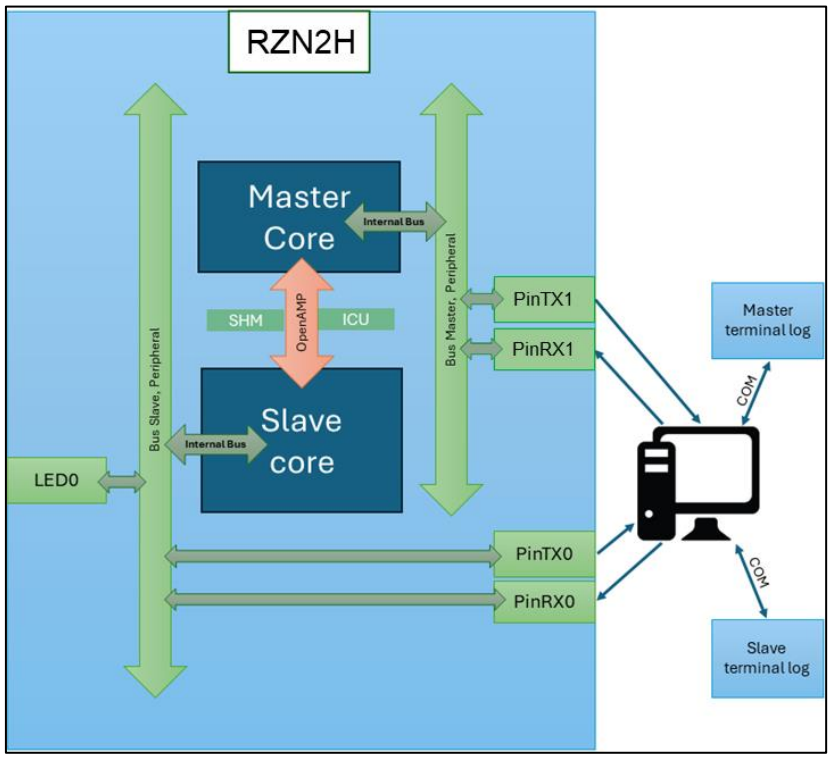
RZ/N2H

Quick Start Guide: RZ/N2H Multiple-Core Application

Introduction

This document explains Sample Program setup procedures for multiple-core (BareMetal/FreeRTOS/Linux running on Cortex®-A55 and Bare-metal/FreeRTOS running on Cortex®-R52) environment and sample program.

This package consists of the **RZ/N2H** Cortex®-R52 Flexible Software Package (hereinafter referred to as **RZ/N2H** CR52/CA55 FSP), the Inter-Processor Communication Feature via OpenAMP for the **RZ/N2H** Board Support Package (hereinafter referred to as RZ/N2H BSP) and the Linux BSP software package.



Here are brief descriptions of each component of RZ/N2H Multiple-core Application:

- RZ/N2H CR52 and CA55 FSP
  The software package consists of production-ready peripheral drivers, Bare-metal, FreeRTOS and portable middleware stacks, and the best in-case HAL drivers with low memory footprint.
- OpenAMP
  The framework includes the software components needed for Asymmetric Multiprocessing (AMP) systems such as Inter-Processor Communication.
- RZ/N2H BSP
  The environment for building the Board Support Package.

Target device

RZ/N2H

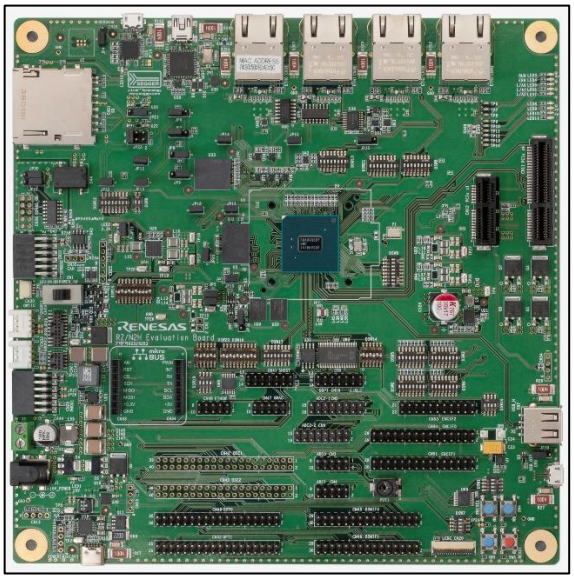
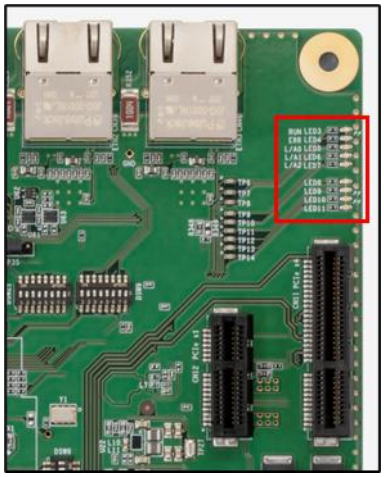
Contents

1. Specifications

Table 1-1. List of the on-chip peripheral modules to be used in this application.

Peripheral module	Usage
Serial Communications Interface (SCI)	Performs standard serial communications sending and receiving console messages.
Interrupt controller (ICU)	Configures interrupt settings; the processor will receive interrupts during buffered serial communications; configure inter-processor interrupt.
General Purpose Input Output (GPIO)	Configures I/O lines used by serial communications.

Table 1-2. List of the equipment to be used in this application.

Peripheral module	Description
Board	RZ/N2H Evaluation Board 
LED	User LED8 on RZ/N2H Evaluation Board 

1.1. Package information

Table 1-3. Package information

Package Name
│ ├──**gcc**
│ │ ├──rzn2h_ca55_0_baremetal_master_ca55_1_baremetal_slave.zip
│ │ │ ├──rzn2h_ca55_0_baremetal_mastercore
│ │ │ └──rzn2h_ca55_1_baremetal_slavecore
│ │ ├──rzn2h_ca55_0_baremetal_master_cr52_0_baremetal_slave.zip
│ │ │ ├──rzn2h_ca55_0_baremetal_mastercore
│ │ │ └──rzn2h_cr52_0_baremetal_slavecore
│ │ ├──rzn2h_ca55_0_baremetal_master_cr52_1_baremetal_slave.zip
│ │ │ ├──rzn2h_ca55_0_baremetal_mastercore
│ │ │ └──rzn2h_cr52_1_baremetal_slavecore
│ │ ├──rzn2h_ca55_1_baremetal_master_ca55_2_baremetal_slave.zip
│ │ │ ├──rzn2h_ca55_0_boot
│ │ │ ├──rzn2h_ca55_1_baremetal_mastercore
│ │ │ └──rzn2h_ca55_2_baremetal_slavecore
│ │ ├──rzn2h_ca55_1_baremetal_master_cr52_1_baremetal_slave.zip
│ │ │ ├──rzn2h_cr52_0_boot
│ │ │ ├──rzn2h_ca55_1_baremetal_mastercore
│ │ │ └──rzn2h_cr52_1_baremetal_slavecore
│ │ ├──rzn2h_cr52_0_baremetal_master_ca55_0_baremetal_slave.zip
│ │ │ ├──rzn2h_cr52_0_baremetal_mastercore
│ │ │ └──rzn2h_ca55_0_baremetal_slavecore
│ │ ├──rzn2h_cr52_0_baremetal_master_ca55_1_baremetal_slave.zip
│ │ │ ├──rzn2h_cr52_0_baremetal_mastercore
│ │ │ └──rzn2h_ca55_1_baremetal_slavecore
│ │ ├──rzn2h_cr52_0_baremetal_master_cr52_1_baremetal_slave.zip
│ │ │ ├──rzn2h_cr52_0_baremetal_mastercore
│ │ │ └──rzn2h_cr52_1_baremetal_slavecore
│ │ ├──rzn2h_cr52_1_baremetal_master_cr52_0_baremetal_slave.zip
│ │ │ ├──rzn2h_cr52_1_baremetal_mastercore
│ │ │ └──rzn2h_cr52_0_baremetal_slavecore
│ │ ├──rzn2h_ca55_0_freertos_master_ca55_1_baremetal_slave.zip
│ │ │ ├──rzn2h_ca55_0_freertos_mastercore
│ │ │ └──rzn2h_ca55_1_baremetal_slavecore

```
│   │   ├──rzn2h_ca55_0_freertos_master_cr52_0_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_freertos_mastercore
│   │   │   └──rzn2h_cr52_0_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_freertos_master_cr52_1_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_0_freertos_mastercore
│   │   │   └──rzn2h_cr52_1_baremetal_slavecore
│   │   ├──rzn2h_cr52_1_freertos_master_cr52_0_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_1_freertos_mastercore
│   │   │   └──rzn2h_cr52_0_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_baremetal_master_cr52_1_freertos_slave.zip
│   │   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   │   └──rzn2h_cr52_1_freertos_slavecore
│   │   ├──rzn2h_ca55_0_baremetal_master_ca55_2_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_2_baremetal_slavecore
│   │   ├──rzn2h_ca55_0_baremetal_master_ca55_3_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_3_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_baremetal_master_ca55_2_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_2_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_baremetal_master_ca55_3_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_3_baremetal_slavecore
│   │   ├── rzn2h_ca55_1_freertos_master_ca55_2_baremetal_slave.zip
│   │   │   ├── rzn2h_ca55_0_boot
│   │   │   ├── rzn2h_ca55_1_freertos_mastercore
│   │   │   └── rzn2h_ca55_2_baremetal_slavecore
│   │   ├── rzn2h_ca55_1_freertos_master_cr52_1_baremetal_slave.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   ├── rzn2h_cr52_1_baremetal_slavecore
│   │   │   └── rzn2h_ca55_1_freertos_mastercore
│   │   ├──gcc_rzn2h_cr52_0_rpmsg_linux_baremetal_demo.zip
│   │   │   ├── gcc_rzn2h_cr52_0_rpmsg_linux_baremetal_demo
│   │   ├──gcc_rzn2h_cr52_1_rpmsg_linux_baremetal_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
```

```
│   │   │   └──gcc_rzn2h_cr52_1_rpmsg_linux_baremetal_demo
│   │   ├──gcc_rzn2h_cr52_0_rpmsg_linux_freertos_demo.zip
│   │   │   └── gcc_rzn2h_cr52_0_rpmsg_linux_freertos_demo
│   │   ├──gcc_rzn2h_cr52_1_rpmsg_linux_freertos_demo.zip
│   │   │   ├──_cr52_0_boot
│   │   │   └── gcc_rzn2h_cr52_1_rpmsg_linux_freertos_demo
│   │   ├──gcc_rzn2h_ca55_1_rpmsg_linux_freertos_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
│   │   │   └── gcc_rzn2h_ca55_1_rpmsg_linux_freertos_demo
│   │   ├──gcc_rzn2h_ca55_2_rpmsg_linux_freertos_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
│   │   │   └── gcc_rzn2h_ca55_2_rpmsg_linux_freertos_demo
│   │   ├──gcc_rzn2h_ca55_3_rpmsg_linux_freertos_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
│   │   │   └── gcc_rzn2h_ca55_3_rpmsg_linux_freertos_demo
│   │   ├──gcc_rzn2h_ca55_1_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──gcc_rzn2h_ca55_1_rpmsg_linux_ baremetal _demo
│   │   ├──gcc_rzn2h_ca55_2_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──gcc_rzn2h_ca55_2_rpmsg_linux_ baremetal _demo
│   │   ├──gcc_rzn2h_ca55_3_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──gcc_rzn2h_ca55_3_rpmsg_linux_ baremetal _demo
│   ├──iar
│   │   ├──rzn2h_ca55_0_baremetal_master_ca55_1_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_1_baremetal_slavecore
│   │   ├──rzn2h_ca55_0_baremetal_master_cr52_0_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_cr52_0_baremetal_slavecore
│   │   ├──rzn2h_ca55_0_baremetal_master_cr52_1_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_cr52_1_baremetal_slavecore
│   │   ├──rzn2h_ca55_1_baremetal_master_ca55_2_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_boot
```

```
│   │   ├──rzn2h_ca55_1_baremetal_mastercore
│   │   └──rzn2h_ca55_2_baremetal_slavecore
│   ├──rzn2h_ca55_1_baremetal_master_cr52_1_baremetal_slave.zip
│   │   ├──rzn2h_cr52_0_boot
│   │   ├──rzn2h_ca55_1_baremetal_mastercore
│   │   └──rzn2h_cr52_1_baremetal_slavecore
│   ├──rzn2h_cr52_0_baremetal_master_ca55_0_baremetal_slave.zip
│   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   └──rzn2h_ca55_0_baremetal_slavecore
│   ├──rzn2h_cr52_0_baremetal_master_ca55_1_baremetal_slave.zip
│   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   └──rzn2h_ca55_1_baremetal_slavecore
│   ├──rzn2h_cr52_0_baremetal_master_cr52_1_baremetal_slave.zip
│   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   └──rzn2h_cr52_1_baremetal_slavecore
│   ├──rzn2h_cr52_1_baremetal_master_cr52_0_baremetal_slave.zip
│   │   ├──rzn2h_cr52_1_baremetal_mastercore
│   │   └──rzn2h_cr52_0_baremetal_slavecore
│   ├──rzn2h_ca55_0_freertos_master_ca55_1_baremetal_slave.zip
│   │   ├──rzn2h_ca55_0_freertos_mastercore
│   │   └──rzn2h_ca55_1_baremetal_slavecore
│   ├──rzn2h_ca55_0_freertos_master_cr52_0_baremetal_slave.zip
│   │   ├──rzn2h_ca55_0_freertos_mastercore
│   │   └──rzn2h_cr52_0_baremetal_slavecore
│   ├──rzn2h_cr52_0_freertos_master_cr52_1_baremetal_slave.zip
│   │   ├──rzn2h_cr52_0_freertos_mastercore
│   │   └──rzn2h_cr52_1_baremetal_slavecore
│   ├──rzn2h_cr52_1_freertos_master_cr52_0_baremetal_slave.zip
│   │   ├──rzn2h_cr52_1_freertos_mastercore
│   │   └──rzn2h_cr52_0_baremetal_slavecore
│   ├──rzn2h_cr52_0_baremetal_master_cr52_1_freertos_slave.zip
│   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   └──rzn2h_cr52_1_freertos_slavecore
│   ├──rzn2h_ca55_0_baremetal_master_ca55_2_baremetal_slave.zip
│   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   └──rzn2h_ca55_2_baremetal_slavecore
```

```
│   │   ├──rzn2h_ca55_0_baremetal_master_ca55_3_baremetal_slave.zip
│   │   │   ├──rzn2h_ca55_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_3_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_baremetal_master_ca55_2_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_2_baremetal_slavecore
│   │   ├──rzn2h_cr52_0_baremetal_master_ca55_3_baremetal_slave.zip
│   │   │   ├──rzn2h_cr52_0_baremetal_mastercore
│   │   │   └──rzn2h_ca55_3_baremetal_slavecore
│   │   ├── rzn2h_ca55_1_freertos_master_ca55_2_baremetal_slave.zip
│   │   │   ├── rzn2h_ca55_0_boot
│   │   │   ├── rzn2h_ca55_1_freertos_mastercore
│   │   │   └── rzn2h_ca55_2_baremetal_slavecore
│   │   ├── rzn2h_ca55_1_freertos_master_cr52_1_baremetal_slave.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   ├── rzn2h_cr52_1_baremetal_slavecore
│   │   │   └── rzn2h_ca55_1_freertos_mastercore
│   │   ├──iar_rzn2h_cr52_0_rpmsg_linux_baremetal_demo.zip
│   │   │   └──iar_rzn2h_cr52_0_rpmsg_linux_baremetal_demo
│   │   ├──iar_rzn2h_cr52_1_rpmsg_linux_baremetal_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_cr52_1_rpmsg_linux_baremetal_demo
│   │   ├──iar_rzn2h_cr52_0_rpmsg_linux_freertos_demo.zip
│   │   │   └──iar_rzn2h_cr52_0_rpmsg_linux_freertos_demo
│   │   ├──iar_rzn2h_cr52_1_rpmsg_linux_freertos_demo.zip
│   │   │   ├──rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_cr52_1_rpmsg_linux_freertos_demo
│   │   ├──iar_rzn2h_ca55_1_rpmsg_linux_freertos_demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_1_rpmsg_linux_freertos_demo
│   │   ├──iar_rzn2h_ca55_2_rpmsg_linux_freertos_demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_2_rpmsg_linux_freertos_demo
│   │   ├──iar_rzn2h_ca55_3_rpmsg_linux_freertos_demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_3_rpmsg_linux_freertos_demo
```

```
│   │   ├──iar_rzn2h_ca55_1_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_1_rpmsg_linux_ baremetal _demo
│   │   ├──iar_rzn2h_ca55_2_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_2_rpmsg_linux_ baremetal _demo
│   │   ├──iar_rzn2h_ca55_3_rpmsg_linux_ baremetal _demo.zip
│   │   │   ├── rzn2h_cr52_0_boot
│   │   │   └──iar_rzn2h_ca55_3_rpmsg_linux_ baremetal _demo
│   ├──Lib_mastercores
│   │   ├──Baremetal
│   │   │   ├──RM_OpenAMP_App
│   │   │   ├──hal_entry.c
│   │   │   └──pin_config.c
│   │   ├──FreeRTOS
│   │   │   ├──RM_OpenAMP_App
│   │   │   ├──hal_entry.c
│   │   │   ├── new_thread0_entry.c
│   │   │   └──pin_config.c
│   ├──Lib_slavecore
│   │   ├──Baremetal
│   │   │   ├──RM_OpenAMP_App
│   │   │   ├──hal_entry.c
│   │   │   └──pin_config.c
│   │   ├──FreeRTOS
│   │   │   ├──RM_OpenAMP_App
│   │   │   ├──hal_entry.c
│   │   │   ├── new_thread0_entry.c
│   │   │   └──pin_config.c
│   ├── Lib_bootproject
│   │   ├──hal_entry.c
│   │   └──pin_config.c
│   ├──Patch
│   │   ├──redefinition_memory_regions_gcc.h
│   │   ├──fsp_ram_execution.ld
│   │   ├──redefinition_memory_regions_iar.h
```

```
│   │   ├──fsp_ram_execution.icf
└──── LICENSE.md
```

Table 1-3. Mapping ZIP files to CPU core roles and platform types

CA55				CR52		Project ZIP file
0	1	2	3	0	1	
M	S	-	-	-	-	rzn2h_ca55_0_baremetal_master_ca55_1_baremetal_slave.zip
M	-	S	-	-	-	rzn2h_ca55_0_baremetal_master_ca55_2_baremetal_slave.zip
M	-	-	S	-	-	rzn2h_ca55_0_baremetal_master_ca55_3_baremetal_slave.zip
M	-	-	-	S	-	rzn2h_ca55_0_baremetal_master_cr52_0_baremetal_slave.zip
M	-	-	-	-	S	rzn2h_ca55_0_baremetal_master_cr52_1_baremetal_slave.zip
B	M	S	-	-	-	rzn2h_ca55_1_baremetal_master_ca55_2_baremetal_slave.zip
-	M	-	-	B	S	rzn2h_ca55_1_baremetal_master_cr52_1_baremetal_slave.zip
S	-	-	-	M	-	rzn2h_cr52_0_baremetal_master_ca55_0_baremetal_slave.zip
-	S	-	-	M	-	rzn2h_cr52_0_baremetal_master_ca55_1_baremetal_slave.zip
-	-	S	-	M	-	rzn2h_cr52_0_baremetal_master_ca55_2_baremetal_slave.zip
-	-	-	S	M	-	rzn2h_cr52_0_baremetal_master_ca55_3_baremetal_slave.zip
-	-	-	-	M	S	rzn2h_cr52_0_baremetal_master_cr52_1_baremetal_slave.zip
-	-	-	-	S	M	rzn2h_cr52_1_baremetal_master_cr52_0_baremetal_slave.zip
M	S	-	-	-	-	rzn2h_ca55_0_freertos_master_ca55_1_baremetal_slave.zip
M	-	-	-	S	-	rzn2h_ca55_0_freertos_master_cr52_0_baremetal_slave.zip
B	M	S	-	-	-	rzn2h_ca55_1_freertos_master_ca55_2_baremetal_slave.zip
-	M	-	-	B	S	rzn2h_ca55_1_freertos_master_cr52_1_baremetal_slave.zip
-	-	-	-	M	S	rzn2h_cr52_0_freertos_master_cr52_1_baremetal_slave.zip
-	-	-	-	S	M	rzn2h_cr52_1_freertos_master_cr52_0_baremetal_slave.zip
-	-	-	-	M	S	rzn2h_cr52_0_baremetal_master_cr52_1_freertos_slave.zip
M	-	-	-	S	-	gcc/iar_rzn2h_cr52_0_rpmsg_linux_baremetal_demo.zip
M	-	-	-	-	S	gcc/iar_rzn2h_cr52_1_rpmsg_linux_baremetal_demo.zip
M	S	-	-	-	-	gcc/iar_rzn2h_ca55_1_rpmsg_linux_ baremetal _demo.zip
M	-	S	-	-	-	gcc/iar _rzn2h_ca55_2_rpmsg_linux_ baremetal _demo.zip
M	-	-	S	-	-	gcc/iar_rzn2h_ca55_3_rpmsg_linux_ baremetal _demo.zip
M	-	-	-	S	-	gcc/iar_rzn2h_cr52_0_rpmsg_linux_freertos_demo.zip
M	-	-	-	-	S	gcc/iar_rzn2h_cr52_1_rpmsg_linux_freertos_demo.zip
M	S	-	-	-	-	gcc/iar_rzn2h_ca55_1_rpmsg_linux_freertos_demo.zip
M	-	S	-	-	-	gcc/iar_rzn2h_ca55_2_rpmsg_linux_freertos_demo.zip
M	-	-	S	-	-	gcc/iar_rzn2h_ca55_3_rpmsg_linux_freertos_demo.zip

Note: The symbols and colors in the above table are defined as follows, where M stands for Master project, S for Slave project, and B for Boot project.

Primary core
Baremetal
FreeRTOS
Linux

1.2. Operating environment

Please refer to and utilize the recommended tools as listed below:

Tool names	Version	Remarks
e² studio	2025-10	Download links: Release v3.1.0 Refer to the Getting Started document for tool installation.
FSP Smart Configurator	2025-10	
GCC Compiler for Cortex-R52	13.3.Rel1	Download links: GCC Compiler for Cortex-R52
GCC Compiler for Cortex-A55	10.3-2021.07	Download links: GCC Compiler for Cortex-A55
SEGGER J-Link	8.60	Download links: SEGGER J-Link
IAR EWARM	9.60.3	Download links: IAR Embedded Workbench for Arm
Tera Term	5.2	Download links: Tera Term Refer to the instructions below to install it.

Please follow the steps below to download the serial port terminal software – Tera Term:

(1) Visit the GitHub Releases Page

- Open your web browser and go to the Tera Term GitHub Releases page.

(2) Download the Installer File

- In the Assets section of the selected version, find the file named something like teraterm-<version>-installer.exe. This is the installer file for Tera Term.
- Click on this file to start the download.

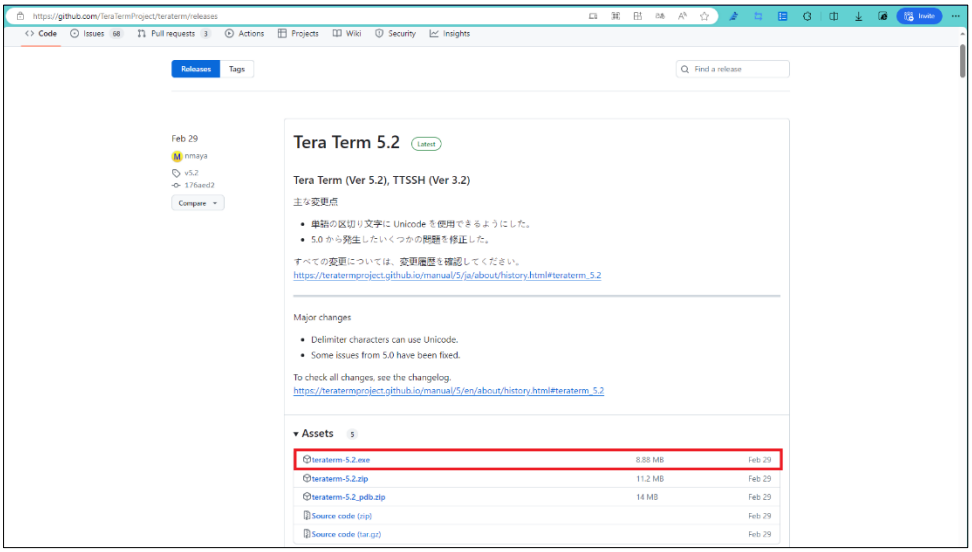


Figure 1-1. The Tera Term GitHub Releases page

(3) Run the Installer

- Once the download is complete, navigate to the location where you saved the file and double-click on the installer to run it.

(4) Install the Software

- Follow the on-screen instructions to complete the installation. You may be asked to agree to the license agreement and choose the installation directory.

Please use the recommended software packages as listed below:

Package name	Version	Remarks
RZ/N2 FSP Package	3.1.0	Download links: Release v3.1.0 Refer to the Getting Started document for tool installation.
Verified Linux Package	5.0.0	Refer to the RZ/N Multi-OS Package Application Note for installation instructions.

2. Verified operation conditions

Table 2-1. Verified Operating Conditions

Item	Contents	
Microprocessor used	RZ/N2H	
FSP version	RZ/N2 FSP v3.1.0	
Integrated Development Environment	e² Studio 2025-10	IAR EW for Arm 9.60.3
C compiler	GNU Arm Embedded 13.3.1.arm-13-24 Compiler Options (except directory path): -mcpu=cortex-r52 -mthumb -mfloat-abi=hard -mfpu=neon-fp-armv8 -fdiagnostics-parseable-fixits -Og -fmessage-length=0 -fsigned-char -ffunction-sections -fdata-sections -fno-strict-aliasing -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -g -o GCC Arm A-Profile (AArch64 bare-metal) 10.3.1.20210621 Compiler Options (except directory path): -mcpu=generic+simd -mcmodel=small -Og -fmessage-length=0 -fsigned-char -ffunction-sections -fdata-sections -fno-strict-aliasing -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -g -o	##IAR Ninja build file #Rules rule COMPILER_XCL command = C$:\iar\ewarm-9.60.3\common\bin\XclFileGenerator.exe $xclcommand -f "$rspfile_name" description = IAR_NEW_TOOL+++COMPILER_XCL+++$out rspfile = $rspfile_name rspfile_content = $flags rule INDEXER command = C$:\iar\ewarm-9.60.3\common\bin\SourceIndexer.exe $flags depfile = $out.dep deps = gcc description = IAR_NEW_TOOL+++INDEXER+++$out rule MAKEBROWSE command = C$:\iar\ewarm-9.60.3\common\bin\makeBrowseData.exe $flags description = IAR_NEW_TOOL+++MAKEBROWSE+++$out rule PDBLINK command = C$:\iar\ewarm-9.60.3\common\bin\PbdLink.exe $flags description = IAR_NEW_TOOL+++PDBLINK+++$out

3. Sample program invocation

3.1. Overview of application behavior

3.1.1. Overview of application for FSP x FSP (Baremetal/ FreeRTOS)

The behavior of the application is as follows:

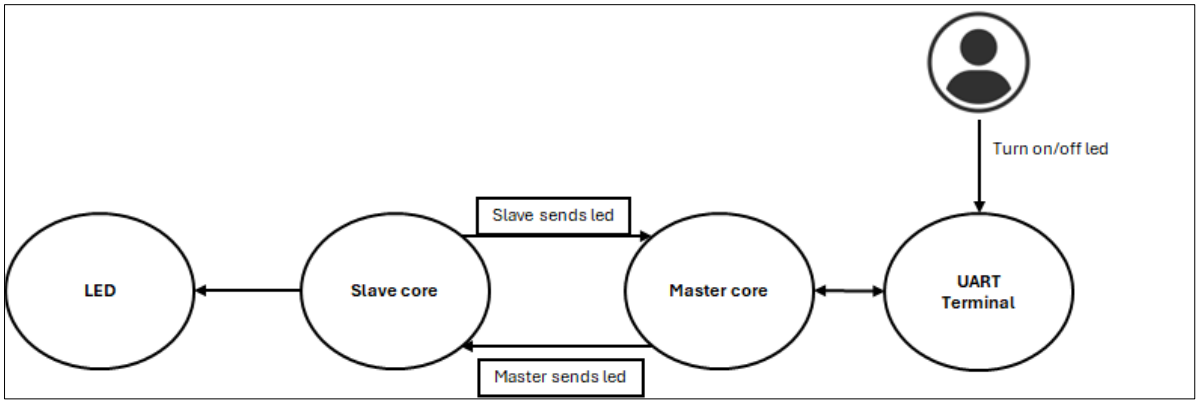


Figure 3-1. The behavior of the application for FSP x FSP

After setting up programming for the master core and slave core, the procedure for this case as below:

(1) Setting up running for the master core and slave core.

(2) UART console of Master core and LED of Slave core can operate to control LED ON and LED OFF.

(3) Master core reads data from UART that is to turn off/on for led as data below:

Data	Action
43210C01	Turn on LED
43210C00	Turn off LED
Anything	Do nothing

(4) After completing receive data, the master core will send states to turn on/off to the slave core.

(5) The slave core receives a state of led from the master core through OpenAMP. The slave core executes turn on/ off LED. Then, the slave core sends back to inform slave's implementation for led to the master core through OpenAMP.

(6) The master core receives slave's implementation for led through OpenAMP and print to console "Slave turned on LED" "Slave turned off LED".

3.1.2. Overview of application for FSP x Linux

The behavior of sample program is as follows:

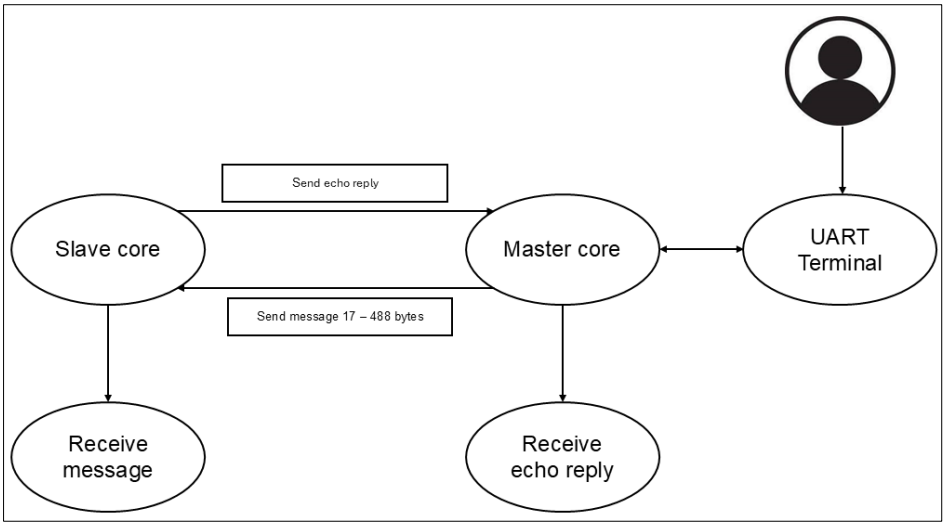


Figure 3-2. The behavior of the application for FSP x Linux

(1) Wait until a communication channel between FSP and Linux is established.

(2) Once the communication channel is established, Linux sample program starts to send the message to FSP while increasing its size from the minimum value 17 to the maximum value 488. At that time, the message like the following should be shown in the console connected to USB to Serial Port of RZN2H board:

```
Sending payload number 148 of size 165
```

(3) When FSP receives the message sent from Linux, the echo reply is sent back to Linux.

(4) When Linux receives the echo reply, the message below should be displayed in the console connected to USB to Serial Port of RZN2H board:

```
echo test: sent: 165
received payload number 148 of size 165
```

(5) After the message which has 488 bytes sized payload is sent from Linux to FSP and FSP sends back the echo reply, the message for terminating the communication channel is sent from Linux to FSP. Then, Linux and FSP sample programs output the following log messages to the corresponding consoles respectively when receiving the termination message.

Termination message on Linux side:

```
**********************************
Test Results: Error count = 0
**********************************
Quitting application .. Echo test end
Stopping application...
```

Termination message on FSP side:

```
De-initializating remoteproc
```

Then, FSP side re-waits for the establishment of connection channel. You can see the following log on the console a short time later:

```
creating remoteproc virtio
initializing rpmsg vdev
```

3.2. Hardware setup

3.2.1. Hardware setup when using e² studio

(1) Connect J-Link OB to RZ/N2H

(2) Connect USB to serial port to RZ/N2H

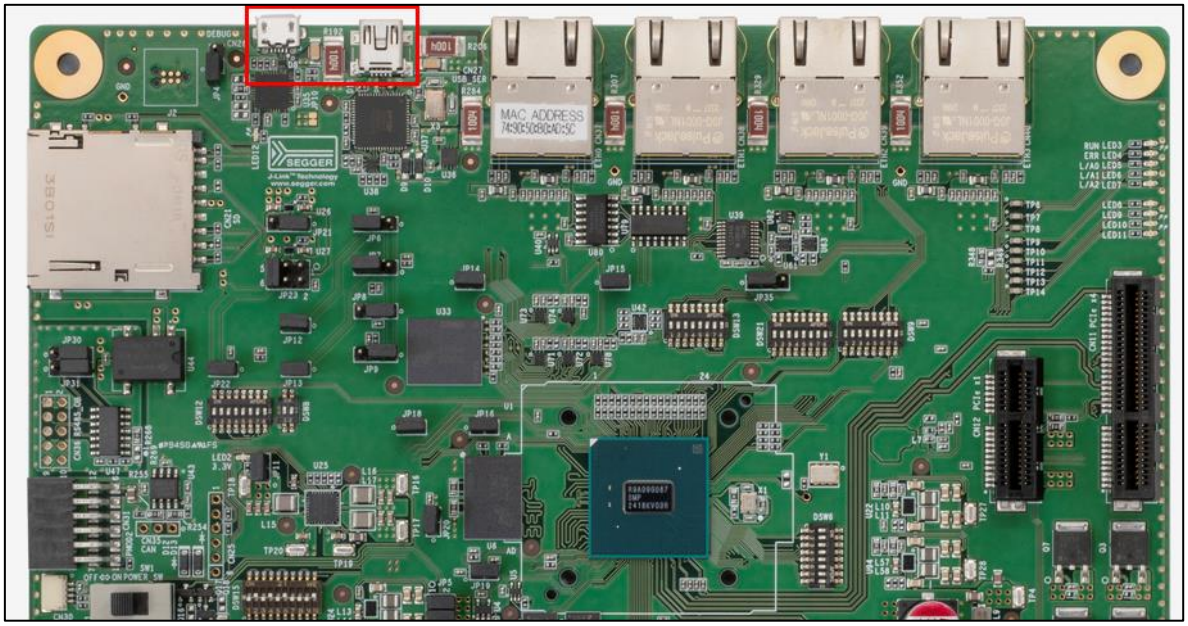


Figure 3-3. Connection J-Link, SCI (UART) to RZ/N2H board

(3) Setup operation mode switch setting for RZ/N2H evaluation board

Table 3-1. Operation mode switch setting

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

(4) Setup LED8 on RZ/N2H Evaluation Board

Table 3-2. Setting LED on RZ/N2H Evaluation Board.

LED	Setting	RZ/N2H Evaluation Board
DSW15.8	OFF	LED8
DSW15.9	OFF	
DSW15.10	ON	

(5) Setup SCI_UART on RZ/N2H Evaluation Board

Table 3-3. Setting signal Connections of USB-to-Serial Conversion on RZ/N2H Evaluation Board.

SCI_UART	Setting	RZ/N2H Evaluation Board
DSW9.1	ON	Reception of UART data 1 for USB-to-serial conversion Transmission of UART data 1 for USB-to-serial conversion
DSW9.2	OFF	
DSW9.3	ON	
DSW9.4	OFF	
DSW5.8	OFF	Reception of UART data 2 for USB-to-serial conversion Transmission of UART data 2 for USB-to-serial conversion
DSW9.5	ON	
DSW9.6	OFF	
DSW9.7	ON	
DSW9.8	OFF	

3.2.2. Hardware setup when using IAR EW for Arm

(1) Connect I-Jet to RZ/N2H

(2) Connect USB to serial port to RZ/N2H

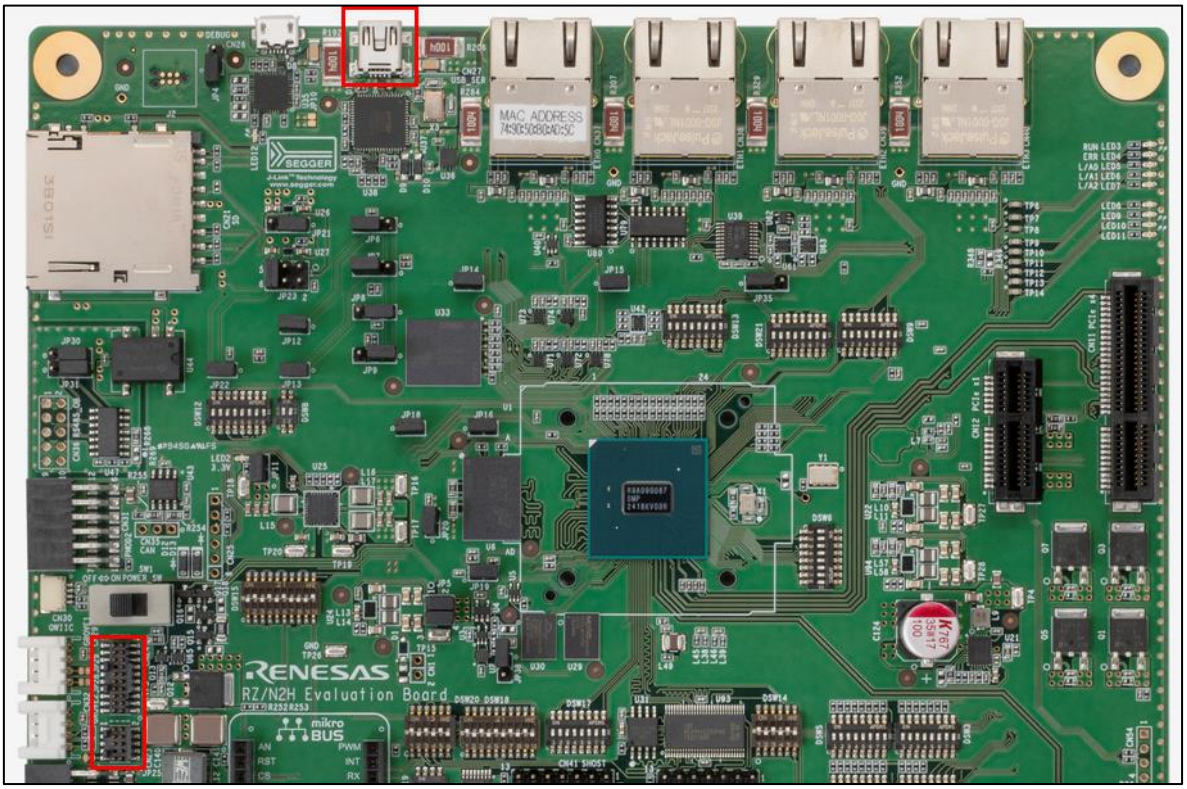


Figure 3-4. Connection I-Jet, SCI (UART) to RZ/N2H board

Note: Setup operation mode, SCI_UART and LED8 are similar when using e^2 studio.

3.3. Sample program setup on e² studio

Please carry out the following procedures for setting up the program running on multiple core.

(1) Extract *rzn2h_multiple_core_application.zip*

(2) Extract the GCC project at:

<*path_to_download*>/rzn2h_multiple_core_application/gcc/<*project_to_execute*>

(3) Open e² studio and click File » Import

(4) Double-click General and select Existing Projects into Workspace as shown in Figure 3-5:

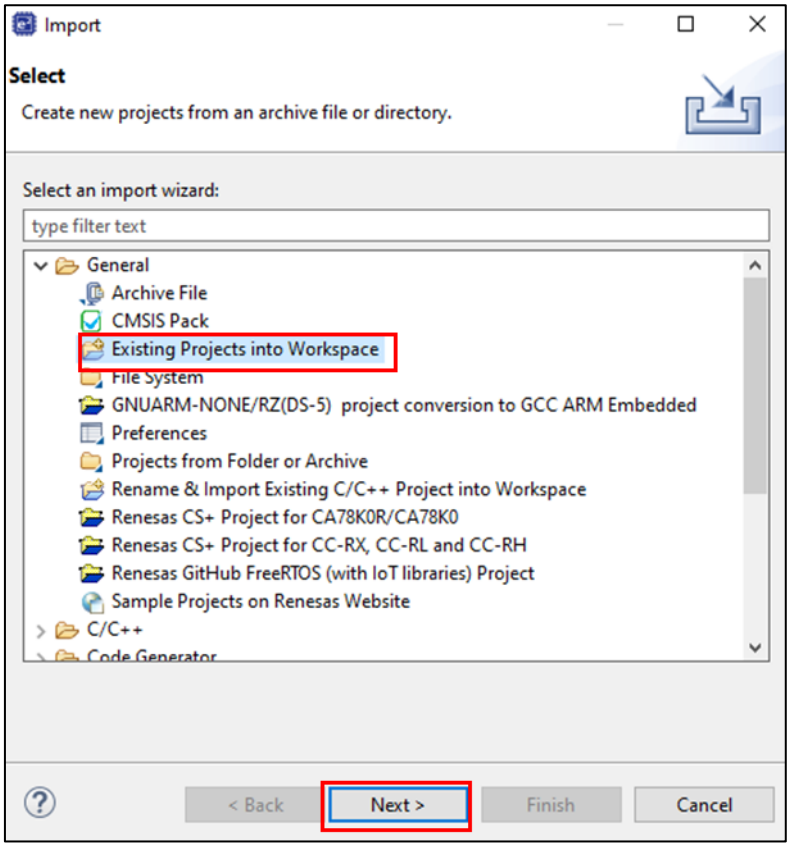


Figure 3-5. Import sample project (1)

(5) Input the path to the folder *rzn2h_<core>_<os>_mastercore* and *rzn2h_<core>_<os>_slavecore*, press Enter key and click Finish button

With *<core>:* ca55_3/ ca55_2/ ca55_1/ ca55_0/ cr52_1/ cr52_0

With *<os>:* baremetal/ freertos

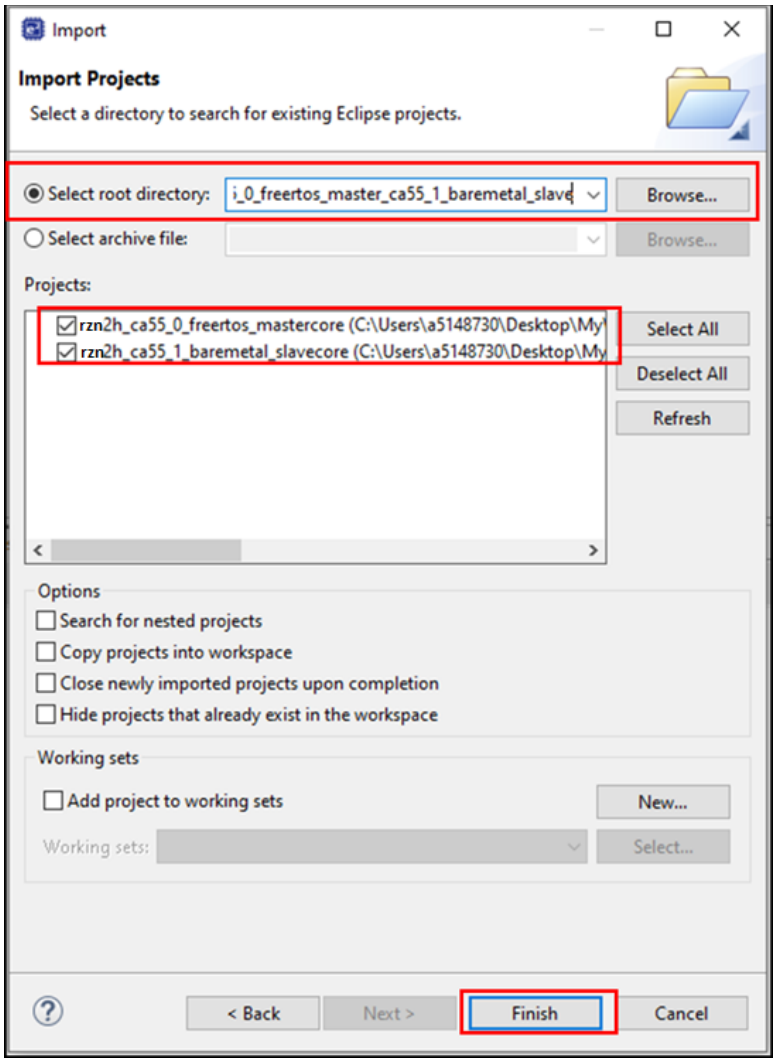


Figure 3-6. Import sample project (2)

(6) Build the project master and slave from Choose Project » Build Project

Please refer to the table below for the priority order when building the project:

Table 3-3. Priority order when building the project.

Priority	Project
1 (Highest)	cr52_0 project/ ca55_0 project (primary core)
2	cr52_1 project
3	ca55_0 project (secondary core)
4	ca55_1 project
5	ca55_2 project
6 (Lowest)	ca55_3 project

(7) Click Debug button in Debug Configuration

- If “A Renesas GDB debug session is already active” window below appears, please press “No” to go ahead.

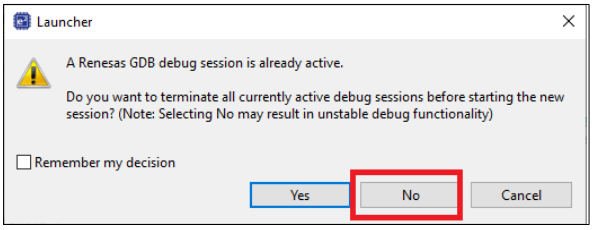


Figure 3-7. Renesas GDB debug session is already active

- If “Proceed with lauch” window below appears, please press “yes” to go ahead.

Figure 3-8. Proceed with launch

Note 1: Similar debug procedure for other combinations.

Note 2: For the FSP x Linux combination, refer to section 4.4 in the Release Note of the RZN Multi OS Package.

(8) Running programing for slave and master

There are 3 constraint as below:

- CR52_0 or CA55_0 must execute before other cores.

Note 3: In the case where CR52_0 or CA55_0 is neither a slave core nor a master core, rzn2h_cr52_0 boot /rzn2h_ca55_0_boot must run before other cores begin execution.

- In the case of CR52_0 or CA55_0 is master core, it should be setup running as procedure below:
  - Run the master core until it reaches the main function.
  - Running slave core programming.
  - Running master core programming again.
- Slave must run before running master as OpenAMP procedure.

(9) Sample application result

- Console of Master/Slave Core: OpenAMP opens successfully.

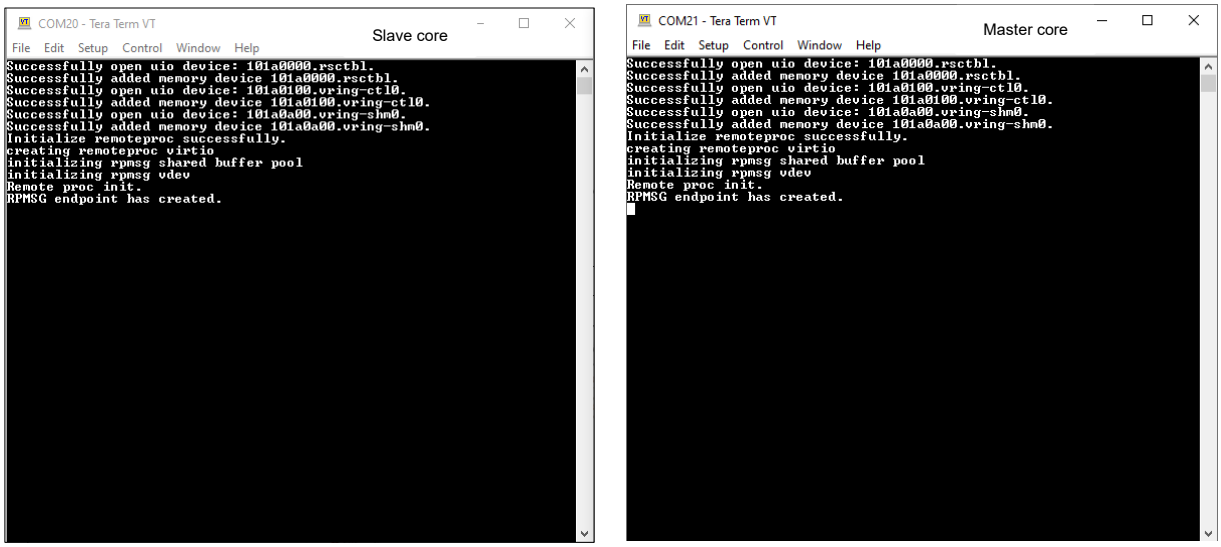


Figure 3-9. TeraTerm screen (Left: Slave Core, Right: Master Core)

- LED ON: Turn,

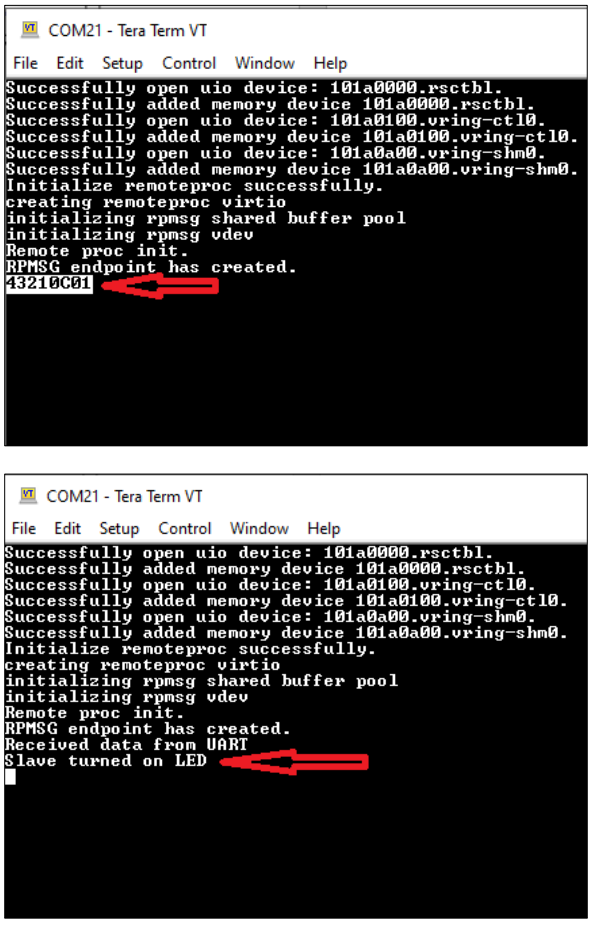


Figure 3-10. Enter 43210C01 into the Master Core TeraTerm screen

- LED OFF: Turn off,

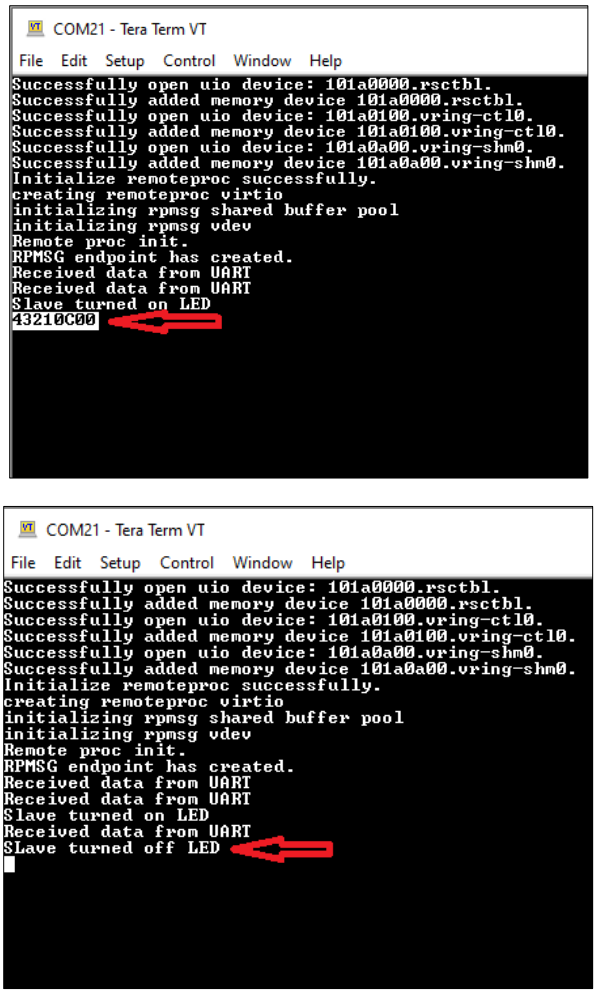


Figure 3-11. Enter 43210C00 into the Master Core TeraTerm screen

Note 4: If you want the input data (e.g., “43210C01” or “43210C00”) to be displayed in Tera Term, enable the Local Echo option under [Setup] >> [Terminal] in Tera Term, as shown in Figure 3-12.

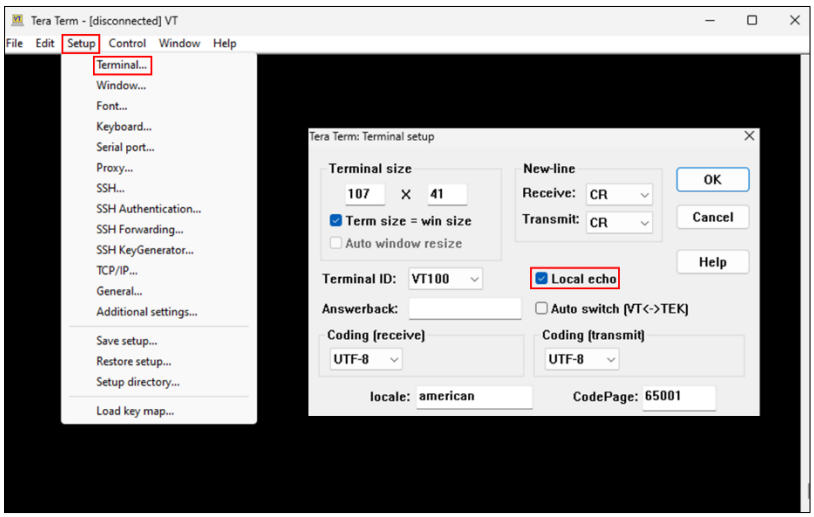


Figure 3-12. Enabling Local Echo to display input data in Tera Term

3.4. Sample program setup on IAR EW for Arm

Please carry out the following procedures for setting up the program running on multiple core.

(1) Extract *rzn2h_multiple_core_application.zip*

(2) Extract the IAR project at:

<path_to_download>/rzn2h_multiple_core_application/iar/<project_to_execute>

(3) Open IAR EW for Arm and click File » Open Worksapce

(4) Navigate to the folder containing the extracted project, select the file with the .eww extension, then click Open

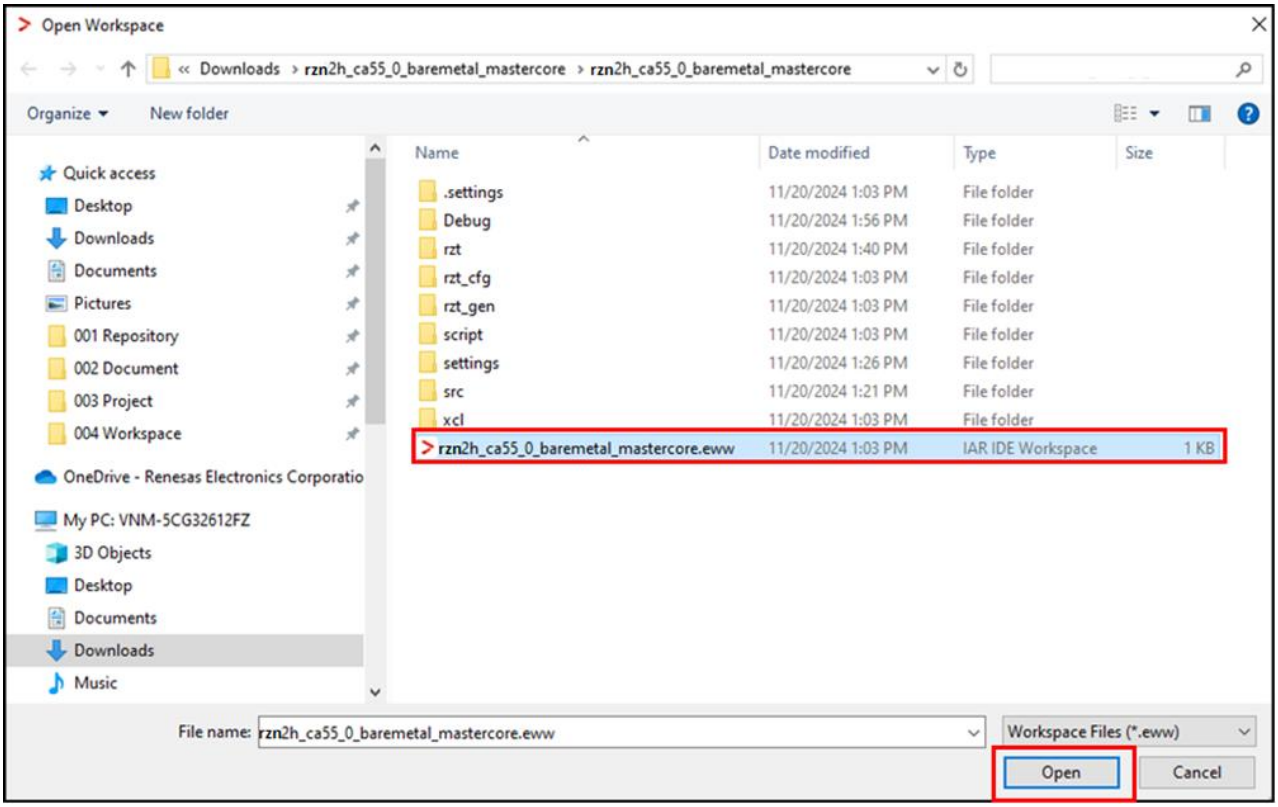


Figure 3-9. Open project on IAR EW for Arm

(5) Build the project from Choose Project >> Rebuild All

Please refer to Table 3.3 for the priority order when building the project.

(6) Click the Download and Debug button on the toolbar to enter debug mode

Figure 3-10. Config for debug

Note 1: For the FSP x Linux combination, refer to section 4.4 in the Release Note of the RZN Multi OS Package.

(7) Running programing for slave and master

There are 3 constraint as below:

- CR52_0 or CA55_0 must execute before other cores.

Note 2: In the case where CR52_0 or CA55_0 is neither a slave core nor a master core, rzn2h_cr52_0 boot /rzn2h_ca55_0_boot must run before other cores begin execution.

- In the case of CR52_0 or CA55_0 is master core, it should be setup running as procedure below:
  - Run the master core until it reaches the main function.
  - Running slave core programming.
  - Running master core programming again.
- Slave must run before running master as OpenAMP procedure.

(8) Sample application result

- Console of Master/Slave Core: OpenAMP opens successfully.

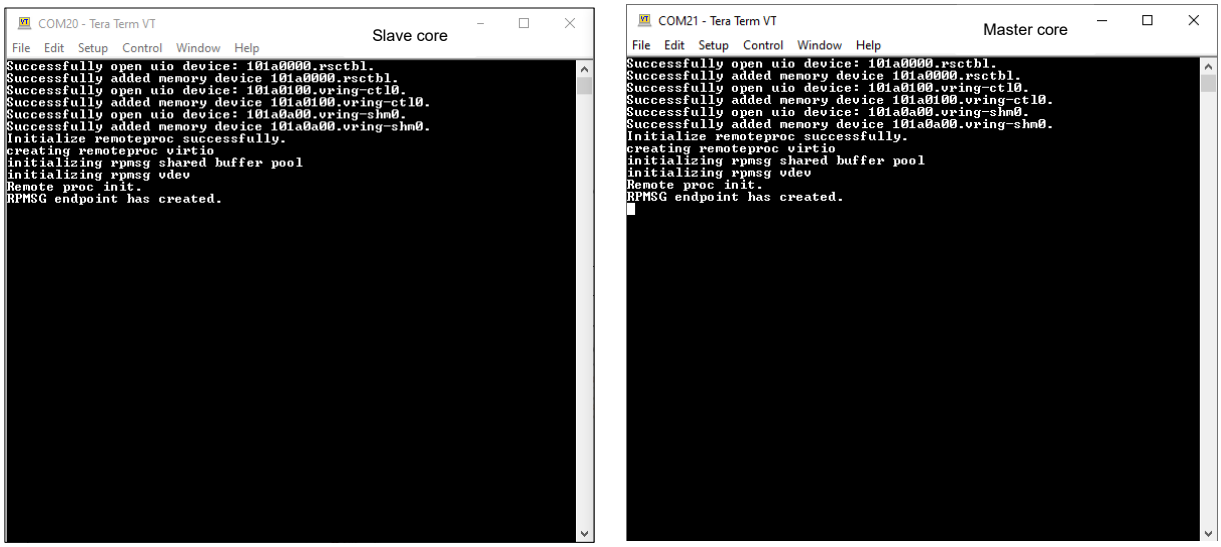


Figure 3-14. TeraTerm screen (Left: Slave Core, Right: Master Core)

- LED ON: Turn,

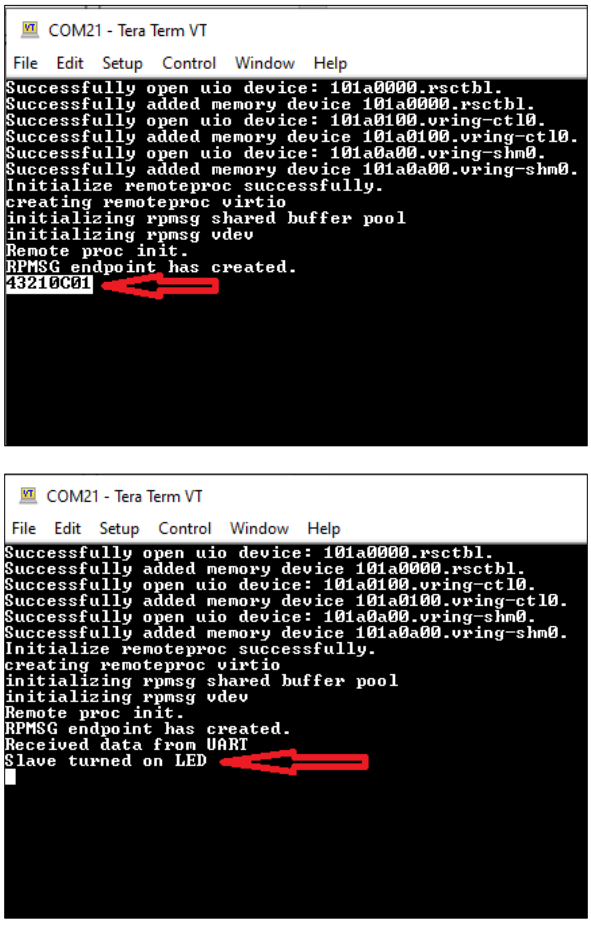


Figure 3-15. Enter 43210C01 into the Master Core TeraTerm screen

- LED OFF: Turn off,

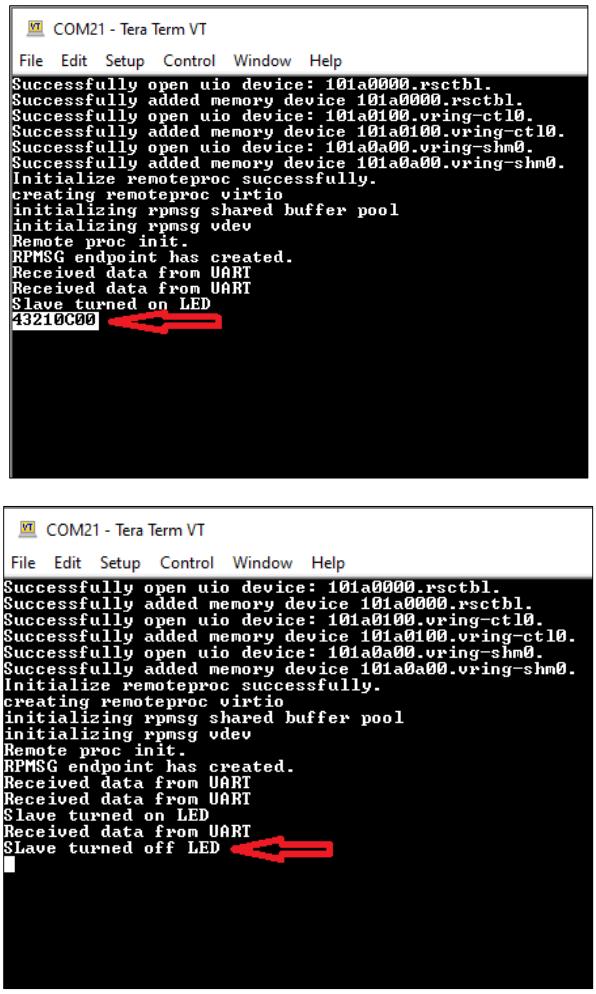


Figure 3-16. Enter 43210C00 into the Master Core TeraTerm screen

Note 3: If you want the input data (e.g., “43210C01” or “43210C00”) to be displayed in Tera Term, enable the Local Echo option under [Setup] >> [Terminal] in Tera Term, as shown in Figure 3-12.

4. Instructions for creating a new project (for unavailable combinations)

* Notes and constraints when creating new combinations that is not supported by the sample projects.

(1) In a combination, **there must be at least one primary core (CR52_0/CA55_0 core)** with a role that can be either a slave or a master core. If neither the slave nor the master core you want to combine includes a primary core, you will need to create a boot project.

Note: A boot project is a project created with the CR52_0 or CA55_0 core.

(2) When starting the creation process, make sure that the primary core project is created first and successfully built in order to generate the smartbundle file for other core projects to use.

Note: The second core project must be linked to the smartbundle file of the first core project. Similarly, the third core project needs to link to the smartbundle file of the second core project, and so on, in the following order: CR52_0, CR52_1, CA55_0, CA55_1, CA55_2, CA55_3.

Example: Creating a combination with CR52_1 (slave) and CA55_1 (master).

- There is no primary core in the combination, so a boot project needs to be created.
- If you choose the CR52_0 core as the boot project, the project order will be: CR52_0 → CR52_1 → CA55_1.
- If you choose the CA55_0 core as the boot project, the project order will be: CA55_0 → CR52_1 → CA55_1.

4.1. Guidelines on e² studio

4.1.1. Baremetal project on e² studio

Refer to the Getting Started document, section "***6. FSP Configuration Users Guide***", to create a project in e² studio.

(1) Notes when creating a new project

(1) In the Device and Tools Selection window, select "*<FSP version>*" for FSP Version from the dropdown list, select "*RZN2H Evaluation Board (RAM execution without flash memory)*" for Board, select "*R9A09G087M44GBG*" for Device, select the desired Core, and choose "*C*" for Language.

(2) For Toolchains, select "*GNU ARM Embedded*" version "*<GNU Toolchain version>*" if using the CR52 core, or select "*GCC ARM A-Profile (AArch64 bare-metal)*" version "*<GCC Toolchain version>*" if using the CA55 core.

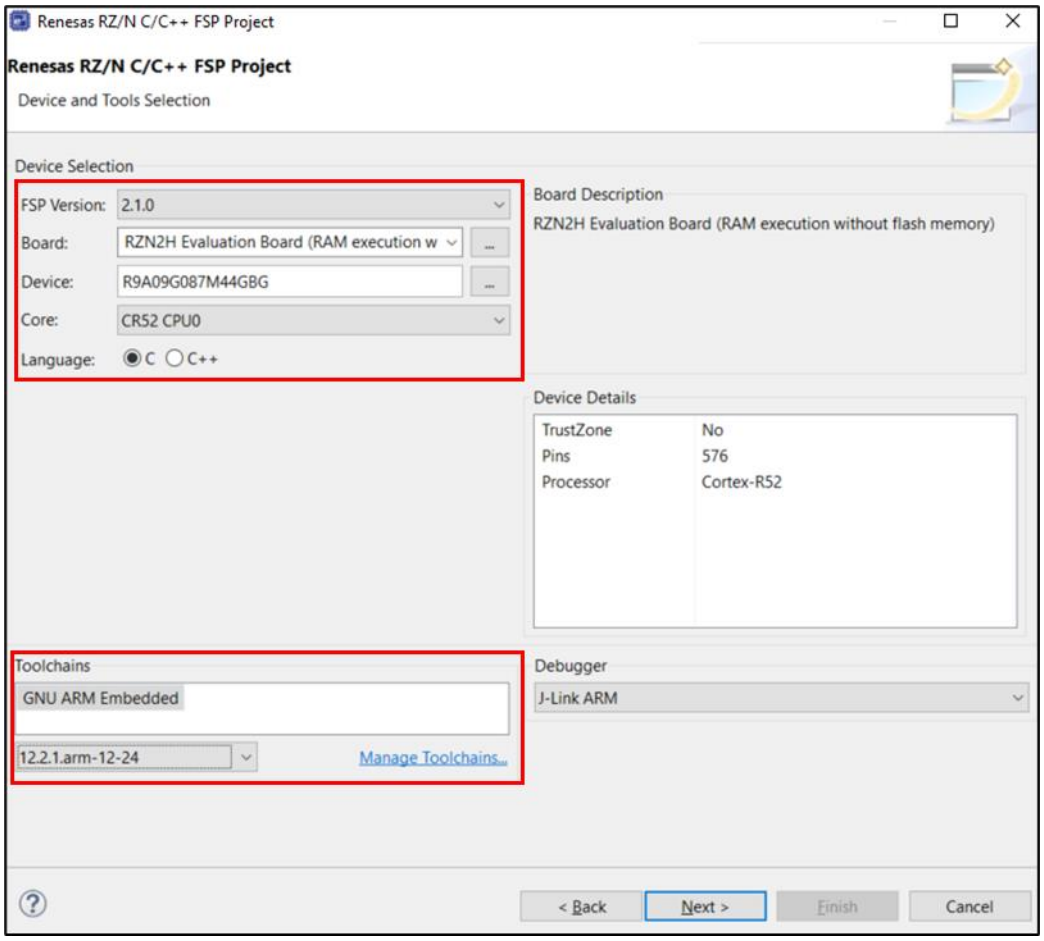


Figure 4-1. Device and Toolchain configuration

(3) In the Preceding Project or Smart Bundle Selection window, select "*None*" if the core project is the primary core or boot project. Select "*Preceding Project*" in all other cases.

(4) Except for the CA55_0 core, its role must be considered carefully. If CA55_0 is not the primary core (in the combination of CR52_0 core and CA55_0 core), select "*Preceding Project*" and link to the CR52_0 project in the dropdown list.

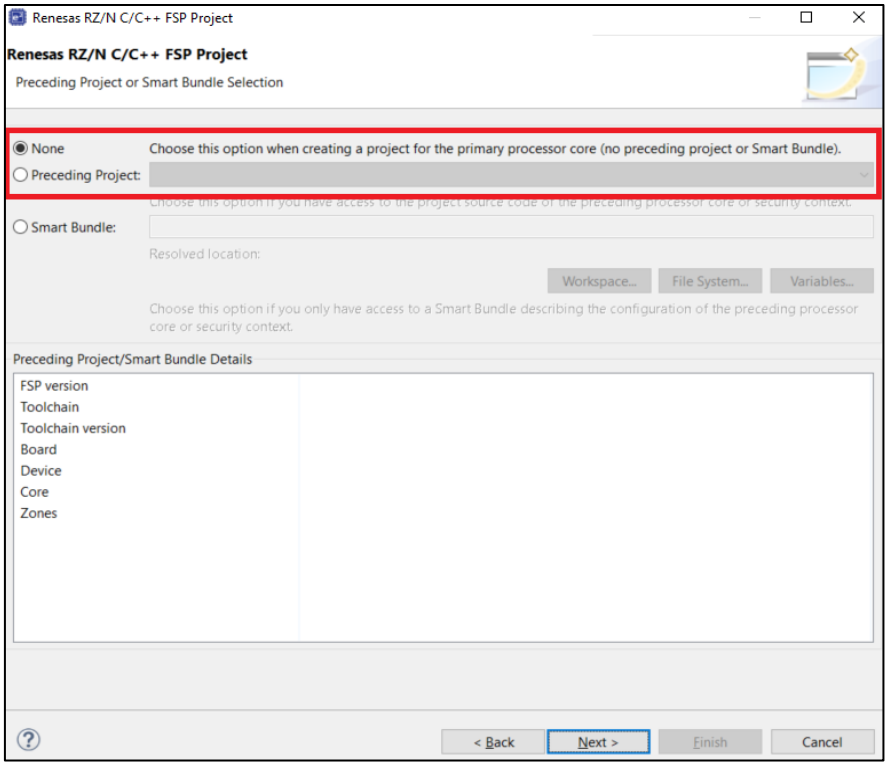


Figure 4-2. Preceding Project or Smart Bundle configuration

(5) In the Build Artifact and RTOS Selection window, select "*No RTOS*" for RTOS Selection.

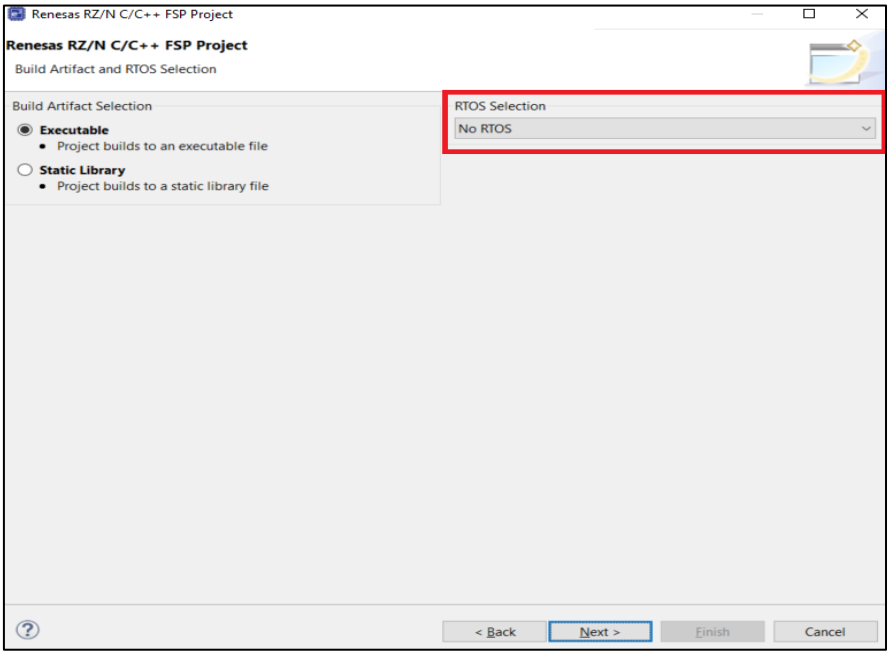


Figure 4-3. Build Artifact and RTOS configuration

(6) In the Project Template Selection window, select "*Bare Metal – Minimal*".

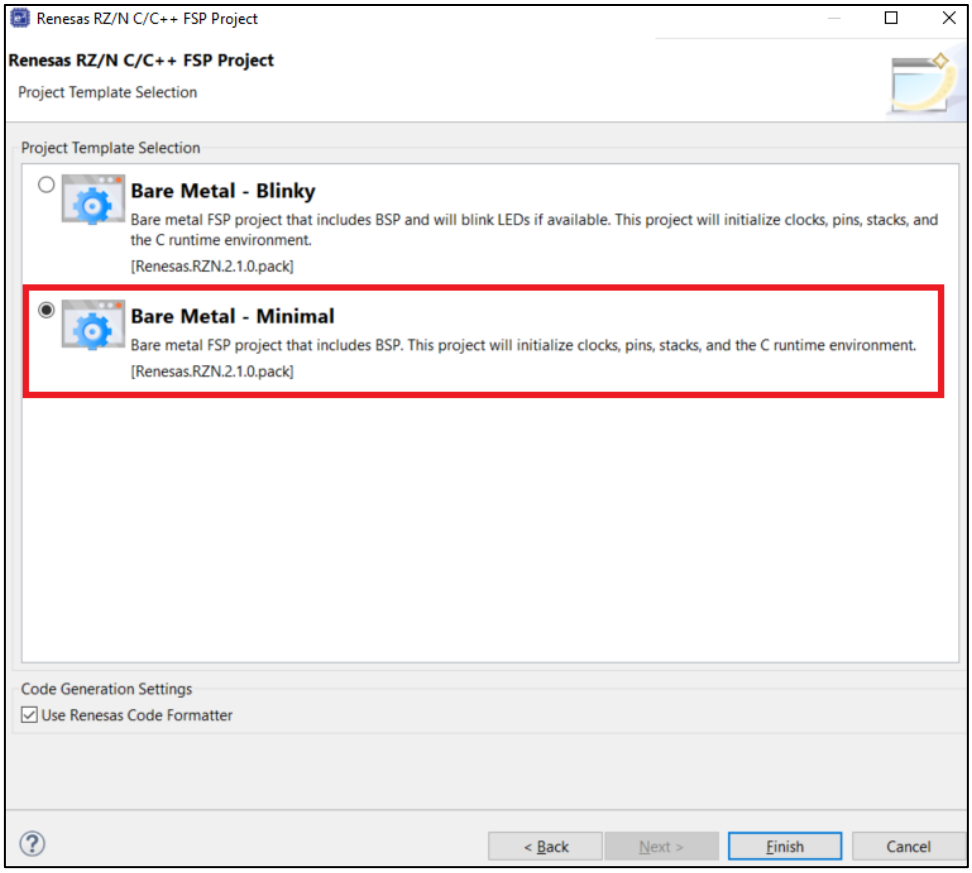


Figure 4-4. Project Template configuration

(2) Configuration in the *configuration.xml* file

(1) Add the following drivers to the Stacks tab.

- OpenAMP
- Inter-CPU IRQ
- UART

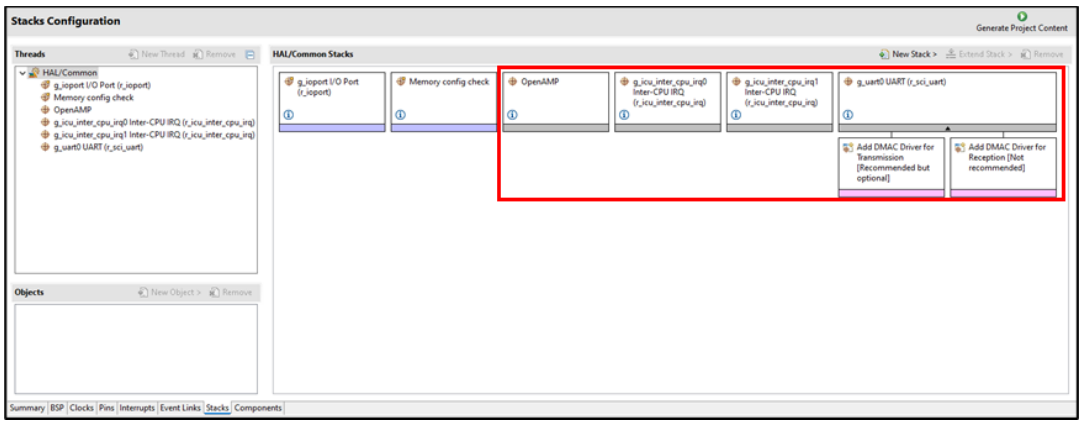


Figure 4-5. Stacks tab configuration

(2) Configure the drivers for the slave core and master core as shown below.

(Note that the boot project will not require configuration for the Stacks tab)

Inter-CPU IRQ			
Element	**Master core**	**Slave core**	**Remarks**
Parameter checking	Enable	Enable	-
g_icu_intercpu_irq0(channel)	**1**	**0**	-
g_icu_intercpu_irq1(channel)	**0**	**1**	-
g_icu_intercpu_irq0(Callback)	NULL	NULL	-
g_icu_intercpu_irq1(Callback)	rzn2_callback	rzn2_callback	-
g_icu_intercpu_irq0 (Software Interrupt Priority)	Disable	Disable	-
g_icu_intercpu_irq1 (Software Interrupt Priority)	30	30	-

UART			
Element	**Master core**	**Slave core**	**Remarks**
Parameter checking	Enable	Enable	-
Channel	1	**0**	-
Callback	uart_callback	uart_callback	-
Receive Interrupt Priority	30	30	-
Transmit Data Empty Interrupt Priority	30	30	-
Transmit End Interrupt Priority	30	30	-
Error Interrupt Priority	30	30	-

(3) Configure the BSP and Clocks tab for projects that are the primary core.

- For CR52 core: In the Properties of the BSP tab, under RZN2H → LPDDR4 SDRAM Subsystem, select "*Enable*" for DDR Initialization and Zero-initialized for DDR memory. Under RZN2H Memory Config → CPU MPU → Region → Region 21, select "*Outer Shareable*" for the Shareability field and choose "*Attribute 3*" for the Attribute Index.

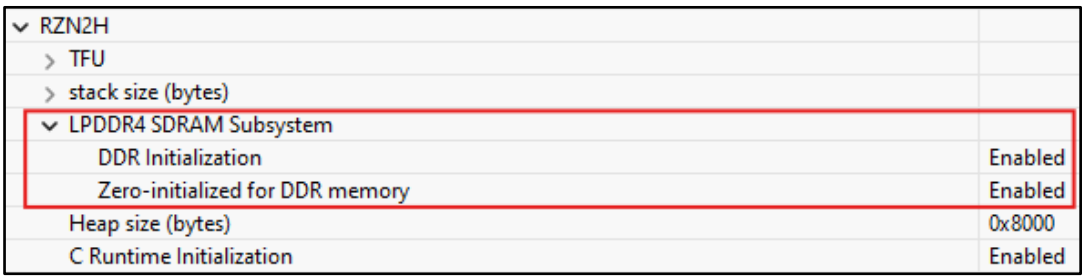


Figure 4-6. BSP tab configuration for CR52 core

Name	DDR mirror
Base	0xC0000000
Limit	0xFFFFFFFF
Sharebility field	Outer Shareable
Access Permission(EL1 / EL0)	ReadWrite / ReadWrite
Execute never	Execute Enable
Attribute Index	Attribute 3
Region enable	Enabled

Figure 4-7. BSP tab configuration for CR52 core

- For CR52 core: In the Properties of the Clocks tab, change from "*PLL2 is standby state*" to "*PLL2 is released from standby state*".

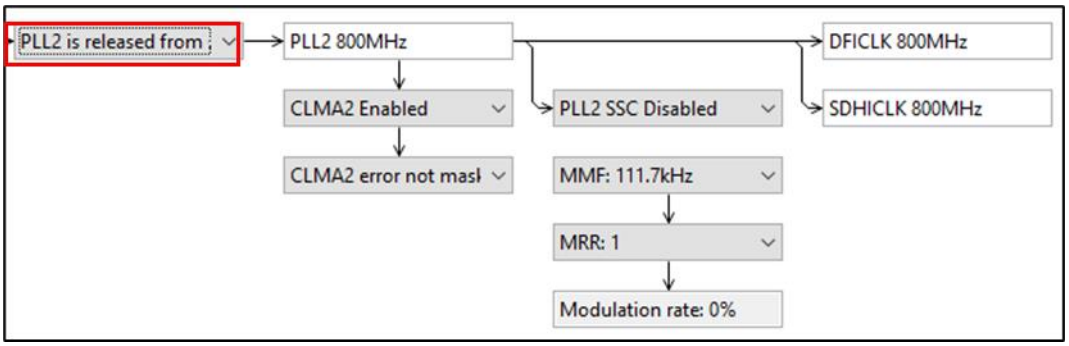


Figure 4-8. Clocks tab configuration for CR52 core

- For CA55 core: In the Properties of the BSP tab, under RZN2H → LPDDR4 SDRAM Subsystem, select "*Enable*" for DDR Initialization and Zero-initialized for DDR memory. Under RZN2H Memory Config → CPU MMU → Region → Region 37, select "*Outer Shareable*" for the Shareability field and choose "*Attribute 3*" for the Attribute Index.

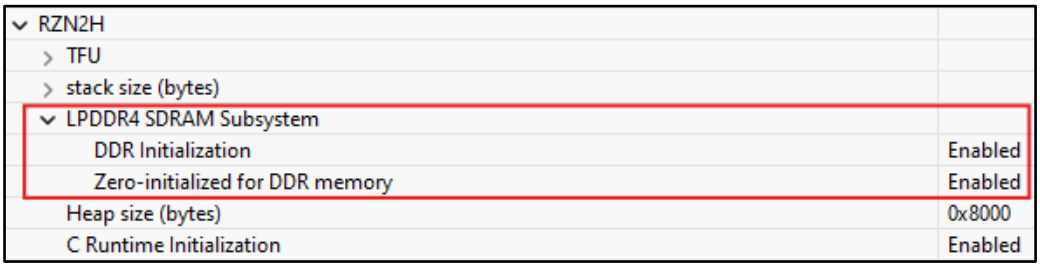


Figure 4-9. BSP tab configuration for CA55 core

Name	DDR mirror
Virtual address	0xC0000000
Physical address	0xC0000000
Size	0x40000000
Unprivileged Execute Never(UXN)	Execute Enable
Privileged Execute Never(PXN)	Execute Enable
Sharebility	Outer Shareable
Access Permissions(EL0/EL1 to 3)	Read and write/Read and write
Security bit(only EL3 and EL1)	Secure
Attribute Index	Attribute 3
Region Enable	Enable

Figure 4-10. BSP tab configuration for CA55 core

- For CA55 core: In the Properties of the Clocks tab, change from "*PLL2 is standby state*" to "*PLL2 is released from standby state*".

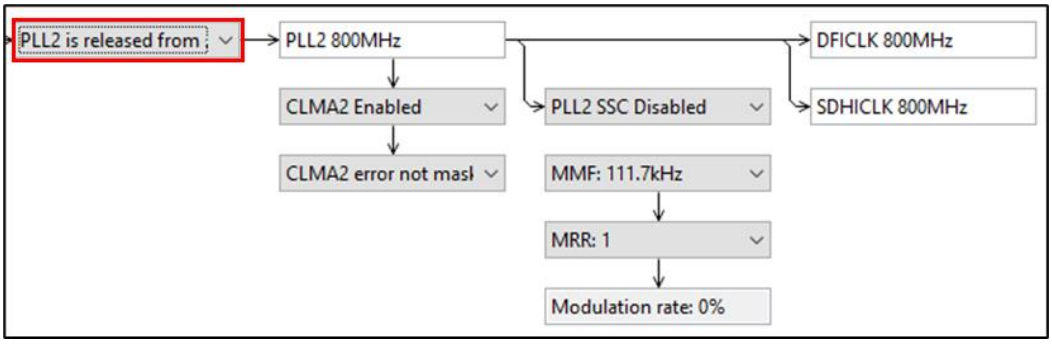


Figure 4-11. Clocks tab configuration for CA55 core

(4) Configure the BSP and Clocks tab for projects that are not the primary core.

- For CR52 core: In the Properties of the BSP tab, under RZN2H Memory Config → CPU MPU → Region → Region 21, select "*Outer Shareable*" for the Shareability field and choose "*Attribute 3*" for the Attribute Index.

Region 15	
Name	DDR mirror
Base	0xC0000000
Limit	0xFFFFFFFF
Sharebility field	Outer Shareable
Access Permission(EL1 / EL0)	ReadWrite / ReadWrite
Execute never	Execute Enable
Attribute Index	Attribute 3
Region enable	Enabled

Figure 4-12. BSP tab configuration for CR52 core

- For CA55 core: In the Properties of the BSP tab, under RZN2H Memory Config → CPU MMU → Region → Region 37, select "*Outer Shareable*" for the Shareability field and choose "*Attribute 3*" for the Attribute Index.

Name	DDR mirror
Virtual address	0xC0000000
Physical address	0xC0000000
Size	0x40000000
Unprivileged Execute Never(UXN)	Execute Enable
Privileged Execute Never(PXN)	Execute Enable
Sharebility	Outer Shareable
Access Permissions(EL0/EL1 to 3)	Read and write/Read and write
Security bit(only EL3 and EL1)	Secure
Attribute Index	Attribute 3
Region Enable	Enable

Figure 4-13. BSP tab configuration for CA55 core

(5) After completing the corresponding configurations above, click the Generate Project Content button.

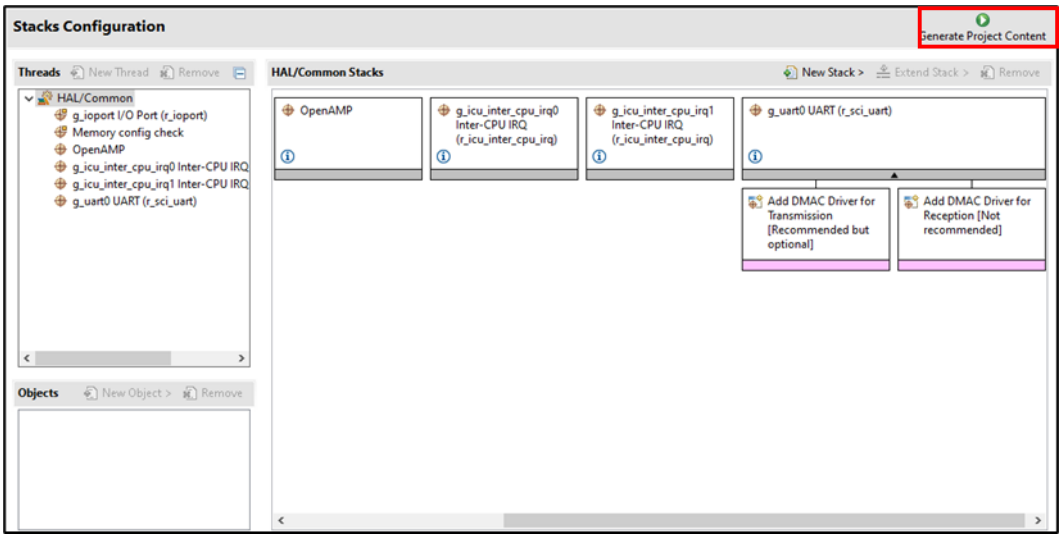


Figure 4-14. Generate project

(3) Copy the Baremetal folder

(1) For the master core project, copy the files in the "*Baremetal*" folder from "*Lib_mastercore*" into the "*src*" directory of the project.

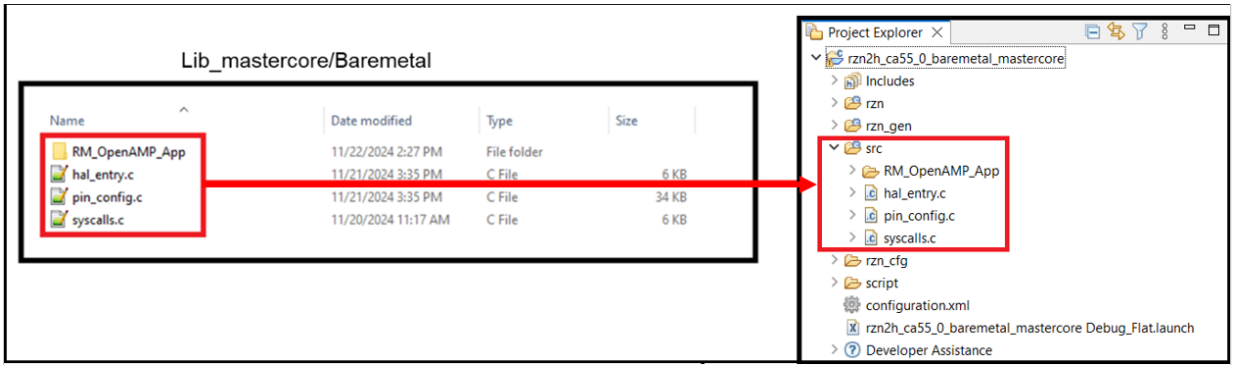


Figure 4-15. Copy the Baremetal folder from Lib_mastercore into the "src" directory

(2) For the slave core project, copy the files in the "*Baremetal*" folder from "*Lib_slavecore*" into the "*src*" directory of the project.

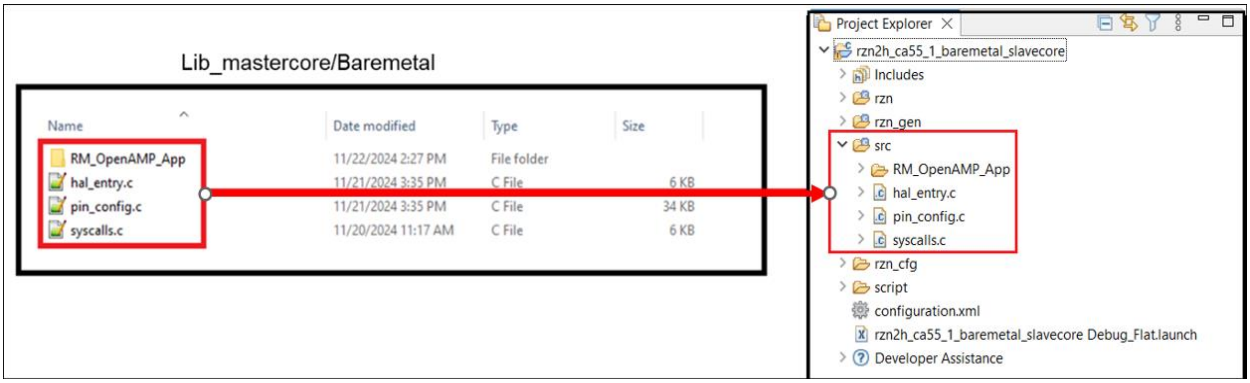


Figure 4-16. Copy the Baremetal folder from Lib_slavecore into the "src" directory

Note: For the boot project, copy the files in the "*Lib_bootproject*" folder into the "*src*" directory of the project.

(4) Copy the Patch folder

- Copy and overwrite the files in the Patch folder into the corresponding paths for both master project and slave project as shown below.

Name file	Path to folder
redefinition_memory_regions_gcc.h	script/
fsp_ram_execution.ld	script/

Example: Overwrite the Patch/ fsp_ram_execution.ld into "rzn2h_ca55_0_baremetal_master_ca55_1_baremetal_slave\rzn2h_ca55_0_baremetal_mastercore/script/ fsp_ram_execution.ld".

(5) Build combination

Refer to step 6 of sections 3.3 of this document.

(6) Configuration the Debug for combination

You need to follow the following steps to invoke Multiple-Core sample program with Segger J-Link.

(1) Configure the debugger for project by clicking » Debugger Configurations ... or by selecting the dropdown menu next to the bug icon and selecting Debugger Configurations ...

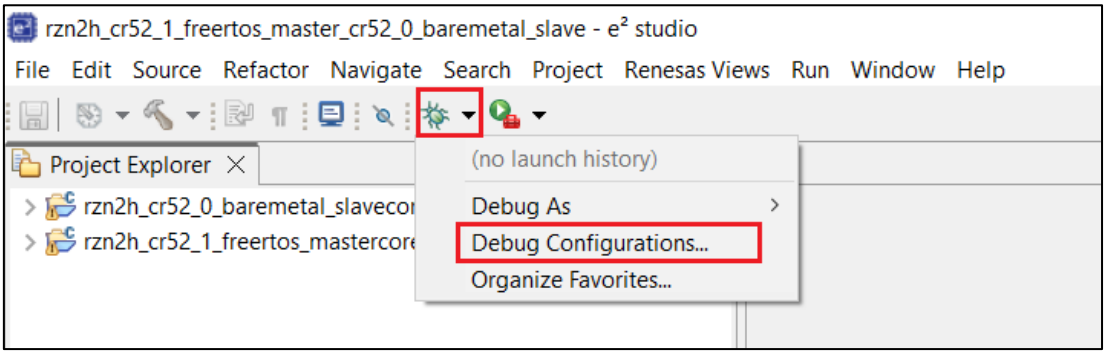


Figure 4-17. Select of Debug Configuration

(2) In the window Debug Configurations double click on "*Renesas GDB Hardware Debugging*" to create new configurations.

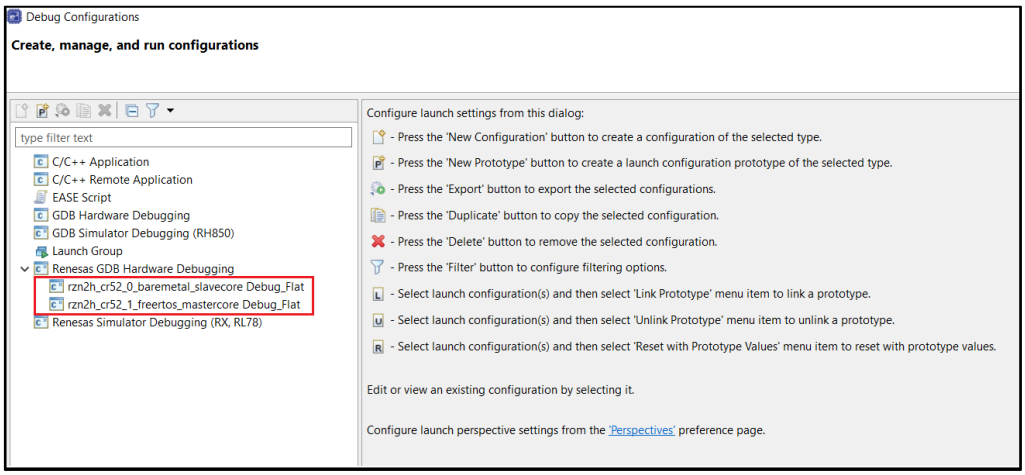


Figure 4-18. Configuration for debugging

- Select Main tab:

  Set *rzn2h_<core>_<os>_mastercore* and *rzn2h_<core>_<os>_slavecore* to Project with Browser button.

  With *<core>:* ca55_3/ ca55_2/ ca55_1/ ca55_0/ cr52_1/ cr52_0

  With *<os>:* baremetal/ freertos

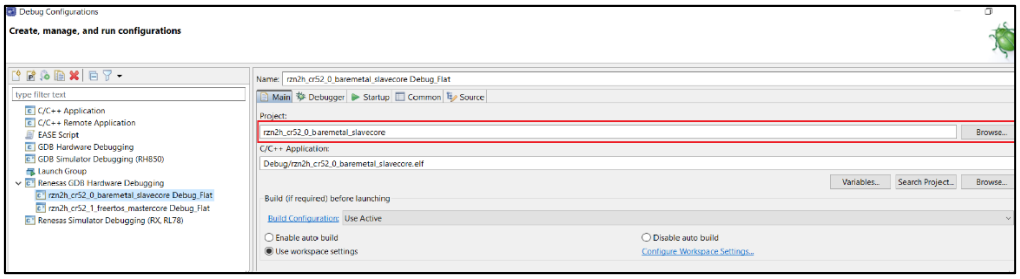


Figure 4-19. Configuration for debugging

- Debug hardware select J-Link ARM.
- Target Device select RZ » RZ/RZN2H » R9A09G087M44_CR52_0/ R9A09G087M44_CR52_1/ R9A09G087M44_CA55_0/R9A09G087M44_CA55_1/R9A09G087M44_CA55_2/R9A09G087M44_A55_3 depending on selection type of cores for master and slave.

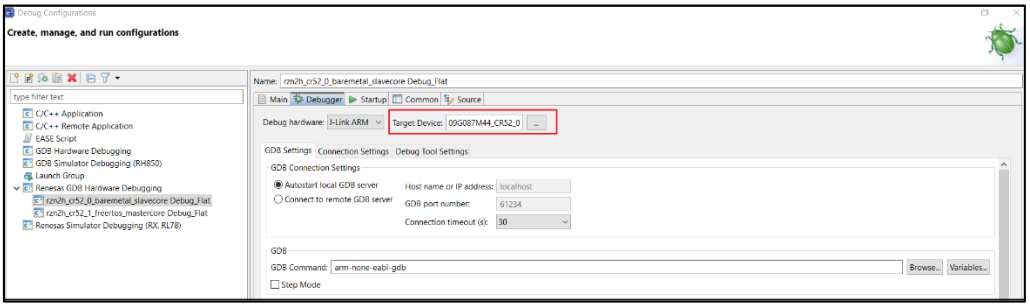


Figure 4-20. Configuration for target device

- Select Connection Settings tab:

  (CR52_0 only) Set CPSR (5bit) after download to “**Yes**”.

  (CR52_1 only) Add “${workspace_loc:/${ProjName}}/script/initialization_TCM.JlinkScript” to the Script File field.

  (CA55 only) Set Reset at the beginning of connection to “**No**”.

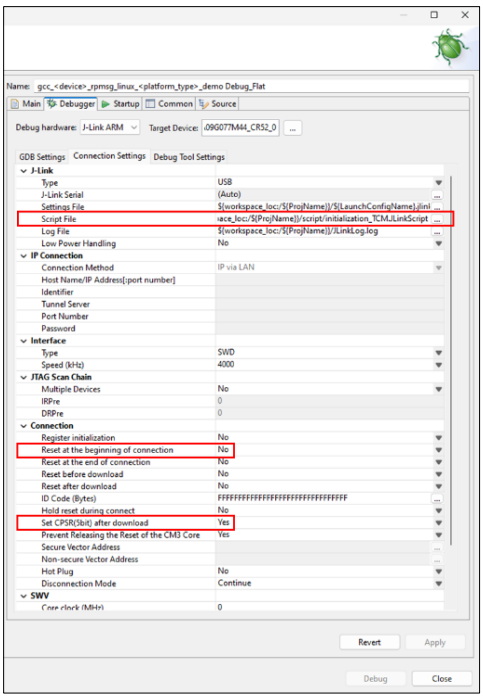


Figure 4-21. Debug Configuration

(3) Click "*Apply*" then "*Debug*" button in Debug Configuration.

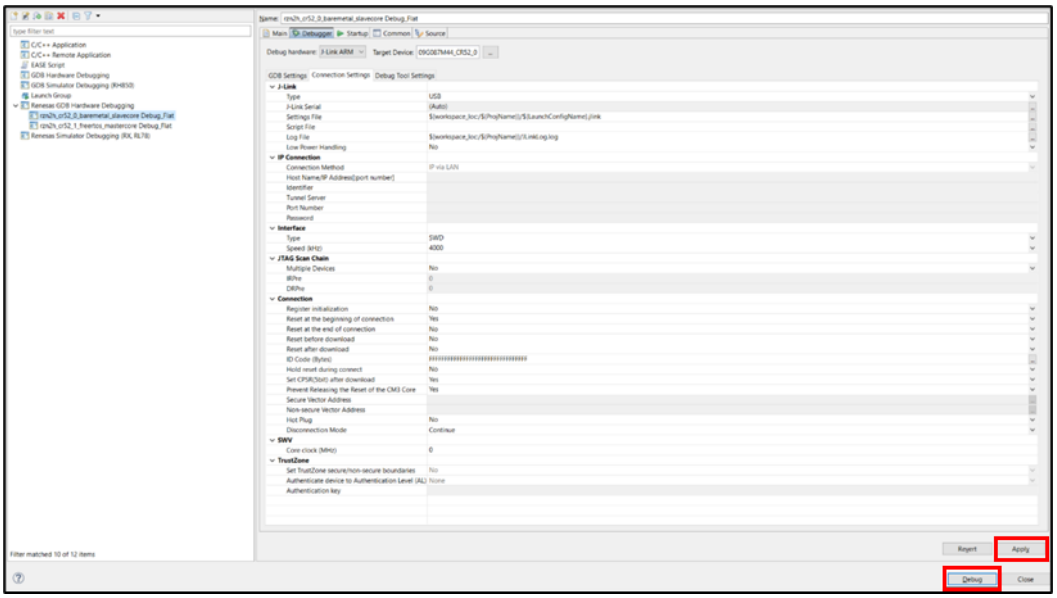


Figure 4-22. Save and run the debugging configuration

(4) Perform the same steps as in step 7 of section 3.3.

4.1.2. FreeRTOS project on e² studio

Follow the same steps as in section "*4.1.1 Baremetal project on e² studio*" with the following notes:

(1) In the Build Artifact and RTOS Selection window, select *(<FSP version>)*" for RTOS Selection

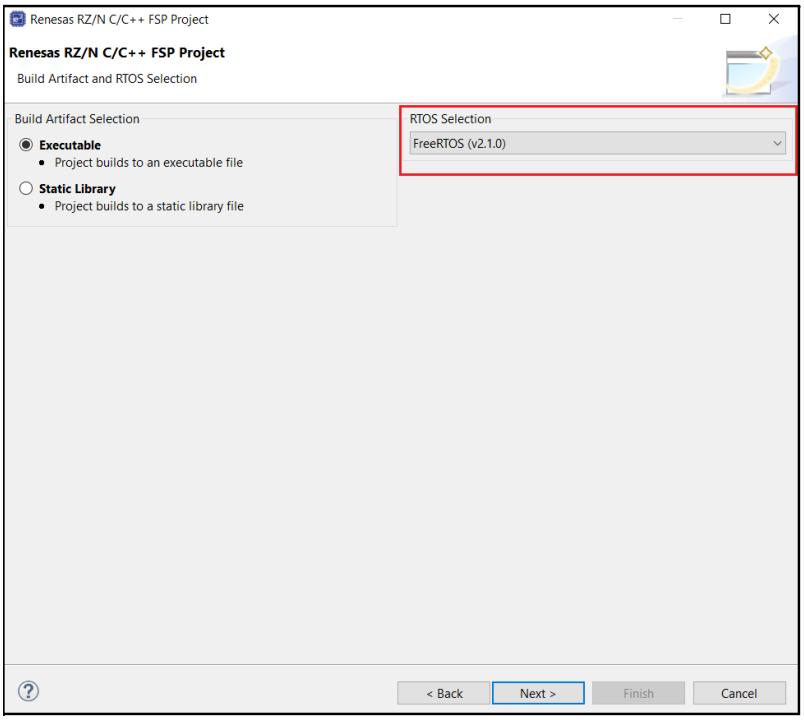


Figure 4-23. Build Artifact and RTOS configuration

(2) In the Project Template Selection window, select "*FreeRTOS – Minimal – Static Allocation*"

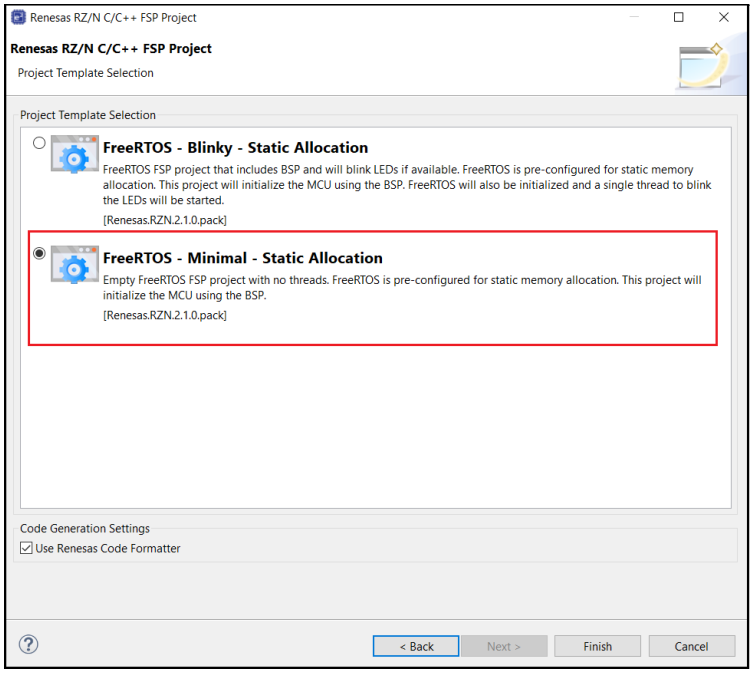


Figure 4-24. Project Template configuration

(3) Add a New Thread for FreeRTOS

In the Stacks tab, under the Threads section, click New Thread.

(4) Copy the FreeRTOS folder

(1) For the master core project, copy the files in the "*FreeRTOS*" folder from "*Lib_mastercore*" into the "*src*" directory of the project.

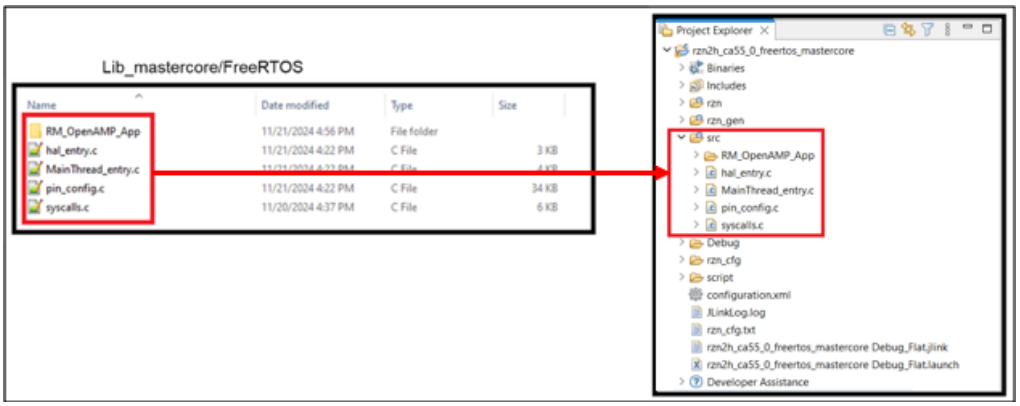


Figure 4-25. Copy the FreeRTOS folder from Lib_mastercore into the "src" directory

(2) For the slave core project, copy the files in the "*FreeRTOS*" folder from "*Lib_slavecore*" into the "*src*" directory of the project.

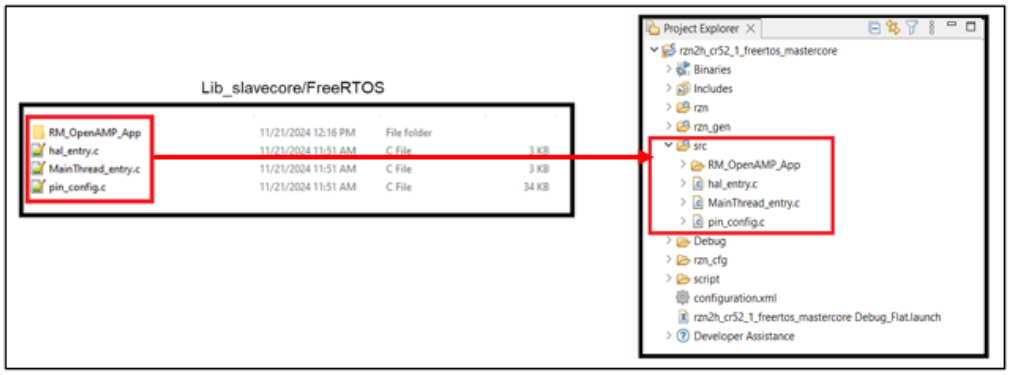


Figure 4-26. Copy the FreeRTOS folder from Lib_mastercore into the "src" directory

Note 1: In a combination, only one core project is allowed to have FreeRTOS.

4.2. Guidelines on IAR EW for Arm

4.2.1. Baremetal project on IAR EW for Arm

Refer to the Getting Started document, section "***5. FSP Smart Configurator User Guide***", to create a project in IAR EW for ARM.

(1) Notes when creating a new project

(1) In the Device and Tools Selection window, select "*<FSP version>*" for FSP Version from the dropdown list, select "*RZN2H Evaluation Board (RAM execution without flash memory)*" for Board, select "*R9A09G087M44GBG*" for Device, select the desired Core, and choose "*C*" for Language.

(2) For IDE Project Type, select "*IAR EWARM [v9.60+]*".

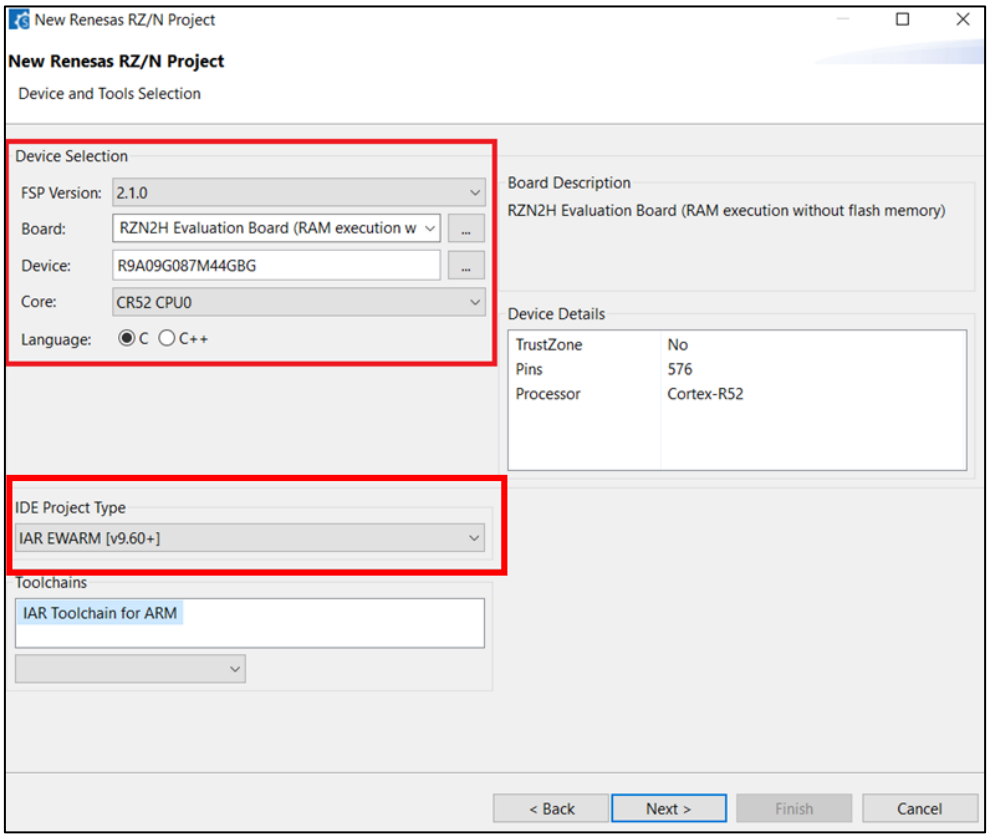


Figure 4-27. Device and Toolchain configuration

(3) In the Existing Smart Bundle Selection window, uncheck "*Use Smart Bundle*" if the core project is the primary core or boot project. Check "*Use Smart Bundle*" in all other cases.

(4) If "*Use Smart Bundle*" is selected, click Browse and navigate to the smart bundle file at "*path_to_project/Debug/Exe/.sbd file*".

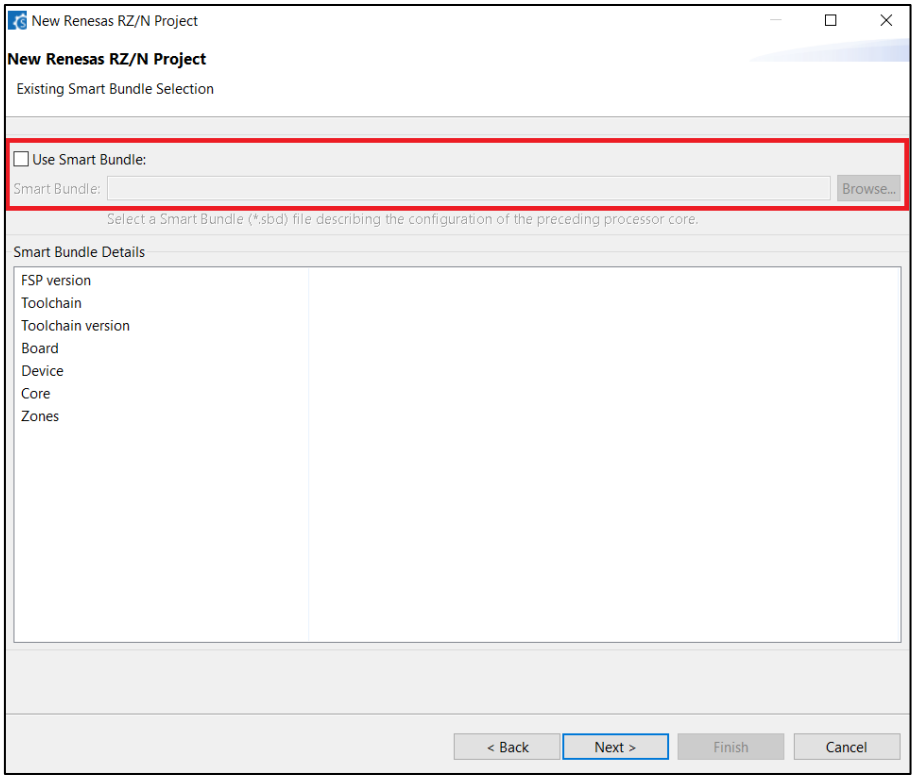


Figure 4-28. Existing Smart Bundle configuration

(5) In the RTOS Selection window, select "*No RTOS*" for RTOS Selection.

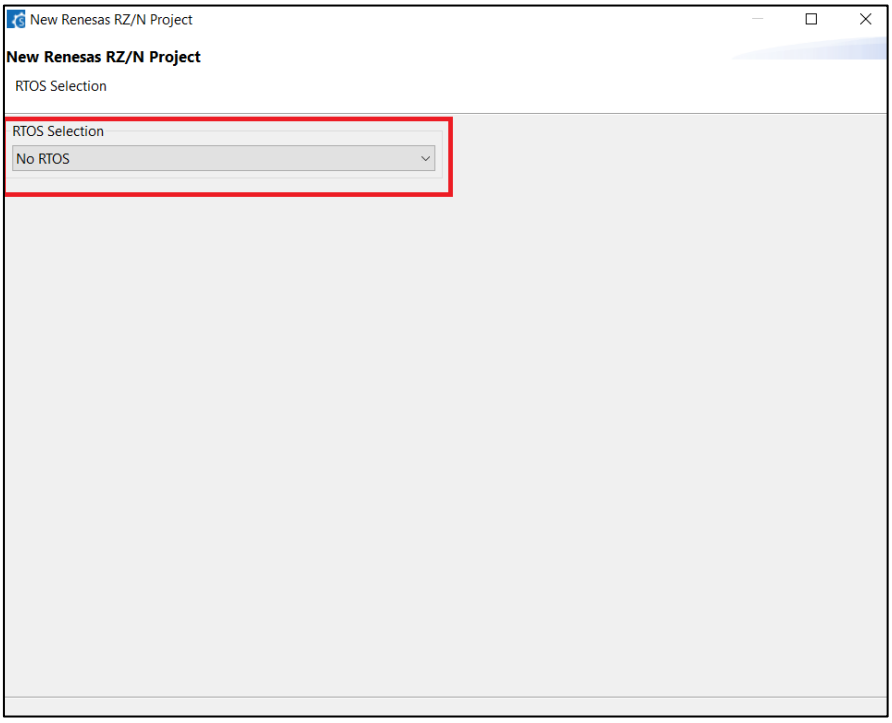


Figure 4-29. RTOS configuration

(6) In the Project Template Selection window, select "*Bare Metal – Minimal*".

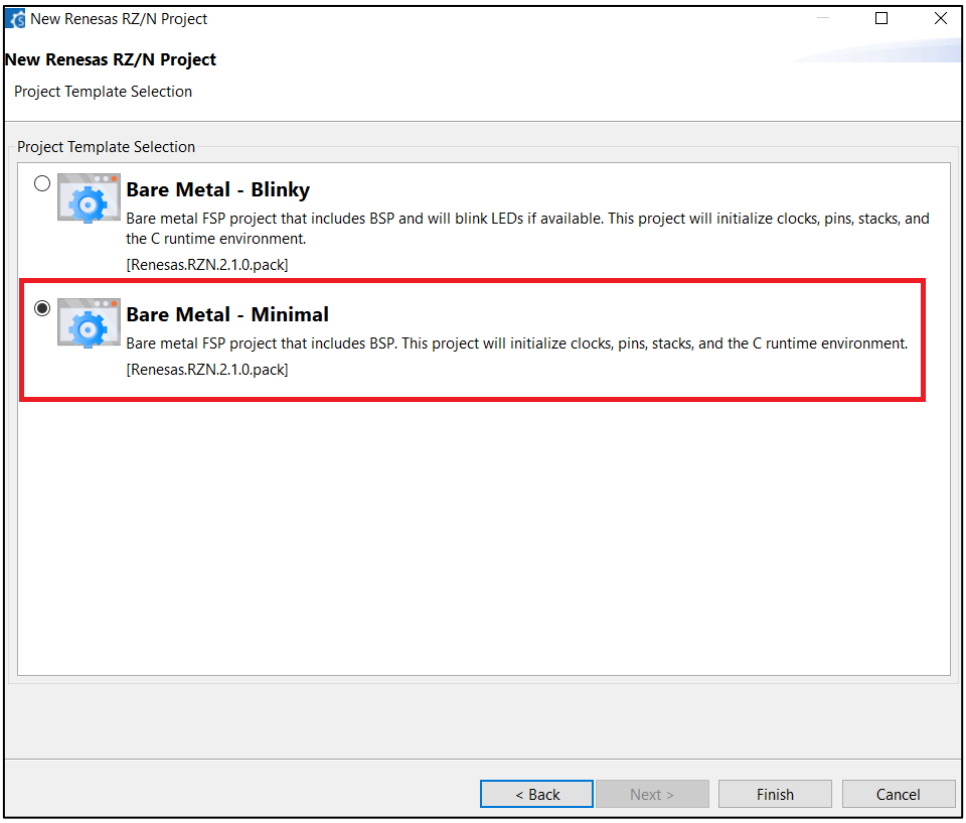


Figure 4-30. Project Template configuration

(2) Configuration for FSP Smart Configurator

Perform the same steps as in step 2 of section 4.1.1.

(3) Copy the Baremetal folder

Perform the same steps as in step 3 of section 4.1.1.

(4) Copy the Patch folder

- Copy and overwrite the files in the Patch folder into the corresponding paths for both master project and slave project as shown below.

Name file	Path to folder
redefinition_memory_regions_iar.h	script/
fsp_ram_execution.icf	script/

(5) Modify the buildinfo.ipcf file

After clicking the Generate Project Content button in the FSP Smart Configurator, please move the files listed below from the "Component" group to another group.

```
<path>rzn/linaro/libmetal/lib/device.c</path>
<path>rzn/linaro/libmetal/lib/dma.c</path>
<path>rzn/linaro/libmetal/lib/init.c</path>
<path>rzn/linaro/libmetal/lib/io.c</path>
<path>rzn/linaro/libmetal/lib/log.c</path>
<path>rzn/linaro/libmetal/lib/shmem.c</path>
```

(6) Build combination

(1) Configure the debugger for project by clicking Options » General Options. In the Target tab, select the Renesas R9A09G087M44_R52_0/ Renesas R9A09G087M44_R52_1/ Renesas R9A09G087M44_A55 device.

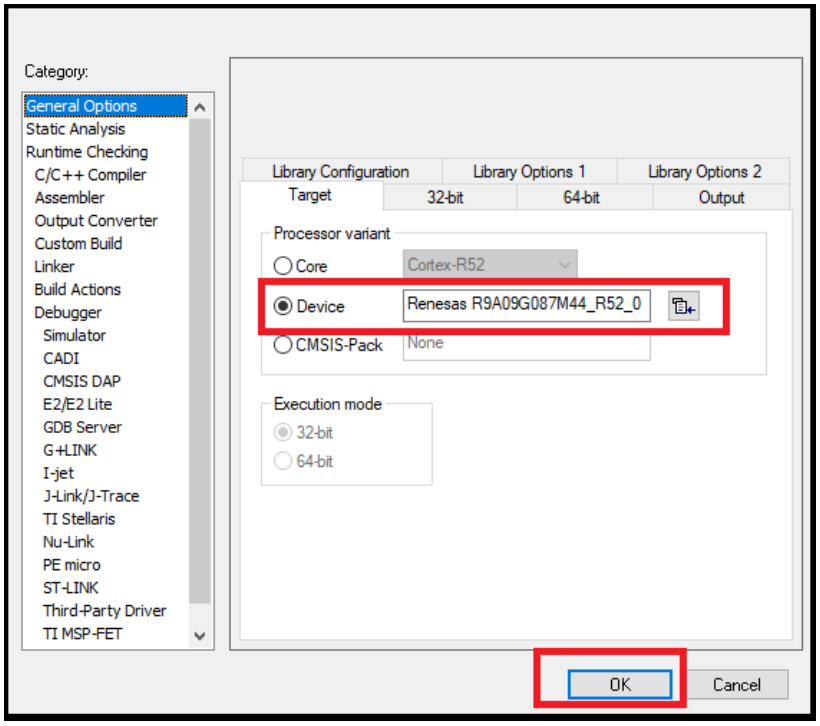


Figure 4-31. Select of device

(2) In the **Build Actions** section, delete Pre-compile to prevent the buildinfo.ipcf file from automatically updating.

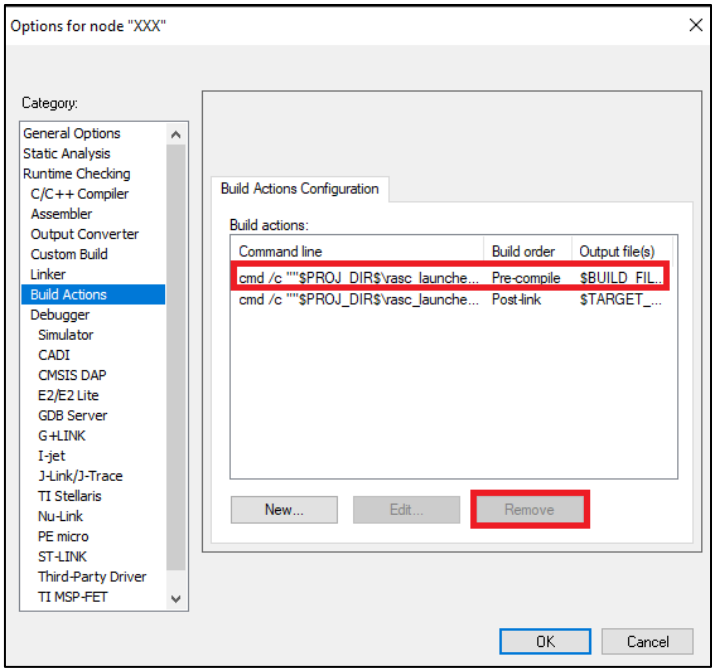


Figure 4-32. Select of device

(3) In the **Debugger** section, under the **Setup** tab, select the **I-Jet** driver.

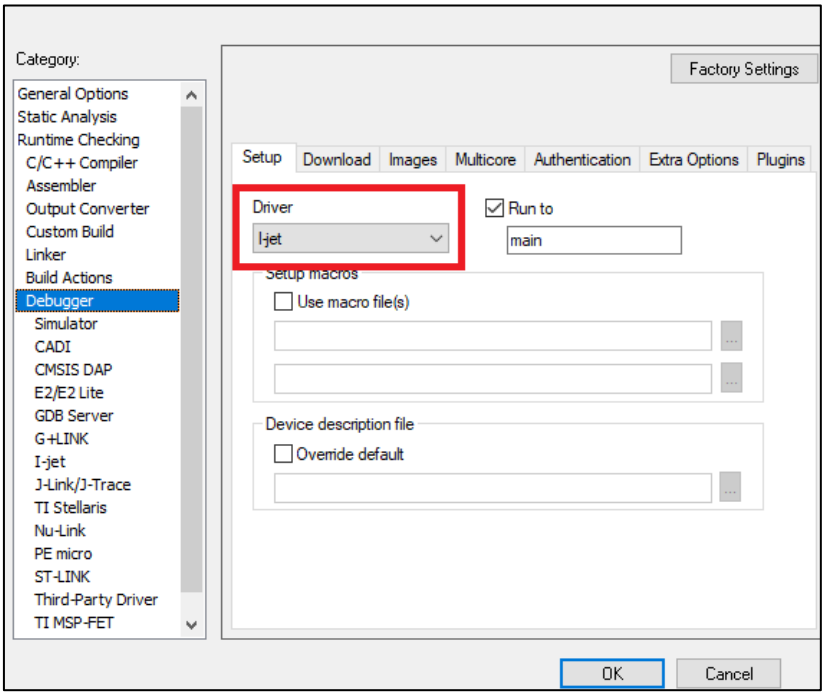


Figure 4-33. Config for debug

Note: Click Debugger > Setup, check Use macro file(s) and add "$PROJ_DIR$\script\initialization_TCM.mac" for CR52_1 core.

(4) In the **I-Jet** section, under the **Setup** tab, select **Hardware (default)** for primary core and **Software** for secondary core.

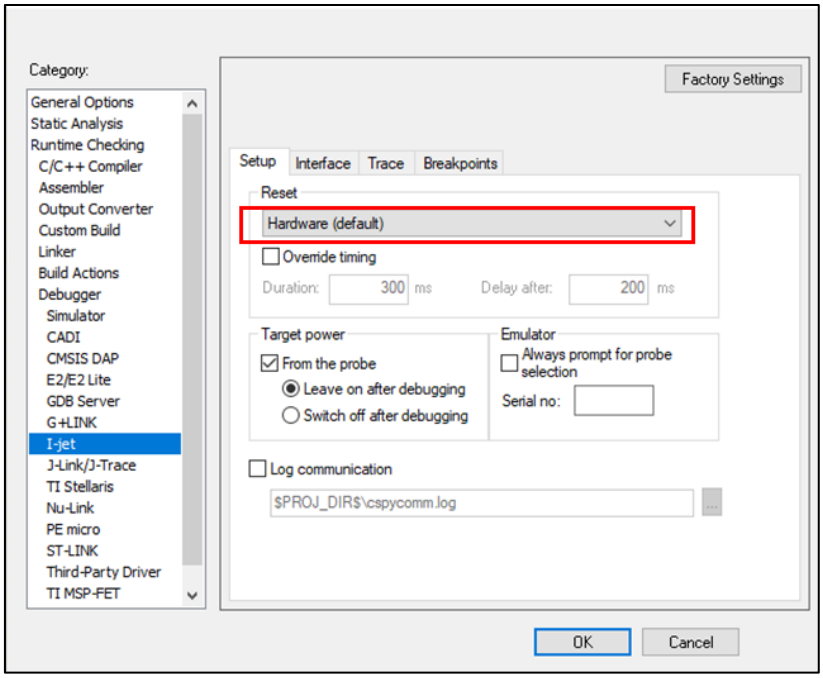


Figure 4-34. Config for debug

(5) If the project is the primary core, in the Debugger section, under the Multicore tab, select Simple for Asymmetric Multicore if the combination includes 2 projects, and Advanced if the combination includes more than 2 projects.

- Partner workspace: $PROJ_DIR$\..\[*secondary_core_project*]\[*secondary_core_project*].eww
- Partner project: [*secondary_core_project_name*]
- Partner configuration: Debug
- *multicore_setup.xml* for the combination of 3 projects.

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

```
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>false</attachToRunningTarget>
</partner>
<partner>
<name>Partner2</name>
<workspace>$PROJ_DIR$\..\[The_name_of_the_third_project]\[The_name_of_the_third_project].eww
</workspace>
<project>[The_name_of_the_third_project]</project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>false</attachToRunningTarget>
</partner>
</sessionSetup>
```

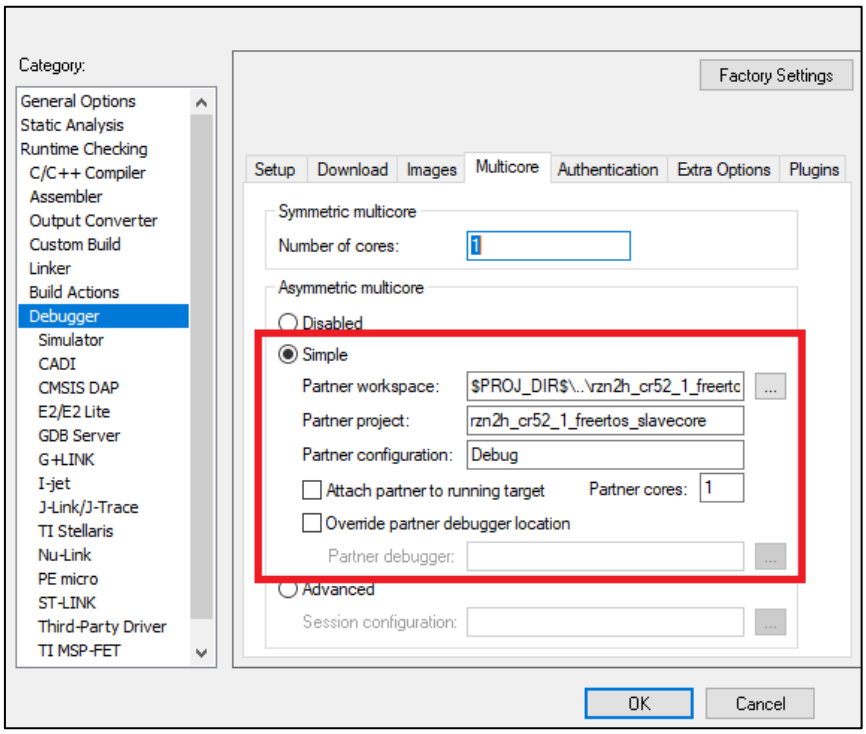


Figure 4-35. Config for debug

(6) If the project is the secondary core, in the Debugger section, under the Multicore tab, select Disabled for Asymmetric Multicore.

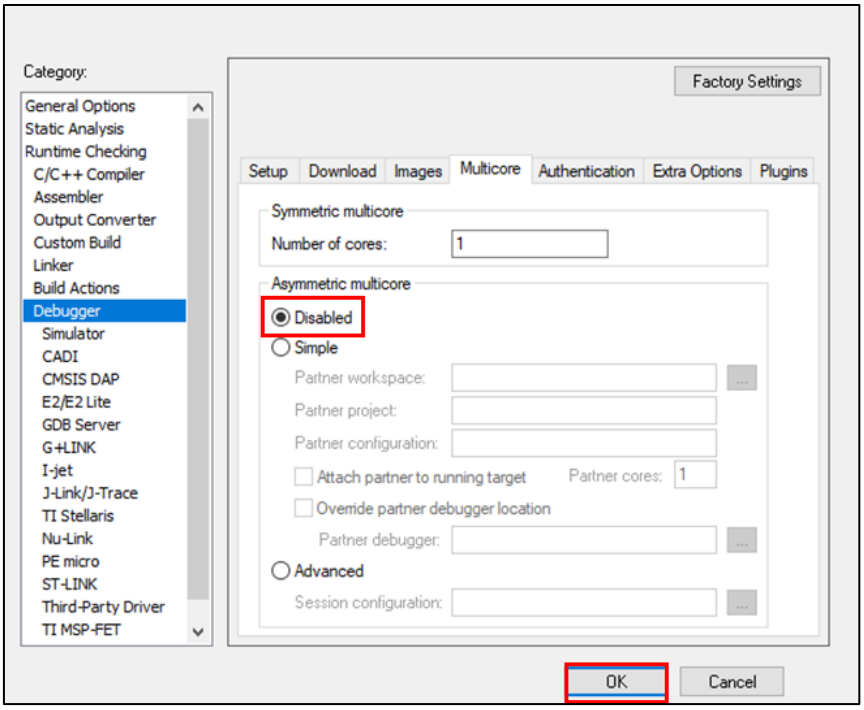


Figure 4-36. Config for debug

(5) Build and Debug Combination

Refer to step 5 and 6 of section 3.4 of this document.

4.2.2. FreeRTOS project on IAR EW for Arm

Follow the same steps as in section "***4.2.1 Baremetal project on IAR EW for Arm***" with the following notes:

(1) In the RTOS Selection window, select "*FreeRTOS (<FSP version>)*" for RTOS Selection

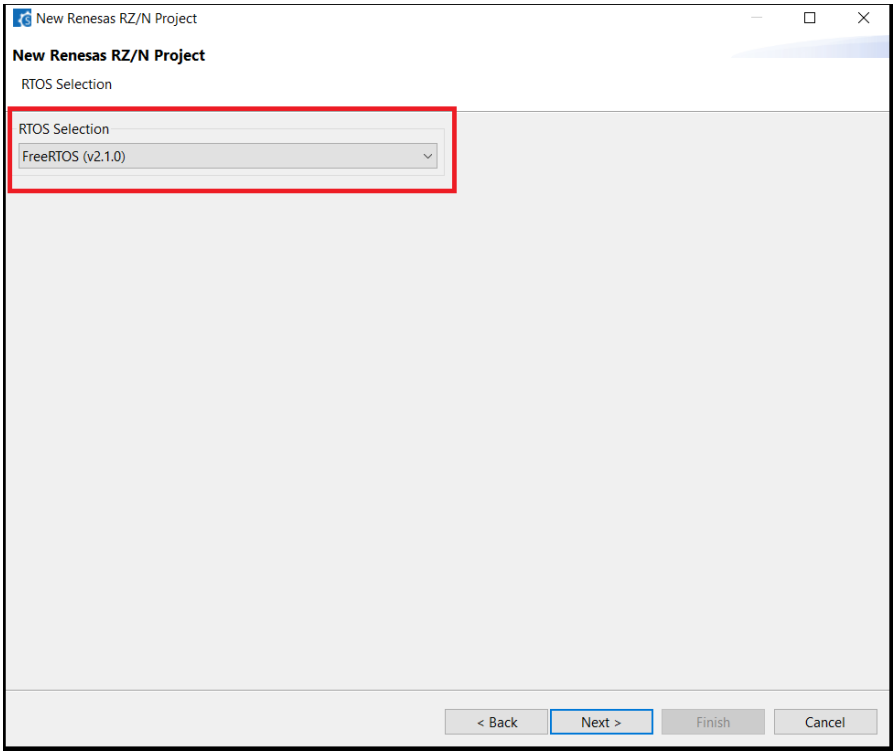


Figure 4-37. RTOS configuration

(2) In the Project Template Selection window, select "*FreeRTOS – Minimal – Static Allocation*"

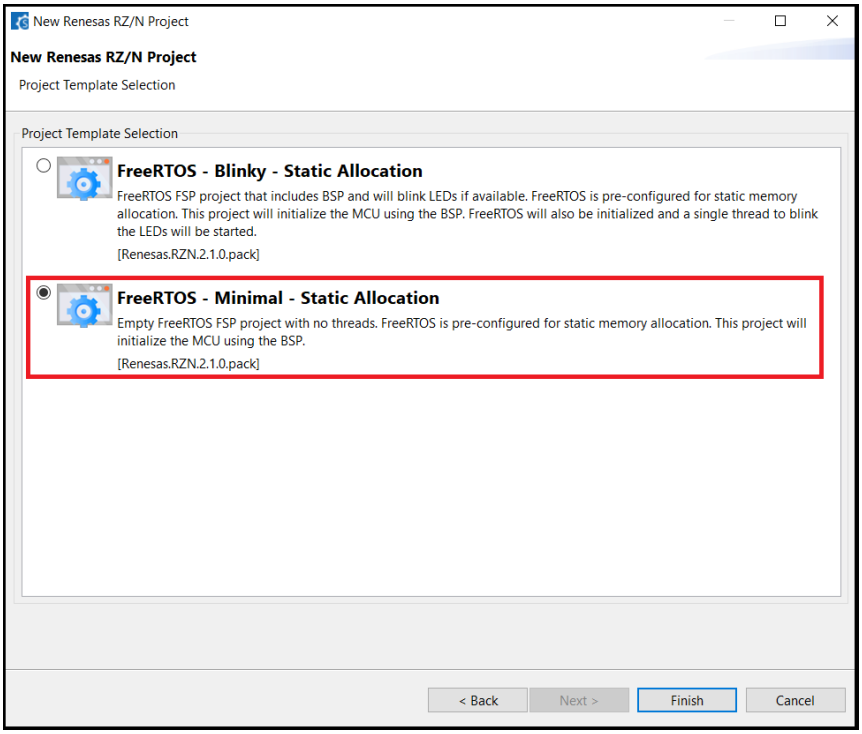


Figure 4-38. Project Template configuration

(3) Add a New Thread for FreeRTOS

In the Stacks tab, under the Threads section, click New Thread.

(4) Copy the FreeRTOS folder

- For the master core project, copy the files in the “*FreeRTOS*” folder from "*Lib_mastercore*" into the “*src*“ directory of the project.

- For the slave core project, copy the files in the “*FreeRTOS*” folder from "*Lib_slavecore*" into the “*src*” directory of the project.

Note 1: In a combination, only one core project is allowed to have FreeRTOS.

5. Reference Documents

- R01AN6434: RZ/N2H Getting Started with Flexible Software Package
- R01UH1039: RZ/N2H User's Manual: Hardware
- R01AN7633: RZ/N Multi-OS Package Application Note

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.23.24	All	• First edition issued.
1.20	May.30.25	6, 8 11	• Supported 5 additional combinations: rzn2h_cr52_0_rpmsg_linux_freertos_demo rzn2h_cr52_1_rpmsg_linux_freertos_demo rzn2h_ca55_1_rpmsg_linux_freertos_demo rzn2h_ca55_2_rpmsg_linux_freertos_demo rzn2h_ca55_3_rpmsg_linux_freertos_demo • Updated Linux BSP Package 1.0.2
1.30	Oct.31.25	6, 9 14	• Supported 3 additional combinations: rzn2h_ca55_1_rpmsg_linux_ baremetal _demo rzn2h_ca55_2_rpmsg_linux_ baremetal _demo rzn2h_ca55_3_rpmsg_linux_ baremetal _demo • Updated FSP Package v3.0.0
3.00	Jan.30.26	-	• Update FSP Package v3.1.0 • Update Linux Verify Package v5.0.0

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 reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

・Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere.
All rights reserved.
・Ethernet is a registered trademark of Fuji Xerox Co. Ltd.
・IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc
・Additionally all product names and service names in this document are a trademark or a registered trademark which belongs to the respective owners. a trademark or a registered trademark which belongs to the respective owners.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

© 2026 Renesas Electronics Corporation. All rights reserved.