gt;         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>          VD Port          <boardsupport <device="" for="" package="">          Board support package for <device>          Board support package for RZT2M</device></boardsupport></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> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></board>	Device: Toolchain: Toolchain Version: FSP Version: Project Type:	
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>	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>	<components information="" version=""></components>

Figure 97 : 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 Flexible Software Package Documentation (<u>RZT</u>, <u>RZN</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&gt;            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> Arm CMSIS Version 5 - Core Board support package for <device> Board support package for RZT2M</device></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	onents		
		Board Support Packag Memory Config Check I/O Port <board> Arm CMSIS Version 5 Board support packag Board support packag</board>	e Common Files ing - Core a for <device> a for RZT2M</device>	<components information="" version=""></components>	· · · · · · · · · · · · · · · · · · ·

Figure 98 : 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.

alle formel	] FSP Configuration ×		
Board	Support Package Configura	ration O Generate Project Cont	ent
		🔯 Restore Defaults	^
Device	Selection		
FSP ve	ersion: <pre></pre>	Board Details Renesas Starter Kit+ for RZ/T2M CPU Board (RAM execution	
Board	RSK+RZT2M (RAM execution v		
Devic	e: R9A07G075M24GBG		
Core:	CR52_0	~	
RTOS:	No RTOS	$\sim$	
			×
Summary	BSP Clocks Pins Interrupts Event L	Links Stacks Components	~
	BSP Clocks Pins Interrupts Event L ties X 🖹 Problems 🏟 Smart Brow		
Proper		wser 🔀 🖬 🖓 🖁 🗖	
Proper	ties 🗙 🔝 Problems 🏟 Smart Brow	wser 🔀 🖬 🖓 🖁 🗖	
Proper RSK+RZ	ties X Problems & Smart Brow	wser 📑 🐨 🖓 🖇 🖻 t flash memory)	
Proper RSK+RZ	ties X Problems Smart Brow	wser 📑 🐨 🖓 🖇 🖻 t flash memory)	
Proper RSK+RZ	Problems Smart Brow     RAM execution without     Property     R9A076075M24GBG	wser Transformer T	
Proper RSK+RZ	Problems Smart Brow     TZM (RAM execution without     Property     R9A076075M24GBG     part_number	wser T Te T III A S IIII A S III A S IIII A S IIIII A S IIIIIII A S IIIII A S IIIIII A S IIIII A S IIIII A S IIIII A S IIIIII A S IIIII A S IIIIII A S IIIIIII A S IIIIIII A S IIIIIII A S IIIIIIII	
Proper RSK+RZ	Property Proper	wser T Te T III & S III A S IIII A S IIIII A S IIIIII A S IIIIIIII	
Proper RSK+RZ	Problems Smart Brow     TZM (RAM execution without     Property     Property     PRA07G075M24GBG     part_number     atcm_size_bytes     btcm_size_bytes	wser	
Proper RSK+RZ	<ul> <li>Caracterization (Caracterization)</li> <li>Property</li> <li>R9A07G075M24GBG</li> <li>part_number</li> <li>atcm_size_bytes</li> <li>bytes_bytes</li> <li>system_ram_size_bytes</li> </ul>	wser	
Proper RSK+RZ	<ul> <li>Problems Smart Brow</li> <li>Property</li> <li>PatorGo75M24GBG</li> <li>part_number</li> <li>atcm_size_bytes</li> <li>btcm_size_bytes</li> <li>system_ram_size_bytes</li> <li>package_style</li> </ul>	vser	

**Figure 99 : 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. e<sup>2</sup> 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.

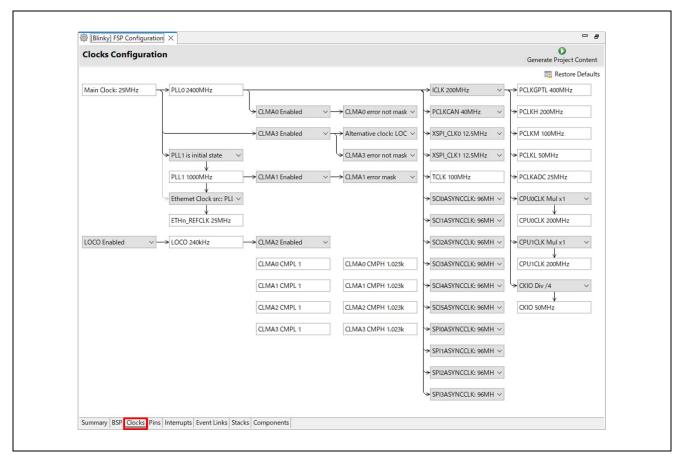
#### Warning:

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 100 : 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.

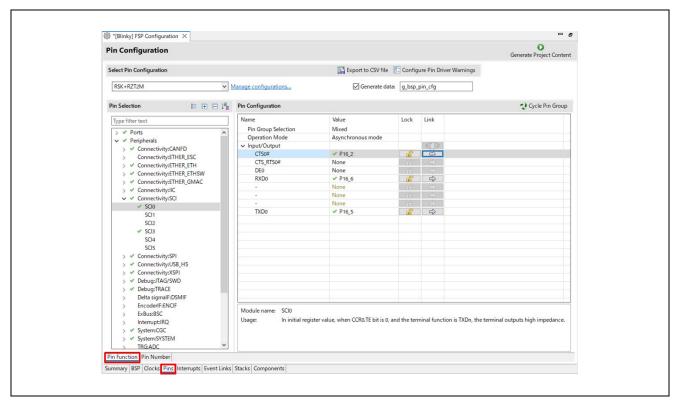


Figure 101: Pin Configuration



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.

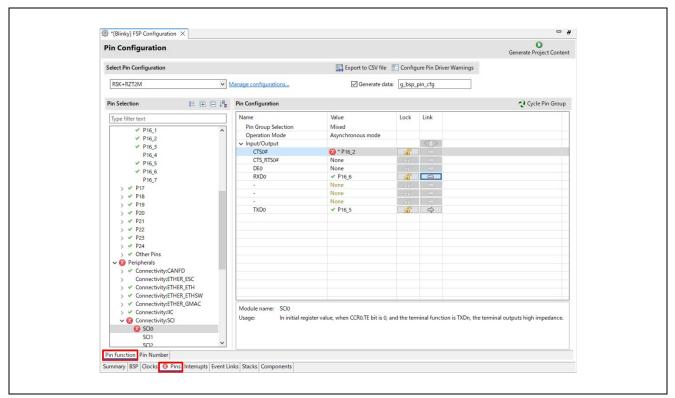


Figure 102: Conflict Checker in Pin Configuration



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

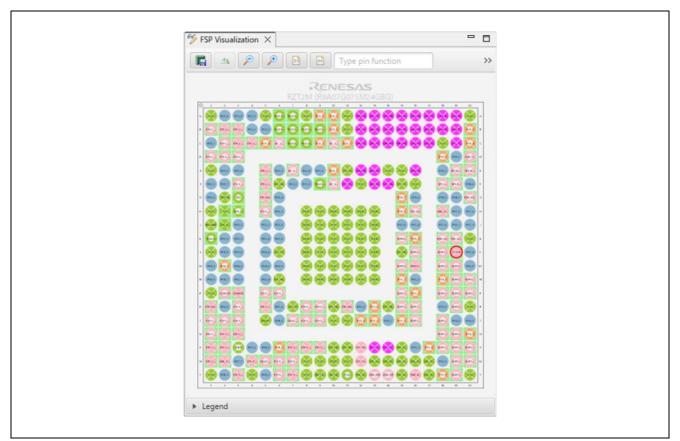


Figure 103: Pin Configurator Package View

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.

	y] FSP Configuration X 📄 rzt_cfg.txt		0	- 0
Stacks	Configuration		Generate Project C	ontent
Threads	🐑 New Thread 🔬 Remove 📄	HAL/Common Stacks	🕢 New Stack > 🚔 Extend Stack > 👔 Re	move
(	HAL/Common	Image: g_ioport I/O Port (r_ioport)       Image: g_ioport I/O Port (r_ioport)	Memory config check	
Objects	🐑 New Object > 🙀 Remove			
Summary	BSP Clocks Pins Interrupts Event Links Stacks	Components		
🔲 Prope	ties 🗙 🔝 Problems 🏶 Smart Browser		📑 🖬 🎖 🖾 🔗 🏛 🎖 🖾 🔗 🖩 🎖 🖾 🔗 🛔	- 0
g_time	0 Timer, Compare Match W (r_cmtw)			
	Property	Value		^
Settings	✓ Common	Value		
	Parameter Checking	Default (BSP)		
	Multiplex Interrupt	Disabled		
	<ul> <li>Module g_timer0 Timer, Compare Match W (r_</li> </ul>			
	> General	cintur,		
	> Output			
	> Input			
	✓ Interrupts			
	Callback	NULL		
	Compare Match Interrupt Priority	Priority 11		
	Compare Match Interrupt Priority Input Capture 0 Interrupt Priority	Priority 11 Disabled		
	Compare Match Interrupt Priority Input Capture 0 Interrupt Priority Input Capture 1 Interrupt Priority			
	Input Capture 0 Interrupt Priority	Disabled		
	Input Capture 0 Interrupt Priority Input Capture 1 Interrupt Priority	Disabled Disabled		
	Input Capture 0 Interrupt Priority Input Capture 1 Interrupt Priority Output Compare 0 Interrupt Priority	Disabled Disabled Disabled		
	Input Capture 0 Interrupt Priority Input Capture 1 Interrupt Priority Output Compare 0 Interrupt Priority Output Compare 1 Interrupt Priority	Disabled Disabled Disabled		
	Input Capture 0 Interrupt Priority Input Capture 1 Interrupt Priority Output Compare 0 Interrupt Priority Output Compare 1 Interrupt Priority	Disabled Disabled Disabled Disabled		Ţ

Figure 104 : 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 > 🔬 F	emove
Event		ISR	
		ISR	
Interrupt 59	Event CMTW0_CMWI (CMTW0 Compare match)	cmtw_cm_int_isr	
Summary BSP	Clocks Pins Interrupts Event Links Stacks Components		

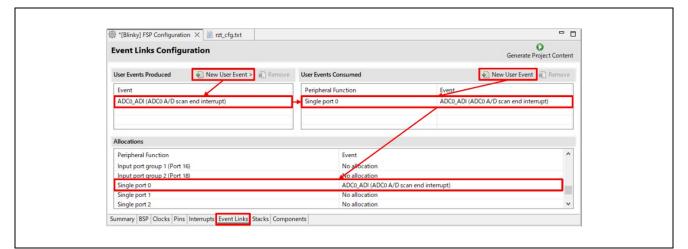
Figure 105 : Configuring Interrupt in Interrupt Tab

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 106 : 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.

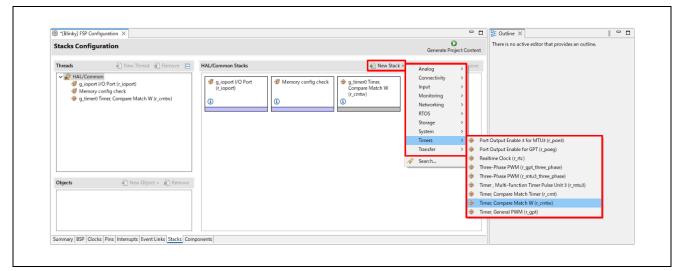


### RZ/T2, RZ/N2

# 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 107 : e<sup>2</sup> 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.

e<sup>2</sup> 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 19 Generate Contents on FSP Configuration**

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.

鬱 *[Blinky] FSP Configuration × 📄 rz	t_cfg.txt		- E
Components Configuration			Generate Project Content
□ □ ◆16		Group by: Vendor 🗸 Filter All	V Search
Component	Version	Description	Variant
✓ 🔐 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	
V 🖗 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
✓ 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			
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 B	200	Poard support package for POA07G07EM26GPG	PoAo7G07EM26GPG

Figure 108 : 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>

e<sup>2</sup> 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

FSP Configuration on  $e^2$  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 on  $e^2$  studio and FSP SC, please see the chapter 6. "FSP Configuration Users Guide".

o FSP Modules

The FSP provides HAL drivers and BSP configured by FSP Configuration on e<sup>2</sup> studio and FSP SC. Regarding their features, usage notes and API references, please see the related file "FSP Documentation".

No.	Title	Targe	t Devic	e				Category
		T2M	T2L	T2ME	T2H	N2L	N2H	
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	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 20 List of Known Issues



No.	Title	Target	Category					
		T2M	T2L	T2ME	T2H	N2L	N2H	
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		FSP Configuration, Stacks
14	The second argument of "r_mtu3" APIs do not match with common API.	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				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



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

No.	Title	Target	Device					Category
		T2M	T2L	T2ME	T2H	N2L	N2H	1
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	1	1	FSP Modules, FreeRTOS+F AT
29	Caution when specifying program placement in linker scripts.				1		1	Others, Linker script
30	In the secondary project for multiprocessing, no error occurs when there is a conflict in a resource used with the preceding project.				1		1	FSP Configuration, Stacks
31	Errors occur when setting ELC in r_gpt module.				1		1	FSP Configuration, Stacks
32	CR52 CPU1 of RZ/T2H and RZ/N2H is implemented to start programs from System SRAM instead of CPU1 ATCM.				1		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		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		1	Others, Address space
35	r_gmac_b module cannot use zero- copy mode.				1		1	FSP Modules, GMAC
36	r_adc module does not support the calibration function.				1			FSP Modules, ADC
37	The USB driver for CA55 project does not work.				1		1	FSP Modules, USB
38	No error returns when entering the virtual addresses that cannot be translated to physical addresses as arguments.				1		1	FSP Modules, xSPI_OSPI, xSPI_QSPI, DMAC
39	The CA55 project with noncache sections aborts when debugging with flash boot mode on IAR EWARM.				1		1	FSP Modules, BSP
40	When changing the duty setting in r_gpt module, there is a possibility the duty may unintentionally become 100%.	1	1	1	1	1	1	FSP Modules, GPT



No.	Title	Target	Device					Category
		T2M	T2L	T2ME	T2H	N2L	N2H	
41	CPU registers save and restore process cannot be performed correctly in FIQ_Handler for CA55 projects.				1		1	FSP Modules, BSP, FreeRTOS
42	MTU3 callback does not occur as expectation.				1		1	FSP Modules, MTU3
43	An undefined error of r_gpt module occurs when building a project.				1		1	FSP Modules, GPT
44	Pin names according to unit and channel numbers are not displayed in r_gpt module configurations.	1	1	1	1	1	1	FSP Configuration, Stacks
45	Using R_GPT_DutyCycleSet() with option both pins A and B cannot work properly.	~	~	~	~			FSP Modules, GPT
46	Parameter checking of R_ETHER_SELECTOR_Open() is not working properly.	~	~	~	~	~	✓	FSP Modules, ETHER_SEL ECTOR
47	Parameter checking feature of R_GMAC_CallbackSet() is not working.	~	~	~	~	~	√	FSP Modules, ETHER_GM AC
48	r_usb_hhid module is not working properly.	~	$\checkmark$	$\checkmark$	$\checkmark$			FSP Modules, USB_HHID

### 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, RZ/N2H
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. 3 Resolved

#### 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" Category: Cate
	0K Cancel



Target       RZ/T21, RZ/N21.         Category       FSP Configuration, BSP         Description       Errors occur when changing board settings from RSK+RZN21 (RAM execution without flash memory) to RZN21. Custom User Board (RAM execution without flash memory) to RZN21. Custom User Board (RAM execution without flash memory) to RZN21. Custom User Board (RAM execution without flash memory) to RZN21. Custom User Board (RAM execution without flash memory) to RZN21. Custom User Board (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Changed from RSK+RZN21 (RAM execution without flash memory)         1.       Interview (Rame and the state of the state	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 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.         Description         Reference of the comparison of the compariso	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 (xSP10 x1 boot mode).</pre> 1. 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 screen is displayed after the build.  Transformer in the successful is the top of the successful is the following screen is displayed after the build.  Transformer is the successful is the successful is the following screen is displayed after the build.  Transformer is the successful is the successful is the following screen is displayed after the build.  Transformer is the successful is	Category	FSP Configuration, BSP								
Board (xSPI0 x1 boot mode)         bsp_mcu_device_pn_cfg.h is not generated and builds error occurs.         Properties       Problems × Image: Second and Seco		<ul> <li>memory) to RZN2L Custom User Boz RSK+RZN2L (RAM execution witho boot mode).</li> <li>Changed from RSK+RZN2L (RA Board (RAM execution without ff In this case, the build is successful. But</li> <li>Propetie Problems Smat Browser Console X Memory COT Build Console [rm2], minimal</li> <li>Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cache.c Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/ran2/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_cader_param. Building file:/ran/fsp/src/bsp/mcu/all/bsp_file.c /ran/fsp/src/bsp/mcu/all/bsp_file.c /ran/fsp/src/bsp/mcu/all/bsp_file.c /ran/fsp/src/bsp/mcu/all/bsp_file.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_file.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu/all/bsp_sp-kc.c Building file:/ran/fsp/src/bsp/mcu</li></ul>	c tion.c /startup.c /system.c imal.srec" (RAM execution lash memory) lash memor	cution with ory) to RZN without fla: og screen is cution	iout fla I2L Cu: sh mem display red a problem. uild. c c c er on project ug/rzn/board/rzn/ rzn/board/rzn/	sh memory) and fr stom User Board ( nory) to RZN2L C ved after the build.	om xSPI0 x1 ustom Us			
□ Properties [*]       Problems × → Smart Browser       □ Console         6 errors, 18 warnings, 227 othes (Filter matched 124 of 251 items)         Description       Resource       Path       Location       Type         • ● frors (6 items)       Path       Location       Type         • ● fatal error. bis_mcu_device, pn_cfg.h: No such file or directory       bsp_mcu_family_cfg.h       /rzn2L_minimal       C/C++ ProbL         • ● make: *** [ran_ger/subdir.mb<22: ran_ger/normon_data.o] Error 1       rzn2L_minimal       C/C++ ProbL       C/C++ ProbL         • ● make: *** [ran_ger/subdir.mb<22: ran_ger/main.o] Error 1       rzn2L_minimal       C/C++ ProbL         • ● make: *** [ran_ger/subdir.mb<22: ran_ger/main.o] Error 1       rzn2L_minimal       C/C++ ProbL         • ● make: *** [ran_ger/subdir.mb<22: ran_ger/main.o] Error 1       rzn2L_minimal       C/C++ ProbL										
6 errors, 18 warnings, 227 others (Filter matched 124 of 251 items)         Description       Resource       Path       Location       Type         ♥ 2 Errors (6 items)       Image: Transperiod (and the ordirectory)       bsp.mcu_family_cfg.ht       /rzn2_minimal/rzn       line 4       C/C++ Probl         ● make: ***[rran_gen/subdir.mik-42: ran_gen/rank_d_ata.o] Error 1       rzn2_minimal       C/C++ Probl       C/C++ Probl         ● make: ***[rran_gen/subdir.mik-42: ran_gen/main.o] Error 1       rzn2_minimal       C/C++ Probl       C/C++ Probl         ● make: ***[rran_gen/subdir.mik-42: ran_gen/main.o] Error 1       rzn2_minimal       C/C++ Probl       C/C++ Probl         ● make: ***[ran_gen/subdir.mik-42: ran_gen/main.o] Error 1       rzn2_minimal       C/C++ Probl       C/C++ Probl         ● make: ***[ran_gen/subdir.mik-42: ran_gen/main.o] Error 1       rzn2_minimal       C/C++ Probl       C/C++ Probl										
Description     Resource     Path     Location     Type <ul> <li>             £ trors (6 items)</li> </ul> <ul> <li>             • fatal error: bsp.mcu.device.pn_cfg.ht: No such file or directory</li> <li>             bsp.mcu.family_cfg.ht</li>             / rzn2_minimal/rzn             line 4             C/C++ ProbL </ul> enake: ****[rzn_gen/subdir.mb42: rzn_gen/common.data.o] Error 1             rzn2_minimal <ul>             C/C++ ProbL             C/C++ ProbL             C/C++ ProbL             C/C++ ProbL             C/C++ ProbL             enake: ***[rzn_gen/subdir.mb42: rzn_gen/main.o] Error 1             zrn2_minimal </ul>										
Image: State error: bsp.mcu_device_pn_cfg.h: No such file or directory       bsp_mcu_family_cfg.h       /rzn2l_minimal/rzn       line 4       C/C++ Probl         Image: State error: Irzn_gen/subdir.mck2: rzn_gen/common_data.o] Error 1       rzn2l_minimal       C/C++ Probl       C/C++ Probl         Image: State error: Irzn_gen/subdir.mck2: rzn_gen/hal_data.o] Error 1       rzn2l_minimal       C/C++ Probl         Image: State error: Irzn_gen/subdir.mck2: rzn_gen/hal_data.o] Error 1       rzn2l_minimal       C/C++ Probl         Image: State error: Irzn_gen/subdir.mck2: rzn_gen/hal_gen/hal_data.o] Error 1       rzn2l_minimal       C/C++ Probl         Image: State error: Irzn_gen/subdir.mck2: rzn_gen/hal_gen/ha		Description	Resource	Path	Location	Туре				
Image: *** [rzn_gen/subdir.mk-42: rzn_gen/common.data.o] Error 1         rzn2[_minimal         C/C++ Probl           Image: *** [rzn_gen/subdir.mk-42: rzn_gen/main.o] Error 1         rzn2[_minimal         C/C++ Probl           Image: *** [rzn_gen/subdir.mk-42: rzn_gen/main.o] Error 1         rzn2[_minimal         C/C++ Probl           Image: *** [rzn_gen/subdir.mk-42: rzn_gen/main.o] Error 1         rzn2[_minimal         C/C++ Probl           Image: *** [rzn_gen/subdir.mk-42: rzn_gen/main.o] Error 1         rzn2[_minimal         C/C++ Probl			bsp. mcu family, cfa.b	/rzn2l_minimal/rzn	line 4	C/C++ Probl				
Image: *** [rzn_gen/subdir.mlc42: rzn_gen/main.o] Error 1         rzn2Lminimal         C/C++ Probl           Image: *** [src/subdir.mlc26: src/hal_entry.o] Error 1         rzn2Lminimal         C/C++ Probl				/iziizi_nimmai/izh.						
make: *** [src/subdir.mk:26: src/hal_entry.o] Error 1 rzn2l_minimal C/C++ Probl		Make: *** [rzn_gen/subdir.mk:42: rzn_gen/hal_data.o] Error 1	rzn2l_minimal							
<b>Workaround</b> Please create a new project to change the board setting to RZN2L Custom User Board.										

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



Title	Pin configuration er	ror occurs in M	PX-IO 16bi	it ope	erating	g mode of "	r_bsc'		
Target	RZ/T2M, RZ/T2L, 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	river Warnings			
	RSK+RZN2L	<ul> <li>Manage configurations.</li> </ul>		erate dat	ta: g_bsp	_pin_cfg			
	Pin Selection $\models \oplus = \downarrow^{a}_{Z}$	Pin Configuration				Cycle Pin	Group		
	bsc X	Name	Value	Lock	Link		^		
	✓ ✓ Ports ✓ ✓ Other Pins ✓ BSCANP	Pin Group Selection	Mixed						
		Operation Mode	MPX-IO 16bit		1000				
		✓ Input/Output			<□>		_		
	✓ ❷ Peripherals	A0 A1	None 8 * None	(T)					
	✓ <sup>©</sup> ExBus:BSC	A1 A2	None None				_		
	😣 BSC	A3	✓ P05 1	ď	4				
		A4	* None		⇒		~		
		<					>		
	Module name: BSC								
	Pin Function Pin Number								
	Summary BSP Clocks <sup>©</sup> 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	ab 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       a unused variable 'tmp' [-Wunused-variable]         49       uint16_t tmp = ETH0_RXD0:         50	
	Properties       Problems       Console ×       Search       Image: Console ×	
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	

### 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 <b>default</b> 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.
	<ul> <li>All check boxes are enabled.</li> </ul>



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 vkplicationTRQHandler (wint32_t ullCCIAR) { f 0 /* Seremable interrupt. */ /* Seremable interrupt. */ /* seremable interrupt. */ eendif bsp_common_interrupt_handler (wint32_t id) { void bsp_common_interrupt_handler (wint32_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 Resolved for RZ/T series devices in RZT FSP v3.0.0

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, RZ/N2H
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
	<pre>g_timer0.p_api-&gt;periodSet() g_timer0.p_api-&gt; InfoGet() g_timer0.p_api-&gt; StatusGet()</pre>
	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 and 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.		
Workaround	<ul> <li>When using "r_poe3" and MTU0-B/MTU6/MTU7 as MTU3 output pin, please follow these four steps:</li> <li>1. Add "Port Output Enable 3 for MTU3 (r_poe3)" on Stacks tab of FSP Configuration.</li> <li>2. Click Generate Project Content button and "r_poe3" code is generated.</li> <li>3. Disable code generating function. After this setting, the code cannot be generated.</li> <li>[For e<sup>2</sup> 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.</li> <li>a. Uncheck "Project Properties &gt; Builders &gt; DDSC Builder"</li> </ul>		
	<ul> <li>Properties for</li> <li>Properties for</li> <li>Uppe filter text</li> <li>Resource Builders</li> <li>C/C++ Build</li> <li>C/C++ Build</li> <li>MCU</li> <li>Project Natures</li> <li>Project References</li> <li>Reneasa QE</li> <li>Run/Debug Settings</li> <li>Task Tags</li> <li>Validation</li> </ul>		
	Apply and Close Cancel      Apply and Close Cancel      Men unchecking it, the following message appears: click OK.      Confirm Disable Builder      M      This is an advanced operation. Disabling a project builder can have many side-effects. Continue?      OK Cancel		
	<pre>[For FSP SC (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. 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_pin_select = POE3_PIN_SELECT_0, .hiz_output_enable = false },</pre>		
	<pre>/* Setting structure for complementary <u>pwm</u> pin. */ static const poe3_complementary_pwm_setting_t g_poe30_complementary_pwm_setting[] = {     {         .pin_setting[0] =         {         .pin_setting[0] =     } }</pre>		

# No. 18 Resolved



# RZ/T2, RZ/N2

.pin_sc { .posit .negat .negat	<pre>{ .positive_pwm_pin_select = POE3_PIN_SELECT_0,      .pin_setting[X] = { .positive_pwm_pin_select = POE3_PIN_SELECT_0,     .negative_pwm_pin_active_level = POE3_ACTIVE_LEVEL_SETTING_NONE,     .negative_pwm_pin_active_level = POE3_ACTIVE_LEVEL_SETTING_NONE,     .hiz_output_enable = false },    </pre>		
For MTU	0-B		
MTU3	Port	Location (Struct [module name]	Replace with
output pi	n	pwm pin setting[])	-
mtioc0b	P14_4	the second pwm_pin_select	<pre>.pwm_pin_select = POE3_PIN_SELECT_1</pre>
mtioc0b	P24_0	the second pwm_pin_select	<pre>.pwm_pin_select = POE3_PIN_SELECT_2</pre>
mtioc0b	P13_3	the second pwm_pin_select	.pwm_pin_select = POE3_PIN_SELECT_3
For MTU	6/MTU7		
MTU3	Port	Location (Struct [module	Replace with
output pi		name]_complementary_pwm_setting[])	.positive pwm pin select
mtioc6b	P21_2	the second pin_setting[0]	= POE3_PIN_SELECT_1,
mtioc6b	P08_5	the <b>second</b> pin_setting[0]	<pre>.positive_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
mtioc6d	P21_4	the <b>second</b> pin_setting[0]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_1,</pre>
mtioc6d	P08_7	the second pin_setting[0]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
mtioc7a	P21_5	the second pin_setting[1]	<pre>.positive_pwm_pin_select = POE3_PIN_SELECT_1,</pre>
mtioc7a	P09_0	the second pin_setting[1]	<pre>.positive_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
mtioc7c	P21_7	the <b>second</b> pin_setting[1]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_1,</pre>
mtioc7c	P09_2	the <b>second</b> pin_setting[1]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
mtioc7b	P21_6	the second pin_setting[2]	<pre>.positive_pwm_pin_select = POE3_PIN_SELECT_1,</pre>
mtioc7b	P09_1	the <b>second</b> pin_setting[2]	<pre>.positive_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
mtioc7d	P22_0	the second pin_setting[2]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_1,</pre>
mtioc7d	P09_3	the second pin_setting[2]	<pre>.negative_pwm_pin_select = POE3_PIN_SELECT_2,</pre>
Note: The first.	ere are two pir	n_setting[X] 's with the same number(X);	



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 generation, values of DSMIFn_ERR additional trigger for the module are generated in rzt_gen/hal_data.c. But the defined values are not defined in FSP, so a build error occurs. Property of r_poe3 stack				
	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)				
	<ul> <li>Additional MTU0 pin control request condition (Always enable POE8# Input)</li> </ul>				
	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				
	CDT Build Console [rzt2]		💥 🤑 👉 💁 📑 🖬 📻 🐘 🛃 📹 🖬 🗸 🖶 🗖		
	/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_DE0ADDMT02	E Pos)   (1U << R	POE3 POECR5 DE1ADDMT0ZE Pos)   0U),		
	ANNNNNNNNNNNNNNNNNNNNNNNN	INNININ			
	<pre>R_POE3_POECR5_D0F0ADDMT02E_POS fix-it:/rzt_gen/hal_data.c":{70:40-70:69}:"R_POE3_POECR5_D0F0ADDMT02E_Pos"/rzt_gen/hal_data.c:70:80: error: 'R_POE3_POECR5_D0E1ADDMT02E_Pos' 70   .mtu@_control_channel_mask = ((10 &lt;&lt; R_POE3_POECR5_D0E0ADDMT02E_Pos)   (10 &lt;&lt; R_POE3_POECR5_D0E0ADDMT02E_Pos)   00),</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				
			POE3_POECR5_D0E0ADDMT0ZE_Pos		
	• R_POE3_POECR5_DE1ADDMT0ZE_Pos to R_POE3_POECR5_D0E1ADDMT0ZE_Pos				
	When using DSMIF 0 ERROR, DSMIF 1	ERROR for	· MTU3/4 additional trigger, modify:		
	R POE3 POECR4 DE0ADDMT34Z				
	to R POE3 POECR4 D0E0ADDMT				
	R_POE3_POECR4_DE1ADDMT34ZE_Pos to				
	R_POE3_POECR4_D0E1ADDMT34	ZE_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	8		
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.			
	Second Category	utput Enable 3 for MTU3 (r_poe3) of St 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. <ul> <li>MTU3 and MTU4 Pin Control</li> <li>MTU3-B (MTIOC3B) and MTU3-D (MTIOC3D)</li> <li>MTU4-B (MTIOC4B) and MTU4-D (MTIOC4D)</li> <li>Use this field to configure for MTU4-A and MTU4-C</li> <li>MTU4-A (MTIOC4A) and MTU4-C (MTIOC4C)</li> <li>Additional MTU3/4 pin control request condition (Always enable POE0# Input)</li> <li>MTU6-B (MTIOC5B) and MTU6-D (MTIOC5D)</li> <li>MTU6-B (MTIOC7B) and MTU6-D (MTIOC5D)</li> <li>MTU7-B (MTIOC7B) and MTU7-D (MTIOC7C)</li> <li>Use this field to configure for MTU7-A and MTU7-C</li> <li>MTU7-A (MTIOC7A) and MTU7-C (MTIOC7C)</li> <li>Use this field to configure for MTU7-B and MTU7-D</li> <li>Additional MTU6/7 pin control request condition (Always enable POE4# Input)</li> </ul>			

#### No. 20 Resolved

# No. 21 Resolved

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 size is 4 bytes.) .data_noncache		
	0x30190005 0x23 ./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. 22 Resolved

### No. 23 Resolved

Issue	Initial values of data placed in some sections were overwritten with 0.		
Target	RZ/N2L		
Category	FSP Modules, BSP		
Description	<ul> <li>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.</li> <li>.dmac_link_mode</li> <li>.shared_noncache_buffer</li> <li>.noncache_buffer</li> </ul>		
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	<ul> <li>When selecting a flash boot mode as "Board" in BSP tab of FSP Configuration, variables placed in the following sections are NOT initialized.</li> <li>Boards for flash boot mode</li> <li>XXXXX (xSPI0 x1 boot mode)</li> </ul>		
	<ul> <li>XXXXX (16-bit bus NOR flash boot mode)</li> <li>RZN2L Custom User Board (xSPI0 x8 boot mode)</li> <li>RZN2L Custom User Board (xSPI1 x1 boot mode)</li> <li>Sections</li> <li>.dmac_link_mode</li> </ul>		
Workaround	<ul> <li>.shared_noncache_buffer</li> <li>.noncache_buffer</li> <li>Please initialize the variables placed in the above sections in the user application.</li> </ul>		

### No. 24 Resolved

#### No. 25 Resolved

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. When setting as below, build errors will occur.		
	Property of r_poeg stack		
	<pre>     Module g_poeg0 Port Output Enable for GPT (r_poeg)     General     V General     GETERGE Fin     GETERGE Fin     GET Output Level     Oscillation Stop     DSMF1 error     DSMF0 error     DSMF1 err</pre>		
	<pre>make -routput-sync -j8 all/rzt_gen/hal_data.c:7:36: 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_DERR0E_1   POEG_TRIGGER_DERR1E_1   POEG_TRIGGER_SOFTWARE),</pre>		
	<pre>fix-it:"/rzt_gen/hal_data.c":{7:36-7:57}:"POEG_TRIGGER_DERR0E"/rzt_gen/hal_data.c:7:60: error: 'POEG_TRIGGER_DERR1E_1' undeclared here (not in a function); did you mean 'POEG_TRIGGER_DERR1E'? 7   POEG_TRIGGER_PIN   POEG_TRIGGER_DERR0E_1   POEG_TRIGGER_DERR1E_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		



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:			
	POEG_TRIGGER_DERR0E_1 = 1U << 18,			

#### No. 26 Resolved

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 Resolved

Issue	Missing constraint for DSMIF error trigger in channel 1 and channel 2.		
Target	RZ/T2M, RZ/T2ME, RZ/T2L, RZ/T2H		
Category	FSP M	Iodules, POEG	
Description	current	tly there are no constraints to prevent conf	SMIF error trigger in all RZT devices. But iguring DSMIF error trigger for channel 1 and
		Property	Value
	Settings	<ul> <li>Module g poeg0 Port Output Enable for GPT (r poeg)</li> </ul>	Voide
	API Info	✓ General	
		✓ Trigger	
		GTETRG Pin	
		GPT Output Level	
		Oscillation Stop	
		DSMIF0 error	
		DSMIF1 error	
		DSMIF0 error 1	
		DSMIF1 error 1	
		Name	a poeg0
		Channel	1
		Group	A
		> Input	
		> Interrupts	
		✓ Pins	
		GTETRGA	<unavailable></unavailable>
Workaround	Use the	e DSMIF error trigger for channel 0 only.	



Issue	FreeRTOS+FAT format process is not executed correctly.
Target	RZ/T2M, RZ/T2ME, RZ/T2L, RZ/T2H, RZ/N2L, RZ/N2H
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	Please add g_format_flag before and after calling the FF_Format function in your application code. extern uint8_t g_format_flag; g_format_flag = 1; err = FF_Format(&disk, 0, pdFALSE, pdFALSE) g_format_flag = 0;

### No. 28

### No. 29 Resolved

Issue	Caution when specifying program placement in linker scripts
Target	RZ/T2H, RZ/N2H
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 } &gt; 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, RZ/N2H
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.

### No. 30

### No. 31 Resolved

Issue	Errors occur when setting ELC in r_gpt module.
Target	RZ/T2H, RZ/N2H
Category	FSP Configuration, Stacks
Description	When setting ELC event trigger source in the stack of r_gpt module, errors will occur and cannot generate the value.  Stacks Configuration  Threads  What Common Stacks  Mat/Common Stacks   Mat/Common Stacks   Mat/Common Stacks
	Objects     Item Voted > 10 Tem Concept       Summary BSP Clocks Pins Interrupts [strent Links @ Stacks Components]       Summary BSP Clocks Pins Interrupts [strent Links @ Stacks Components]       Spokens @ Console @ Properties × @ Smart Browser @ Smart Manual & Search to Debug       @ Line OT Timer, General PWM (r.gpt)       > Common       API Info       > Console [strend Timer, General PWM (r.gpt)       > Central       > Output       > Input
	Interrupts     I
Workaround	<ul> <li>To use the ELC trigger source for r_gpt module in code without configuring it in the FSP Configurator, follow these steps.</li> <li>1. Add the r_elc stack in the Stacks tab of FSP Configuration and include the r_elc header in your application.</li> <li>2. Call the following functions to initially set up the r_elc module. <ul> <li>i. R_ELC_Open();</li> <li>ii. R_ELC_Enable();</li> </ul> </li> <li>3. Use the R_ELC_LinkSet(); function to configure the event triggers.</li> </ul>
	<pre>Example: R_ELC_Open(&amp;g_gpt_test_elc_ctrl, &amp;g_elc_cfg); R_ELC_Enable(&amp;g_gpt_test_elc_ctrl); R_ELC_LinkSet(&amp;g_gpt_test_elc_ctrl, ELC_PERIPHERAL_GPT00_A, ELC_EVENT_INTCPU0);</pre>



Issue	CR52 CPU1 of RZ/T2H and RZ/N2H is implemented to run program from System SRAM instead of CPU1 ATCM.
Target	RZ/T2H(CR52), RZ/N2H(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. 32 Resolved for RZ/T series devices in RZT FSP v3.0.0

#### No. 33

Issue	No Error Occurs when entering out-of-range values for window parameters in r_pcie_ep and r_pcie_rc module configurations.			
Target	RZ/T2H, RZ/N2H			
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.			
	Configuration item	r_pcie_ep	r_pcie_rc	
	AXI Window Base	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 the input value conditions for each item.         AXI Window Base and PCIe Window Base         - "greater than or equal to 0" and "address is 4Kbyte aligned"         AXI Window Mask and PCIe Window Mask         - "greater than or equal to 0", 'the lower 12 bits are 1', and "the 63rd bit is 0"         AXI Window Destination and PCIe Window Destination         - "greater than or equal to 0" and "address is 4Kbyte aligned"         MSI Receive Window Mask         - "greater than or equal to 0" and "the lower 2 bits are 1"         MSI Receive Window Address         - Align according to "MSI Receive Window Mask".			



Issue	DDR and PCIE0/1 memory cannot be used in secondary (or later) projects with flash boot mode.
Target	RZ/T2H, RZ/N2H
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, RZ/N2H
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.

### No. 36 Resolved

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			
	<pre>/* Release module stop for ADC12 */ R_BSP_RegisterProtectDisable(BSP_REG_PROTECT_LPC_RESET); R_BSP_MODULE_START(FSP_IP_ADC12, 2); R_BSP_RegisterProtectEnable(BSP_REG_PROTECT_LPC_RESET);</pre>			
	<pre>R_BSP_SoftwareDelay(1, BSP_DELAY_UNITS_MICROSECONDS); /* Write ADCCALCTL.CAL bit to 1 to start calibration. */ R_ADC122-&gt;ADCALCTL_b.CAL = ADC_ADCALCTL_SET_CAL;</pre>			
	<pre>/* Poll ADCCALCTL.CAL_RDY bit until it is changed to 1. */ FSP_HARDWARE_REGISTER_WAIT(R_ADC122-&gt;ADCALCTL_b.CAL_RDY, 1U); /* Confirm ADCCALCTL.CAL_ERR bit is 0.*/ FSP_HARDWARE_REGISTER_WAIT(R_ADC122-&gt;ADCALCTL_b.CAL_ERR, 0U); /* Write ADCCALCTL.CAL bit to 0 */ R_ADC122-&gt;ADCALCTL_b.CAL = 0U;</pre>			
	<pre>/* Initializing the ADC module */ R_ADC_Open(&amp;g_adc0_ctrl,&amp;g_adc0_cfg );</pre>			



Issue	The USB driver for CA55 project does not work.		
Target	RZ/T2H(CA55), RZ/N2H(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	<pre>Please modify \rztNsp\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_CF6_CORE_CASS)     /* Converts a physical address to a virtual address. */     if (FSP_SUCCESS != R_BSP_MmuPatoVa(paddr, &amp;vaddr, BSP_MMU_CONVERSION_NON_CACHE))     {         /* On error, returns the physical address without conversion. */         vaddr = paddr;     }     #else /* #if defined(BSP_CF6_CORE_CASS) */     return vaddr;         /* End of function r_usb_pa_to_va() */ 2. Modify the r_usb_va_to_pa function.     uint64_t r_usb_va_to_pa (uint64_t vaddr)     {         uint64_t paddr = 0;     #if defined(BSP_CF6_CORE_CASS)         /* Converts a virtual address to a physical address. */         if (FSP_SUCCESS != R_BSP_MmuVatoPa(vaddr, &amp;paddr))     {         uint64_t r_usb_va_to_pa (uint64_t vaddr)         {         uint64_t r_aub_va_to_pa (uint64_t vaddr)         {         uint64_t r_aub_va_to_pa (vaddr, &amp;paddr))         {             /* On error, returns the virtual address without conversion. */             paddr;         }         #else /* #if defined(BSP_CF6_CORE_CASS) */         return vaddr;         /* End of function r_usb_pa_to_va() */         2. Modify the r_usb_va_to_pa (uint64_t vaddr)         {             uint64_t r_aub_va_to_pa (uint64_t vaddr)         {             uint64_t r_aub_va_to_pa (uint64_t vaddr)         }         /* On error, returns the virtual address. */         if (FSP_SUCCESS != R_BSP_MmuVatoPa(vaddr, &amp;paddr)))         {             /* On error, returns the virtual address without conversion. */             paddr = vaddr;         #ust ef if defined(BSP_CF6_CORE_CASS) */         paddr = vaddr;         #ust ef if defined(BSP_CF6_CORE_CASS) */         paddr = vaddr;         #ust ef if defined(SSP_CF6_CORE_CASS) */         paddr = vaddr;         #ust ef if defined(SSP_CF6_CORE_CASS) */         paddr = vaddr;         #ust ef if defined(SSP_CF6_CORE_CASS) */         paddr =</pre>		
	<pre>} #else /* #if defined(BSP_CFG_CORE_CA55) */</pre>		

#### No. 37 Resolved



Issue	No error returns when entering the virtual addresses that cannot be translated to physical addresses as arguments.		
Target	RZ/T2H(CA55), RZ/N2H(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.		

### No. 38 Resolved



Issue	The CA55 project with noncache sections aborts when debugging with flash boot mode on IAR EWARM			
Target	RZ/T2H(CA55), RZ/N2H(CA55)			
Category	FSP Modules, BSP			
Description	When performing debugging of a flash boot CA55 project with noncache sections on IAR EWARM, executing the CA55 project will abort. This is due to cache initialization.			
Workaround	<pre>Follow the steps below 1. Add " set_ICIALLU(0);" and "ISB();" to bsp_memory_protect_setting in     XXX/fsp/src/bsp/cmsis/Device/RENESAS/Source/ca/system_core.c. (XXX= rzt, rzn) void bsp_memory_protect_setting (void) {     bsp_mmu_configure();     R_BSP_CacheInvalidateAll();     R_BSP_CacheEnableMemoryProtect();     R_BSP_CacheEnableInst();     r_set_ICIALLU(0);    ISB(); 2. Move "set_ICIALLU(0);" in R_BSP_CacheInvalidateAll in     XXX/fsp/src/bsp/mcu/all/bsp_cache.c (XXX= rzt, rzn) to just before "asm volatile("ISB     SY");" at the end.</pre>			
	<pre>b 1 /, to the provide state of the second state of the second</pre>			

#### No. 39 Resolved



Issue	When changing the duty setting in r_gpt module, there is a possibility the duty may unintentionally become 100%.		
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L, RZ/N2H		
Category	FSP Modules, GPT		
Description	When changing the PWM period with "R_GPT_PeriodSet()" and the duty with "R_GPT_DutyCycleSet()" while the GPT is running, the duty may unintentionally become 100% depending on the both setting values. The reason is that when "gpt_calculate_duty_cycle()" in "R_GPT_DutyCycleSet()" performs a comparison calculation of the duty and period, the old period value of GTPR register is mistakenly referenced instead of the current period value of GTPBR (buffer) register.		
Workaround			

### No. 40 Resolved

### No. 41 Resolved

Issue	CPU registers save and restore process cannot be performed correctly in FIQ_Handler for CA55 projects.		
Target	RZ/T2H(CA55), RZ/N2H(CA55)		
Category	FSP Modules, BSP, FreeRTOS		
Description	Some of CPU registers are not saved and restored in FIQ_Handler. Therefore, the value of registers may be partially corrupted before or after the FIQ interrupt occurs.		
Workaround	<pre>may be partially corrupted before or after the FIQ interrupt occurs.  • When NOT using FreeRTOS Please modify XXX/fsp/src/bsp/cmsis/Device/RENESAS/Source/ca/startup_core.c (XXX= rzt, rzn) as follows. WEAK void FIQ_Handler (void) {    asm volatile (         "STP x30, XZR, [SP, #-0x10]! \n"         "STP x28, x29, [SP, #-0x10]! \n"         "STP x26, x27, [SP, #-0x10]! \n"         "STP x16, x17, [SP, #-0x10]! \n"         "STP x12, x13, [SP, #-0x10]! \n"         "STP x12, x13, [SP, #-0x10]! \n"         "STP x10, x11, [SP, #-0x10]! \n"         "STP x4, x5, [SP, #-0x10]! \n"         "STP x6, x1, [SP, #-0x10]! \n"</pre>		



# RZ/T2, RZ/N2

	"MRS x0, FPCR	\n"
	"MRS x1, FPSR	\n"
	"STP x0, x1, [SP, #-0x10]!	\n"
	"STP q30, q31, [SP, #-0x20]!	\n"
	"STP q28, q29, [SP, #-0x20]!	\ <b>n</b> "
	"STP q26, q27, [SP, #-0x20]!	\n"
	"STP q24, q25, [SP, #-0x20]!	\n"
		\n"
	"STP q22, q23, [SP, #-0x20]!	
	"STP q20, q21, [SP, #-0x20]!	\n"
	"STP q18, q19, [SP, #-0x20]!	\n"
	"STP q16, q17, [SP, #-0x20]!	\n"
	"STP q14, q15, [SP, #-0x20]!	\n"
	"STP q12, q13, [SP, #-0x20]!	\n"
	"STP q10, q11, [SP, #-0x20]!	\n"
	"STP q8, q9, [SP, #-0x20]!	\ <b>n</b> "
	"STP q6, q7, [SP, #-0x20]!	\n"
	"STP q4, q5, [SP, #-0x20]!	\n"
	"STP q2, q3, [SP, #-0x20]!	\n"
	"STP q0, q1, [SP, #-0x20]!	\n"
#	endif	
~	~~~~~~	
#	ifFPU_USED	
	"LDP q0, q1, [sp], #0x20	\n"
	"LDP q2, q3, [sp], #0x20	\n"
	"LDP q4, q5, [sp], #0x20	\n"
	"LDP q6, q7, [sp], #0x20	\n"
		\n"
		\n"
	"LDP q10, q11, [sp], #0x20	•
	"LDP q12, q13, [sp], #0x20	\n"
	"LDP q14, q15, [sp], #0x20	\n"
	"LDP q16, q17, [sp], #0x20	\n"
	"LDP q18, q19, [sp], #0x20	\n"
	"LDP q20, q21, [sp], #0x20	\n"
	"LDP q22, q23, [sp], #0x20	\n"
	"LDP q24, q25, [sp], #0x20	\ <b>n</b> "
	"LDP q26, q27, [sp], #0x20	\n"
	"LDP q28, q29, [sp], #0x20	\n"
		\n"
	"LDP x0, x1, [sp], #0x10	\n"
	"MSR FPCR, x0	\n"
	"MSR FPSR, x1	\n"
#	endif	
	"LDP x0, x1, [sp], #0x10	\n"
	"LDP x2, x3, [sp], #0x10	\n"
	"LDP x4, x5, [sp], #0x10	\n"
	"LDP x6, x7, [sp], #0x10	\n"
	"LDP x8, x9, [sp], #0x10	\n"
	"LDP x10, x11, [sp], #0x10	\n"
	"LDP x12, x13, [sp], #0x10	\n"
	"LDP x14, x15, [sp], #0x10	\n"
	"LDP x16, x17, [sp], #0x10	\n"
	"LDP x18, x19, [sp], #0x10	\n"
	"LDP x20, x21, [sp], #0x10	\n"
	"LDP x22, x23, [sp], #0x10	\n"
	"LDP x24, x25, [sp], #0x10	\n"
	"LDP x26, x27, [sp], #0x10	\n"
	"LDP x28, x29, [sp], #0x10	\n"
		•
	"LDP x30, XZR, [sp], #0x10	\n"
	UCDET	
	"ERET	\n"
	<pre>::: "memory");</pre>	
}		
•	When using Treerer 05	
	Please modify XXX/fsp/src/rm	_freertos_port/ca/port.c (XXX= rzt, rzn) as follows.
В	<pre>SP_ATTRIBUTE_STACKLESS void FIQ_Handle</pre>	er (void)
{		
	/* Save volatile registers. */	
	asm volatile (	
	"STP x30, XZR, [SP, #-0x10]!	\n"
	"STP x28, x29, [SP, #-0x10]!	\ <b>n</b> "
	"STP x26, x27, [SP, #-0x10]!	\n"
	"STP x24, x25, [SP, #-0x10]!	\n"
	"STP x22, x23, [SP, #-0x10]!	\n"
		\n"
	"STP x20, x21, [SP, #-0x10]!	
	"STP x18, x19, [SP, #-0x10]!	\n" \"
	"STP x16, x17, [SP, #-0x10]!	\n"
	"STP x14, x15, [SP, #-0x10]!	\n"
	"STP x12, x13, [SP, #-0x10]!	\n"
	"STP x10, x11, [SP, #-0x10]!	\n"
	"STP x8, x9, [SP, #-0x10]!	\n"



[		V - U	
	"STP x6, x7, [SP, #-0x10]!	\n"	
	"STP x4, x5, [SP, #-0x10]!	\n"	
	"STP x2, x3, [SP, #-0x10]!	\n"	
	"STP x0, x1, [SP, #-0x10]!	\n"	
	#ifFPU_USED		
	"MRS x0, FPCR	\n"	
		· · · · · · · · · · · · · · · · · · ·	
	"MRS x1, FPSR	\n"	
	"STP x0, x1, [SP, #-0x10]!	\n"	
	"STP q30, q31, [SP, #-0x20]!	\n"	
	"STP q28, q29, [SP, #-0x20]!	\n"	
	"STP q26, q27, [SP, #-0x20]!	\n"	
	"STP q24, q25, [SP, #-0x20]!	\n"	
	"STP q22, q23, [SP, #-0x20]!	\n"	
	"STP q20, q21, [SP, #-0x20]!	\n"	
	"STP q18, q19, [SP, #-0x20]!	\n"	
	"STP q16, q17, [SP, #-0x20]!	\n"	
	"STP q14, q15, [SP, #-0x20]!	\n"	
	"STP q12, q13, [SP, #-0x20]!	\n"	
	"STP q10, q11, [SP, #-0x20]!	\n"	
		\n"	
	"STP q8, q9, [SP, #-0x20]!	· · · · · · · · · · · · · · · · · · ·	
	"STP q6, q7, [SP, #-0x20]!	\n"	
	"STP q4, q5, [SP, #-0x20]!	\n"	
	"STP q2, q3, [SP, #-0x20]!	\n"	
	"STP q0, q1, [SP, #-0x20]!	\n"	
	#endif		
	/* Save the SPSR and ELR. */		
	~~~~~~		
	"DSB SY	\n"	
	"ISB SY	\n"	
	וכ מכד	/11	
	<pre>#ifFPU_USED</pre>		
		\n"	
		· · · · · · · · · · · · · · · · · · ·	
	"LDP q2, q3, [sp], #0x20	\n"	
	"LDP q4, q5, [sp], #0x20	\n"	
	"LDP q6, q7, [sp], #0x20	\n"	
		· · · · · · · · · · · · · · · · · · ·	
	"LDP q8, q9, [sp], #0x20	\n"	
	"LDP q10, q11, [sp], #0x20	\n"	
	"LDP q12, q13, [sp], #0x20	\n"	
	LDF 412, 415, [5p], #0x20		
	"LDP q14, q15, [sp], #0x20	\n"	
	"LDP q16, q17, [sp], #0x20	\n"	
		\n"	
	"LDP q18, q19, [sp], #0x20		
	"LDP q20, q21, [sp], #0x20	\n"	
	"LDP q22, q23, [sp], #0x20	\n"	
	"UDD =24 =25 [sp]; "0x20	· · · · · · · · · · · · · · · · · · ·	
	"LDP q24, q25, [sp], #0x20	\n"	
	"LDP q26, q27, [sp], #0x20	\n"	
	"LDP q28, q29, [sp], #0x20	\n"	
		· · · · · · · · · · · · · · · · · · ·	
	"LDP q30, q31, [sp], #0x20	\n"	
	"LDP x0, x1, [sp], #0x10	\n"	
		\n"	
	"MSR FPCR, x0	· · · · · · · · · · · · · · · · · · ·	
	"MSR FPSR, x1	\n"	
	#endif		
	"LDD v0 v1 [cm] #0.10	\n"	
	"LDP x0, x1, [sp], #0x10	\n"	
	"LDP x2, x3, [sp], #0x10	\n"	
	"LDP x4, x5, [sp], #0x10	\n"	
	"LDP x6, x7, [sp], #0x10	\n"	
	"LDP x8, x9, [sp], #0x10	\n"	
	"LDP x10, x11, [sp], #0x10	\n"	
	"LDP x12, x13, [sp], #0x10	\n"	
	"LDP x14, x15, [sp], #0x10	\n"	
	"LDP x16, x17, [sp], #0x10	\n"	
	"LDP x18, x19, [sp], #0x10	\n"	
	"LDP x20, x21, [sp], #0x10	\n"	
	"LDP x22, x23, [sp], #0x10	\n"	
	"LDP x24, x25, [sp], #0x10	\n"	
	"LDP x26, x27, [sp], #0x10	\n"	
	"LDP x28, x29, [sp], #0x10	\n"	
	"LDP x30, XZR, [sp], #0x10	\n"	
		1.1	
	::: "memory");		
	/* Save the context of the current tas	c and select a new task to run. */	
	~~~~~		
	}		
	BSP_ATTRIBUTE_STACKLESS void Exit_IRQ_No_Co	ontext_Switch (void)	
	{		
	"DSB SY	\n"	
	"ISB SY	`. \n"	
	#if EDIL USED		
	<pre>#ifFPU_USED</pre>		
	"LDP q0, q1, [sp], #0x20	\n"	
	"LDP q2, q3, [sp], #0x20	\n"	



"LDP	q4, q5, [sp], #0x20	\n"
"LDP	q6, q7, [sp], #0x20	\n"
"LDP	q8, q9, [sp], #0x20	\n"
"LDP	q10, q11, [sp], #0x20	\n"
"LDP	q12, q13, [sp], #0x20	\n"
"LDP	q14, q15, [sp], #0x20	\n"
"LDP	q16, q17, [sp], #0x20	\n"
"LDP	q18, q19, [sp], #0x20	\n"
"LDP	q20, q21, [sp], #0x20	\n"
"LDP	q22, q23, [sp], #0x20	\n"
"LDP	q24, q25, [sp], #0x20	\n"
"LDP	q26, q27, [sp], #0x20	\n"
"LDP	q28, q29, [sp], #0x20	\n"
"LDP	q30, q31, [sp], #0x20	\n"
"LDP	x0, x1, [sp], #0x10	\n"
"MSR	FPCR, x0	\n"
"MSR	FPSR, x1	\n"
#endif		
"LDP	x0, x1, [sp], #0x10	\n"
"LDP	x2, x3, [sp], #0x10	\n"
"LDP	x4, x5, [sp], #0x10	\n"
"LDP	x6, x7, [sp], #0x10	\n"
"LDP	x8, x9, [sp], #0x10	\n"
"LDP	x10, x11, [sp], #0x10	\n"
"LDP	x12, x13, [sp], #0x10	\n"
"LDP	x14, x15, [sp], #0x10	\n"
"LDP	x16, x17, [sp], #0x10	\n"
"LDP	x18, x19, [sp], #0x10	\n"
"LDP	x20, x21, [sp], #0x10	\n"
	x22, x23, [sp], #0x10	\ <b>n</b> "
"LDP	x24, x25, [sp], #0x10	\n"
"LDP	x26, x27, [sp], #0x10	\n"
"LDP	x28, x29, [sp], #0x10	\ <b>n</b> "
"LDP	x30, XZR, [sp], #0x10	\n"
"ERET	-	\n"
	'memory");	
}		



# No. 42 Resolved

Issue	MTU3 callback does not occur as expectation.
Target	RZ/T2H, RZ/N2H
Category	FSP Modules, MTU3
Description	Callback does not always occur just like when you specify NULL in the configurator settings of MTU3.
	Image: Section of the section of th
	Interpret (FT_SE) Callect Callect Collect Callect Ca
Workaround	<pre>Please use the Callback Set API from: R_MTU3_CallbackSet() For reference on how to use the API, please refer to MTU3 Examples below: Step1: Define the Callback Function void mtu3_callback_set_test(timer_callback_args_t * p_args) { //Callback function to be implemented by the user } Step2: Sequence call the API /* Open the timer module */ R_MTU3_Open(&amp;g_timer0_ctrl, &amp;g_timer0_cfg); /* Use R_MTU3_CallbackSet */</pre>
	<pre>R_MTU3_CallbackSet (timer_ctrl_t * const p_ctrl,</pre>



### No. 43 Resolved

Issue	An undefined error of r_gpt module occurs when building a project.				
Target	RZ/T2H, RZ/N2H				
Category	FSP Modules, GPT				
Description	Undefined error of "gpt_counter_underflow_isr" occurs when "Pin Output Support" is set to anything other than "Enabled with Extra Features" regardless of whether Pin Output is used or not.				
Workaround	Please always set the "Pin Output Support" to "Enabled with Extra Features" when configuring r gpt module.				
	Timer, General PWM (r_gpt)				
	Property  Common	Value			
	Parameter Checking	Default (BSP)			
	Pin Output Support	Enabled with Extra Features			
	Write Protect Enable	Disabled			
	Multiplex Interrupt	Disabled			

#### No. 44

Issue	Pin names according to unit and channel numbers are not displayed in r_gpt module configurations.							
Target	<b>RZ/T2M, RZ/T2L</b> ,	RZ/T2ME, RZ/	Г2Н, RZ/N2I	2, RZ/N2H				
Category	FSP Configuration, Stacks							
Description	When setting unit and not change or not app		r in r_gpt moo	lule, pin names i	n the Pins of stack p	roperty o		
	g_timer0 Timer, General PWM (r_gpt)		RZ/T2M	imer0 Timer, General PWM (r_gpt)	RZ/T2H			
	Settings Property  Affinite  Common Parametr Checking Product Support With Protect Stable Multiples Internet General Name Common Common Mode General Name Common Mode Find Common Mode Find Common Mode Find Common Mode Find Common Find Find Common Find Find Common Find Find Find Find Find Find Find Fin	g_timer0 0 Periodic 0x10000000 Raw Counts		ting Property Parameter Onesking P Proventer Onesking P Product Support Write Protect Enable Multiple Interrupt > Output > Interrupti > Interrupti > Interruption >				
Workaround	Please use the Pins ta Pin Configuration	b in the FSP Cor	nfiguration to		0			
			<b>m</b> .	Generate Project Content				
	Select Pin Configuration RZT2H Evaluation Board	<ul> <li>Manage configurations</li> </ul>		W file 🚺 Configure Pin Driver Warnii e data: g_bsp_pin_cfg	ngs			
	Pin Selection			g_osp_pm_org	Cycle Pin Group			
		Name	Value	Lock Link	Cycle Pin Gloup			
	GPT         ×           V # Peripherals         •           GP100         GP101           GP102         GP103           GP103         GP104           GP106         GP107           GP109         CP109           PIn Function         Pin Number	Name Pin Group Selection Operation Mode ✓ Input/Output GTADSM00_0 GTADSM0_0 GTIOC00_08 GTIOC00_08 GTIOC00_1A GTIOC00_1B ← Module name: GPT00	Value Mixed Disabled None None None None		×			



### No. 45 Resolved

Issue	Using R_GPT_DutyCycleSet() with option both pins A and B cannot work properly.				
Target	RZ/T2H, RZ/T2M, RZ/T2ME, RZ/T2L				
Category	FSP Modules, GPT				
Description	When updating the duty cycle, GPT_IO_PIN_GTIOCA_AND_GTIOCB cannot be used to set both pins at the same time.				
Workaround	To change the duty cycle during runtime, each pin must be updated separately. Example: R_GPT_DutyCycleSet(&g_timer0_ctrl, duty_cycle_counts, GPT_IO_PIN_GTIOCA); R_GPT_DutyCycleSet(&g_timer0_ctrl, duty_cycle_counts, GPT_IO_PIN_GTIOCB);				

#### No. 46 Resolved for RZ/T series devices in RZT FSP v3.0.0

Issue	Parameter checking of R_ETHER_SELECTOR_Open() is not working properly.			
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L, RZ/N2H			
Category	FSP Modules, ETHER_SELECTOR			
Description	When using the R_ETHER_SELECTOR_Open() with parameter checking enabled, the configuration pointer is used before being assigned. This causes parameter checking to work incorrectly, sometimes leading to exceptions.			
Workaround	Disable parameter checking when using ether selector module.			

### No. 47 Resolved for RZ/T series devices in RZT FSP v3.0.0

Issue	Parameter checking feature of R_GMAC_CallbackSet() is not working.					
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L, RZ/N2H					
Category	FSP Modules, ETHER_GMAC					
Description	<ul> <li>Parameter check feature of R_GMAC_CallbackSet() does not work because the parameter checking macro name generated by the MDF is inconsistent with the macro referenced in the actual source code.</li> <li>Macro generated by MDF: GMAC_CFG_PARAM_CHECKING_ENABLE</li> <li>Macro used in source code: ETHER_CFG_PARAM_CHECKING_ENABLE</li> <li>Note: Parameter check feature of other APIs work properly.</li> </ul>					
Workaround	Step 1: In project Properties->Builders, uncheck DDSC Builder option (this step make user can modify "r_ether_cfg.h" file)         Step 2: In "XXX_cfg/fsp_cfg/r_ether_cfg.h"(XXX = rzt, rzn), manually add the macro definition below.         #define ETHER_CFG_PARAM_CHECKING_ENABLE ((1))					



Issue	r_usb_hhid module is not working properly.					
Target	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H					
Category	FSP Modules, USB_HHID					
Category Description Workaround	It is possible that HOST does not respond to the received data because it cannot get the transfer size correctly, so even if the transfer is completed without error, it may not be successful.         In the report, when debuging with a usb analyzer, errors are reported as belows:         Image: the transfer is completed without error, it may not be successful.         In the report, when debuging with a usb analyzer, errors are reported as belows:         Image: the transfer is completed without error, it may not be successful.         Image: the transfer is completed without error, it may not be successful.         Image: the transfer is completed without error, it may not be successful.         Image: the transfer is completed without error, it may not be successful.         Image: the transfer is completed without error, it may not be successful.         Image: the transfer is completed without error.         Image: th					
	After update: if ((((uint16_t) p_instance_ctrl->device_address) << USB_DEVADDRBIT) ==					
	<pre>(uint16_t) (g_usb_pipe_table[p_instance_ctrl-&gt;module_number][pipe_no].pipe_maxp &amp; USB_DEVSEL)) {     (*p_pipe) = (uint16_t) ((*p_pipe)   (uint16_t) 1 &lt;&lt; pipe_no); }</pre>					

### No. 48 Resolved



# Appendix. Tool Software Limitations

This section describes the limitations regarding the tool software (e<sup>2</sup> studio, FSP SC) to create and debug FSP projects.

No	Title	Target Device						Category
•		T2M	T2L	T2ME	T2H	N2L	N2H	
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 <sup>2</sup> studio connection first.	1	1			1		e <sup>2</sup> 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 <sup>2</sup> studio
4	The user program cannot be stopped immediately after the device boot process.	1	1	1	1	1	~	e <sup>2</sup> studio
5	When using e <sup>2</sup> 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			<i>✓</i>		e <sup>2</sup> studio
6	The Memory Region Usage of ATCM displayed in the Memory Usage window of e <sup>2</sup> studio is smaller than the actual size by memory region usage of DUMMY.	1	1			<i>✓</i>		e <sup>2</sup> 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	1	e <sup>2</sup> 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		1	e <sup>2</sup> studio
10	When adding the CallbackSet function using the Developer Assistance feature, the second argument needs to be changed.	1	1	1	1	1	1	e <sup>2</sup> 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 <sup>2</sup> studio.				1		1	e <sup>2</sup> studio
14	Unable to restart debugging immediately after debugging ends of CA55 RAM execution without flash memory project in e <sup>2</sup> studio.				1		1	e <sup>2</sup> studio
15	Unable to re-download CA55 binary file.				1		1	IAR EWARM

Table 21	List of	Tool	Software	Limitations
I able #1	LISCOL	1001	Solution	Limitations



No.	Title		et Devi		Category			
		T2M	T2L	T2ME	T2H	N2L	N2H	
16	Unable to access the upper 32-bit address area in memory view.				1		1	e <sup>2</sup> studio
17	Unable to create a CMake project using FSP SC.	1	✓	1	1	1	1	FSP SC
18	The secondary project aborts when debugging multicore with flash boot mode.	1		1	1		1	IAR EWARM
19	Build errors occur in CA55 projects when install e <sup>2</sup> studio as the Current user.				1		1	e <sup>2</sup> studio
20	Wrong core name when create a project CA55 with FSP SC.						1	FSP SC
21	Build is failed when adding the OpenAMP.				1		1	FSP SC
22	The bundle file (.sbd) may not be generated during build.	1	1	1	1	1	1	e <sup>2</sup> studio
23	When implementing CMT interrupts in RZ/T2H CR52 or RZ/N2H CR52 project, an unintended source file is displayed during debugging.				1		1	e <sup>2</sup> studio

### No. 1 Resolved

Limitation	When installing, please install into the default installation folder specified by installer.
<b>Target Device</b>	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/N2L
Category	SC, FSP SC
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 <sup>2</sup> studio connection first.	
Target	RZ/T2M, RZ/T2L, RZ/N2L	
Category	e <sup>2</sup> studio	
Description	If the reset button is pressed on the board while connected with e <sup>2</sup> 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 <sup>2</sup> 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, RZ/N2H	
Category	e <sup>2</sup> 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.	



Limitation	When using e <sup>2</sup> 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 <sup>2</sup> 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.		
	v202210230745 User All Users < Back Next > OK Cancel		

#### No. 5 Invalid

### No. 6 Resolved

Limitation	The Memory Region Usage of ATCM displayed in the Memory Usage window of e <sup>2</sup> studio is smaller than the actual size by Memory Region Usage of DUMMY.	
Target	RZ/T2M, RZ/T2L, RZ/N2L	
Category	e <sup>2</sup> studio	
Description	The Memory Region Usage of DUMMY shown in the Memory Usage window is the region used by the system. The DUMMY is placed in ATCM, however Memory Region Usage of ATCM does NOT include its size. Therefore, please note that the Memory Region Usage of ATCM displayed is smaller than the actual size by the Memory Region Usage of DUMMY.	
	Virtual Console 🔗 Search 🔋 Memory Usage × 🖉 🎓 🖓 🕼 🔗 🖓 🥵 🥞 🖉	
	Memory Region Usage:	
	ATCM 6% 512KB 33KB used	
	BTCM 44 % 64KB 28KB used	
	DUMMY 100 % 88 88 used	
	SYSTEM_RAM 0% 2048KB 0B used	
	< · · · · · · · · · · · · · · · · · · ·	



Ν	о.	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, RZ/N2H
Category	e <sup>2</sup> 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 Category: Runthe Checking C(C++ Complet Assertion Cutom Build Runthe Checking C(C++ Complet Runthe Checking C(C++ Complet Run
	OK Cancel



Limitation	The Device Memory Usage of CPU1 in the Memory Usage window does not work properly.	
Target	RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H	
Category	e <sup>2</sup> 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.	

### 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, RZ/N2H	
Category	e <sup>2</sup> studio, SC	
Description	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.	
	<pre>status = R_SCI_SPI_CallbackSet(&amp;g_spi0_ctrl, spi_callback_args_t, p_context, p_callback_memory); It needs to replace the second argument with "p_callback".</pre>	
	<pre>status = R_SCI_SPI_CallbackSet(&amp;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 projects of multiprocessing, it is necessary to add " macro_param cpu1_enable=1" in the " <b>command line options (one per line</b> )" of CPU1 project.	

### 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, RZ/N2H	
Category	FSP SC	
Description	If install path of FSP SC is different between creating project and executing project, build of a project is failed.	
	Please reselect the execution path by following the steps below.	
	1. Launch FSP SC.	
	2. Close the window to create a new project.	
	3. Click on File -> Open and select configuration.xml in your project.	
	4. Click "Generate Project Content".	
	5. Save the project and close FSP SC.	
	6. Open the project with EWARM.	



Limitation	Unable to debug CA55 flash boot project with e <sup>2</sup> studio.		
Target Device	RZ/T2H(CA55), RZ/N2H(CA55)		
Category	e <sup>2</sup> studio		
Description	When debugging with e <sup>2</sup> 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. 13 Resolved

### No. 14 Resolved

Limitation	Unable to restart debugging immediately after debugging ends of CA55 RAM execution without flash memory project in e2 studio.		
<b>Target Device</b>	RZ/T2H(CA55), RZ/N2H(CA55)		
Category	e <sup>2</sup> studio		
Description	When debugging with e <sup>2</sup> 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 but (red) on the board.		

### No. 15

Limitation	Unable to re-download CA55 binary file.		
<b>Target Device</b>	RZ/T2H(CA55), RZ/N2H(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), RZ/N2H(CA55)			
Category	e <sup>2</sup> studio			
Description	<ul> <li>In the Memory view of e<sup>2</sup> studio, data in the upper 32-bit address area cannot be display. When debugging the CA55 project, please make the following settings when accessing to 32-bit address.</li> <li>Open Debug Configurations of CA55 project.</li> <li>Select Renesas GDB Hardware Debugging &gt; [project name] Debug_Flat.</li> <li>Select Startup and specify the command in Run Commands. Referring to the folloc command, specify the address area you want to access, and you will be able to access?-bit address area.</li> <li>"mem 0x200000000 0x300000000 rw"</li> </ul>	he upper		
	Create, manage, and run configurations			
	C 🖻 🕫 🗟 🗶 🖻 🏹 👻 Name: Blinky Debug Flat			
	Image: Second Secon			
	C GOB Hardware Debuggin (RH     C GOB Hardware Debuggin (RH     C Lounch Group     C S Blinky Debug, RR     C Renesss GDB Hardware Debug     C Blinky Debug, RR     C Renesss Simulator Debugging (			
	Run Commands			
	Filter matched 11 of 13 items     Regert     Apply			
	O     Debug     Close			
	Before adding the command         Image: Address			
	Image: Memory ×         Image: Me			
	Monitors 💠 🗱 🍇 0x200000000 : 0x200000000 <hex integer=""> 🗙 💠 New Renderings</hex>			
	• 0x20000000          Address              0 - 3               4 - 7               8 - B               C - F                 • 00000000000               FFFFFFF                 A               A               A               A               A               A               A               A               A               A               A               A               A               C               A               A                         A			
	000000200000020 FFFFFFF FFFFFFF FFFFFFF FFFFFFFF			
	0000000200000000 FFFFFFF FFFFFFF FFFFFFFF			
	000000200000000 FFFFFFF FFFFFFF FFFFFFFF			
	0000000200000070 FFFFFFF FFFFFFFFFFFFFFF			
	0000000200000080 FFFFFFF FFFFFFF FFFFFFFF FFFFFFF			



### No. 17 Resolved

Limitation	Unable to create a CMake project using FSP SC.				
Target Device	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L, RZ/N2H				
Category	FSP SC				
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.				
	There is no workaround.				

#### No. 18

Limitation	The secondary project aborts when debugging multicore with flash boot mode.				
Target Device	RZ/T2M, RZ/T2ME, RZ/T2H, RZ/N2H				
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 SUJO į 🏪				
	Disabled (no reset)				
	Software T				
	Hardware				
	Custom				
	3. Run the secondary project.				



Limitation	Build errors occur in CA55 projects when install e <sup>2</sup> studio as the Current user.		
<b>Target Device</b>	RZ/T2H(CA55), RZ/N2H(CA55)		
Category	e <sup>2</sup> studio		
Description	If you install e <sup>2</sup> studio as the Current user, an error will occur when building a CA55 project. As a workaround, install e <sup>2</sup> studio as All Users.		

# No. 20

Limitation	Wrong core name when create a project CA55 with FSP SC.         RZ/N2H(CA55)         FSP SC		
Target Device			
Category			
Description	When you create a project CA55 with FSP SC the buildinfo.ipcf even though you select CA55 core.	—	

### No. 21

Limitation	Build is failed when adding the OpenAMP.			
Target Device	RZ/T2H, RZ/N2H			
Category	FSP SC			
Description	When adding OpenAMP in Stacks tab, source browser occurs error and build is failed due to conflict of same file name. After clicking the Generate Project Content button in the FSP SC, please move the files listed below from the "Component" group to another group in [project name]/buildinfo.ipcf. <path>rzt/linaro/libmetal/lib/device.c</path> <path>rzt/linaro/libmetal/lib/init.c</path> <path>rzt/linaro/libmetal/lib/init.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path>rzt/linaro/libmetal/lib/io.c</path> <path io.c<="" lib="" libmetal="" linaro="" path="" rzt=""> <path io.c<="" lib="" libmetal="" linaro="" path="" rzt=""> therefore the rzt/linaro/libmetal/lib/io.c</path> therefore the rzt/linaro/libmetal/lib/io.c</path>			
	<pre><pre><pre><pre>control indecaring (control indecaring) (control indecaring (contr</pre></pre></pre></pre>			



Limitation	The bundle file (.sbd) may not be generated during build.		
Target Device	RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H, RZ/N2L, RZ/N2H		
Category	e <sup>2</sup> studio		
Description	<ul> <li>In e<sup>2</sup> studio, when you import and build an existing project, the bundle file (.sbd) may not be generated.</li> <li>As a workaround, right-click the project in Project Explorer, click Properties &gt; C/C++ Build &gt; Refresh Policy, and verify that the projects displayed in Resources match the projects listed in the upper left.</li> <li>If they are different, click the project in Resources and click Delete</li> </ul>		
	type filter text     > Resource   Builders   > C/C++ Build     > Configuration:     Debug [Active]     > Manage Configurations     Project References   Renesas QE   Run/Debug Settings   Task Tags   > Validation     Add Exception     Edit Exception     Delete		
	Image: Cancel     Next, click Add Resource, check the correct project, and click OK.     Resource Selection     Select a resource to refresh:     Image: Cancel     Select a resource to refresh:     Image: Cancel     Select All        Deselect All     Image: Cancel              Finally, click Apply and Close. By making this setting, an .sbd file will be generated during the build.		



Limitation	When implementing CMT interrupts in RZ/T2H CR52 or RZ/N2H CR52 project, an unintended source file is displayed during debugging.		
Target Device	RZ/T2H (CR52), RZ/N2H (CR52)		
Category	e <sup>2</sup> studio		
Description	When you create a Blinky project for RZ/T2H CR52 or RZ/N2H CR52 project, add a CMT stack, and define a CMT callback function, the command to jump to hal_entry() during debugging may not display the hal_entry.c source file, and an unintended file may be displayed. As a workaround, right-click the project in Project Explorer, click <b>Properties</b> > <b>C/C++ Build</b> > <b>Settings</b> > <b>Tool Settings</b> > <b>Debugging</b> > <b>Default format</b> , and change "Toolchain default" to "dwarf-4".		
	Properties for CMT_Check — 🗆 🗙		
	type filter text Settings $\diamond \bullet \bullet \bullet \bullet$		
	> Resource       ^         Builders       C/C++ Build         C/C++ Build       Configuration:         Debug [Active]       ~         Manage Configurations		
	Environment JSON Compilation Da 🛞 Tool Settings 🛞 Toolchain 🛞 Device 🎤 Build Steps 😤 Build Artifact 扇 Binary Parsers 😣 Error Parsers		
	Logging         Settings         Tool Chain Editor         > C/C++ General         Project Natures         Project References         Renesas QE         Run/Debug Settings         Task Tags         > Validation         > Warnings         @ Senerate gprof information (-p)         @ Generate gprof information (-pg)         Other debugging flags         @ Warnings         @ Includes         @ Warnings         @ Senerate gprof information (-pg)         Other debugging flags         @ Warnings         @ W		
	Apply and Close     Cancel		



# 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 (e<sup>2</sup> studio or IAR EWARM) to debug the user program from its beginning.

#### 1. (Both e<sup>2</sup> 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 e<sup>2</sup> studio, 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.

#### Note for multiprocessing projects:

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

IDE	Core	Boot Mode	Position at which the loop part should be inserted.
e <sup>2</sup> studio	CR52	xSPI boot	First line in system_init() function.
IAR EWARM		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 (         " movw r0, #0</pre>
			<pre>" MOVT r0, #0 \n" " MCR p15, #4, r0, c1, c0, #1 \n" /* Write r0 to HACTLR */ ::[bsp_hactlr_bit_1] "i" (BSP_HACTLR_BIT_L) : "memory");</pre>
	CA55	xSPI boot	<pre>First line in system_init() function. BSP_ATTRIBUTE_STACKLESS void system_init (void) {     #if 1 // Software loops are only needed when debugging.    asm volatile (         " mov x1, #0 \n"         " movk x2, #0x68bf \n"         " movk x2, #0x478, LSL #16 \n"         " movk x2, #0x478, LSL #16 \n"         " movk x2, #0x478, LSL #16 \n"         " adds x1, x1, #1 \n"         " adds x1, x1, #1 \n"         " cmp x1, x2 \n"         " b.ne software_loop \n"         ::: "memory"); #endif         /* g_bsp_software_reset_occurred = false */        asm volatile (             "MOV x0, %0 \n"             "MOV w1, #0 \n"             "STRB w1, [x0] \n"             :::"r" (&amp;g_bsp_software_reset_occurred) : "memory"); </pre>



Note:

The required waiting time varies in proportion to the size of the executable file of the project using FSP. Therefore, when the executable file size is large, the number of loop processes added above should be adjusted.

In this process, the loop count is expressed in hexadecimal, and the 32-bit loop count is divided into upper 16-bit and lower 16-bit and set in a general-purpose register. The following shows the procedure for changing the loop count of CR52 core from 50000000 to 100000000, which is twice the number of loops.

- 1. Convert the loop count from decimal to hexadecimal. 10000000d = 0x5f5 e0ff
- 2. Replace the operand of movw\* to the lower 16-bit value. movw r1, #0xe0ff
- 3. Replace the operand of movt\* to the upper 16-bit value. movt r1, #0x5f5

\* In CA55 core, opcodes are different.

### 2. (IAR EWARM) (RZ/N2H Only) Override the board files.

When RZ/N2H in xspi boot mode, it's required to override the .board files for RZ/T2H (*FlashRZT2H\_EVB\_A55.board*, *FlashRZT2H\_EVB\_R52.board*) to flash the board.

Please reselect the execution path by following the steps bellow:

- i. Click on **Project** and then click on **Option...** to open project option window.
- ii. Select Debugger category and Download Tab.
- iii. Enable "Override default .board file"
- iv. Select path to override the .board files for RZ/T2H
  - CR52 project: TOOLKIT\_DIR\$\config\flashloader\Renesas\FlashRZT2H\_EVB\_R52.board
  - CA55 project: TOOLKIT DIR\$\config\flashloader\Renesas\FlashRZT2H EVB A55.board

Static Analysis         Runtime Checking         C/C++ Compiler         Assembler         Output Converter         Custom Build         Linker         Build Actions         Debugger         Simulator         CADI         GDB Server         GDB Server         GDB Server         GDB Server         Ti Stellaris         Nu-Link         PE micro         ST-LINK         Third-Party Driver         TI MSP-FET	
--	--

Figure 109 : Project Options – Debugger (RZ/N2H Only)

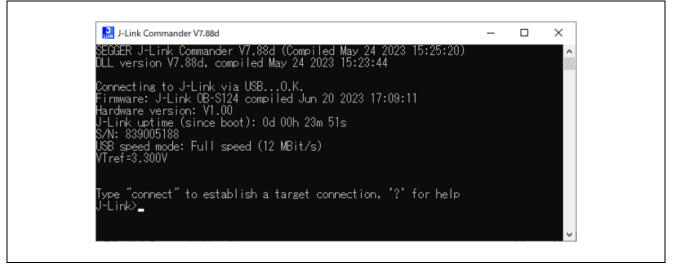


### Appendix. How to Erase Flash Memory

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 (e<sup>2</sup> 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 110 : 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 22 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
RZN2H Evaluation Board	R9A09G087M44_R52_0

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



🔜 J-Link Commander V7.88d	_	×
Type "connect" to establish a target connection, '?' f J-Link>connect Please specify device / core. <default>: R9A07G075M24_</default>		
Type '?' for selection dialog Device>R9A07G075M24 CPU0	<u>_</u> CI 00	
Please specify target interface: J) JTAG (Default)		
S) SWD T) cJTAG TIF>S		
Specify target interface speed [kHz]. <default>: 4000 Speed&gt;</default>	kHz	
Device "R9A07G075M24_CPU0" selected.		

Figure 111 : Initial Setup for Connecting to the Device

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

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: EL0: AArch32		^
FPU support: Single + Double + Conversion Add. info (CPU temp. halted) Current exception level: EL2 Exception level AArch usage:		
Current exception level: EL2 Exception level AArch usage:		
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 112 : 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)



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	0x60000000	0x63FFFFFF
	16-bit bus	CS0	0x70000000	0x71FFFFFF
RZN2H Evaluation Board	xSPI0 x1	xSPI0 CS0	0x40000000	0x47FFFFFF
	xSPI1 x1	xSPI1 CS0	0x50000000	0x57FFFFFF

Table 23 External Address Space to Be Used in Each Boot Mode



Figure 113 : Specify Erase Range

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

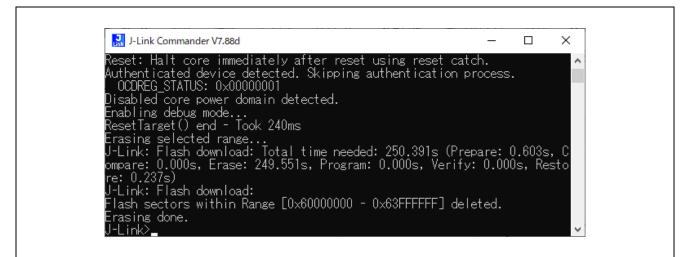


Figure 114: Message of Flash Memory Erase Complete

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 115 : Open Workspace for IAR EWARM

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

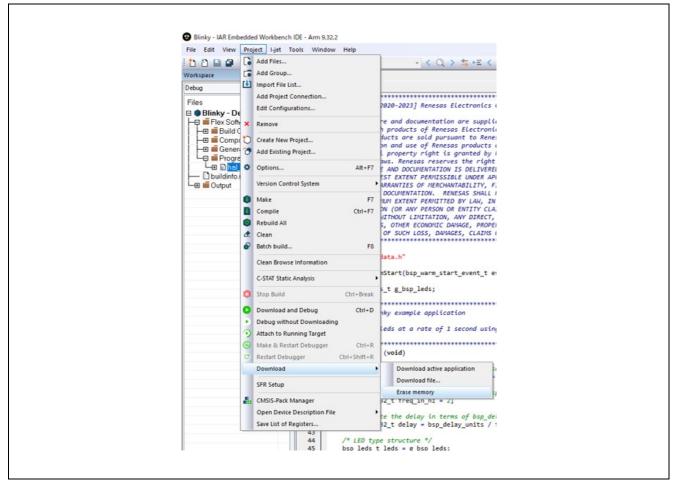


Figure 116 : Select Erase memory Command



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 117 : 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 118 : 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\JAR Systems\Embedded Workbench 9.32.2\	
arm/config/tlashloader/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\\AR 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 Hjet driver	
	~
Build Debug Log	

Figure 119 : 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 Project Content
Select Pin Configuration	🔛 Export to CSV file 🛛 Configure Pin Driver Warnings
RSK+RZT2M     Manage       Image: Configuration     Image: Configuration       Pin Selection     Image: Configuration       Type filter text     Multiple Pin Configuration       Image: Configuration     Image: Configuration <th>Management r import/export external file  Add  Remove  X</th>	Management r import/export external file  Add  Remove  X

Figure 120: How to Rename Pin Configuration



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

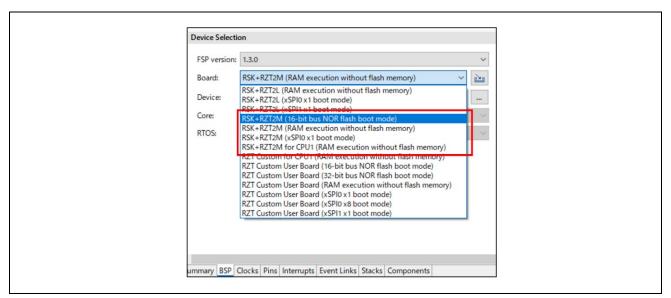


Figure 121: Change the Boot Mode in the BSP Tab

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

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

FSP version:	1.3.0	$\sim$
board.	1.3.0	_
Device:	R9A07G075M24GBG	
Core:	CR52_0	$\sim$
RTOS:	No RTOS	$\sim$

Figure 122: 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  V Manage	
Generate data: g_bsp_pin_cfg	

Figure 123: Uncheck Generate data in the Pins Tab

5. Select the regenerated configuration for the board.

🔅 *[sample] FSP Configuration $ imes$	
Pin Configuration	
Select Pin Configuration	
RSK+RZT2M_OLD	✓ Manage
RSK+RZT2M_OLD	
R9A07G075M24GBG.pincfg RSK+RZT2M	

Figure 124: Select the Regenerated Configuration for the Board

6. Check "Generate data" again and enter "g\_bsp\_pin\_cfg" as the name.

Image: Image
Pin Configuration
Select Pin Configuration
RSK+RZT2M V
Generate data: g_bsp_pin_cfg

Figure 125: Check Generate data again



# Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e<sup>2</sup> 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.

(Same core type in N and M): In multiprocessing, the same type of cores in the N and M projects(N, M = pri(primary), sec(secondary), ter(tertiary)).

(Different core types in N and M): In multiprocessing, the different type of cores in the N and M projects(N, M = pri(primary), sec(secondary), ter(tertiary)).

# For RZ/T2 FSP v3.0.0

## Multiprocessing with 2 cores for RZ/T devices

- 1. Create a primary project according to the procedures in 4.3 Create a New Project for Blinky.
  - (RZ/T2H CA55 Core1) To blink the LED on the RZ/T2H CA55 Core1, you need to change the pin configuration of the primary project in the 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. Build the secondary project according to the procedures in 4.4.1 Build.
  - (Flash boot) The following object files are output to the Debug folder of the secondary project.

Device	Project core	Object files	Note
RZ/T2M	CR52	secondary_CR52.0	
RZ/T2ME		secondary_noncache_CR52.o	When using a noncache sections
RZ/T2H	CR52	secondary_atcm_CR52_0.o	For CR52 CPU0
		secondary_btcm_CR52_0.o	For CR52 CPU0
		secondary_atcm_CR52_1.o	For CR52 CPU1
		secondary_btcm_CR52_1.o	For CR52 CPU1
		secondary_systemram_CR52.o	For a multi-core project with 3 or more cores
		secondary_CR52.0	When placing a program in System SRAM
			instead of TCM
		secondary_noncache_CR52.o	When using a noncache sections
	CA55	secondary_CA55.o	
		secondary noncache CA55.o	When using a noncache sections

#### Table 24 Object files Output to the Debug Folder of the Secondary Project



- 6. (Flash boot) (Different core types in pri and sec) The following additional properties must be set.
  - i. In the Properties window of the primary project, click C/C++ Build > Settings > Build Steps.
  - ii. Add **Command(s)** at **Pre-build steps**. sh ../script/prebuild.sh ../../[the secondary project name]/Debug

Properties for Blinky_cpu0_p	rimary	- <b>D</b> X
type filter text	Settings	← ▼ ⇔ ▼ 8
> Resource Builders V C/C++ Build Build Variables Environment	Configuration: Debug [ Active ]	Manage Configurations
JSON Compilation Datab Logging	🛞 Tool Settings 🛞 Toolchain 🛞 Device 🎤 Build Steps 🟆 Build Artifact 🖥	Binary Parsers "
Settings Tool Chain Editor	Command(s):	
> C/C++ General > MCU	sh/script/prebuild.sh//Blinky_cpu1_secondary/Debug	~
Project Natures Project References	Description:	
Renesas QE		
Run/Debug Settings Task Tags	Post-build steps Command(s):	
> Validation		~
	Description:	
		~
	<	~
< >	×	,
?	App	ly and Close Cancel

Figure 126 e<sup>2</sup> studio Build Setting for the Primary Project (Flash Boot) (Different core types in pri and sec)

7. (Flash boot) (Different core types in pri and sec) Build the primary project according to the procedures in 4.4.1 Build. The following object files are output to the Debug folder of the secondary project.

Primary project core	Secondary project core	Object files	Note
CA55	CR52	secondary_atcm_aarch64_CR52_0.o	For CR52 CPU0
		secondary btcm aarch64 CR52 0.0	For CR52 CPU0
		secondary atcm aarch64 CR52 1.0	For CR52 CPU1
		secondary btcm aarch64 CR52 1.0	For CR52 CPU1
		secondary_systemram_aarch64_CR52.o	For a multi-core project with 3 or more
			cores
		secondary_aarch64_CR52.o	When placing a program in System SRAM instead of TCM
		secondary_aarch64_noncache_CR52.o	When using a noncache sections
CR52	CA55	secondary_aarch32_CA55.o	
		secondary_noncache_aarch32_CA55.o	When using a noncache sections

Table 25 Obi	iect files Output to	o the Debug Folder	• of the Secondar	v Proiect
	feet mes output t	o the Debug I bluer	of the Secondar	y I I Ujece



- 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 file paths of object files in the secondary project to **Other objects**.

Table	26	Example	of File	Paths to	Be	Set to	Other	Objects
								J

Device	Primary project core	Secondary project core	File path	Note
RZ/T2M RZ/T2ME	CR52 CPU0	CR52 CPU1	\${workspace_loc:/Blinky_cpu1_secondary/Debug/secondary_CR52.o}	
			\${workspace_loc:/Blinky_cpu1_secondary/Debug/secondary_noncache_CR52.0}	Import only if file is output
RZ/T2H	CR52 CPU0	CR52 CPU1	<pre>\${workspace_loc:/Blinky_cpu1_secondary/Debug/seco ndary atcm_CR52_l.o}</pre>	
			\${workspace_loc:/Blinky_cpu1_secondary/Debug/secondary_btcm_CR52_1.o}	
			\${workspace_loc:/Blinky_cpu1_secondary/Debug/secondary_noncache_CR52.0}	Import only if file is output
	CA55 Core0	CA55 Core1	<pre>\${workspace_loc:/Blinky_cpu1_secondary/Debug/seco ndary_CA55.o}</pre>	
			<pre>\${workspace_loc:/Blinky_cpu1_secondary/Debug/seco ndary noncache CA55.o}</pre>	Import only if file is output

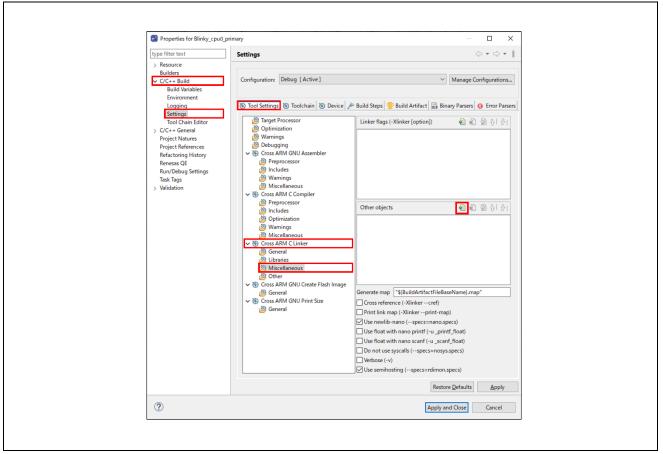


Figure 127 e<sup>2</sup> studio Build Setting for the Primary Project (Part 1)



#### Figure 128 e<sup>2</sup> studio Build Setting for the Primary Project (Part 2)

- 9. (Flash boot) Build the primary project.
- 10. Debug the projects according to the procedures in 4.7 Debug and Run for Multiprocessing.
  - Check the debug configuration in the No. 3 procedure of 4.5.2 Debug Steps.

Item	Project core	Primary project		Secondary o project	or later
		RAM exec	Flash boot	RAM exec	Flash boot
Reset at the beginning of connection	CR52 CPU0	Yes		No	
	CA55 Core0				
	Other cores			Yes	
Reset after download	-	No	Yes	No	
Set CPSR(5bit) after download	CR52 CPU0	Yes	No	Yes	
	Other cores	No		No	



Create, manage, and run configurations  Create, manage, and run configurations  Main Constant of the state of	Image: Secondary Debug. Flat         Image: Secondary Debug. Flat <td< th=""><th>Image: Image: Image:</th><th>Image: Image: Image:</th><th>Debug Configurations</th><th></th><th></th><th></th></td<>	Image:	Image:	Debug Configurations			
Type filter text         Image: Constant Participation           © CC++ Application         Debug hardware           © CGB Settings         Connection Settings           © CGB Advance Debugging ()         Debug hardware           © Lance Line Application         Part LAN           © Reness CB Hardware Debugging ()         Debug hardware           © Reness CB Hardware Debugging ()         Mumber           Part Streep ()         Streep ()           V TAG Scan Chain         Mumber           Perser at the beginning of connection         No           V TAG Scan Chain         No           Publication         V           Reness Simulator Debugging         Multiple Devices           No         V           Reference ()         Device ()           Provent Networe ()         No           V TAG Scan Chain         No	Type filter text         Image: Constant Participation           © CC++ Application         Debug hardware           © CGB Settings         Connection Settings           © CGB Advance Debugging ()         Debug hardware           © Lance Line Application         Part LAN           © Reness CB Hardware Debugging ()         Debug hardware           © Reness CB Hardware Debugging ()         Mumber           Part Streep ()         Streep ()           V TAG Scan Chain         Mumber           Perser at the beginning of connection         No           V TAG Scan Chain         No           Publication         V           Reness Simulator Debugging         Multiple Devices           No         V           Reference ()         Device ()           Provent Networe ()         No           V TAG Scan Chain         No	Type filter text           © OCC++ Application           © OCD Absorption           © Secondary Debug Table           © Absorption           © Debug Name Comparison           © Debug Name Comparison           © Debug Name Debug Table           © Debug Name Debug Na	Type filter text         Image: Construction Statup         Statup         Statup         Construction           Image: Construction Statup         Image: Construction Statup         Target Device         RAdoSG77M44_CR5         Image: Construction Statup           Image: Construction Statup         Image: Construction Statup         Debug hardware         Image: Construction Statup         Image: Constatup         Image: Constatup	Create, manage, and run configu	rations		Ť
Image: Construction       Debug hardware       Jelink ABM ▼       Target Device       RA03G077M44_CKS       Image: Construction         Image: Construction       Debug hardware       Jelink ABM ▼       Target Device       RA03G077M44_CKS       Image: Construction         Image: Construction       Debug hardware       Debug hardware       Debug hardware       Debug hardware         Image: Construction       Debug hardware       Debug hardware       Debug hardware       Image: Construction         Image: Construction       Debug hardware       Debug hardware       Image: Construction       Image: Construction       Image: Construction         Image: Construction       Debug hardware       Image: Construction       Image: Constructio	Image: Construction       Debug hardware       Jelink ABM ▼       Target Device       RA03G077M44_CKS       Image: Construction         Image: Construction       Debug hardware       Jelink ABM ▼       Target Device       RA03G077M44_CKS       Image: Construction         Image: Construction       Debug hardware       Debug hardware       Debug hardware       Debug hardware         Image: Construction       Debug hardware       Debug hardware       Debug hardware       Image: Construction         Image: Construction       Debug hardware       Debug hardware       Image: Construction       Image: Construction       Image: Construction         Image: Construction       Debug hardware       Image: Construction       Image: Constructio	C C++ Application         E C5 C++ Renote Application         B ASS Script         C G0B Andware Debugging         C G0B Senulator Debugging         C G0B Andware Debugging         C Bass Script         C G0B Andware Debugging         C Bass Script         C G0B Andware Debugging         C Bass Script         C Bass Script </th <th>C C++ Application         C C++ Renote Application         B ASS Script         G GB Standware Debugging         G GB Sender Debugging         G GB Standware Debugging         G BA Andware Debugging         G BA Standware Debugging</th> <th>📑 🖻 🗫 🗎 🗮 🏹 🕶</th> <th>Name: Secondary Debug_Flat</th> <th></th> <th></th>	C C++ Application         C C++ Renote Application         B ASS Script         G GB Standware Debugging         G GB Sender Debugging         G GB Standware Debugging         G BA Andware Debugging         G BA Standware Debugging	📑 🖻 🗫 🗎 🗮 🏹 🕶	Name: Secondary Debug_Flat		
CCC++ Remote Application     CCCC++ Remote Application     CCCCC++ Remote Application     CCCCCC++ Remote Application     CCCCC++ Remote Application     CCCCCCC+++++++++++++++++++++++++++	CCC++ Remote Application     CCCC++ Remote Application     CCCCC++ Remote Application     CCCCCC++ Remote Application     CCCCC++ Remote Application     CCCCCCC+++++++++++++++++++++++++++	Image: Construction is Constructed in the provided in theter in theter in the provided in theter in the provid	C/C++ Application         C/C++ Rendet Application         A EAS Stript         C/DB Hardware         C/DB Strings	type filter text	Main Sebugger > Startup	Common	
CORD CDP OPCOD Debugging ()       Connection Method       IP via LAN       Image: Connection Method       Image: Connection Monection Monec	CORD CDP OPCOD Debugging ()       Connection Method       IP via LAN       Image: Connection Method       Image: Connection Monection Monec	COB® CDP Obcugging ()       Connection Method       P via LAN       ✓         COB® CopenCCD Debugging ()       Hot Shame/P Address(port number)       Hot Shame/P Address(port number)       Hot Shame/P Address(port number)         Discond Spletation       Lianch Group       Fortamer Line Application       Prise LAN       ✓         Premeta Java Application       Lianch Group       Prise Lanch Copencies       No       ✓         Premeta Java Application       Fortamer Line Server       Fortamer Line Server       Fortamer Line Server       Fortamer Line Server         Premeta Java Application       Fortamer Line Server       Fortamer Line Server       Fortamer Line Server       Fortamer Line Server         Premeta Java Application       Fortamer Line Server       Fortamer Line Server       Fortamer Line Server         Premeta Java Application       Vinteface       SWD       ✓         Type       SWD       ✓       Second Line Server         Prevent Releasing of connection       No       ✓       Multiple Devices         Prevent Releasing of connection       No       ✓       Feet after Gownload       No         Prevent Releasing the Reset of the CM3 Core       Yes       ✓       Secure Vector Address       ✓         Hold reset during connection Mode       Continue       ✓       ✓	Conscion Mehod P via LAN  Conscion Port Number  Port Num	C/C++ Remote Application EASE Script GDB Hardware Debugging	Debug hardware: J-Link ARM V Target Device:	R9A09G077M44_CR5	
C       GOB Simulator Obeloging ()         Java Applet       Java Applet         Q       Remess CDB Handmar Debu         C       Primary Debug flat         C       Remess Simulator Debugging ()         V       Interface         Q       V         Q       Reness Simulator Debugging ()         V       Interface         Q       V         Reness Simulator Debugging ()       V         V       Interface         Q       OBPre       O         V       DiPre       O         V       DiPre       O         V       Rest at the beginning of connection       No         V       Rest at the beginning of connection       No <t< td=""><td>C       GOB Simulator Obeloging ()         Java Applet       Java Applet         Q       Remess CDB Handmar Debu         C       Primary Debug flat         C       Remess Simulator Debugging ()         V       Interface         Q       V         Q       Reness Simulator Debugging ()         V       Interface         Q       V         Reness Simulator Debugging ()       V         V       Interface         Q       OBPre       O         V       DiPre       O         V       DiPre       O         V       Rest at the beginning of connection       No         V       Rest at the beginning of connection       No         <t< td=""><td>© 300 Simulator Obeloging ()       I/va Application         Ø Java Application       Journal Server         © Lunch Group       Personor         © Remote Java Application       Port Number         © Remote Java Application       Vinterface         Start Remote Java Application       Vinterface         © Remese Java Application       Vinterface         © Remese Java Application       Vinterface         Start Remease Simulator Debuggiti       Vinterface         © Remese Java Application       Vinterface         V Interface       SWD         © Remese Java Application       Vinterface         V ITAG Scan Chain       Vinterface         © Remese Java Application Connection       No         Peset Athered Connection       No         Peset Athered Connection       No         V Di Code (Bynel)       Vinterfiftererererefiftererererererererererererererererererer</td><td>Codd Simulator Debugging ()     I/va Application       I/va Application     Joint Application       Construction     Port Number       Person Linux Application     Port Number       Primary Debug Flat     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Person Linux Application     V       V Interfac     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     O       Primary Debug Flat     O</td><td>C GDB OpenOCD Debugging</td><td></td><td></td><td>× •</td></t<></td></t<>	C       GOB Simulator Obeloging ()         Java Applet       Java Applet         Q       Remess CDB Handmar Debu         C       Primary Debug flat         C       Remess Simulator Debugging ()         V       Interface         Q       V         Q       Reness Simulator Debugging ()         V       Interface         Q       V         Reness Simulator Debugging ()       V         V       Interface         Q       OBPre       O         V       DiPre       O         V       DiPre       O         V       Rest at the beginning of connection       No         V       Rest at the beginning of connection       No <t< td=""><td>© 300 Simulator Obeloging ()       I/va Application         Ø Java Application       Journal Server         © Lunch Group       Personor         © Remote Java Application       Port Number         © Remote Java Application       Vinterface         Start Remote Java Application       Vinterface         © Remese Java Application       Vinterface         © Remese Java Application       Vinterface         Start Remease Simulator Debuggiti       Vinterface         © Remese Java Application       Vinterface         V Interface       SWD         © Remese Java Application       Vinterface         V ITAG Scan Chain       Vinterface         © Remese Java Application Connection       No         Peset Athered Connection       No         Peset Athered Connection       No         V Di Code (Bynel)       Vinterfiftererererefiftererererererererererererererererererer</td><td>Codd Simulator Debugging ()     I/va Application       I/va Application     Joint Application       Construction     Port Number       Person Linux Application     Port Number       Primary Debug Flat     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Person Linux Application     V       V Interfac     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     O       Primary Debug Flat     O</td><td>C GDB OpenOCD Debugging</td><td></td><td></td><td>× •</td></t<>	© 300 Simulator Obeloging ()       I/va Application         Ø Java Application       Journal Server         © Lunch Group       Personor         © Remote Java Application       Port Number         © Remote Java Application       Vinterface         Start Remote Java Application       Vinterface         © Remese Java Application       Vinterface         © Remese Java Application       Vinterface         Start Remease Simulator Debuggiti       Vinterface         © Remese Java Application       Vinterface         V Interface       SWD         © Remese Java Application       Vinterface         V ITAG Scan Chain       Vinterface         © Remese Java Application Connection       No         Peset Athered Connection       No         Peset Athered Connection       No         V Di Code (Bynel)       Vinterfiftererererefiftererererererererererererererererererer	Codd Simulator Debugging ()     I/va Application       I/va Application     Joint Application       Construction     Port Number       Person Linux Application     Port Number       Primary Debug Flat     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Person Linux Application     V       V Interfac     SVD       Speed (Hz)     4000       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     SVD       Primary Debug Flat     O	C GDB OpenOCD Debugging			× •
■ Java Applet       Java Applet         Java Applet       Tume Sever         ■ Reneas GGB Hardware Deb       For Number         ■ Reneas GGB Hardware Deb       For Number         ● Secondary Debug Fixit       Type         ● Secondary Debug Fixit       Symbol         ● Neness Simulator Debuggit       That Secondary         ● Renease Simulator Debuggit       Multiple Device       No         ● DRPre       0         ○ DRPre       0         ● Renet at the beginning of connection       No         ■ Register initialization       No         ● Rest at the beginning of connection       No<	■ Java Applet       Java Applet         Java Applet       Tume Sever         ■ Reneas GGB Hardware Deb       For Number         ■ Reneas GGB Hardware Deb       For Number         ● Secondary Debug Fixit       Type         ● Secondary Debug Fixit       Symbol         ● Neness Simulator Debuggit       That Secondary         ● Renease Simulator Debuggit       Multiple Device       No         ● DRPre       0         ○ DRPre       0         ● Renet at the beginning of connection       No         ■ Register initialization       No         ● Rest at the beginning of connection       No<	Image: Start Star	■ Java Applet       ■ Reneas GGB Hardware Deb       ● Primary Debug Flat       ● Secondary Debug Flat       ● Secondary Debug Flat       ● Reneas Simulator Debug flat       ● DRPre       ● DCRPre				
↓ Lunch Group       Port Number         ② Reneas SG0 Hardware Obt       Password         ● The Reneas SG0 Hardware Obt       > Inteffece         ③ Secondary Debug Jiat       Type         ③ Reneas Simulator Debuggit       > Inteffece         ● The Reneas Simulator Debuggit       No         ● Reneas Simulator Debuggit       > Inteffece         ● DRFe       0         ○ DRFre       0         ● Renease the mod pictometion       > No         ● Renease the mod pictometion       > No         ● Renease the mod pictometion       > No         ● Disclose (bjts)       + Renease the mod pictometion         ● Renease the mod pictometion       > No         ● Disclose (bjts)       + Rest et dire download         No       > No         ● Disclose (bjts)       + Rest et dire download         Non-secure Vector Address       >         Hold Plag       No       >	↓ Lunch Group       Port Number         ② Reneas SG0 Hardware Obt       Password         ● The Reneas SG0 Hardware Obt       > Inteffece         ③ Secondary Debug Jiat       Type         ③ Reneas Simulator Debuggit       > Inteffece         ● The Reneas Simulator Debuggit       No         ● Reneas Simulator Debuggit       > Inteffece         ● DRFe       0         ○ DRFre       0         ● Renease the mod pictometion       > No         ● Renease the mod pictometion       > No         ● Renease the mod pictometion       > No         ● Disclose (bjts)       + Renease the mod pictometion         ● Renease the mod pictometion       > No         ● Disclose (bjts)       + Rest et dire download         No       > No         ● Disclose (bjts)       + Rest et dire download         Non-secure Vector Address       >         Hold Plag       No       >	Lunch Group Q. Renets Na Application Prev Number Pars Number	Lunch Group Reneas SGB Hardware Deb Part Number P				
<ul> <li></li></ul>	<ul> <li></li></ul>		<ul> <li>Remote Java Application         <ul> <li>Passance</li> <li>Privaty Debug Fat</li> <li>Speci (H12)</li> <li>Speci (H12)</li> <li>Reneass Simulator Debuggir</li> <li>INS Can Chain</li> <li>Multiple Devices</li> <li>No</li> <li>INFPe</li> <li>DROW</li> <li>Rest at the exploring of connection</li> <li>No</li> <li>Rest at the connolad</li> <li>No</li> <li>Rest at the connolad</li> <li>No</li> <li>Secure Vector Address</li> <li>Hold rest during connect</li> <li>No-</li> <li>Secure Vector Address</li> <li>Hold Plug</li> <li>No</li> <li>Disconnection Mode</li> <li>Continue</li> <li>V</li> </ul> </li> </ul>		Tunnel Server		
▼       Reneas GDB Hardware Deb         ■       Primary Debug Jiat         ▼       Secondary Debug Jiat         ▼       TAG Scan Chain         Multiple Device       No         ■       Primary Debug Jiat         ■       Multiple Device         ■       No         ■       DRPre         ■       DRPre         ■       Register initialization         ■       Register initediation         ■	▼       Reneas GDB Hardware Deb         ■       Primary Debug Jiat         ▼       Secondary Debug Jiat         ▼       TAG Scan Chain         Multiple Device       No         ■       Primary Debug Jiat         ■       Multiple Device         ■       No         ■       DRPre         ■       DRPre         ■       Register initialization         ■       Register initediation         ■	▼       Reneass CBC Hardware Deb         ●       Interface         ▼       Secondary Debug; Flat         ●       Secondary Debug; Flat         ●       Interface         ●       Interface         Secondary Debug; Flat       No         ●       Interface         ●       Interface         ●       Interface         ●       Secondary Debug; Flat         ●       No         ●       Interface         ●       No         ●       DRPre         0       OFfre         0       No         ●       Rest at the deal of connection         No       ∨         Perset at the deal of connection       No         ●       No         ●       Exet at the deal of connection         No       ∨         Perset at the download       No         ∨       No         ●       Exet at the download         No       ∨         ■       Exet at the download         No       ∨         ■       Exet of the CM3 Core         ●       Exet of the CM3 Core         ●	▼     Reneas GDB Hardware Deb       ●     Primary Debug Jia       ▼     Secondary Debug Jia       ▼     TAG Scan Chain       Multiple Devices     No       ▼     No       ■     Brees as Simulator Debug Jia       ■     No       ■     Bree       ■     Brees as Simulator Debug Jia       ■     No       ■     Brees at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the beginning of connection       No     ■       ■     Breest at the		Port Number		
Primary Debug Flat     Speci (kHz)     Vita	Primary Debug Flat     Speci (kHz)     Vita	Pinary Debug Fiat     Speci (ktz)     Spe	Primary Debug Flat Process Simulator Debuggi Renease Simulator Debuggi Register initialization Register initializatio	Remote Java Application	Password		
Secondary Debugg Fiel     Speed (kHz)	Secondary Debugg Fiel     Speed (kHz)	Secondary Debugg Fielt       Speed (kHz)       4000       ∨         Speed (kHz)       4000       ∨         Multiple Devices       No       ∨         BPre       0       DBPre       0         Connection       No       ∨         Reset at the beginning of connection       No       ∨         Reset at the consol       No       ∨         Reset at the consol connection       No       ∨         Breet at the consol connection       No       ∨         Hold reset during connect       No       ∨         Secure Vector Address       ∨ <td>Secondary Debugg Fiel     Speed (Hz) 4000     V     JAG Scan Dalin     Multiple Devices No     Montple Devices No     Rest at the beginning of connection     Rest at the beginnit of connection</td> <td></td> <td>✓ Interface</td> <td></td> <td></td>	Secondary Debugg Fiel     Speed (Hz) 4000     V     JAG Scan Dalin     Multiple Devices No     Montple Devices No     Rest at the beginning of connection     Rest at the beginnit of connection		✓ Interface		
Renesas Simulator Debuggi	Renesas Simulator Debuggi	C Renesas Simulator Debuggir   V TAG Scan Chain   Multiple Devices   No   BiPre   O DR?re   Connection   Register initialization   No   Rest at the explaining of connection   No   V Rest at the explaining of connection   No   V Rest at the explaining of connection   No   V Rest at the devices   No   V Rest at the download   No   V Dicode (bytes)   Prevent Releasing the Rest of the CM3 Core   Yes   Secure Netor Address   Non-secure Vector Address   Hot Plug   Disconnection Mode	C Renesas Simulator Debuggir    TAG Scan Chain    Mittiple Devices   No   BitPe   ORIVE   Connection   Registeri initialization   No   Rest at the expining of connection   No   Prevent Releasing connect   No   Secure Vector Address   Non-secure Vector Address   Non-secure Vector Address   Non-secure Vector Address   Norte Disconnection Mode   Continue		Туре	SWD	~
Multiple Devices     No     v       INPre     0     DRTret       DRTret     0     V       Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the download     No     v       Rest at the download     No     v       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFF     univ       Hold read during connect     No     v       Secure Ketor Address     V     V       Non-secure Vector Address     V     V       Hold Pulg     No     V	Multiple Devices     No     v       INPre     0     DRTret       DRTret     0     V       Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the download     No     v       Rest at the download     No     v       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFF     univ       Hold read during connect     No     v       Secure Ketor Address     V     V       Non-secure Vector Address     V     V       Hold Pulg     No     V	Multiple Devices     No     v       IRPre     0     0       DRFre     0     0       V Connection     No     v       Register institutation     No     v       Reset at the beginning of connection     No     v       Reset at the deal of connection     No     v       Reset at the deal of connection     No     v       Reset at the deal of connection     No     v       Reset at the download     No     v       ID Code (Bytes)     Fffffffffffffffffffffffffffffffffffff	Multiple Devices     No       INPre     0       DRive     0       Connection     No       Register initialization     No       Register initialization     No       Register initialization     No       Register initialization     No       Register at the beginning of connection     No       Register at the download     No       V     Register download       No     V       Register download     No       V     Disconnect in the construction       No     V       Register download     No       V     Prevent Releasing the Register download       V     Prevent Releasing the Register download       V     Secure Vector Address       Non-secure Vector Address       Non-secure Vector Address       Not Plug     No       Disconnection Mode     Continue		Speed (kHz)	4000	~
IBPre         0           DBPre         0           Connection         No           Reset at the beginning of connection         No           Reset at the beginning of connection         No           Reset at the dot connection         No           Preset at the dot connection         No           V         Reset at the dot connection           No         V           Reset at the dot connection         No           V         Reset at the dot connection           No         V           Reset at the dot connection         No           V         Operation           No         V           Reset at the dot connection         No           V         Operation           No         V           Reset at the dot connection         No           V         Secure Vector Address           Non-secure Vector Address         V           Hot Plug         No	IBPre         0           DBPre         0           Connection         No           Reset at the beginning of connection         No           Reset at the beginning of connection         No           Reset at the dot connection         No           Preset at the dot connection         No           V         Reset at the dot connection           No         V           Reset at the dot connection         No           V         Reset at the dot connection           No         V           Reset at the dot connection         No           V         Operation           No         V           Reset at the dot connection         No           V         Operation           No         V           Reset at the dot connection         No           V         Secure Vector Address           Non-secure Vector Address         V           Hot Plug         No	IBPre 0 DPPre 0 Connection No  Register initialization No  Register initialization No  Rest at the beginning of connection No  Rest at the editor download No  Rest at the doff of connection No  Rest at at dre download No  Rest at at dre download No  Rest at at dre download No  PEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	BPre     O     DPPre     O     Connection     Register initialization     No     Rest at the beginning of connection     No     Rest at the deginning of connection     Rest at the deginning of connection     No     Rest at the deginning of connection	C Renesas Simulator Debuggir	✓ JTAG Scan Chain		
DBPre     0       Connection     No       Register initialization     No       Rest at the beginning of connection     No       Rest at the dorf connection     No       Rest at the dorf connection     No       Peest at dorf dorf dorf     No       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	DBPre     0       Connection     No       Register initialization     No       Rest at the beginning of connection     No       Rest at the dorf connection     No       Rest at the dorf connection     No       Peest at dorf dorf dorf     No       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	DBPre     0       Connection     No       Register initialization     No       Register initinitiali	DBPre     0       Connection     No       Register initialization     No       Rest at the beginning of connection     No       Rest at the download     No       V     Rest at download       No     V       Rest at download     No       V     Rest at download       No     V       Rest at download     No       V     Rest at download       No     V       Rest set during connect     No       V     Rest set during connect       No     V       Secure Vector Address     V       Non-secure Vector Address     V       Nor-secure Vector Address     V       Disconnection Mode     Continue				~
Connection     No       Reset at the beginning of connection     No       Reset at the beginning of connection     No       Reset at the connection     No       Reset atter download     No       No     V       Reset atter download     No       No     V       Reset atter download     No       V     V       Discle Etytes;     V       Secure Vector Address     Ves       Non-secure Vector Address     Ves       Hot Plug     No	Connection     No       Reset at the beginning of connection     No       Rest at the beginning of connection     No       Rest at the construction     No       Rest atter download     No       No     V       Rest atter download     No       V     To Cost Efficiency       No     V       Rest atter download     No       V     To Cost Efficiency       No     V       Secure Vector Address     Ves       Non-secure Vector Address     No       Hot Plug     No	Connection     No       Register initialization     No       Rest at the beginning of connection     No       Rest at the end of connection     No       Rest at the decision     No       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Connection     Register initialization     Register initinitialization     Register initialization     Regist				
Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the and of connection     No     v       Rest at dre and of connection     No     v       Rest atter download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the and of connection     No     v       Rest at dre and of connection     No     v       Rest atter download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Register initialization     No     v       Rest at the segning of connection     No     v       Rest at the end of connection     No     v       Rest after download     No     v       ID Code (lybrc)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the end of connection     No     v       Rest at the download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			0	
Rest at the beginning of connection     No     v       Rest after end of connection     No     v       Rest after download     No     v       Dicode (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the beginning of connection     No     v       Rest after end of connection     No     v       Rest after download     No     v       Dicode (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the beginning of connection     No     v       Rest at the beginning of connection     No     v       Rest before download     No     v       Dicate (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the beginning of connection No      Peet at the beginning of connection No      Peet at the connection No      Peet atter download No      Peet atter atter download No      Peet atter download No      Peet atter d				
Rest at the end of connection     No     v       Rest before download     No     v       Rest at the download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the end of connection     No     v       Rest before download     No     v       Rest at the download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the end of connection     No     v       Rest after download     No     v       ID Code (lyhet)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the smd of connection No v Rest before download No v Rest before download No v ID Code (Bytes) FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Rest before download     No     v       Rest and download     No     v       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest before download     No     v       Rest and download     No     v       ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reset before download     No     v       Reset after download     No     v       DC Gode (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest Hefore download     No     Ve     Rest Hefore download     No     Ve     D Code (Bytes)     D Code (Bytes)     Hold reset download     No     Ve				
Reset after download     No     Y       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reset after download     No     Y       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reset after download     No     Y       ID Cost lighter)     IF IFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reet after download     No     Y       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	ID Code (bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Code (Bytes)     FIFIFFIFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Hold reset during connect No v Set CPSRISbit after download Ves v Prevent Releasing the Reset of the CM3 Core Ves v Secure Vector Address Non-secure Vector Address Hot Plug No v	Hold reset during connect No v Set CPSRISbit after download Ves v Prevent Releasing the Reset of the CM3 Core Ves v Secure Vector Address Non-secure Vector Address Hot Plug No v	Hold reset during connect     No     V       Set CPSIBDI after download     Ves     V       Prevent Releasing the Reset of the CM3 Core     Ves     V       Secure Vector Address     V     V       Non-secure Vector Address     V     V       Disconnection Mode     Continue     V	Hold reset during connect     No     V       Get CPSRIght after download     Yes     V       Preven Releasing the Reset of the CM3 core     Yes     V       Secure Vector Address     No     V       Non-secure Vector Address     No     V       Disconnection Mode     Continue     V				~
Set CPSR Sbit) After download         Yes         Yes           Prevent Releasing the Reset of the CM3 Core         Yes         Yes           Secure Yestor Address         Hot Prug         No	Set CPSR Sbit) After download         Yes         Yes           Prevent Releasing the Reset of the CM3 Core         Yes         Yes           Secure Yestor Address         Hot Prug         No	Set CPSR(Sbit) after download     Ves     V       Prevent Releasing the Rest of the CM3 Core     Yes     V       Secure Vector Address     V     V       Non-secure Vector Address     V     V       Hote Plug     No     V       Disconnection Mode     Continue     V	Set CPSRIsbit after download         Yes         Yes           Prevent Releasing the Reset of the CM3 Core         Yes         Yes           Secure Yes         Yes         Yes           Non-secure Yestor Address         Yes         Yes           Hot Plug         No         Yes           Disconnection Mode         Continue         Yes				
Prevent Releasing the Reset of the CM3 Core     Yes     v       Secure Vector Address     Image: Comparison of the CM3 Core     Image: Comparison of the CM3 Core       Non-secure Vector Address     Image: Comparison of the CM3 Core     Image: Comparison of the CM3 Core       Hot Plug     No     v	Prevent Releasing the Reset of the CM3 Core     Yes     v       Secure Vector Address     Image: Comparison of the CM3 Core     Image: Comparison of the CM3 Core       Non-secure Vector Address     Image: Comparison of the CM3 Core     Image: Comparison of the CM3 Core       Hot Plug     No     v	Prevent Releasing the Rest of the CM3 Core Ves v Secure Vector Address Non-secure Vector Address Hot Plug No v Disconnection Mode Continue v v	Prevent Releasing the Rest of the CM3 Core Ves      Secure Vector Address     Non-secure Vector Address     Non-secure Vector Address     No      Disconnection Mode      Continue      v v				
Secure Vector Address Non-secure Vector Address Hot Plug No v	Secure Vector Address Non-secure Vector Address Hot Plug No v	Secure Vector Address Address Address Continue Volume Vector Address Continue Volume V	Secure Vector Address				~
Non-secure Vector Address	Non-secure Vector Address	Non-secure Vector Address Hete Plug No v Disconnection Mode Continue v v	Non-secure Vector Address Hote Plug No v Disconnection Mode Continue v v			Yes	
Hot Plug No 🗸	Hot Plug No 🗸	Hot Plug No v Disconnection Mode Continue v v	Hot Plug No v Disconnection Mode Continue v v				
		Disconnection Mode Continue v v	Disconnection Mode Continue v v				
Disconnection Mode Continue V V	Disconnection Mode Continue v v		C >				
			Period Apply		Disconnection Mode	Continue	~ ~

Figure 129 e<sup>2</sup> studio Debug Configurations for Multiprocessing



- (Flash boot) The flash boot mode differs from RAM execution without flash memory. Both the primary and secondary project binaries are downloaded to the device when connecting debugger with the primary project.
- 11. When changing the project and debugging it again, follow these steps.
  - i. Build the primary project (No. 3).
  - ii. Build the secondary project (No. 5).
  - iii. (Flash boot) Build the primary project again (No. 9).
  - iv. Debug the projects (No. 10).



### Multiprocessing with 3 or more cores for RZ/T devices

This section shows how to perform multi-core debugging using three cores. If more than 4 cores are used, add the process of creating a project, building it and one previous project after No. 3.

- 1. Create and build the primary and secondary projects according to Multiprocessing with 2 cores for RZ/T devices No. 1 to No. 6.
- 2. Create a tertiary project using the bundle file (.sbd) of the secondary project according to the procedures in 4.3 Create a New Project for Blinky.
- 3. Build the tertiary project according to the procedures in 4.4.1 Build.
  - (Flash boot) The following object files are output to the Debug folder of the secondary project.

Device	Project core	Object files	Note
RZ/T2H	CR52	secondary_atcm_CR52_0.o	For CR52 CPU0
		secondary_btcm_CR52_0.o	For CR52 CPU0
		secondary_atcm_CR52_1.o	For CR52 CPU1
		secondary_btcm_CR52_1.o	For CR52 CPU1
		secondary_systemram_CR52.o	For a multi-core project with 3 or more cores
		secondary_CR52.o	When placing a program in System SRAM
			instead of TCM
		secondary_noncache_CR52.o	When using a noncache sections
	CA55	secondary_CA55.o	
		secondary_noncache_CA55.o	When using a noncache sections

#### Table 28 Object files Output to the Debug Folder of the Secondary Project

- 4. (Flash boot) (Different core types in sec and ter) The following additional properties must be set.
  - i. In the Properties window of the secondary project, click C/C++ Build > Settings > Build Steps.

#### ii. Add **Command(s)** at **Pre-build steps**.

- sh ../script/prebuild.sh ../../[the tertiary project name]/Debug
- 5. (Flash boot) (Different core types in sec and ter) Build the secondary project according to the procedures in 4.4.1 Build. The following object files are output to the Debug folder of the tertiary project.

Table 29 Object files O	utnut to the Debug	Folder of the T	Certiary Project
Table 29 Object files O	ulput to the Debug	roluer of the 1	ertiary rroject

Secondary project core	Tertiary project core	Object files	Note
CA55	CR52	secondary_atcm_aarch64_CR52_0.o	For CR52 CPU0
		secondary_btcm_aarch64_CR52_0.o	For CR52 CPU0
		secondary_atcm_aarch64_CR52_1.o	For CR52 CPU1
		secondary_btcm_aarch64_CR52_1.o	For CR52 CPU1
		secondary_systemram_aarch64_CR52.o	For a multi-core project with 3 or more
			cores
		secondary_aarch64_CR52.o	When placing a program in System
			SRAM instead of TCM
		secondary_aarch64_noncache_CR52.o	When using a noncache sections
CR52	CA55	secondary_aarch32_CA55.o	
		secondary_noncache_aarch32_CA55.o	When using a noncache sections



- 6. (Flash boot) Set the following additional properties to the secondary project.
  - i. In the Properties window of the secondary project, click C/C++ Build > Settings > Tool Settings > Cross ARM C Linker > Miscellaneous.
  - ii. Set file paths of object files in the tertiary project to **Other objects**.

Table .	<b>30 Example o</b>	f File Paths	to Be Set to	Other Ol	bjects for 1	the Secondary	Project

Device	Primary project core	Secondary project core	Tertiary project core	File path	Note
RZ/T2H	CA55	CR52	CR52	{workspace loc:/Blinky tertiary/Debug/secon	Import only if
	Core0	CPU0	CPU1	dary noncache CR52.0}	file is output*
	CR52	CA55	CA55	\${workspace_loc:/Blinky_tertiary/Debug/secon	
	CPU0	Core0	Corel	dary_CA55.o}	
				\${workspace_loc:/Blinky_tertiary/Debug/secon	Import only if
				dary_noncache_CA55.o}	file is output
	CR52	<b>CR52</b>	CA55	\${workspace_loc:/Blinky_tertiary/Debug/secon	
	CPU0	CPU1	Core0	dary_aarch32_CA55.o}	
				\${workspace_loc:/Blinky_tertiary/Debug/secon	Import only if
				dary_noncache_aarch32_CA55.o}	file is output

\* Object files for TCM such as secondary\_atcm\_CR52\_1.0 will not work properly if they are imported into a project other than the primary project. Even object files from tertiary projects must be imported into the primary project.

- 7. (Flash boot) Build the secondary project.
- 8. (Flash boot) Set the following additional properties to the primary project.

Table 31 Evam	nla of Fila Path	to Ro Sot to (	Other Objects f	or the Primary Project
Table 31 Exam	ріе от г пе г аш	to be set to t	Julier Objects in	or the Frimary Froject

Device	Primary project core	Secondary project core	Tertiary project core	File path	Note
RZ/T2H	CA55	CR52	CR52	\${workspace_loc:/Blinky_secondary/Debug/sec	
	Core0	CPU0	CPU1	ondary_atcm_aarch64_CR52_0.o}	
				\${workspace_loc:/Blinky_secondary/Debug/sec	
				ondary_btcm_aarch64_CR52_0.o}	
				\${workspace_loc:/Blinky_secondary/Debug/sec	Import only if
				ondary_noncache_aarch64_CR52.o}	file is output
				\${workspace_loc:/Blinky_tertiary/Debug/secon	
				dary_atcm_aarch64_CR52_1.o}	
				\${workspace_loc:/Blinky_tertiary/Debug/secon	
				dary_btcm_aarch64_CR52_1.o}	
	<b>CR52</b>	CA55	CA55	\${workspace_loc:/Blinky_secondary/Debug/sec	
	CPU0	Core0	Core1	ondary_aarch32_CA55.o}	
				\${workspace_loc:/Blinky_secondary/Debug/sec	Import only if
				ondary_noncache_aarch32_CA55.o}	file is output
	<b>CR52</b>	CR52	CA55	\${workspace_loc:/Blinky_secondary/Debug/sec	
	CPU0	CPU1	Core0	ondary_atcm_CR52_1.o}	
				\${workspace_loc:/Blinky_secondary/Debug/sec	
				ondary_btcm_CR52_1.o	
				\${workspace_loc:/Blinky_secondary/Debug/sec	
				ondary_systemram_CR52.0}	
				\${workspace_loc:/Blinky_secondary/Debug/sec	Import only if
				ondary_noncache_CR52.o}	file is output



#### Note:

If the primary project is CA55 and the secondary and tertiary projects are CR52, the following additional properties must be set.

i. In the Properties window of the primary project, click C/C++ Build > Settings > Build Steps.
ii. Add Command(s) at Pre-build steps.

- 9. (Flash boot) Build the primary project.
- 10. Debug the projects according to the procedures in Multiprocessing with 2 cores for RZ/T devices No. 10. Connections are made in the order primary, secondary, tertiary. The tertiary project is connected in the same procedure as secondary project.
- 11. When changing the project and debugging it again, follow these steps.
  - i. Build the primary project (Multiprocessing with 2 cores for RZ/T devices No. 3).
  - ii. Build the secondary project (Multiprocessing with 2 cores for RZ/T devices No. 5).
  - iii. Build the tertiary project (Multiprocessing with 3 or more cores for RZ/T devices No. 3).
  - iv. (Flash boot) Build the secondary project (Multiprocessing with 3 or more cores for RZ/T devices No. 7).
  - v. (Flash boot) Build the primary project again (Multiprocessing with 3 or more cores for RZ/T devices No. 9).
  - vi. Debug the projects (Multiprocessing with 3 or more cores for RZ/T devices No. 10).



sh ../script/prebuild.sh ../../[the secondary project name]/Debug && sh ../script/prebuild.sh ../../[the tertiary project name]/Debug

# For RZ/N2 FSP v2.2.0

## Multiprocessing with 2 cores for RZ/N devices

- 1. Create a primary project according to the procedures in 4.3 Create a New Project for Blinky.
- 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) (Same core type in pri and sec) 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-none-eabi-objcopy -I binary -O elf32-littlearm -B arm --rename-
      - section .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 --rename-section .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o

bype filter text Settings     > Resource   Builders   CCC++ Suild   Build Variables   Environment   Logging   Tool Chain Editor   > CC++ Suild Steps   Build Variables   Environment   Logging   Tool Chain Editor   > CC++ Suild Steps   Build Variables   Environment   Logging   Tool Chain Editor   > CC++ Studies   Project Natures   Project Natures   Project Natures   Project Natures   Project Natures   Project Natures   Post-build steps   Command(s):   Description:   Command(s):   Post-build steps   Command(s):   Description:   Command(s):   Description:   Post-build steps Command(s): Description: Description: Command(s): Description: Command(s): Description: Command(s): Description: Command(s): Description: Command(s): <p< th=""><th>Properties for Blinky_cpu1_se</th><th></th><th>- 0</th><th></th></p<>	Properties for Blinky_cpu1_se		- 0	
Restore <u>D</u> efaults <u>Apply</u>	> Resource     Builders     VC++ Build     Build Variables     Environment     Logging     Settings:     Tool Chain Editor     VC++ General     Project Natures     Project Natures     Renesas QE     Run/Debug Settings     Task Tags	Configuration: Debug [ Active ]  Tool Settings Toolchain Device Per-build steps Command(s): Description: Post-build steps Command(s): armrename-section.data=.secondary.alloc,data,n	eadonly, load, contents secondary, bin secondary,	
			Restore Defaults	pply

Figure 130 e<sup>2</sup> studio Build Setting for the Secondary Project (Flash Boot) (Same core type in pri and sec)



i.

- 7. (Flash boot) (Different core types in pri and sec) 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.
  - 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 --renamesection .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 path of secondary.o in the secondary project to **Other objects**. e.g. \${workspace\_loc:/Blinky\_secondary/Debug/secondary.o}

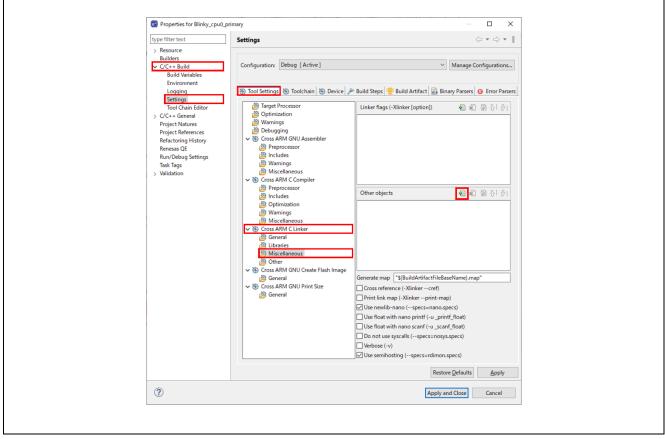


Figure 131 e<sup>2</sup> studio Build Setting for the Primary Project (Part 1)



Figure 132 e<sup>2</sup> 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
      - Set CPSR(5bit) after download: Yes

Create, manage, and run configurations	Image: Secondary Debug,Flat         Ype filter text         Image: Secondary Debug,Flat         Image: Secondary Debug         Image: Secondary Debug <th>Debug Configurations</th> <th></th> <th></th> <th></th>	Debug Configurations			
Type filter text             Main       DBogge       Startup       Source       Common             Main       DBogge       Startup       Source       Common            COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Applet        Debug hardware: Hink ARM       Target Device       Restore Source       Common            COC+ Applet Cochagging COC Applet Deva Applet Deva Applet Deva Applet Deva Applet Deva Addressiport number Deva Addressiport number Deva Addressiport number Deva Addressignet Comection Method            Pvia LAN           Target Deva Source Deva Addressignet Comection Method             Comection Port Number Port Number Port Number Port Number Port Number Port Source Comection Multiple Devices         No           V             Pressord V Intefrace Port Number Port Number	Type filter text             Main       DBogge       Startup       Source       Common             Main       DBogge       Startup       Source       Common            COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Application COC+ Applet        Debug hardware: Hink ARM       Target Device       Restore Source       Common            COC+ Applet Cochagging COC Applet Deva Applet Deva Applet Deva Applet Deva Applet Deva Addressiport number Deva Addressiport number Deva Addressiport number Deva Addressignet Comection Method            Pvia LAN           Target Deva Source Deva Addressignet Comection Method             Comection Port Number Port Number Port Number Port Number Port Number Port Source Comection Multiple Devices         No           V             Pressord V Intefrace Port Number Port Number	Create, manage, and run configu	rations		Ť
	Main Stablesgie & Startup & Source Common                  C/C1+ Application             C/C3+ Application             C/C4+ Application             Comention Method             Privia LAN             V             Anam/P Address(port number)             Link Amm/P Address(port number)             Link Amm/P Address(port number)             Link Amm/P Address(port number)             Link Amm/P Address(port number)             Comention Method             Privia LAN             V             Marking Pbblight             Comention Method             Privia LAN             V             Password             V             Interface             Sype             Sype             Sympt             Sype             Sympt             Sype             Syp		Name: Secondary Debug Flat		1
CC++ Application     CC++ Renote Application     CC+++ Renote Application     CC++++++++++++++++++++++++++++++++	CC++ Application     CC++ Renote Application     CC+++ Renote Application     CC++++++++++++++++++++++++++++++++		-		
C G60 OpenCCD Debugging ()       Connection Method       IP via LAN       ✓         C G60 SpenCCD Debugging ()       Het Shame/IP Address(port number)       Het Shame/IP Address(port number)         Max Applet       Iwan Applet       Iwan Applet         C more class Application       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet Class         C more class Applet class       Type       SWD       V         Second Number       V       Iwan Applet Class       V         Multiple Devices       No       V       V         BPre       0       V       Rest at the class of connection       V         Rest before download       No       V       V       Rest at the class of connection <td>C G60 OpenCCD Debugging ()       Connection Method       IP via LAN       ✓         C G60 SpenCCD Debugging ()       Het Shame/IP Address(port number)       Het Shame/IP Address(port number)         Max Applet       Iwan Applet       Iwan Applet         C more class Application       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet Class         C more class Applet class       Type       SWD       V         Second Number       V       Iwan Applet Class       V         Multiple Devices       No       V       V         BPre       0       V       Rest at the class of connection       V         Rest before download       No       V       V       Rest at the class of connection<td>C/C++ Application C/C++ Remote Application EASE Script</td><td>Debug hardware: J-Link ARM V Target Device:</td><td>R9A09G077M44_CR5</td><td></td></td>	C G60 OpenCCD Debugging ()       Connection Method       IP via LAN       ✓         C G60 SpenCCD Debugging ()       Het Shame/IP Address(port number)       Het Shame/IP Address(port number)         Max Applet       Iwan Applet       Iwan Applet         C more class Application       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet         C more class Applet class       Fort Number       Iwan Applet Class         C more class Applet class       Type       SWD       V         Second Number       V       Iwan Applet Class       V         Multiple Devices       No       V       V         BPre       0       V       Rest at the class of connection       V         Rest before download       No       V       V       Rest at the class of connection <td>C/C++ Application C/C++ Remote Application EASE Script</td> <td>Debug hardware: J-Link ARM V Target Device:</td> <td>R9A09G077M44_CR5</td> <td></td>	C/C++ Application C/C++ Remote Application EASE Script	Debug hardware: J-Link ARM V Target Device:	R9A09G077M44_CR5	
COB Simulator Debugging ()       Host Name/IP Address(port number)         Java Application       Host Name/IP Address(port number)         Lauch Group       Turnel Server         Paramot Streer       Paramot Streer         Paramot Streer       ViterSce         © Pinnay Debug Jiki       Speed Liki?         © Reneas Simulator Debuggin       Multiple Devices         No Brie       0         OR/re       0         DR/re       0         OR/re       0         OR/re       0         V TRA Scan Chain       No         Register initialization       No         Register initialization       No         Register initialization       No         Rest at the end of connection       No         V Edd (Rythys)       FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	COB Simulator Debugging ()       Host Name/IP Address(port number)         Java Application       Host Name/IP Address(port number)         Lauch Group       Turnel Server         Paramot Streer       Paramot Streer         Paramot Streer       ViterSce         © Pinnay Debug Jiki       Speed Liki?         © Reneas Simulator Debuggin       Multiple Devices         No Brie       0         OR/re       0         DR/re       0         OR/re       0         OR/re       0         V TRA Scan Chain       No         Register initialization       No         Register initialization       No         Register initialization       No         Rest at the end of connection       No         V Edd (Rythys)       FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				~ ^
Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Se	Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Filter       Image: Second any Debug: Filter         Image: Second any Debug: Se	GDB Simulator Debugging (I			
I Java Application       Iumel Server         Q Laurch Group       Person Comparison         Q Pennas Dollardware Dollar	I Java Application       Iumel Server         Q Laurch Group       Person Comparison         Q Pennas Dollardware Dollar				
▼       Interface         ♥       Interface         ♥       Interface         ♥       Interface         Secondary Debug Flatt       Trad Scan Chain         ●       Interface         ♥       Interface         ●       Interface         Secondary Debug Flatt       Vacan Chain         ●       Interface         ●       Intenter         ● <t< td=""><td>✓       Interact         ✓       Interact         ✓       Interact         ✓       Secondary Debug Flat         ✓       Secondary Debug Flat         ✓       Interact         ✓       Interact         Ø       No         ✓       Interact         Ø       <td< td=""><td></td><td>Port Number</td><td></td><td></td></td<></td></t<>	✓       Interact         ✓       Interact         ✓       Interact         ✓       Secondary Debug Flat         ✓       Secondary Debug Flat         ✓       Interact         ✓       Interact         Ø       No         ✓       Interact         Ø       Interact         Ø <td< td=""><td></td><td>Port Number</td><td></td><td></td></td<>		Port Number		
Primary Debug Flat     Secondary Debug Flat     Speed (kHz) 4000     V     TAG Scan Chain     Multiple Devices No     V     TAG Scan Chain     Register initialization     Register initinitialization     Register initialization     Register     Regis	Primary Debug Flat     Secondary Debug Flat     Speed (kHz) 4000     V     TAG Scan Chain     Multiple Devices No     V     TAG Scan Chain     Register initialization     Register initinitialization     Register initialization     Register     Regis		Password		
C) Secondary Debugg Flat     Speci (kHz)     4000     ▼       C) Reneass Simulator Debuggi     VFAG Scan Chain     ■       Multiple Device     No     ▼       BPre     0     ■       DRPre     0     ■       Register initialization     No     ▼       Register information     No     ▼       Register information     No     ▼       Register information     No     ▼       Register information     No     ▼       Secon Vector Address     ■	C) Secondary Debugg Flat     Speci (kHz)     4000     ▼       C) Reneass Simulator Debuggi     VFAG Scan Chain     ■       Multiple Device     No     ▼       BPre     0     ■       DRPre     0     ■       Register initialization     No     ▼       Register information     No     ▼       Register information     No     ▼       Register information     No     ▼       Register information     No     ▼       Secon Vector Address     ■		✓ Interface		
Reness Simulator Debuggir    TAG Scan Chain Mittiple Devices No Bit Pe O Office Office Office Office Connection Reset at the beginning of connection No Reset at the dot of connection No Volume Prevent Releasing the Reset of the CMS Core Yes Secure Vector Address Hot Plug No Volume Volume No Volume Volume No Volume <td>Reness Simulator Debuggir    If AG San Chain Multiple Devices No No BiPe Oncetton Register initialization Register initialization No Secure Vector Address Horizon Address Horizon Address Horizon Address No V</td> <td></td> <td>Туре</td> <td></td> <td>~</td>	Reness Simulator Debuggir    If AG San Chain Multiple Devices No No BiPe Oncetton Register initialization Register initialization No Secure Vector Address Horizon Address Horizon Address Horizon Address No V		Туре		~
Multiple Devices     No     v       IRPre     0     0       DRFre     0     0       * Connection     *       Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the dot of connection     No     v       Rest at the dot of connection     No     v       Rest at the dot of connection     No     v       Rest at download     No     v       DCode (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Multiple Devices     No     v       IRPre     0     0       DRFre     0     0       * Connection     *       Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the dot of connection     No     v       Rest at the dot of connection     No     v       Rest at the dot of connection     No     v       Rest at download     No     v       DCode (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			4000	~
IBPre     0       DBPre     0       Connection     0       Register initialization     No       Preset at the beginning of connection     No       Rest at the disconnection     No       Preset at the beginning of connection     No       Preset at the disconnection     No       Preset at the disconnection     No       Preset at the disconnection     No       UCode (Bytes)     1       Hold reset during connect     No       Verse     V       Prevent Releasing the Reset of the CM3 Core       Yes     V       Secure Vector Address       Non-secure Vector Address       Hold Plug     No	IBPre     0       DBPre     0       Register initialization     No       Rest at the beginning of connection     No       Rest at the end of connection     No       Rest at the decision     No       Prevent Gifters     Fifthfffffffffffffffffffffffffffffffffff	C Renesas Simulator Debuggir			
DBPre     0       Connection     No       Register initialization     No       Rest at the beginning of connection     No       Best at the beginning of connection     No       Rest at the download     No       DCote (bytes)     Fffffffffffffffffffffffffffffffffffff	DRPre     0       Connection     No       Reset at the beginning of connection     No       Reset at the dorigon of connection     No       Reset at doring connection     No       ID Cost (Bytes)     IFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				~
✓ Connection     No       Reset at the beginning of connection     No     ∨       Rest at the dot connection     No     ∨       Rest dot dot dot connection     No     ∨       D Cocie (Gytes)     HIFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Connection     No     v       Reset at the beginning of connection     No     v       Rest at the dire direction     No     v       Rest dire direction     No     v       Rest direction download     No     v       Dicade (Bytes)     IFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the and of connection     No     v       Rest at the download     No     v       ID code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Register initialization     No     v       Rest at the beginning of connection     No     v       Rest at the and of connection     No     v       Rest at the and of connection     No     v       Rest at the download     No     v       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF			0	
Rest at the beginning of connection     No     V       Brest at the dright connection     No     V       Rest after download     No     V       D Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest at the beginning of connection     No     V       Brest bit end of connection     No     V       Rest bit end of connection     No     V       Rest after download     No     V       D Code (bytes)     Hiffffffffffffffffffffffffffffffffffff				
Rest at the end of connection     No     V       Rest before download     No     V       Rest at the end of connection     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reset at the end of connection     No     V       Reset before download     No     V       Reset at the download     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Rest before download     No     V       Rest before download     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Rest before download     No     V       Rest and download     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Rest after download     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Reset after download     No     V       ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
ID Code (Bytes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	ID Code (Bynes)     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
Hold reset during connect     No     V       Get CPSR(bit) after download     Yes     V       Prevent Relaxing the Exet of the CM3 Core     Yes     V       Secure Vector Address     V       Non-secure Vector Address     V       Hot Plug     No     V	Hold reset during connect     No     Ves       Set CFSR Sbit after download     Ves     Ves       Prevent Releasing the Breat of the CM3 Core     Ves     Ves       Secure Vector Address     Hot Plug     No				
Set CPSRUShit after download         Yes         Yes           Prevent Releasing the Reset of the CM3 Core         Yes         Yes           Secure Vector Address         Yes         Yes           Hot Plug         No         Yes	Set CPSR/Sbit after download Ves V Prevent Releasing the Rest of the CM3 Core Ves V Secure Vector Address Hick Plug No V				
Prevent Releasing the Reset of the CM3 Core         Yes         V           Secure Vector Address         Image: Comparison of Compariso	Prevent Releasing the Reset of the CM3 Core Yes v Secure Vector Address Non-secure Vector Address Hot Plug No v				
Secure Vector Address Non-secure Vector Address Hoto Plug No v	Secure Vector Address Non-secure Vector Address Hiel Plug No v				
Non-secure Vector Address	Non-secure Vector Address Hot Plug No V			16	
Hot Plug No 🗸	Hot Plug No V				
				No	
				Revert	Apply
C > Reyert Apply	Revert Apply	Filter matched 14 of 16 items		_	
C >> Filter matched 14 of 16 items Regent Apply	Revert Apply				
Filter matched 14 of 16 items Reyert Apply	Filter matched 14 of 16 items Reyert Apply	(?)		Debug	Close
Filtermatched 14 of 16 items Revert Apply.	Filter matched 14 of 16 items Reyert Apply			2	

Figure 133 e<sup>2</sup> 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. 3).
  - ii. (RAM exec) Build the secondary project (No. 5).
  - iii. (Flash boot) (Same core type in pri and sec) Build the secondary project (No. 6).
  - iv. (Flash boot) (Different core types in pri and sec) Create an object file and set it to the secondary project (No. 7).
  - v. Build the primary project again (No. 9).
  - vi. Debug the projects (No. 10).

## Multiprocessing with 3 or more cores for RZ/N devices

This section shows how to perform multi-core debugging using three cores. If more than 4 cores are used, add the process of creating a project, building it and one previous project after No. 4.

- 1. Create and build the primary and secondary projects according to Multiprocessing with 2 cores for RZ/N devices No. 1 to No. 7.
- 2. Create a tertiary project using the bundle file (.sbd) of the secondary project according to the procedures in 4.3 Create a New Project for Blinky.
- 3. (RAM exec) Build the tertiary project according to the procedures in 4.4.1 Build.
- 4. (Flash boot) (Same core type in pri and ter) Build the tertiary 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-none-eabi-objcopy -I binary -O elf32-littlearm -B arm --rename-

 $section\ .data = . secondary, alloc, data, readonly, load, contents\ secondary. bin\ secondary. o$ 

• (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 -rename-

- section .data=.secondary,alloc,data,readonly,load,contents secondary.bin secondary.o
- 5. (Flash boot) (Different core types in pri and ter) Create an object file and set it to the tertiary project.
  - i. Build the tertiary 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 tertiary project, click C/C++ Build > Settings > Build Steps.
    - b. Add Command(s) at Post-build steps.
    - (The primary project uses CR52, and the tertiary project uses CA55) aarch64-none-elf-objcopy -I elf64-littleaarch64 -O binary \${ProjName}.elf secondary.bin
    - (The primary project uses CA55, and the tertiary 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 tertiary project, move it to the Debug folder of the primary project.
  - 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 tertiary 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 tertiary project uses CR52) aarch64-none-elf-objcopy -I binary -O elf64-littleaarch64 -B aarch64 --renamesection .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 tertiary project.
- 6. (Flash boot) Set the following additional properties to the secondary project.
  - i. In the Properties window of the secondary project, click C/C++ Build > Settings > Tool Settings > Cross ARM C Linker > Miscellaneous.
  - ii. Set a file path of secondary.o in the tertiary project to Other objects.
    - e.g. \${workspace\_loc:/Blinky\_tertiary/Debug/secondary.o}
- 7. Build the secondary project.
- 8. (Flash boot) Set the following additional properties to the primary project (Multiprocessing with 2 cores for RZ/N devices No. 8)
- 9. Build the primary project.
- 10. Debug the projects according to the procedures in Multiprocessing with 2 cores for RZ/N devices No. 10. Connections are made in the order primary, secondary, tertialy. The tertialy project is connected in the same procedure as secondary project.
- 11. When changing the project and debugging it again, follow these steps.
  - i. Build the primary project (Multiprocessing with 2 cores for RZ/N devices No. 3).
  - ii. (RAM exec) Build the secondary project (Multiprocessing with 2 cores for RZ/N devices No. 5).
  - iii. (RAM exec) Build the tertiary project (Multiprocessing with 3 or more cores for RZ/N devices No. 3).
  - iv. (Flash boot) (Same core type in pri and ter) Build the tertiary project (Multiprocessing with 3 or more cores for RZ/N devices No. 4).
  - v. (Flash boot) (Different core types in pri and ter) Create an object file and set it to the tertiary project (Multiprocessing with 3 or more cores for RZ/N devices No. 5).
  - vi. Build the secondary project (Multiprocessing with 3 or more cores for RZ/N devices No. 7).
  - vii. Build the primary project again (Multiprocessing with 3 or more cores for RZ/N devices No. 9).
  - viii. Debug the project (Multiprocessing with 3 or more cores for RZ/N devices No. 10)



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

5.3 Using FSP SC 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.

# For RZ/T2 FSP v3.0.0

## Multiprocessing with 2 cores for RZ/T devices

- 1. Create projects according to the procedures in 5.3.2 Create a New Project.
  - (RZ/T2H CA55 Core1) To blink the LED on the RZ/T2H CA55 Core1, you need to change the pin configuration of the primary project in the FSP SC 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. The following additional properties must be set.
  - i. Click **Project** > **Options...**.
  - ii. (Flash boot) Click **Build Actions** > **Build Actions Configuration** > **New** and set it as follows:
    - a. Command line
      - ielftool --bin-multi=0x0-0x7FFFF;0x102000-0x10FFFF; region\_CACHE\_start\_region\_CACHE\_end\_; region\_NONCACHE\_INIT\_start\_region\_NONCACHE\_INIT\_end\_\_\$TARGET\_BPATH\$.out \$TARGET\_BPATH\$.bin
    - b. Build order
      - Run after linking



Category:			
General Options			
Static Analysis			
Runtime Checking	Build Actions Configuration		
C/C++ Compiler Assembler			
Output Converter	Build actions:	New Build Action	×
Custom Build	Command line Build order Output file		
Linker	cmd /c ""\$PROJ_DIR\$¥rasc_launch Pre-com \$BUILD_FIL	Command line:	
Build Actions	cmd /c ""\$PROJ_DIR\$¥rasc_launch Post-link \$TARGET_B	artregion_NONCACHE_INIT_end \$TARGET_BPATH\$.out \$TARGET_	BPATH\$.bin
Debugger		Output files (one per line):	
Simulator			~
CADI			
CMSIS DAP			
E2/E2 Lite			$\sim$
GDB Server		Input files (one per line):	
G+LINK			~
I-jet			
J-Link/J-Trace TI Stellaris			
Nu-Link	New Edit Remove		
PE micro		Maddan dhashann	
CT LINK Y		Working directory: \$PROJ_DIR\$	
		\$PROJ_DIR\$	
		Build order: Run after linking	~

Figure 134 IAR EWARM Project Options for the Secondary Project in Flash Boot Mode

- 6. Build the secondary project according to the procedures in 5.3.3.2 Build for Multiprocessing No. 2.
  - (Flash boot) The following binary files are output to the Debug\Exe folder of the secondary project.

Device	Project	Binary files	Note
	core		
RZ/T2M	CR52	xxxxx-yyyyy.bin	Binary files of programs placed in cache section of System SRAM.
RZ/T2ME		xxxxx-zzzzz.bin	Binary files of programs placed in noncache section of System SRAM.
			When using a noncache sections.
RZ/T2H	CR52	xxxxx-0x0.bin	Binary files of programs placed in ATCM.
		xxxxx-0x102000.bin	Binary files of programs placed in BTCM.
		xxxxx-yyyyy.bin	Binary files of programs placed in cache section of System SRAM.
		xxxxx-zzzzz.bin	Binary files of programs placed in noncache section of System SRAM.
			When using a noncache sections.
	CA55	xxxxx-yyyyy.bin	Binary files of programs placed in cache section of System SRAM.
		xxxxx-zzzzz.bin	Binary files of programs placed in noncache section of System SRAM.
			When using a noncache sections.

Table 32 Binary Files Output to the Debug\Exe folder of the Secondary Project

xxxxx: the secondary project name

yyyyy: The start address of the ache section of System SRAM

zzzz: The start address of the noncache section of System SRAM

- 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:



Device	Primary	Secondary	Keep symbols	File	Align	Note
	project	project	(one per line),		_	
	core	core	Symbol, Section			
RZ/T2M	CR52	CR52	SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	
RZ/T2ME	CPU0	CPU1		ry\Debug\Exe\Blinky_secondar		
				y-0x10000000.bin		
			SECONDARY_N	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	Import only if
			ONCACHE	ry\Debug\Exe\Blinky_secondar		file is output
				y-0x10180040.bin		
RZ/T2H	CR52	CR52	SECONDARY_A	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	
	CPU0	CPU1	TCM_CR521	ry\Debug\Exe\Blinky_secondar		
				y-0x0.bin		
			SECONDARY_B	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	
			TCM_CR521	ry\Debug\Exe\Blinky_secondar		
				y-0x102000.bin		
			SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	
				ry\Debug\Exe\Blinky_secondar		
				y-0x10000000.bin		
			SECONDARY_N	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	Import only if
			ONCACHE	ry\Debug\Exe\Blinky_secondar		file is output
				y-0x10180040.bin		
	CA55	CA55	SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	
	Core0	Corel		ry\Debug\Exe\Blinky_secondar		
				y-0x10020000.bin		
			SECONDARY_N	<pre>\$PROJ_DIR\$\\Blinky_seconda</pre>	8	Import only if
			ONCACHE	ry\Debug\Exe\Blinky_secondar		file is output
				y-0x10180040.bin		

Table 33 Example of Set to "Keep symbols" and "Raw binary image"

General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Converter Output Converter Custom Build Unker Build Actions Debugger Simulator CADI CMSIS DAP EZ/E2 Lite GD8 Server G+LINK I-jet J-Link/J-Trace File: File: File: Symbol: Section: Align: Content of the symbol: Section: Align: File: Symbol: Section: Align: Symbol: Section: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Section: Align: Section: Section: Section: Align: Section: Section: Section: Section: Align: Section: Section: Section: Align: Section: Section: Section: S	Static Analysis Static Analysis Static Analysis Static Analysis Static Analysis Untime Checking C/C++ Compler Config Library Input Optimizations Advanced Output List Config Library Input Optimizations Advanced Output List SECONDARY SECO	Category:						Factor	Settings	
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 I-jet J-Link/J-Trace TI Stellaris Nu-Link File: Sumbol: Section: Align: Sumbol: Section: Align:	Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Univer Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Litie GDB Server G+LINK I-jet J.Link/J-Trace TI Stellaris NuLink PE mirro File: Symbol: Section: Align: File: Symbol: Section: Align: File: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align:									
C/C++ Compler Assembler Output Converter Custom Build Unker Build Actons Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G 4LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE mirco File: Symbol: Section: Align: Symbol: Section: Align:	c/C++ Compler Assembler Output Converter Output Converter Simulator CADI CADI CADI CADI CADI CADI CADI CADI									
Additional Cutput Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server GDB Server GLIDNK I-jet J-Link/J-Trace Eile: Symbol: Section: Align: Symbol: Section: Align:	Auctimed Univer Custom Build Univer Custom Build Univer Sections Debugger Simulator CADI CAD									
Custom Build Linker Build Actions Debugger Simulator CADI CANSIS DAP E2/E2/LUte GDB Server GHUNK I-jet J-Link/J-Trace File: SPROJ_DIR\$¥ \$Secondary¥Debugit Simulos Section: Align: File: Symbol: Section: Align: Ti Stellaris Nu-Link PE mirco Ti Nu	Custom Build Linker SECONDARY Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server GDB Server GDB Server E2/E2 Lite GDB Server E1/E1		Config	Library	Input	Optimizations	Advanced	Output	List	
Custom Build Univer SECONDARY SECOND	Custom Build Linker Build Actions Debugger Simulator CADI CADI CMSIS DAP E2/E2 Line GDB Server GH1NK L-jet J-Ink/J-Trace TI Stellaris Nu-Link PE micro File: Symbol: Section: Align: Symbol: Section: Align: Bay binary image J-Ink/J-Trace File: Symbol: Section: Align: Bay binary image File: Symbol: Section: Align: Bay binary image File: Symbol: Section: Align: Bay binary image	Output Converter	Koon our	bala (ana	por line)					
Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server GDB Server GHLNK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro File: Symbol: Section: Align: Symbol: Section: Align:	Build Actions Debugger Simulator CADI CADI CADI CADI CADI CADI CADI CADI				per inte)					
Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server GDB Server GHLNK I-jet J-Link/J-Trace <u>File:</u> Symbol: Section: Align: SPROJ_DIR\$¥.,¥Secondary¥Debugì [DNDARY] DNDARY Baw binary image <u>File:</u> Symbol: Section: Align: T IStellaris File: Symbol: Section: Align: T IStellaris Symbol: Section: Align: Symbol: Section: Align: File: Symbol: Section: Align: File: Symbol: Section: Align: C INDARY	Debugger Simulator CADI CADI CMSIS DAP E2/E2 Lite GDB Server GHLINK I-jet J-Ink/J-Trace Eile: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Baw binary image File: Symbol: Section: Align: Baw binary image File: Symbol: Section: Align: Baw binary image Symbol: Section: Align: Baw binary image Symbol: Section: Align: Baw binary image Symbol: Section: Align: Symbol: Section: Section: Align: Symbol: Section: Section: Section: Section: Align: Symbol: Section:		SECONE	JARY						
Simulator CADI CADI CMSIS DAP EZ/E2 Lite GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro File: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align:	Simulator CADI CMSIS DAP E2/E2 Lite GOB Server G+LINK I-jet J-Link/J-Trace Ti Stellaris Nu-Link PE micro File: Symbol: Section: Align: SPROJ_DIR\$¥.#Secondary*Debugi CONDARY 0NDARY 8 File: Symbol: Section: Align:									
CADI CMSIS DAP E2/E2 Lite GOB Server G+LINK I-jet J-Link/J-Trace Ti Stellaris Nu-Link PE micro Fjle: Symbol: Section: Align: JONDARY B Fjle: Symbol: Section: Align: JONDARY B	CADI CMSIS DAP E2/E2 Lite G96 Server G+LINK I-jet J-Ink/J-Trace TI Stellaris Nu-Link PE micro File: Symbol: Section: Align: IONDARY 8 File: Symbol: Section: Align: IONDARY 8									
CMSIS DAP E2/E2 Lite GDB Server GHLINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro T INF File: Symbol: Section: Align: SPROJ_DIRSY./FSecondary¥Debugit	CMSIS DAP E2/E2 Lite GDB Server GH1NK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro T : Lite FIIe: Symbol: Section: Align: SPROJ_DIRS¥.,#Secondary¥Debuga File: Symbol: Section: Align:									
E2/E2 Lite GDB Server G-LINK I-jet TI Stellaris Nu-Link PE miroo T I Stellaris Nu-Link FIle: Symbol: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Section: Symbol: Section: Symbol: Section: Align: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Section: Symbol: Section: Section: Section: Symbol: Section: Sectio	E2,E2 Lite GD8 Server G+LINK I-jet J-Link/J-Trace Ti Stellaris Nu-Link PE micro Ti UP File: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align:									
GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro TI Stellaris Nu-Link Fjle: Symbol: Section: Symbol: Section: Symbol: Section: Fjle: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Symbol: Section: Align: Symbol: Section: Symbol: Secti	GDB Server G+LINK L-jet J-Link/J-Trace Ti Stellaris Nu-Link PE micro Ti UT File: Symbol: Symbol: Section: Align: SPROJ_DIRS¥.,¥Secondary¥Debugà  Symbol: Section: Align: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Symbol: Section: Align: Bailing: Bailin									
G +LINK I-jet J-Link/J-Trace Ti Stellaris Nu-Link PE micro Ti Stellaris Nu-Link FIe: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Align: Symbol: Section: Align: Align: Symbol: Section: Align: Align: Align: Symbol: Section: Align: Align: Symbol: Section: Align: Align: Symbol: Section: Align: Align: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Section: Symbol: Symbol: Symbol: Section: Symbol	G+LINK I-jet J-lnk/J-Trace Ti Stellaris Nu-Link PE micro File: Symbol: Section: Align: SPROJ_DIRS¥_#Secondary¥Debugì  Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align:									
I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro T Inter Symbol: Section: Align: SPROJ_DIRS¥.,#Secondary¥Debuga DNDARY ONDARY 8 Fjle: Symbol: Section: Align: T Inter Symbol: Section: Align:	I-jet J-Lnk/J-Trace Eile: Symbol: Section: Align: SPROJ_DR\$X_#Secondary¥Debugi								~	
J-Link/J-Trace TI Stellaris Nu-Link PE micro TI Itel Symbol: Section: Align: SPROJ_DIR\$¥.,\#Secondary¥Debug\u00e3 File: Symbol: Section: Align: Symbol: Section: Align: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align: Symbol: Section: Align:	J-Link/J-Trace TI Stellaris Nu-Link PE micro File: Symbol: Section: Align: CONDARY SPROJ_DIR\$#.,#Secondary#Debuga  CONDARY Section: Align: Symbol: Section: Align:								_	
TI Stellaris Nu-Link PE micro Ti Div File: Symbol: Section: Align: Specific Specific Section: Align:	TI Stellaris Life: Symbol: Section: Align: Nu-Link SPROJ_DIRS¥.,\#Secondary¥Debug\ : :ONDARY ONDARY 8 PE micro File: Symbol: Section: Align:			nary image						
PE micro File: Symbol: Section: Align:	PE micro File: Symbol: Section: Align:									
		Nu-Link	\$PRO	J_DIR\$¥¥S	econdary	/¥Debug¥	CONDARY	ONDARY	8	
			File:				Symbol: Se	ection:	Align:	
		PE micro								

Figure 135 IAR EWARM Project Options for the Primary Project in Flash Boot Mode



#### Note:

Only two binary files can be imported with "Raw binary image". If you want to import more binary files, follow these steps.

a. Click Linker > Extra Options and add "--image\_input=[the secondary project path]\Debug\Exe\[the secondary project name]-[address].bin,[Symbol],[Section],[Align]" to Command line options: (one per line).

e.g.

--image\_input=\$PROJ\_DIR\$\..\Blinky\_secondary\Debug\Exe\Blinky\_secondary-0x0.bin,SECONDA RY\_ATCM\_CR521,SECONDARY\_ATCM\_CR521,8

- b. Click Linker > Input and set it as follows:
  - Keep symbols: (one per line)
    - Set the same as **symbol** of the **Command line options** set in **Extra Options**. e.g. SECONDARY\_ATCM\_CR521



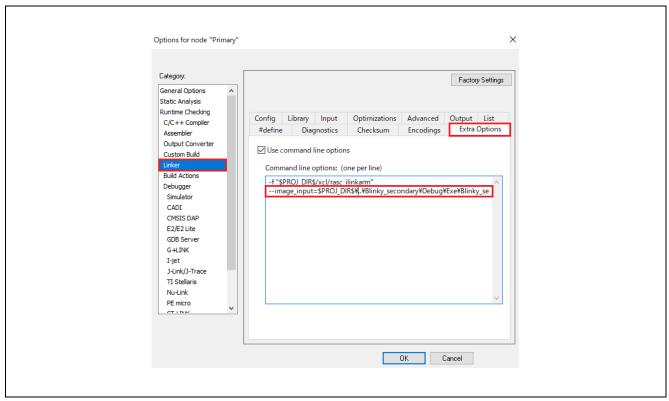


Figure 136 IAR EWARM Project Options for the Primary Project in Flash Boot Mode (Extra Options)

- 8. (Flash boot) 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.

 $e.g. \ PROJ_DIR \ .. \ Blinky\_primary \ Debug \ Exe \ Blinky\_primary.out$ 

File Ed	Edit View Pr	mary - Master - IAR Embedded Workbench IDE jjett l-jet Tools Window Help		
Workspace	Blinky_nor Flex Softv B Build Q B Compt B Compt	Add CMakeLists.ht to Project Force Reconfiguration Add CMakeConnector	- < Q > ≒ + = < ♥ products of Renesas Electronics lucts are sold pursuant to Renesa products of Renesas products are products of Renesas products are products of Renesas products are in and use of Renesas products are in an occurrent set of the interval of the renesative set of the set of the renesative set of the renesative set of the set of the renesative set of t	
		Make & Restart Debugger Ctri+R Restart Debugger Ctri+Shift+R Download • SFR Setup	occamblu. Longunga routing tokes r Download ative application ied Download file	

## Figure 137 IAR EWARM Download File

- 10. Enable settings for multicore debugging in 5.3.5 Debug for Multiprocessing No. 1 and No. 2.
- 11. (RAM exec) Debug the projects according to the procedures in 5.3.5 Debug for Multiprocessing No. 3 and after.
- 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.



Files       Add Project Connection       h products of Renesas Elect         Edit Configurations       Hucks are soid pursuant to on and use of Renesas produ         H = Files Soft       Baildot       property right is granted         H = Baildot       Baildot       products of Renesas produ         H = Baildot       Property right is granted         H = Baildot       products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         H = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the r         M = Baildot       Products of Renesas reserves the reserves the r         M = Baildot       Products of Renesas reserves the reserves the reserves the reserves t	
---	--

### Figure 138 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. 3).
  - iii. Build the secondary project (No. 6).
  - iv. (Flash boot) Build the primary project again (No. 8).
  - v. (Flash boot) Download the file (No. 9).
  - vi. (Flash boot) Enable asymmetric multicore setting (No. 10).
  - vii. (RAM exec) Debug the projects (No. 11).
  - viii. (Flash boot) Debug the projects (No. 12).



## Multiprocessing with 3 or more cores for RZ/T devices

This section shows how to perform multi-core debugging using three cores. If more than 4 cores are used, add the process of creating a project, building it, specifying a raw binary image to build one previous project after No. 3, and add the project information in multicore\_setup.xml.

- 1. Create and build the primary and secondary projects according to Multiprocessing with 2 cores for RZ/T devices No. 1 to No. 6.
- 2. Create a tertiary project using the bundle file (.sbd) of the secondary project according to the procedures in 5.3.2 Create a New Project.
- 3. Build the tertiary project according to Multiprocessing with 2 cores for RZ/T device No. 5 and No. 6. The project option settings for the tertiary project are the same as for the secondary project.
- 4. (Flash boot) Set the following additional options to the secondary project.
  - i. Click **Project** > **Options...**.
  - ii. Click **Linker** > **Input** and set it as follows:

Device	Primary project core	Secondary project core	Tertiary project core	Keep symbols (one per line), Symbol, Section	File	Align	Note
RZ/T2H	CA55 Core0	CR52 CPU0	CR52 CPU1	SECONDARY_ NONCACHE	<pre>\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x10180080.bin</pre>	8	Import only if file is output*
	CR52 CPU0	CA55 Core0	CA55 Core1	SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x10020000.bin</pre>	8	
				SECONDARY_ NONCACHE	<pre>\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x10180080.bin</pre>	8	Import only if file is output
	CR52 CPU0	CR52 CPU1	CA55 Core0	SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x10000000.bin</pre>	8	
				SECONDARY_ NONCACHE	\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x10180080.bin	8	Import only if file is output

#### Table 34 Example of Set to "Keep symbols" and "Raw binary image" for the Secondary Project

\* Binary files for TCM such as Blinky\_secondary\_0x0.bin will not work properly if they are imported into a project other than the primary project. Even binary files from tertiary projects must be imported into the primary project.

- 5. (Flash boot) Build the secondary project according to the procedures in 5.3.3.1 Build.
- 6. (Flash boot) Set the following additional options to the primary project according to Multiprocessing with 2 cores for RZ/T device No. 7.
  - i. Click **Project** > **Options...**.
  - ii. Click **Linker** > **Input** and set it as follows:



Device	Primary project core	project core	Tertiary project core	(one per line), Symbol, Section	File	Align	Note
RZ/T2H	CA55 Core0	CR52 CPU0	CR52 CPU1	SECONDARY_ ATCM_CR520	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary-0x0.bin</pre>	8	
				SECONDARY_ BTCM_CR520	\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary-0x102000.bin	8	
				SECONDARY_ ATCM_CR521	<pre>\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x0.bin</pre>	8	
				SECONDARY_ BTCM_CR521	\$PROJ_DIR\$\\Blinky_ter tiary\Debug\Exe\Blinky_te rtiary-0x102000.bin	8	
				SECONDARY_ NONCACHE	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary- 0x10180040.bin</pre>	8	Import only if file is output
	CR52 CPU0	CA55 Core0	CA55 Core1	SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary- 0x10000000.bin</pre>	8	
				SECONDARY_ NONCACHE	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary- 0x10180040.bin</pre>	8	Import only if file is output
	CR52 CPU0	CR52 CPU1	CA55 Core0	SECONDARY_ ATCM_CR521	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary-0x0.bin</pre>	8	
				SECONDARY_ BTCM_CR521	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary-0x102000.bin</pre>	8	
				SECONDARY	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary- 0x10000000.bin</pre>	8	
				SECONDARY_ NONCACHE	<pre>\$PROJ_DIR\$\\Blinky_se condary\Debug\Exe\Blink y_secondary- 0x10180040.bin</pre>	8	Import only if file is output

## Table 35 Example of Set to "Keep symbols" and "Raw binary image" for the Primary Project



7. Create multicore\_setup.xml and store it in the primary project.

multicore setup.xml

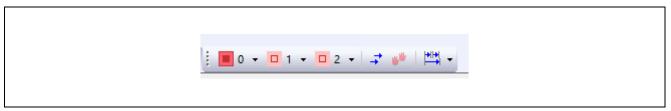
```
<?xml version="1.0" encoding="utf-8"?>
<sessionSetup>
<partner>
<name>Partner0</name>
<workspace>$WS_PATH$</workspace>
<project>$PROJ_PATH$</project></project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
</partner>
<partner>
<name>Partner1</name>
<workspace>$PROJ_DIR$\..\Blinky_secondary\Blinky_secondary.eww</workspace>
<project>Blinky_secondary</project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>false</attachToRunningTarget>
</partner>
<partner>
<name>Partner2</name>
<workspace>$PROJ_DIR$\..\Blinky_tertiary\Blinky_tertiary.eww</workspace>
<project>Blinky_tertiary</project></project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>false</attachToRunningTarget>
</partner>
</sessionSetup>
```

The project information to be debugged is described in the <sessionSetup> tag. The <partner> tag settings are as follows:

- <name> Arbitrary name.
- <workspace> Location of workspace starting from the primary project location. The primary project only set "\$WS\_PATH\$".
- <project> Project name. The primary project only set "\$PROJ\_PATH\$".
- <config> Debug
- <numberOfCores> 1
- (Except the primary project) <attachToRunningTarget> false
- 8. (Flash boot) Build the primary project according to the procedures in 5.3.3.1 Build.
- 9. (Flash boot) In the primary project, click **Project** > **Download** > **Download file...** and select out file of the primary project according to Multiprocessing with 2 cores for RZ/T device No. 9.
- 10. Enable settings for multicore debugging.
  - i. Open the primary project and close the secondary and later projects on IAR EWARM.
  - ii. Set the following in the primary project before debugging:
    - a. Click **Project** > **Options...**.
    - b. 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
        - Advanced
          - Session configuration: \$PROJ\_DIR\$\multicore\_setup.xml
- 11. (RAM exec) Debug the projects according to the procedures in 5.3.5 Debug for Multiprocessing No. 3 and after.



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. As shown in Figure 139, it is according to the setting in multicore\_setup.xml. The primary project is 0, the secondary project is 1, and the tertiary project is 2.



### Figure 139 IAR EWARM Running Projects

- 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 (Multiprocessing with 2 cores for RZ/T device No. 3).
  - iii. Build the secondary project (Multiprocessing with 2 cores for RZ/T device No. 6).
  - iv. Build the tertiary project (Multiprocessing with 3 or more cores for RZ/T devices No. 1)
  - iii. (Flash boot) Build the secondary project again (Multiprocessing with 2 cores for RZ/T device No. 6).
  - iv. (Flash boot) Build the primary project again (Multiprocessing with 2 cores for RZ/T device No. 1).
  - v. (Flash boot) Download the file (Multiprocessing with 2 cores for RZ/T device No. 9).
  - vi. (Flash boot) Enable asymmetric multicore setting (Multiprocessing with 3 or more cores for RZ/T devices No. 10)
  - vii. (RAM exec) Debug the projects (Multiprocessing with 3 or more cores for RZ/T devices No. 11)
  - viii. (Flash boot) Debug the projects (Multiprocessing with 3 or more cores for RZ/T devices No. 12)



## For RZ/N2 FSP v2.2.0

## Multiprocessing with 2 cores for RZ/N devices

- 1. Create projects according to the procedures in 5.3.2 Create a New Project.
- 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. For RZ/N2H, the board file override is also required.
- 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**.

Static Analysis	Factory Settings
Runtme Checking C/C++ Compler Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/C2 Lite GDB Server	Output         Output format:         Raw binary         Qutput file         Override default         Blinky_nor_cpu1_secondary.bin
G +LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PErsirco	v

### Figure 140 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  Static Analysis					Factory Setting	gs	
C/C++ Compiler	#define Config	Diagnostics Library Input	Checksum Optimizations	Encodings Advanced	Extra Option Output List	s	
Output Converter Custom Build Linker	<u>K</u> eep sym SECOND	bols: (one per line ARY	,		^		
Build Actions Debugger Simulator							
CADI CMSIS DAP E2/E2 Lite							
GDB Server G+LINK I-jet	- Raw bin	ary image			~		
J-Link/J-Trace TI Stellaris Nu-Link	<u>F</u> ile: \$PRO.	J_DIR\$¥¥Seconda	ry¥Debugà	CONDARY	ection: <u>A</u> lign: DNDARY 8		
PE micro	File:			Symbol: Se	ection: A <u>l</u> ign:		
	<b>.</b>			0K C	ancel		

Figure 141 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.

 $e.g. \PROJ_DIR\..\Blinky\_primary\Debug\Exe\Blinky\_primary.out$ 

w Pro	Add CMakelists.bt to Project Force Reconfiguration Add CMake Connector	E-Am 9.50.1
c	Download SFR Setup	Download file pps Erase memory
	CMSIS-Pack Manager	57ACKLESS VOId r_55p_sottware_del

Figure 142 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.

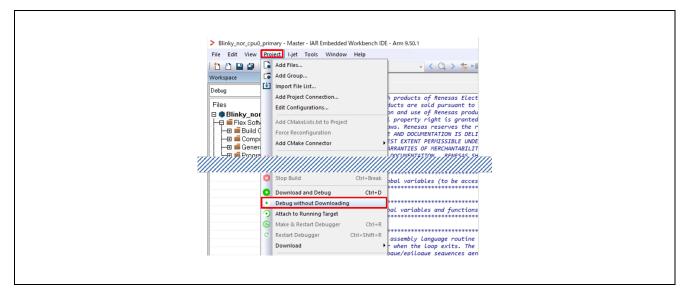


Figure 143 IAR EWARM Debug without Downloading

- 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. 3).
  - 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).

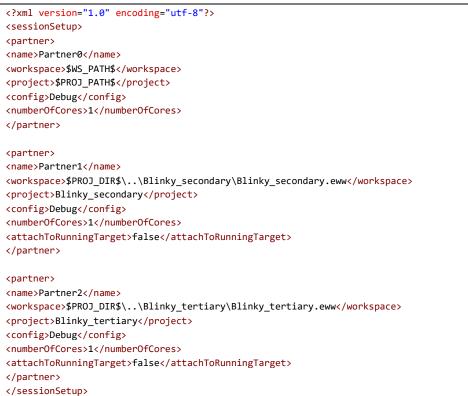
### Multiprocessing with 3 or more cores for RZ/N devices

This section shows how to perform multi-core debugging using three cores. If more than 4 cores are used, add the process of creating a project, building it, specifying a raw binary image to build one previous project after No. 3, and add the project information in multicore\_setup.xml.

- 1. Create and build the primary and secondary projects according to Multiprocessing with 2 cores for RZ/N devices No. 1 to No. 6.
- 2. Create a tertiary project using the bundle file (.sbd) of the secondary project according to the procedures in 5.3.2 Create a New Project.
- 3. Build the tertiary project according to Multiprocessing with 2 cores for RZ/N devices No. 5 and No. 6. The project option settings for the tertiary project are the same as for the secondary project.



- 4. (Flash boot) Set the following additional options to the secondary 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 tertiary project.
         e.g. \$PROJ\_DIR\$\..\Blinky\_tertiary\Debug\Exe\Blinky\_tertiary.bin
      - Symbol: SECONDARY
      - Section: SECONDARY
      - Align: 8
- 5. Build the secondary project according to the procedures in 5.3.3.1 Build.
- 6. (Flash boot) Set the following additional options to the primary project according to Multiprocessing with 2 cores for RZ/N devices No. 7.
  - 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
- 7. Create multicore\_setup.xml and store it in the primary project.
  - multicore\_setup.xml



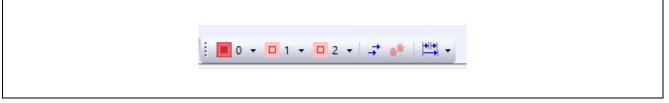


The project information to be debugged is described in the <sessionSetup> tag. The <partner> tag settings are as follows:

- <name> Arbitrary name.
- <workspace> Location of workspace starting from the primary project location. The primary project only set "\$WS\_PATH\$".
- <project> Project name. The primary project only set "\$PROJ\_PATH\$".
- <config> Debug
- <numberOfCores> 1
- (Except the primary project) <attachToRunningTarget> false
- Build the primary project according to the procedures in 5.3.3.1 Build.
- 9. Debug the projects

8.

- i. Open the primary project and close the secondary and later projects on IAR EWARM.
- ii. Set the following in the primary project before debugging:
  - a. Click **Project** > **Options...**.
  - b. 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
      - Advanced
        - Session configuration: \$PROJ\_DIR\$\multicore\_setup.xml
- iii. Follow steps according to 5.3.5 Debug for Multiprocessing No. 3 to No. 7. As shown in Figure 139, it is according to the setting in multicore\_setup.xml. The primary project is 0, the secondary project is 1, and the tertiary project is 2.



### Figure 144 IAR EWARM Running Projects

- 10. 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 and secondary projects (Multiprocessing with 2 cores for RZ/N devices No. 3)
  - iii. Build the secondary project (Multiprocessing with 2 cores for RZ/N devices No. 5 and No. 6).
  - iv. Build the tertiary project (Multiprocessing with 3 or more cores for RZ/N devices No. 3).
  - v. Build the secondary project again (Multiprocessing with 2 cores for RZ/N devices No. 5 and No. 6).
  - vi. Build the primary project again (Multiprocessing with 2 cores for RZ/N devices No. 3).
  - vii. (Flash boot) Debug the projects (Multiprocessing with 3 or more cores for RZ/N devices No. 9).



# **Revision History**

		Descript	tion
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.25	Updated "e <sup>2</sup> studio Prerequisites"
			• Updated the Windows PC requirements.
		p.47	Update "Prerequisites" section
			• Added the note regarding the patch for debugging RZ/N2L FSP project on EWARM.
		p.48	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.74	Updated "Selecting a Board and Toolchain" section.
			• Added the detailed explanation how to select Board and Device for creating a FSP project.
		p.88	Updated "Appendix. Known Issues" chapter.
		p.98	Updated "Appendix. Tool Software Limitations" section.
		-	Added "Appendix. How to update J-Link DLL files in e <sup>2</sup> studio" chapter.
		p.126	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 <sup>2</sup> studio.
		p.45	Added "5.3.2.2 NOTE: Configure IAR EWARM Project [RZ/T2M, RZ/T2L]" section.
		p.88	Updated "Appendix. Known Issues" chapter.
			Remove some limitations regarding RZ/T2M



Rev.         Date         Page         Summary           Updated "Appendix. Tool Software Limitations" section.         •         Added new limitation of e <sup>2</sup> studio regarding J-Link OB S12 version.           •         Added the link to explain how to update J-Link DLL.           •         Added the link to explain how to update J-Link DLL.           •         Removed "Appendix. How to update J-Link DLL files in e <sup>2</sup> studio           1.03         Dec.23.22         •           Updated documentation for RZ/N2L FSP v1.1.0.         •           Removed contents for RZ/N2L FSP v1.0.0         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 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.         •	
<ul> <li>Added new limitation of e<sup>2</sup> studio regarding J-Link OB S12 version.</li> <li>Added the link to explain how to update J-Link DLL.</li> <li>Added the link to explain how to update J-Link DLL files in e<sup>2</sup> stud</li> <li>Dec.23.22</li> <li>Updated documentation for RZ/N2L FSP v1.1.0.</li> <li>Removed contents for RZ/N2L FSP v1.0.0</li> <li>All</li> <li>Updated minor issues.</li> <li>Fixed minor typo.</li> <li>p.69</li> <li>Updated "Appendix. Known Issues" chapter.</li> <li>Removed some limitations regarding RZ/N2L</li> <li>p.73</li> <li>Updated "Appendix. Tool Software Limitations" section.</li> <li>Removed some limitations regarding RZ/N2L</li> <li>p.75</li> <li>Updated "Appendix. How to Debug FSP Project with Flash Boot chapter.</li> <li>Added new procedure for RZ/N2L FSP v1.1.0.</li> <li>1.04</li> <li>Mar.23.23</li> <li>Mat</li> </ul>	
version.•Added the link to explain how to update J-Link DLL.•-Removed "Appendix. How to update J-Link DLL files in e² stud1.03Dec.23.22•Updated documentation for RZ/N2L FSP v1.1.0.•Removed contents for RZ/N2L FSP v1.0.0•AllUpdated minor issues.•Fixed minor typo.•P.69Updated "Appendix. Known Issues" chapter.•Removed some limitations regarding RZ/N2LP.73Updated "Appendix. Tool Software Limitations" section.Removed some limitations regarding RZ/N2LP.75Updated "Appendix. How to Debug FSP Project with Flash Boot chapter.•Added new procedure for RZ/N2L FSP v1.1.0.1.04Mar.23.23AllUpdated documentation for RZ/T2 FSP v1.2.0.	
Image: state of the state of	io" chapter.
-       Removed "Appendix. How to update J-Link DLL files in e <sup>2</sup> stud         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.       •         Dec.23.22       -       •         All       Updated minor issues.       •         •       Fixed minor typo.       •         Dec.23.22       -       •         All       Updated minor issues.       •         •       Fixed minor typo.       •         Dec.23.23       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 chapter.       •         •       Added new procedure for RZ/N2L FSP v1.1.0.         1.04       Mar.23.23       All	io" chapter.
1.03Dec.23.22-Updated documentation for RZ/N2L FSP v1.1.0. • Removed contents for RZ/N2L FSP v1.0.0AllUpdated minor issues. • Fixed minor typo.P.69Updated "Appendix. Known Issues" chapter. • Removed some limitations regarding RZ/N2LP.73Updated "Appendix. Tool Software Limitations" section. 	io" chapter.
Image: Constraint of the second sec	
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.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 chapter.         • Added new procedure for RZ/N2L FSP v1.1.0.         1.04       Mar.23.23	
Image: second	
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.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 chapter.         • Added new procedure for RZ/N2L FSP v1.1.0.         1.04       Mar.23.23	
Image: Provide a section of the sec	
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 chapter.         1.04       Mar.23.23         All       Updated documentation for RZ/T2 FSP v1.2.0.	
Removed some limitations regarding RZ/N2L         p.75       Updated "Appendix. How to Debug FSP Project with Flash Boot chapter.         • Added new procedure for RZ/N2L FSP v1.1.0.         1.04       Mar.23.23         All	
p.75       Updated "Appendix. How to Debug FSP Project with Flash Boot chapter.         1.04       Mar.23.23       All         Updated documentation for RZ/T2 FSP v1.2.0.	
1.04       Mar.23.23       All       Updated documentation for RZ/T2 FSP v1.2.0.	
1.04Mar.23.23AllUpdated documentation for RZ/T2 FSP v1.2.0.	t Mode"
1	
• 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 the FSP version and the FSP version for RZ/T2M and the FSP version and th	nd RZ/T2L.
• Removed the notification that J-Link OB S124 requires the update to debug RZ/T2M FSP project.	firmware
p.10-12 Updated "2.4.1 RSK+RZT2M" section.	
• Added explanation that other boot mode board settings refe User's Manual.	er to the RSK
• 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 refe User's Manual.	r to the RSK
• 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.	



		Descripti	on
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 <sup>2</sup> 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 <sup>2</sup> 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 <sup>2</sup> 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



		Descripti	on
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 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 e <sup>2</sup> studio 2023-04
1.06	Sep.8.23	All	Updated documentation for RZ/T2 FSP v1.3.0.
			<ul> <li>Removed contents for RZ/T2 FSP v1.2.0</li> </ul>
			• Changed GNU ARM Embedded Toolchain to version 12.2.1.arm-12-24.
		p.6	Updated "1.3.2 FSP Documentation" section.
		1	• 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 e <sup>2</sup> 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.
		1	• 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 <sup>2</sup> 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 <sup>2</sup> 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.



		Descript	Description					
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.					



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



		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.
		p.6	Updated "1.3.2 FSP Documentation" section.
			• Removed Note for RZ/N2L about the documentation issue.
		p.8	Updated "2 Set up Evaluation Board" chapter.
			Changed the boards designations.
		p.9	Updated "2.3.1 SEGGER J-Link" section.
			Added how to update J-Link firmware.
		p.16	Added "2.4.4 RZ/T2H Evaluation Board" section.
		p.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.
		p.31	Added "4.4.2 Build for Multiprocessing" section.
		p.32	Updated "4.5.2 Debug Steps" section.
			• Added tables describing the settings for each project.
			Added a note for debugging flash boot project.
		p.37	Updated "4.7 Debug and Run for Multiprocessing" section.
			• Clarified explanation of step 2.
			• Added supplemental information on behavior to step 5.
		p.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.
		p.51	Added "5.3.2.1 NOTE: Configure IAR EWARM Project [RZ/T2H]" section.
		p.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.
		p.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		ion	
Rev.	Date	Page Summary		
		p.58	Updated "5.3.4 Download & Debug the Project" section.	
			• Added a note for debugging flash boot project.	
		p.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.	
		p.67	Updated "6.2 Create a Project" section.	
			Updated tool screen images and menu names.	
		p.71	Added "6.2.4 Duplication of Resources" section.	
		p.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.	
		p.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.	
		p.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.	
		p.127	Added "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e <sup>2</sup> studio" section.	
		p.132	Added "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM" section.	
1.13	Dec.23.24	All	Updated documentation for RZ/N2 FSP v2.1.0.	
			• Removed contents for RZ/N2 FSP v2.0.0.	
		All	Added the RZ/N2H as target device.	
		p.22	Added "2.5.2 RZ/N2H Evaluation Board" section.	
		p.57	Renamed "NOTE: Configure IAR EWARM Project [RZ/T2H and RZ/N2H]" section.	
		p.86	Updated "Appendix. Known Issues" section.	
		-	• Resolved issues No. 21, No. 22, No. 23, No. 24.	
			• Added RZ/N2H as target device of No. 3, No. 14, No. 27 to No. 36.	
			• Added RZ/N2L as target device of No. 27, No. 28.	
			• Added new issues No. 38 to No. 43.	
			• Replaced Smart Configuration with SC.	



	Description		ion
Rev.	Date	Page	Summary
		p.113	Updated "Appendix. Tool Software Limitations" chapter.
			• Added RZ/N2H as target device of No. 4, No. 7, No. 9, No. 10 and No. 12 to No. 19.
			• Added new issues No. 20, No. 21.
			• Resolved issues No. 6.
			• Updated the description of issues No. 12.
			Replaced Smart Configuration with SC.
		p.125	Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section.
			• Moved Note for multiprocessing projects to No. 1 in the same chapter.
			• Removed CA55 NOR flash boot from the table in No. 1.
			• Added No. 2 for RZ/N2H.
			Added note for debug FSP project.
		p.127	Updated "Appendix. How to Erase Flash Memory"
			• Added Device Type Name for RZ/N2H.
			• Added External Address Space for RZ/N2H.
1.14	Feb.28.25	All	Updated documentation for RZ/T2 FSP v2.3.0.
			• Removed contents for RZ/T2 FSP v2.2.0
			• Unified wording for Smart Configurator(SC) and IAR I-jet.
		p.25	Updated "3.2.1 Windows PC Requirements" section.
			• Updated Windows PC requirements to use e <sup>2</sup> studio.
		p.25	Updated "3.2.3 Choosing a Toolchain" section.
			• Added a table of toolchain version for each FSP.
		p.26	Updated "4.3 Create a New Project for Blinky" section.
			• Added a table of the project creation procedure.
			• Added a table of selecting a bundle file to procedure No. 9.
		p.47	Updated "5.3.2 Create a New Project" section.
			• Added a table of the project creation procedure.
			• Updated a description of selecting a bundle file in procedure No. 9.
			• Added a table of selecting a bundle file to procedure No. 9.
		p.55	Updated "5.3.2.1 NOTE: Configure IAR EWARM Project [RZ/T2H and RZ/N2H]" section.
			• Updated a description of error conditions.
			• Removed step 5 to set Build Actions.
		p.86	Updated "Appendix. Known Issues" chapter.
			• Resolved issues No. 3, No. 25 to 27, No. 29, No. 31, No. 36 to 43 for RZ/T series devices.
			• Added new issue No. 44.
		p.118	Updated "Appendix. Tool Software Limitations" chapter.
			• Resolved limitation No. 17 for RZ/T series devices.
			• Added new limitation No. 22.



	Descriptio		)n
Rev.	Date	Page	Summary
		p.141	<ul> <li>Updated "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e<sup>2</sup> studio" chapter.</li> <li>Added "Multiprocessing with 3 or more cores" section.</li> </ul>
		p.146	<ul> <li>Updated "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM" chapter.</li> <li>Added "</li> <li>Multiprocessing with 3 or more cores" section.</li> </ul>
1.15	Apr.25.25	All	<ul> <li>Updated documentation for RZ/N2 FSP v2.2.0.</li> <li>Removed contents for RZ/N2 FSP v2.1.0</li> </ul>
		p.86	<ul> <li>Updated "Appendix. Known Issues" chapter.</li> <li>Resolved issues No. 3, No. 27, No. 29, No. 31, No. 37 to 43.</li> <li>Added new issue No. 45, No. 46, No. 47, No. 48.</li> </ul>
		p.118	<ul> <li>Updated "Appendix. Tool Software Limitations" chapter.</li> <li>Removed RZ/N2H as target device from No. 13, No. 14.</li> <li>Resolved limitation No. 17.</li> <li>Added new limitation No. 23.</li> </ul>
		p.131	<ul> <li>Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" section.</li> <li>Updated a description of CA55 xSPI boot from the table in No. 1</li> </ul>
		p.147	Updated "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e <sup>2</sup> studio" chapter.
1.16	Jul.9.25	All	<ul> <li>Updated "Multiprocessing with 3 or more cores" section for flash boot.</li> <li>Updated documentation for RZ/T2 FSP v3.0.0.</li> <li>Removed contents for RZ/T2 FSP v2.3.0</li> </ul>
		p.25	<ul> <li>Updated "3.2.3 Choosing a Toolchain" section.</li> <li>Added version name displayed in arm Developer website.</li> </ul>
		p.35	<ul><li>Updated "4.5.2 Debug Steps" section.</li><li>Added script file setting for TCM initialization.</li></ul>
		p.60	<ul><li>Updated "5.3.3.2 Build for Multiprocessing" section.</li><li>Added macro file setting for TCM initialization.</li></ul>
		p.88	<ul> <li>Updated "Appendix. Known Issues" chapter.</li> <li>Resolved issues No. 14, No. 32, No. 46, No. 47 for RZ/T series devices.</li> <li>Resolved issues No.45, No. 48.</li> </ul>
		p.122	Updated "Appendix. Tool Software Limitations" chapter. • Resolved limitations No. 13, No. 14.
		p.135	<ul> <li>Updated "Appendix. How to Debug FSP Project with Flash Boot Mode" chapter.</li> <li>Removed column e<sup>2</sup> studio 2022-04 2024-07.</li> <li>Updated black text code in system_init().</li> <li>Changed the number of loops for waiting to 1.5x.</li> </ul>
		p.145	<ul> <li>Updated "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for e<sup>2</sup> studio" chapter.</li> <li>Added procedure for RZ/T2 FSP v3.0.0 to meet that specification.</li> </ul>



		Description	
Rev.	Date	Page	Summary
		p.158	<ul> <li>Updated "Appendix. How to Create and Debug FSP Projects for Multiprocessing in All Cases for IAR EWARM" chapter.</li> <li>Added procedure for RZ/T2 FSP v3.0.0 to meet that specification.</li> </ul>



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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

#### 6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

## Notice

- 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: <u>www.renesas.com/contact/</u>.