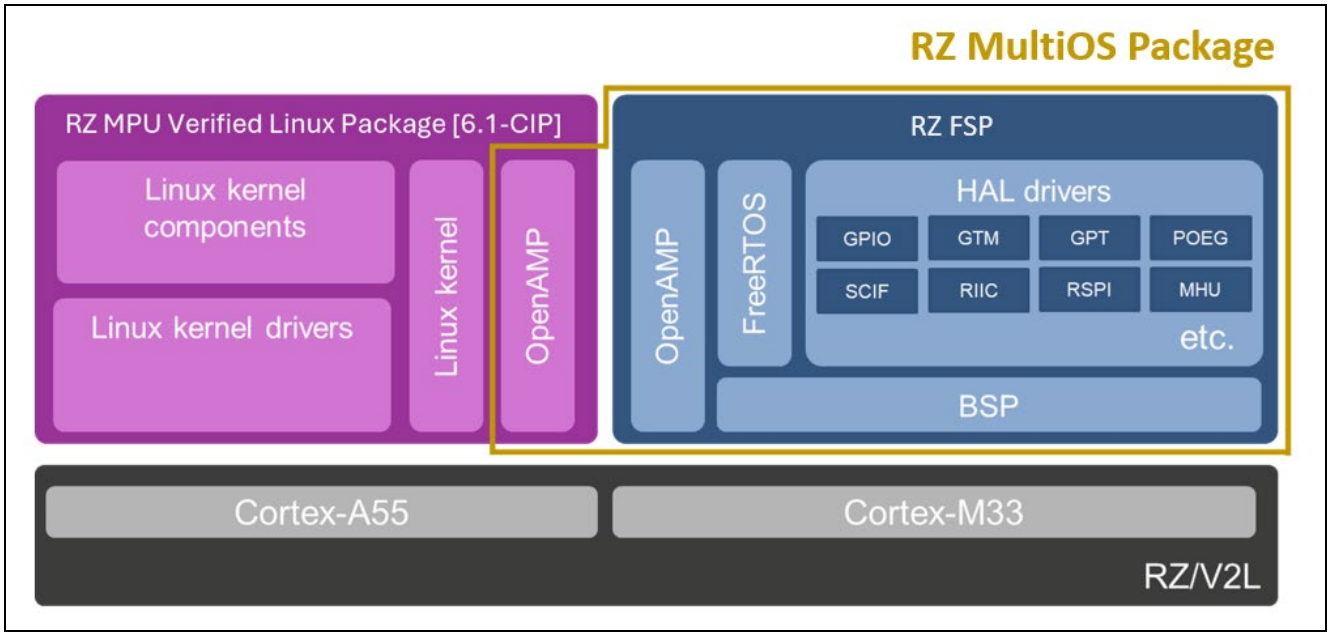
RZ/V2L

Quick Start Guide for RZ Multi-OS Package

Introduction

This document outlines the procedure for integrating the RZ Multi-OS Package into the RZ/V Verified Linux Package (referred to as VLP) for the RZ/V2L. By integrating the Multi-OS Package, users can efficiently establish a Multi-OS environment wherein Linux operates on the Cortex®-A55 and FreeRTOS/BareMetal runs on the Cortex-M33, with support for Inter-Processor Communication between these CPU cores.

This package requires the RZ Flexible Software Package (FSP) for an RTOS/BareMetal environment. The figure below illustrates the software stack for integrating the RZ Multi-OS Package with the RZ/V2L:



Here are brief descriptions of each component included in RZ Multi-OS Package:

- RZ FSP
  This software package consists of production ready peripheral drivers, FreeRTOS and portable middleware stacks and best in-case HAL drivers with low memory footprint.
- OpenAMP
  The framework includes the software components required for Asymmetric Multiprocessing (AMP) systems, such as Inter-Processor Communication.

Target Device

RZ/V2L

Contents

1. Specifications

Table 1-1 lists the on-chip peripheral modules to be used in this package.

Table 1-1 Peripheral modules to be used in this package

Peripheral module	Usage
Message Handling Unit (MHU)	Configure Inter-Processor Interrupt.
Serial Communications Interface with FIFO (SCIFA)	Perform standard serial communications sending and receiving console messages.
Interrupt controller (INTC)	Handle the following types of interrupts as shown below for example: • Processors should receive interrupts during buffered serial communications. • MHU module fires Inter-Processor Interrupt.
General Purpose Input Output (GPIO)	Configure I/O lines used by serial communications.
General Timer (GTM)	Configure the tick for FreeRTOS.

2. Verified Operation Conditions

Table 2-1 shows the verified operation conditions.

Table 2-1 Verified Operation Conditions

Item	Contents
Integrated Development Environment	e^2 studio 2025-12 or later
Toolchain	GNU Arm Toolchain 13.3.Rel1 AArch32 bare-metal target (arm-none-eabi)
Dependent Software	• RZ Flexible Software Package (FSP) v4.0.0 • RZ/V Verified Linux Package (VLP) v3.0.7-update4 • RZ MPU Verified Linux Package (VLP) v4.0.1

3. Sample Program Setup

3.1 Flexible Software Package Setup

Multi-OS Package expects RZ/V Flexible Software Package (FSP) to be installed in advance. For details on the installation, please refer to Getting Started with RZ/V Flexible Software Package.

3.2 Integration of OpenAMP related stuff

3.2.1 Using RZ MPU VLP v4.0.1

This section describes how to integrate OpenAMP related stuff to RZ MPU Verified Linux Package [6.1-CIP] (hereinafter referred to as VLP). The steps are based on **Linux Start-up Guide** included in **RZ VLP v4.0.1**.

1. Follow the procedure stated from the beginning of **2.2 Building Images** to **(4) Add layers** of **Linux Start-up Guide**.
2. Download Multi-OS Package (r01an8260ej0400-rz-multi-os-pkg.zip) to a working directory and run the commands stated below:

```
$ cd ~/rzv_vlp_<pkg ver>
$ unzip <Multi-OS Dir>/r01an8260ej0400-rz-multi-os-pkg.zip
$ tar zxvf r01an8260ej0400-rz-multi-os-pkg/meta-rz-features_multi-os_v4.0.0.tar.gz
```

Here, <pkg_ver> and <Multi-OS Dir> indicate the version number of VLP (e.g. v4.0.1) and the path to the directory where Multi-OS Package is placed respectively.

3. Add the layer for Multi-OS Package.

```
$ cd build
$ bash ../meta-rz-features/meta-rz-multi-os/add_meta_layer.sh
```

(Optional for remoteproc support)

4. Uncomment the following line in meta-rz-features/meta-rz-multi-os/meta-rzv2l-vlp4/conf/layer.conf for enabling remoteproc support:

```
MACHINE_FEATURES_append = " CM33_REMOTEPROC"
```

5. Start a build as described in **(6) Start a build** of **2.2 Building Images** as shown below:

```
$ MACHINE=<board> bitbake core-image-<target>
```

For details on the allowable value of <board> and <target>, please refer to **Linux Start-up Guide**.

3.2.2 Using RZ/V VLP v3.0.7

This section describes how to integrate OpenAMP related stuff to RZ/V Verified Linux Package [5.10-CIP] (hereinafter referred to as VLP). The steps are based on **SMARC EVK of RZ/V2L Linux Start-up Guide** (hereinafter referred to as **Linux Start-up Guide**) included in **RZ/V2L VLP v3.0.7-update3**.

1. Follow the procedure stated from the beginning of **2.2 Building Images** to **(4) Add layers** of **Linux Start-up Guide**.
2. Download Multi-OS Package (r01an8260ej0400-rz-multi-os-pkg.zip) to a working directory and run the commands stated below:

```
$ cd ~/rzv_vlp_<pkg ver>
$ unzip <Multi-OS Dir>/r01an8260ej0400-rz-multi-os-pkg.zip
$ tar zxvf r01an8260ej0400-rz-multi-os-pkg/meta-rz-features_multi-os_v4.0.0.tar.gz
```

Here, <pkg_ver> and <Multi-OS Dir> indicate the version number of VLP (e.g. v3.0.7) and the path to the directory where Multi-OS Package is placed respectively.

3. Add the layer for Multi-OS Package.

```
$ cd build
$ bash ../meta-rz-features/meta-rz-multi-os/add_meta_layer.sh
```

(Optional for remoteproc support)

4. Uncomment the following line in meta-rz-features/meta-rz-multi-os/meta-rzv2l-vlp3/conf/layer.conf for enabling remoteproc support:

```
MACHINE_FEATURES_append = " CM33_REMOTEPROC"
```

5. Start a build as described in **(6) Start a build** of **2.2 Building Images** as shown below:

```
$ MACHINE=<board> bitbake core-image-<target>
```

For details on the allowable value of <board> and <target>, please refer to **Linux Start-up Guide**.

3.3 Note for integration

On RZ/V Linux, the peripherals which are NOT enabled enter Module Standby Mode after Linux kernel is booted up. That means the peripherals used on CM33 side might stop working after that. To avoid such a situation, Multi-OS package incorporates the patch listed below:

- 0003-clk-renesas-rzv2l-Set-SCIF2-OSTM2.patch for RZ/V2L

This patch prevents SCIF channel 2 and GTM channel 2 used in RPMsg demo program from entering Module Standby Mode. If you have any other peripherals which you would like to stop entering Module Standby implicitly, please apply the patch as shown below:

```
drivers/clk/renesas/r9a07g044-cpg.c | 2 ++
 1 file changed, 2 insertions(+)

diff --git a/drivers/clk/renesas/r9a07g044-cpg.c
b/drivers/clk/renesas/r9a07g044-cpg.c
index e27caa075af7..9e904d2e8180 100644
--- a/drivers/clk/renesas/r9a07g044-cpg.c
+++ b/drivers/clk/renesas/r9a07g044-cpg.c
@@ -449,6 +449,# @@ static const unsigned int r9a07g044_crit_mod_clks[]
__initconst = {
        MOD_CLK_BASE + R9A07G044_IA55_PCLK,
        MOD_CLK_BASE + R9A07G044_IA55_CLK,
        MOD_CLK_BASE + R9A07G044_DMAC_ACLK,
+        MOD_CLK_BASE + R9A07G044_SCIF2_CLK_PCK,
+        MOD_CLK_BASE + R9A07G044_OSTM2_PCLK,
+        MOD_CLK_BASE + R9A07G044_xxxx,
 };
```

With respect to the allowable value for **xxxx**, please refer to the link below:

- https://github.com/renesas-rz/rz_linux-cip/blob/f12a03d8f2cf971734190c35aeaac98dde815b2d/drivers/clk/renesas/r9a07g044-cpg.c#L244-L411

3.4 Deployment of RZ/V2L VLP

With respect to the deployment of Linux kernel, device tree and root filesystem for RZ/V2L, please refer to SMARC EVK of RZ/V2L Linux Start-up Guide.

4. Sample Program Invocation on RZ/V2L SMARC EVK

4.1 Hardware Setup

1. Connect SEGGER J-Link to RZ/V2L SMARC EVK. For details, please refer to Getting Started with RZ/V Flexible Software Package.
2. Connect Pmod USBUART to the upper side of Pmod 1 of SMARC Carrier Board as shown below for securing the console for the program running on CM33.

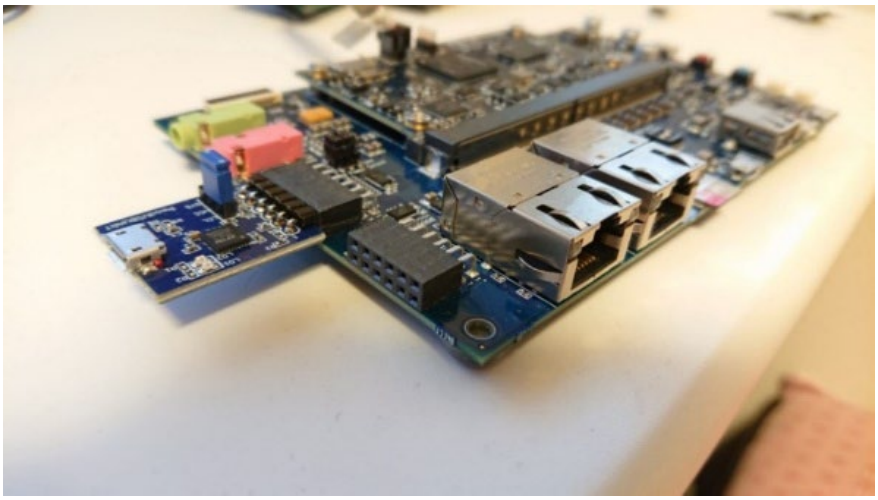


Figure 4-1. Connection between RZ/V2L SMARC EVK and Pmod USBUART

4.2 CM33 sample project Setup

This section describes how to set up CM33 sample project.

1. Extract **r01an8260ej0400-rz-multi-os-pkg.zip** on your development PC.
2. Extract **rzv2l_cm33_rpmsg_linux-rtos_example.zip** included in the unzipped r01an8260ej0400-rz-multi-os-pkg.zip.
3. Open e^2 studio 2025-01 and click **File** > **Import**.
4. Double-click **General** and select **Existing Projects into Workspace** as shown in Figure 4-2:

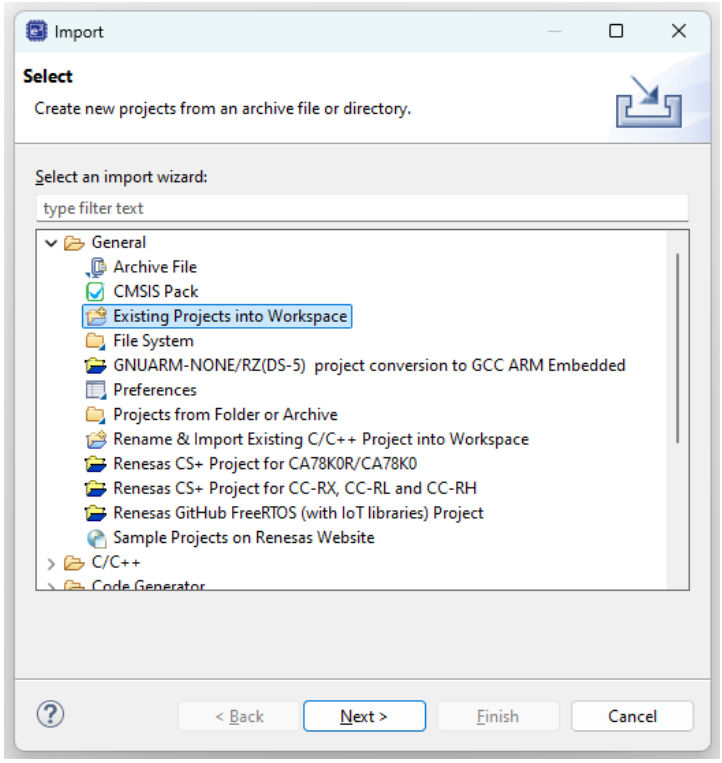


Figure 4-2. Import sample project (1)

5. Input the path to the directory **rzv2l_cm33_rpmsg_linux-rtos_example**, press **Enter** key and click **Finish** button.

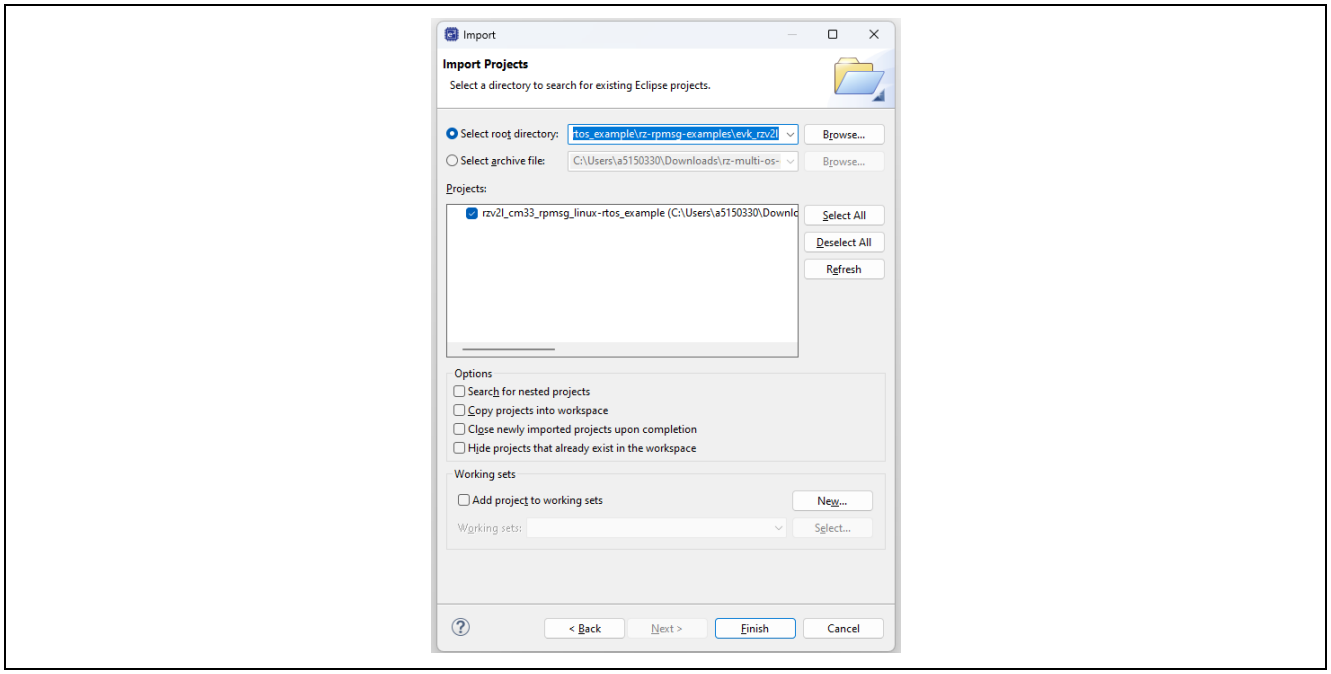


Figure 4-3. Import sample project (2)

(Optional for configuring RPMsg channel)

6. By default, RPMsg channel 0 is configured to be used on CM33. If you would like to use channel 1, please open the property of MainTask#0 on FSP Smart Configurator, specify 1 for Thread Context, and push **General Project Content** button. If **Generate Project Content** pop-up is shown, click **Proceed** to reflect the change to the source code.

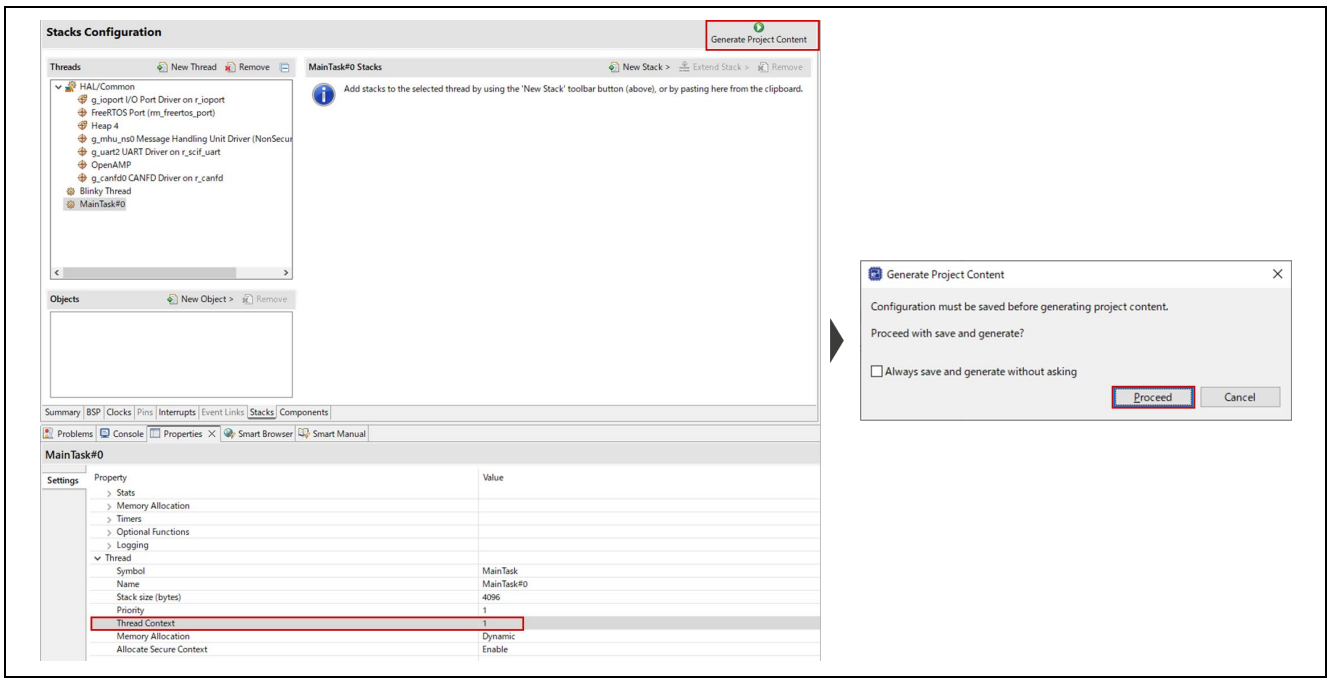


Figure 4-4. RPMsg Channel Setting

(Optional for remoteproc support)

7. Change the value for **ENABLE_REMOTEPROC** defined in **platform_info.h** from 0 to 1 as shown below:

```
#define ENABLE_REMOTEPROC (1U)
```

8. Build the project from **Choose Project** > **Build Project**.
9. The following files should be generated in **Debug** and/or **Release** directory in accordance with the active Build Configuration if there is no build failures.
   - rzv2l_cm33_rpmsg_linux-rtos_example.elf
   - rzv2l_cm33_rpmsg_linux-rtos_example_non_secure_code.bin
   - rzv2l_cm33_rpmsg_linux-rtos_example_non_secure_vector.bin
   - rzv2l_cm33_rpmsg_linux-rtos_example_secure_code.bin
   - rzv2l_cm33_rpmsg_linux-rtos_example_secure_vector.bin

4.3 CM33 sample program invocation

4.3.1 CM33 sample program invocation with SEGGER J-Link

You need to follow the steps stated below to invoke CM33 sample program with SEGGER J-Link:

1. Click the debug button indicated by a red arrow as shown below:

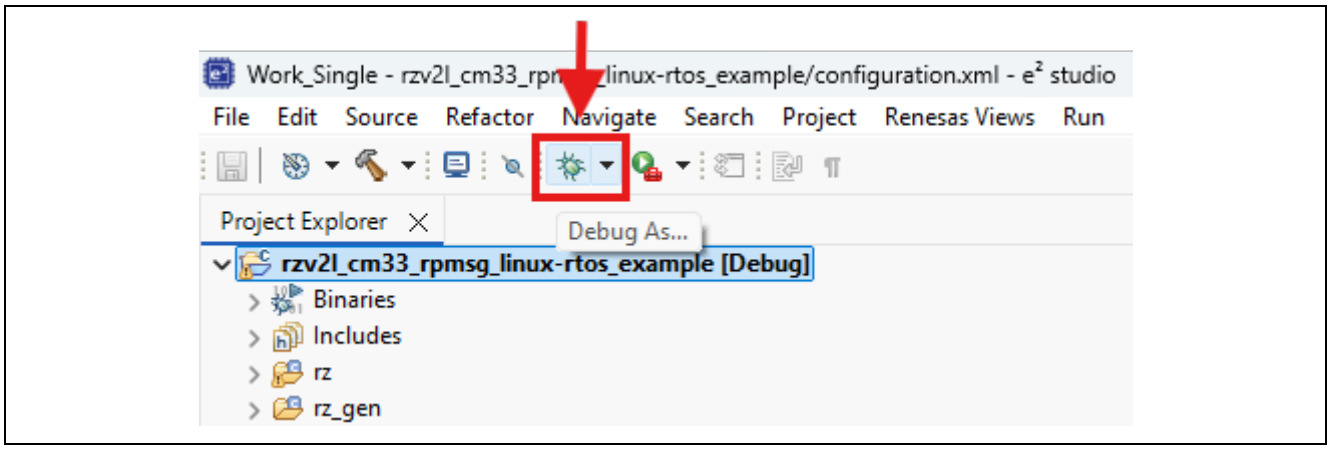


Figure 4-5. Selection of Debug Configuration

2. Choose **rzv2l_cm33_rpmsg_linux-rtos_example Debug_Flat** or **rzv2l_cm33_rpmsg_linux-rtos_example Release_Flat** from the drop-down list indicated in Figure 4-6.

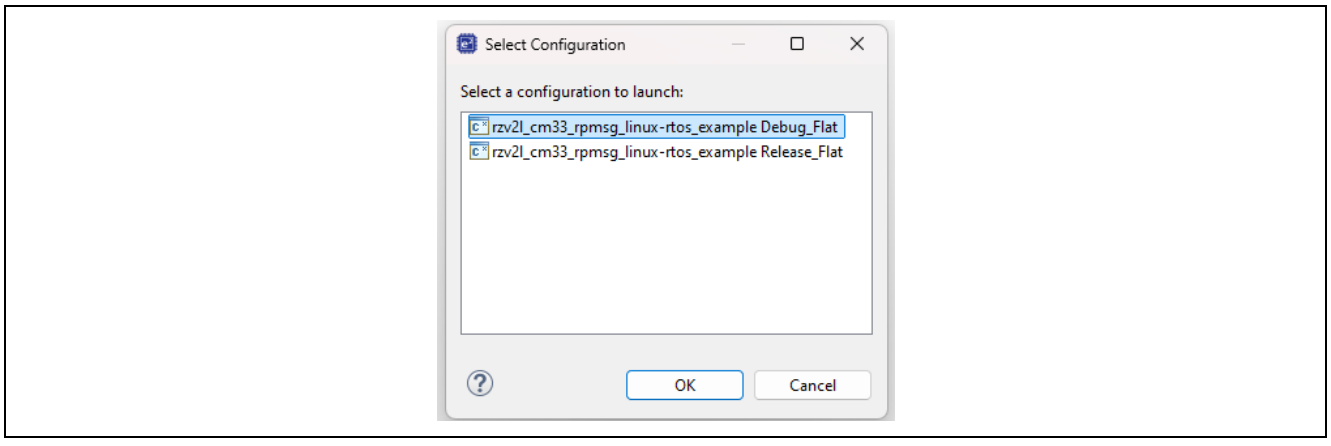


Figure 4-6. Debug Function Launch

If **Confirm Perspective Switch** window below appears, press **Switch** to go ahead.

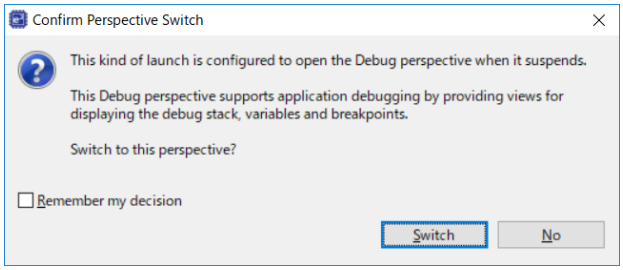


Figure 4-7. Confirm Perspective Switch window

3. When Debug Perspective is opened, Program Counter (PC) should be located at the top of Warm_Reset_S function. Then, you need to press the button indicated by a red arrow in Figure 4-8.

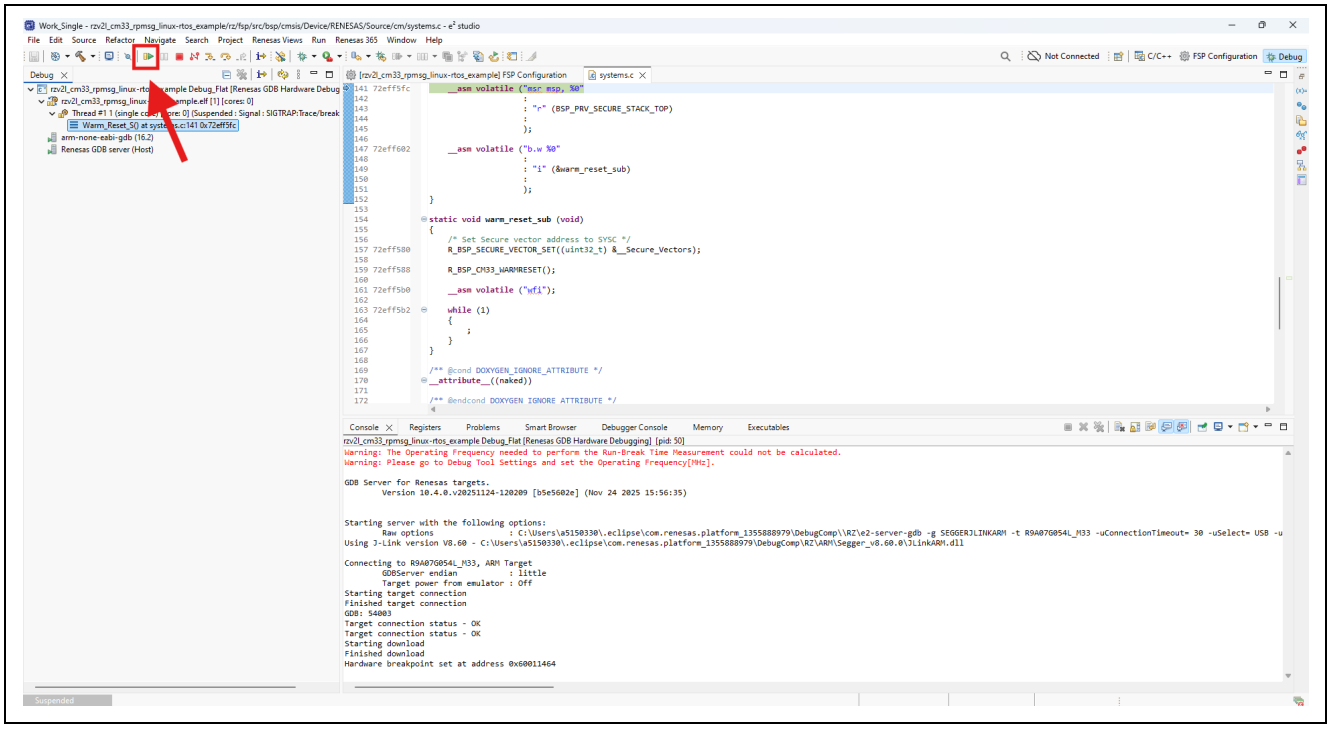


Figure 4-8. How to start to debug sample project (1)

4. Program stops at the top of **main** function. So, please click the same button as the previous step.

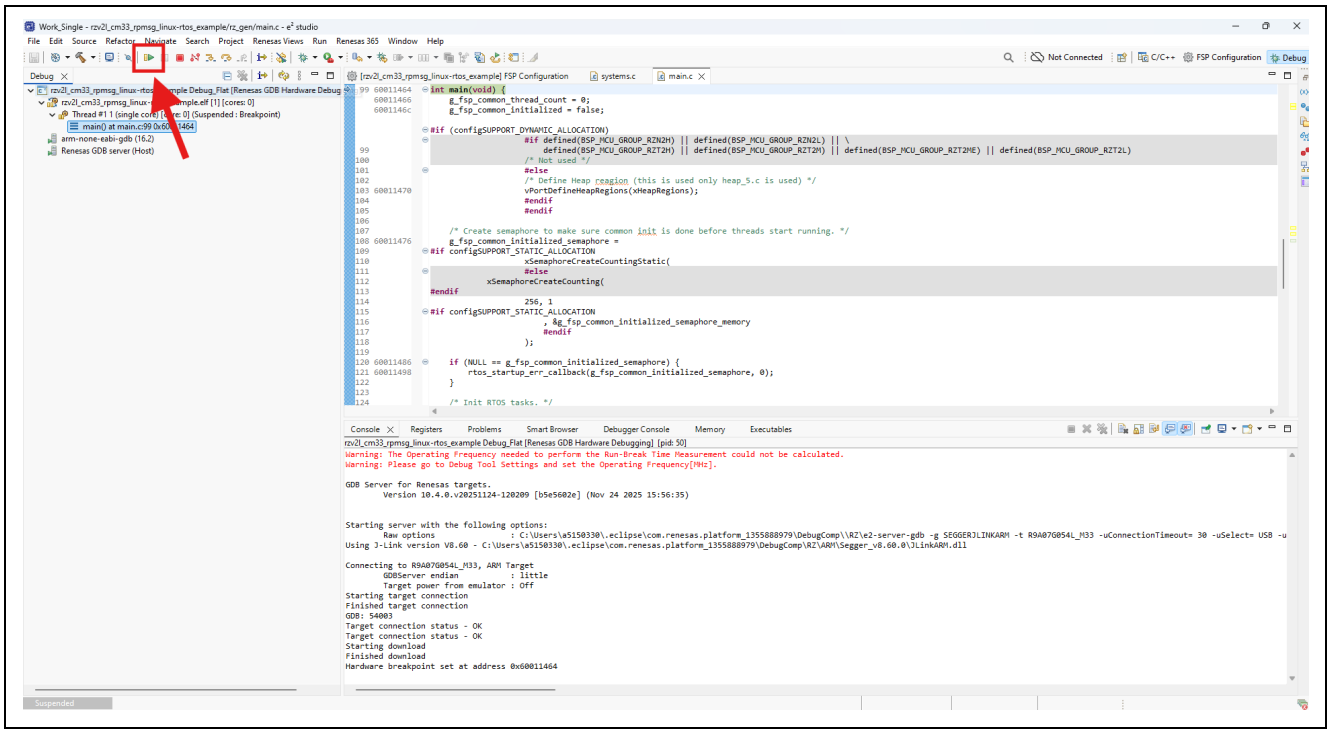


Figure 4-9. How to start to debug sample project (2)

5. Now that CM33 sample project has started, the following message is shown on the console associated with Pmod USBUART:

```
Successfully probed IPI device
Successfully open uio device: 42F00000.rsctbl.
Successfully added memory device 42F00000.rsctbl.
Successfully open uio device: 43000000.vring-ctl0.
Successfully added memory device 43000000.vring-ctl0.
Successfully open uio device: 43200000.vring-shm0.
Successfully added memory device 43200000.vring-shm0.
Initialize remoteproc successfully.
creating remoteproc virtio
initializing rpmsg vdev
```

Please note that CM33 sample program is waiting for the establishment of RPMsg channel between CM33 and CA55.

4.3.2 CM33 sample program invocation with u-boot

You can invoke CM33 sample program from u-boot by following the procedure described below:

1. Copy the binary files generated at 8 of section 4.2 to microSD card.
2. Insert the microSD card into CN10 of SMARC Carrier Board.
3. Thun on SMARC EVK by pressing the Power button for a few seconds.
4. Hit any key to stop autoboot within 3 seconds after the following message is shown in the console connected to CN14 of SMRAC Carrier Board:

```
U-Boot 2021.10 (Dec 15 2023 - 06:47:44 +0000)

CPU:   Renesas Electronics CPU rev 1.0
Model: smarc-rzv2l
DRAM:  1.9 GiB
WDT:   watchdog@0000000012800800
WDT:   Started with servicing (60s timeout)
MMC:   sd@11c00000: 0, sd@11c10000: 1
Loading Environment from MMC... OK
In:    serial@1004b800
Out:   serial@1004b800
Err:   serial@1004b800
U-boot WDT started!
Net:   eth0: ethernet@11c20000
Hit any key to stop autoboot: 2
=>
```

5. Load the binary files you copied at step1 from microSD card to RAM by executing the commands stated below on the console. Here, N stands for the partition number in which you stored the binary files.

```
dcache off
mmc dev 1
fatload mmc 1:N 0x0001FF80 rzv2l_cm33_rpmsg_linux-rtos_example_secure_vector.bin
fatload mmc 1:N 0x42EFF440 rzv2l_cm33_rpmsg_linux-rtos_example_secure_code.bin
fatload mmc 1:N 0x00010000 rzv2l_cm33_rpmsg_linux-
rtos_example_non_secure_vector.bin
fatload mmc 1:N 0x40010000 rzv2l_cm33_rpmsg_linux-rtos_example_non_secure_code.bin
cm33 start_debug 0x1001FF80 0x00010000
dcache on
```

6. CM33 sample program is now started.

Note:
In Step 5, start_debug is a boot method that allows a debugger to connect to the Cortex-M33.
If you do not want to allow debugger connection, please replace start_debug with start_normal when executing.
Note that there is no difference in execution speed regardless of which command is used.

4.3.3 CM33 sample program invocation with remoteproc

Here is the procedure for invoking CM33 sample program with remoteproc:

1. Booting up Linux by following **5. Booting and Running Linux** of **SMARC EVK of RZ/V2L Linux Start-up Guide**.
2. Invoke the command stated below to specify the sample program to be loaded:

```
root@smarc-rzv2l:~# echo rzv2l_cm33_rpmsg_linux-rtos_example.elf >
/sys/class/remoteproc/remoteproc0/firmware
```

3. Kick CM33 by invoking the command below:

```
root@smarc-rzv2l:~# echo start > /sys/class/remoteproc/remoteproc0/state
```

If CM33 sample program starts to work successfully, the following message should be shown on Linux console:

```
root@smarc-rzv2l:~# echo start > /sys/class/remoteproc/remoteproc0/state
[ 737.289773] remoteprocremoteproc0: powering up cm33
[ 737.348226] remoteprocremoteproc0: Booting fwimage rzv2l_cm33_rpmsg_linux-rtos_example.elf, size
1347660
[ 737.356732] remoteprocremoteproc0: unsupported resource 4
[ 737.366255] remoteproc0#vdev0buffer: assigned reserved memory node vdev0buffer@0x43200000
[ 737.374784] remoteproc0#vdev0buffer: registered virtio0 (type 7)
[ 737.380989] remoteprocremoteproc0: remote processor cm33 is now up
```

4.4 CA55 sample program invocation

This section describes how to invoke CA55 sample program.

1. Boot up Linux by executing the following command on u-boot:

```
=> run bootcmd
```

2. Login as **root**.

```
smarc-rzv2l login: root
```

3. Run CA55 sample program by executing the following command:

```
root@smarc-rzv2l:~# rpmsg_sample_client
```

4. Then, you can see the following message on the console relative to CN14 of SMARC carrier board. Be sure that you invoke CM33 sample program in advance.

```
Successfully probed IPI device
metal: info:      metal_uio_dev_open: No IRQ for device 42f00000.rsctbl.
Successfully open uio device: 42f00000.rsctbl.
Successfully added memory device 42f00000.rsctbl.
metal: info:      metal_uio_dev_open: No IRQ for device 43000000.vring-ctl0.
Successfully open uio device: 43000000.vring-ctl0.
Successfully added memory device 43000000.vring-ctl0.
metal: info:      metal_uio_dev_open: No IRQ for device 43200000.vring-shm0.
Successfully open uio device: 43200000.vring-shm0.
Successfully added memory device 43200000.vring-shm0.
metal: info:      metal_uio_dev_open: No IRQ for device 43100000.vring-ctl1.
Successfully open uio device: 43100000.vring-ctl1.
Successfully added memory device 43100000.vring-ctl1.
metal: info:      metal_uio_dev_open: No IRQ for device 43500000.vring-shm1.
Successfully open uio device: 43500000.vring-shm1.
Successfully added memory device 43500000.vring-shm1.
metal: info:      metal_uio_dev_open: No IRQ for device 42f01000.mhu-shm.
Successfully open uio device: 42f01000.mhu-shm.
Successfully added memory device 42f01000.mhu-shm.
Initialize remoteproc successfully.
Initialize remoteproc successfully.

****************************************
*   rpmsg communication sample program   *
****************************************

1. communicate with RZ/V2L CM33 ch0
2. communicate with RZ/V2L CM33 ch1

e. exit

please input
>
```

5. Type 1 if RPMsg channel 0 is used on CM33 RPMsg sample program. Also, type 2 if RPMsg channel 1 is used on the sample program.

4.5 Overview of sample program behavior

The behavior of sample program is stated below:

1. Wait until a communication channel between CA55 and CM33 is established.
2. Once the communication channel is established, CA55 sample program starts to send the message to CM33 by incrementing its size from the minimum value 17 to the maximum value 488. When sending the message, the following log should be shown on Linux console:

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

3. When CM33 receives the message sent from CA55, the echo reply is sent back to CA55.
4. When CA55 receives the echo reply, the message below should be displayed in the console:

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

5. After the message which has 488 bytes sized payload is sent from CA55 to CM33 and CM33 sends back the echo reply, the message for terminating the communication channel is sent from CA55 to CM33.

Then, CA55 and CM33 sample programs output the following log messages to the corresponding consoles respectively when receiving the termination message.

- **On CA55 side:**

```
****************************************
*   rpmsg communication sample program   *
****************************************

1. communicate with RZ/V2L CM33 ch0
2. communicate with RZ/V2L CM33 ch1

e. exit

please input
>
```

If you would like to quit the application, please type **e**.

- **On CM33 side:**

```
De-initializating remoteproc
```

6. Finally, CM33 sample program 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
```

5. Remoteproc support

This chapter describes how to invoke the CM33 firmware with remoteproc on Linux.

5.1 Setup of CM33 related stuff

Here are the steps to develop your own project based on the projects that support remoteproc included in this package.

1. Import the following projects as base projects.

   — rzv2l_cm33_rpmsg_linux-rtos_example

2. Change the project name (if you need).
   In Project Explorer, right-click on the project name and select “Rename...”

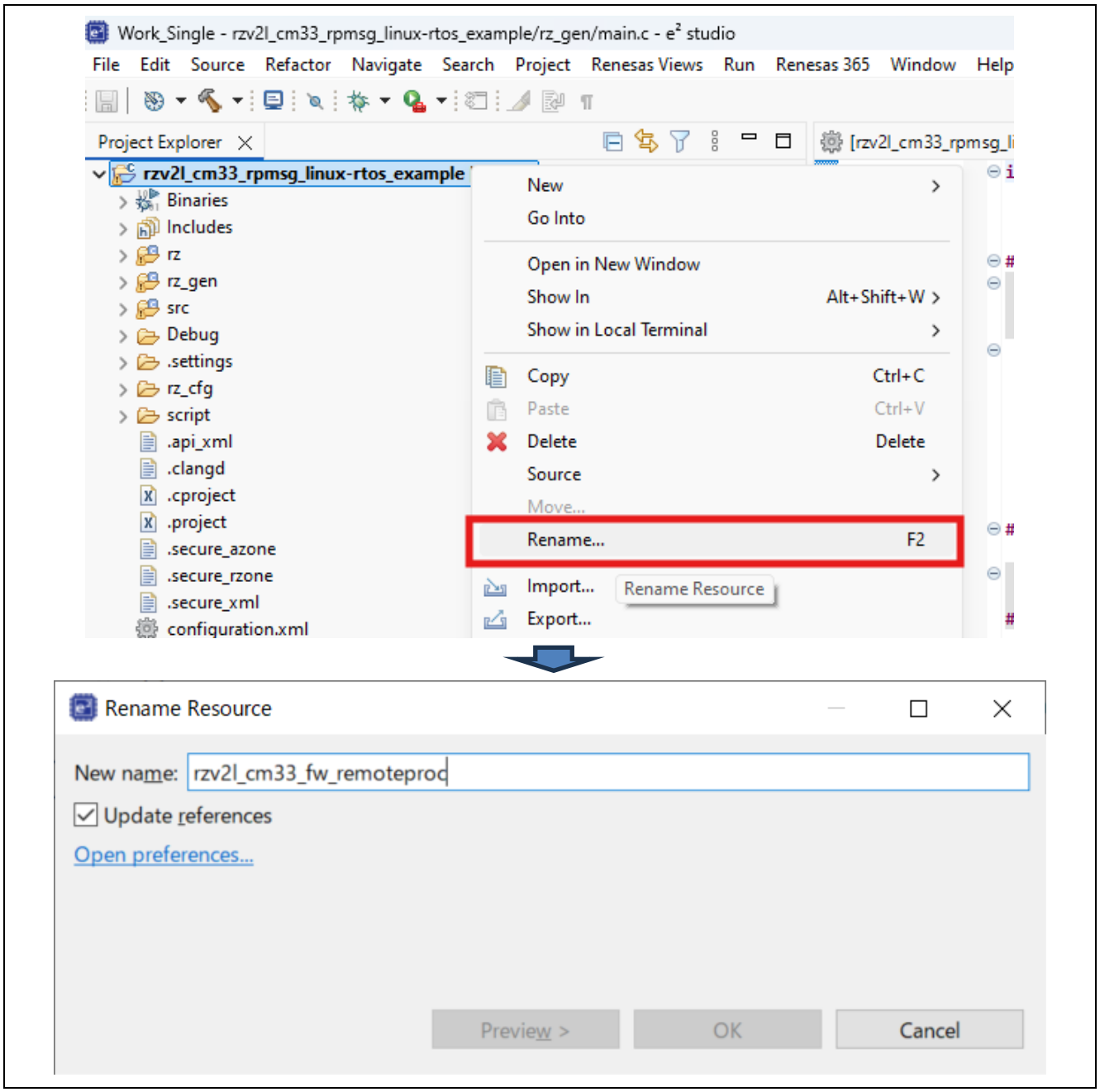


Figure 7-1. Rename project

3. Implement application code.
   Replace the contents of **MainTask_entry** function in **src/MainTask_entry.c** with the following and implement the application code.

```
/*
* Copyright (c) 2020 - 2024 Renesas Electronics Corporation and/or its affiliates
*
* SPDX-License-Identifier: BSD-3-Clause
*/

#include "MainTask.h"
/* MainTask#0 entry function */
/* pvParameters contains TaskHandle_t */
#include "openamp/open_amp.h"
#include "platform_info.h"
#include "rsc_table.h"

void MainTask_entry(void *pvParameters)
{
   unsigned long        proc_id = *((unsigned long*)pvParameters);
   unsigned long        rsc_id  = *((unsigned long*)pvParameters);
   struct rpmsg_device * rpdev;
   void                * platform;
   int ret;

   ret = init_system();
   if (ret)
   {
      LPERROR("Failed to init remoteproc device.\n");
      goto err1;
   }

   ret = platform_init(proc_id, rsc_id, &platform);
   if (ret)
   {
      LPERROR("Failed to create remoteproc device.\n");
      goto err1;
   }

   // Add your code here

err1:
   vTaskDelete(NULL);

}
```

Figure 7-2. Implement application code.

4. Enable remoteproc.
   Change the value of the **ENABLE_REMOTEPROC** macro in **src/platform_info.h** to **1**.

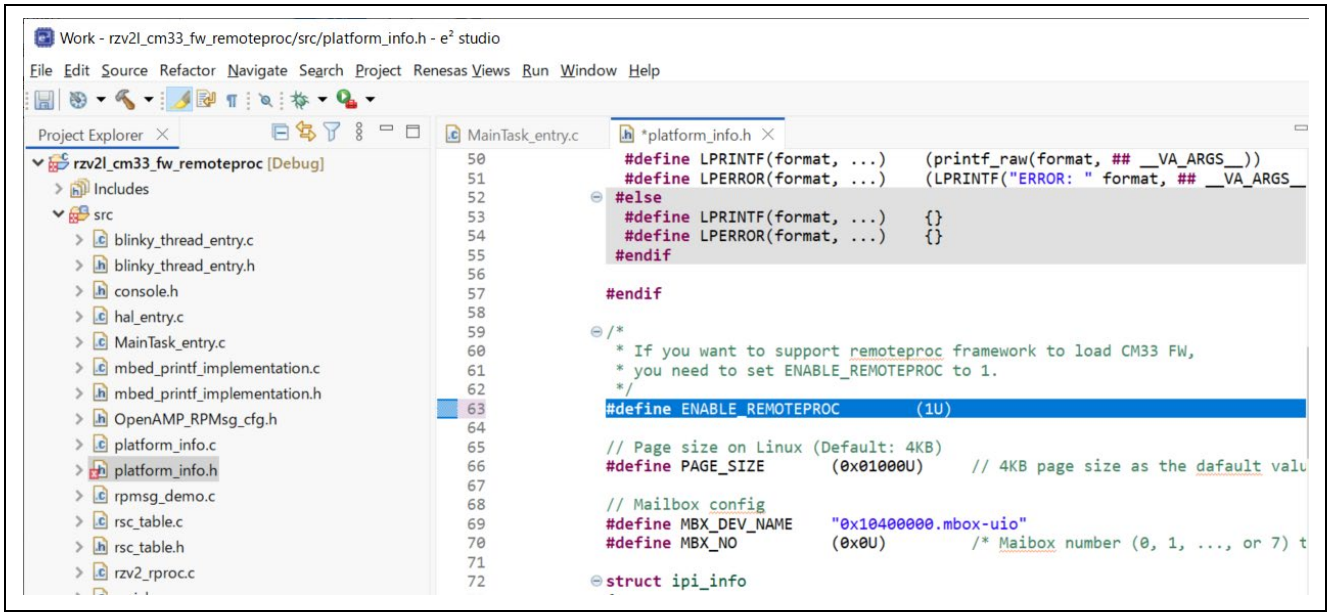


Figure 7-3. Enable remoteproc.

5. Build the project

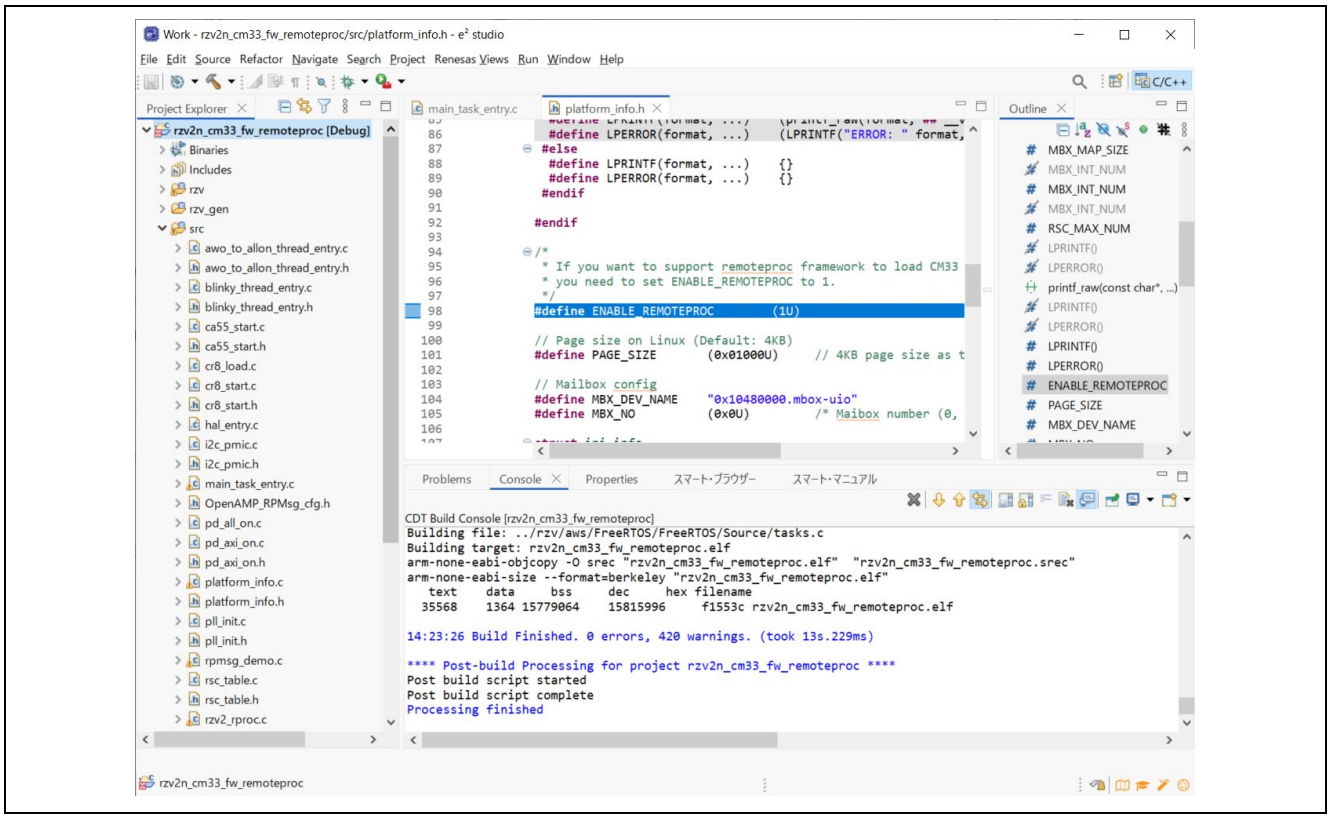


Figure 7-4. Build the project.

6. Copy the image to rootfs.
   Copy the elf file you created in the previous step to the **/lib/firmware** folder on the SD card that contains the Linux rootfs you created in **3.4 Deployment of RZ/V2L VLP**.

5.2 How to run CM33 firmware from remoteproc

Follow the steps in chapter **4.3.3 CM33 sample program invocation with remoteproc**.

However, change the echo argument in step 2 to match the file name of the file you copied in step 6 in the previous chapter.

6. Reference Documents

- R01AN6240: RZ/V2L, RZ/V2H, RZ/V2H Getting Started with Flexible Software Package
- R01US0617: SMARC EVK of RZ/V2L Linux Start-up Guide
- R01UH0964: RZ/G2L, RZ/V2L SMARC Module Board User's Manual
- R01UH0966: RZ SMARC Series Carrier Board User's Manual

Revision History

Rev.	Date	Description	
		Page	Summary
3.00	Mar.11.2025	-	Extract RZ/V2L related description from Release Note of RZ/V Multi-OS Package v2.1.0.
3.10	May.22.2025	-	Updated to align with RZ/V Multi-OS Package v3.1.0.
3.11	Jun.13.2025	-	Update Multi-OS Package version to 3.1.1.
3.20	Aug.29.2025	-	Updated to align with RZ/V Verified Linux Package v3.0.7-update3.
3.30	Jan.13.2026	-	Updated to align with RZ MPU Verified Linux Package v4.0.1.
4.00	Mar.31.2026	-	Update MultiOS Package version to v4.0.0.

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.

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/.