

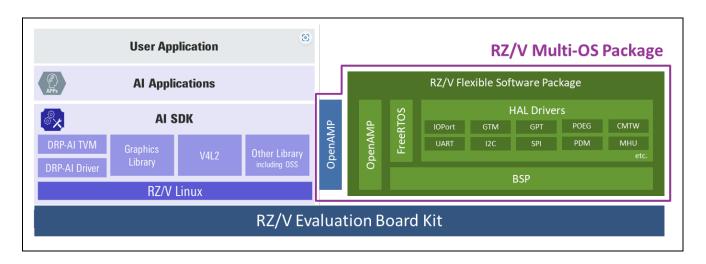
## RZ/V2N

## Quick Start Guide for RZ/V Multi-OS Package

#### Introduction

This document outlines the procedure for integrating the RZ/V Multi-OS Package into RZ/V2N AI SDK. 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/V Flexible Software Package (FSP) for an RTOS/BareMetal environment. The figure below illustrates the software stack of the RZ/V Multi-OS Package with the RZ/V2N:



Here are brief descriptions of each component related to RZ/V Multi-OS Package:

RZ/V 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/V2N



## Contents

1.	Specifications	3
2.	Verified Operation Conditions	3
3.	Sample Program Setup	3
3.1	Flexible Software Package Setup	3
3.2	Integration of OpenAMP related stuff	3
3.3	Note for Integration	4
3.4	Deployment of RZ/V2N AI SDK	5
4.	Sample Program Invocation on RZ/V2N	5
4.1	Hardware setup	5
4.2	CM33 Sample Program Setup	5
4.3	CM33 Sample Project Invocation for communicating with Linux	6
4.3.1	1 CM33 Sample Project Invocation using Segger J-Link	6
4.3.2	2 CM33 Sample Program Invocation from u-boot	8
4.3.3	3 CM33 Sample Program Invocation with remoteproc	9
4.4	CA55 Sample Program Invocation	9
4.5	Overview of Sample Program's behavior	10
5.	CA55 1.8GHz configuration support at CA55 cold boot mode	11
5.1	Setup of CA55 related stuff	11
5.2	Setup of CM33 related stuff	11
5.3	Deployment of Build Artifacts to RZ/V2N EVK	13
6.	CM33 cold boot support	16
6.1	Setup of CA55 related stuff	16
6.2	Deployment of CA55 Build Artifacts to RZ/V2N EVK	16
6.3	Setup of CM33 related stuff	17
7.	Reference Documents	20
Rev	rision History	21



#### 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)	<ul> <li>Handle the following types of interrupts as shown below for example:</li> <li>Processors should receive interrupts during buffered serial communications.</li> <li>MHU module fires Inter-Processor Interrupt.</li> </ul>
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.

Item	Contents
Integrated Development Environment	e <sup>2</sup> studio 2025-01
Toolchain	GNU Arm Toolchain 13.3.Rel1 AArch32 bare-metal target (arm-none-eabi)
Dependent Software	<ul> <li>RZ/V Flexible Software Package (FSP) v3.1.0</li> <li>RZ/V2N AI Software Development Kit (SDK) v5.00</li> </ul>

#### 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 <u>Getting Started with RZ/V Flexible Software Package</u>.

#### 3.2 Integration of OpenAMP related stuff

This section describes how to integrate OpenAMP related stuff into **RZ/V2N AI Software Development Kit** (hereinafter referred to as **RZ/V2N AI SDK**).

- 1. Carry out Step 1, Step 2 and 1-11 of Step 3 stated in How to build RZ/V2N AI SDK Source Code showcased at <u>AI Applications and AI SDK of RZ/V series</u>.
- 2. Download Multi-OS Package (r01an7254ej0311-rzv-multi-os-pkg.zip) to a working directory and run the commands stated below:

```
$ cd ${YOCTO WORK}
```

```
$ unzip <Multi-OS Dir>/r01an7254ej0311-rzv-multi-os-pkg.zip
```

\$ tar zxvf r01an7254ej0311-rzv-multi-os-pkg/meta-rz-features\_multi-os\_v3.1.1.tar.gz

Here, <Multi-OS Dir> stands for the path to the directory where Multi-OS Package is placed.

3. Add the layer for Multi-OS Package.

```
$ cd build
$ bitbake-layers add-layer ../meta-rz-features/meta-rz-multi-os/meta-rzv2n
```

#### (Optional for remoteproc support)



4. Apply the modification below to **meta-rz-features/meta-rz-multi-os/meta-rzv2n/recipeskernel/linux/linux-renesas\_5.10.bbappend** for enabling remoteproc support:

```
$ FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
$ ENABLE REMOTEPROC = "1"
```

#### (Optional for CM33 invocation from u-boot)

 Uncomment the following 2nd line stated in meta-rz-features/meta-rz-multi-os/metarzv2h/conf/layer.conf. For details on CM33 invocation from u-boot, please see 4.3.2.

```
#MACHINE_FEATURES_append = " RZV2H_CM33_BOOT"
MACHINE_FEATURES_append = " SRAM_REGION_ACCESS"
#MACHINE_FEATURES_append = " CM33_FIRMWARE_LOAD"
#MACHINE_FEATURES_append = " CA55_CPU_CLOCKUP"
```

6. Carry out **13 of Step 3** of **How to build RZ/V2N AI SDK Source Code** for building Linux kernel related stuff.

#### 3.3 Note for Integration

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 at that time. To avoid such a situation, Multi-OS Package incorporates the patch below:

• 0002-Set-OSTM-for-MCPU-and-RCPU.patch

This patch prevents GTM 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 update the patch as shown below:

```
diff --git a/drivers/clk/renesas/r9a09g056-cpg.c b/drivers/clk/renesas/r9a09g056-cpg.c
index 03a7be492353..cdb2c6bf659e 100644
--- a/drivers/clk/renesas/r9a09g056-cpg.c
+++ b/drivers/clk/renesas/r9a09g056-cpg.c
00 -886,6 +886,13 00 static const unsigned int r9a09g056 crit mod clks[] initconst = {
 MOD CLK BASE + R9A09G056 ACPU DMAC 1 ACLK,
 MOD CLK BASE + R9A09G056 RCPU DMAC 0 ACLK,
 MOD CLK BASE + R9A09G056 RCPU DMAC 1 ACLK,
+ MOD CLK BASE + R9A09G056 MCPU OSTMO PCLK,
+ MOD_CLK_BASE + R9A09G056_MCPU_OSTM1_PCLK,
+ MOD CLK BASE + R9A09G056 RCPU OSTM0 PCLK,
+ MOD CLK BASE + R9A09G056 RCPU OSTM1 PCLK,
+ MOD CLK BASE + R9A09G056 RCPU OSTM2 PCLK,
+ MOD_CLK_BASE + R9A09G056_RCPU_OSTM3_PCLK,
+ MOD CLK BASE + R9A09G056 **** PCLK,
 };
static struct clk mon r9a09g056 clk mon[] = {
```

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

 <u>https://github.com/renesas-rz/rz\_linux-</u> cip/blob/e619ed8241f998b0e7f6e70feab606272e27655a/drivers/clk/renesas/r9a09g056-cpg.c#L325-L770



#### 3.4 Deployment of RZ/V2N AI SDK

With respect to the deployment of Linux kernel, device tree and root filesystem for RZ/V2N, please refer to <u>https://renesas-rz.github.io/rzv\_ai\_sdk/5.00/</u>.

#### 4. Sample Program Invocation on RZ/V2N

#### 4.1 Hardware setup

Connect J-Link to RZ/V2N EVK. For details, please refer to <u>Getting Started with RZ/V Flexible Software</u> Package.

#### 4.2 CM33 Sample Program Setup

Here are the procedures for setting up the sample program running on CM33.

- 1. Extract r01an7254ej0311-rzv-multi-os-pkg.zip on your development PC.
- 2. Extract rzv2n\_cm33\_rpmsg\_linux-rtos\_demo.zip included in r01an7254ej0311-rzv-multi-os-pkg.zip.
- 3. Open e<sup>2</sup> studio 2025-01 and click **File > Import**.
- 4. Double-click General and select **Existing Projects into Workspace** as shown in Figure 4-1:

🖸 Import – 🗆 🗙	
Select Create new projects from an archive file or directory.	
Select an import wizard:	
type filter text	
Constant of the second se	
()     ()	

Figure 4-1. Import sample project (1)

5. Input the path to the directory where sample project you would like to import to **Select root directory**, press **Enter** key and click **Finish** button.

Figure 4-2. Import sample project (2)

(Optional)

6. By default, RPMsg channel 0 configured to be used on CM33. If you would like to change the channel, you need to open the property of **MainTask#0** on FSP Smart Configurator, specify the channel number



you would like to use for **Thread Control** and push **Generate Project Content** button. If **Generate Project Content** pop-up is shown, click **Proceed** to reflect the changes to source code.

Stacks Config	guration			Generate Project Content		
hreads	🕢 New Thread 🙀 Remove 📋	MainTask#0 Stacks	🔊 New Stack > 🚊 🗄	itend Stack > 🔬 Remove		
<ul> <li>FreeRTO</li> <li>Heap 4</li> <li>g_mhu_</li> <li>g_uart2</li> <li>OpenAM</li> </ul>	nt VO Port Driver on r_joport OS Port (m_freetos_port) not Message Handling Unit Driver (NonSecur UART Driver on r_scif_uart MP do CANFD Driver on r_canfd ead	Add stacks to the selected thread by	using the 'New Stack' toolbar button (above), or by pasti	ng here from the clipboard.		
					Generate Project Content	
ojects	🚯 New Object > 🔬 Remove				Configuration must be saved before generating proje	ct content.
					Proceed with save and generate?	
					Always save and generate without asking	
					E CONTRACTOR OF CONTRACTOR	
mmany BSD Close	ocks Pins Interrupts Event Links Stacks Com	anante			L	Proceed Cancel
	Console 🔲 Properties 🗙 🏟 Smart Browser	44 Smart Manual				
ainTask#0						
ettings Propert	rty		Value			
	Stats					
> N	Memory Allocation					
	Timers					
	Optional Functions					
	Logging					
✓ Three						
	Symbol		MainTask			
	Name		MainTask#0			
	Stack size (bytes)		4096			
	Priority		1			
	Thread Context Memory Allocation		1 Dynamic			
			Dynamic Enable			
A	Allocate Secure Context		Enable			

Figure 4-3. RPMsg Channel Setting

- 7. Build the project from Choose Project > Build Project.
- 8. If building project is successfully completed, build artifacts as listed below should be generated in **Debug** or **Release** directory of the project you imported in accordance with the active Build Configuration.
  - rzv2n\_cm33\_rpmsg\_linux-rtos\_demo.elf
  - rzv2n\_cm33\_rpmsg\_linux-rtos\_demo.bin

#### 4.3 CM33 Sample Project Invocation for communicating with Linux

#### 4.3.1 CM33 Sample Project Invocation using Segger J-Link

Carry out the procedure shown below for invoking CM33 sample project using J-Link:

1. Click Debug button as shown below:

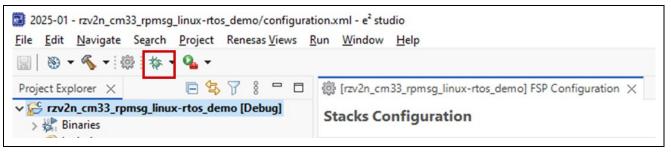


Figure 4-4. Selection of Debug Configuration

2. If the following **Select Configuration** window is shown, choose rzv2n\_cm33\_rpmsg\_linuxrtos\_demo\_Debug\_Flat or rzv2n\_cm33\_rpmsg\_linux-rtos\_demo\_Release\_Flat in accordance with the build configuration you are using and click **OK**.



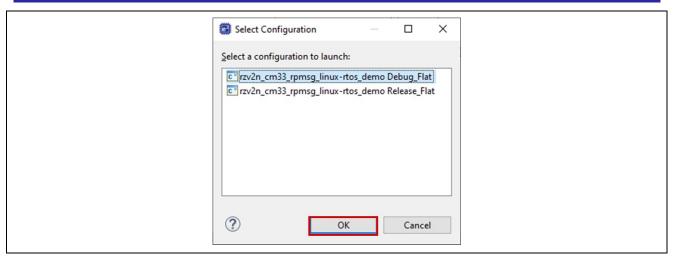


Figure 4-5. Debug Perspective Launch (1)

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

Confirm Perspective Switch	×
This kind of launch is configured to open the Debug perspective when it suspends.	
This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.	
Switch to this perspective?	
□ <u>R</u> emember my decision	
<u>Switch</u>	

Figure 4-6. Debug Perspective Launch (2)

3. When **Debug Perspective** is opened, Program Counter (PC) should be located at the top of **Entry\_Function\_S**. Then, press the button shown in Figure 4.7.

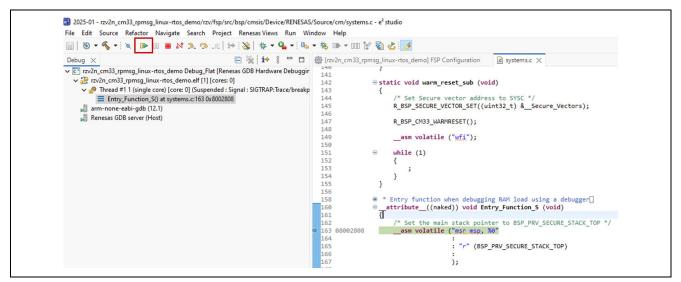


Figure 4-7. How to start to debug Sample Project (1)



4. PC should be stopped at the top of **main** function. Then, click the same button in the previous step to continue.

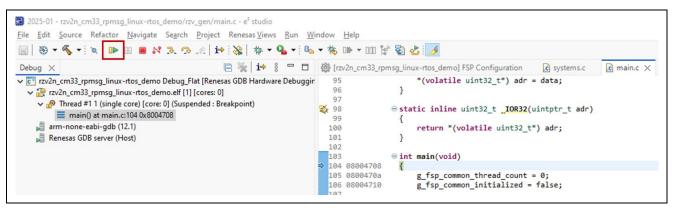


Figure 4-8. How to start to debug Sample Project (2)

5. CM33 Sample Project starts to work and wait for the launch of CA55 RPMsg Sample Program.

#### 4.3.2 CM33 Sample Program Invocation from u-boot

Here is the procedure for invoking CM33 Sample Program from u-boot:

- 1. Place rzv2n\_cm33\_rpmsg\_linux-rtos\_demo.bin together with Linux kernel and Device Tree Blob.
- 2. Insert SD card into SD0 of RZ/V2N EVK.
- Turn on RZ/V2N EVK. Then, you should see the following message on the console associated with CN12 of RZ/V2N EVK.

```
U-Boot 2021.10 (Nov 21 2023 - 11:34:15 +0000)
CPU: Renesas Electronics CPU rev 1.2
Model: Renesas EVK based on r9a09g056
DRAM: 7.9 GiB
MMC: mmc@15c00000: 0, mmc@15c10000: 1
Loading Environment from SPIFlash... SF: Detected mt25qu512a with page size
256 Bytes, erase size 64 KiB, total 64 MiB
OK
In: serial@11c01400
Out: serial@11c01400
Err: serial@11c01400
Net:
Error: ethernet@15c30000 address not set.
No ethernet found.
Hit any key to stop autoboot: 0
=>
```

4. Hit any key within 3 sec. to stop autoboot.5. Carry out the following setup on u-boot to kick CM33.

```
=> setenv cm33start 'dcache off; mw.l 0x10420D2C 0x02000000; mw.l 0x1043080c
0x08003000; mw.l 0x10430810 0x18003000; mw.l 0x10420604 0x00040004; mw.l
0x10420C1C 0x00003100; mw.l 0x10420C0C 0x00000001; mw.l 0x10420904 0x00380008;
mw.l 0x10420904 0x00380038; ext4load mmc 0:2 0x08001e00 boot/
rzv2n_cm33_rpmsg_linux-rtos_demo.bin; mw.l 0x10420C0C 0x00000000; dcache on'
=> saveenv
=> run cm33start
```



#### 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 RZ/V2N EVK Getting Started.
- 2. Invoke the command stated below to specify the sample program to be loaded:

```
root@rzv2n-evk:~# echo rzv2n_cm33_rpmsg_linux-rtos_demo.elf >
/sys/class/remoteproc/remoteproc0/firmware
```

3. Kick CM33 by invoking the command below:

```
root@rzv2n-evk:~# 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@rzv2n-evk:~# echo start > /sys/class/remoteproc/remoteproc0/state
[ 737.289773] remoteprocremoteproc0: powering up cm33
[ 737.348226] remoteprocremoteproc0: Booting fwimage rzv2n_cm33_rpmsg_linux-rtos_demo, 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 RPMsg sample program running on CA55 side.

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

```
=> run bootcmd
```

2. Login as root.

```
rzv2n-evk login: root
```

3. Invoke RPMsg sample program as shown below:

root@rzv2n-evk:~# rpmsg\_sample\_client

4. If the sample program started to work successfully, you can see the following message on Linux console:



```
metal: info: metal uio dev open: No IRQ for device 10480000.mbox-uio.
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/V2N CM33 ch0
2. communicate with RZ/V2N CM33 ch1
e. exit
please input
>
```

- 5. Type **1** if RPMsg channel 0 is used on CM33 RPMsg sample program under default setting. Otherwise, type **2** for starting the communication.
- 6. By typing e, the sample program should be terminated with the message shown below:

```
please input
> e
[xxx] 42f00000.rsctbl closed
[xxx] 4300000.vring-ctl0 closed
[xxx] 43200000.vring-shm0 closed
[xxx] 43100000.vring-ctl1 closed
[xxx] 43500000.vring-shm1 closed
[xxx] 42f01000.bhu-shm closed
```

#### 4.5 Overview of Sample Program's behavior

This section describes the overview of sample program's behavior.

- 1. When the CA55 sample program is successfully executed, the communication channel between CA55 and CM33 is established.
- The CA55 sample program starts to send the message to CM33 with incrementing the message size from the minimum value 17 to the maximum value 488. During the communication, the message as shown below is displayed on your console:

```
[xxx] Sending payload number 148 of size 165
```



- 3. When CM33 sample program receives the message sent from CA55, the echo reply is sent back to CA55 sample program.
- 4. When CA55 receives the echo reply, the message below should be displayed on your console:

```
[xxx] received payload number 148 of size 165
```

5. After the 488-byte sized payload is sent from CA55 to CM33 and CM33 sends back the echo reply, the message indicating the termination of the communication channel is sent from CA55 to CM33. Then, the CA55 sample program outputs the following log messages to your console:

#### 5. CA55 1.8GHz configuration support at CA55 cold boot mode

This chapter describes how to configure 1.8GHz as operational frequence of CA55 at CA55 cold boot.

#### 5.1 Setup of CA55 related stuff

1. Uncomment the following lines in meta-rz-features/meta-rz-multi-os/meta-rzv2n/conf/layer.conf

#MACHINE\_FEATURES\_append = " RZV2N\_CM33\_BOOT" #MACHINE\_FEATURES\_append = " SRAM\_REGION\_ACCESS" MACHINE\_FEATURES\_append = " CM33\_FIRMWARE\_LOAD" MACHINE\_FEATURES\_append = " CA55\_CPU\_CLOCKUP"

Be sure NOT to uncomment the 1st line for CM33 cold boot support.

2. Rebuild TrustedFirmware-A as shown below:

```
MACHINE=rzv2n-evk bitbake trusted-firmware-a -c cleansstate
MACHINE=rzv2n-evk bitbake firmware-pack -c cleansstate
MACHINE=rzv2n-evk bitbake core-image-weston
```

3. Deploy build artifacts to SD card by following <u>Renesas RZ/V AI | The best solution for starting your AI</u> <u>applications.</u>

#### 5.2 Setup of CM33 related stuff

- 1. Import or create RZ/V FSP project for CM33.
- 2. Open configurator.xml in the project and choose BSP tab.
- 3. Configure **Clock up for CA55** properties as Enable. Also, enabled **Launch CA55(core0)** if you would like to configure operational at CM33 cold boot mode.
- 4. Click Generate Project Content to reflect the changes to your project.



	• % • @ * • % •	Run Renesas AI Window				
Project E	xplorer ×	8 7 8 - 0	*[ca55_clockup	_support] FSP Configuration $\times$		
	5_clockup_support		Board Sup	port Package Configuration		Generate Project Co
> 😂 r > 😂 r						Restore De
> 😂 s			Device Selec	tion		
>00	Debug		Device Selec	tion		
> @ .	zv_cfg		FSP version	3.1.0	~	Board Details
> 🗁 s			Board:	RZ/V2N Evaluation Kit v		
	ca55_clockup_support Debug_Flat.launch	1				
	configuration.xml		Device:	R9A09G056N48GBG		
> (?) [	Developer Assistance		Core:	Core 4(CM33_0)		
			RTOS:	FreeRTOS	4	
				Clocks Pins Interrupts Event Links Linker		
Propertie		hansteres -			📑 🐨 🖾 🛷 🕴 s	
	rs × Problems Smart Browser					r 🖬 🖓 🖾 🛷 🕴 🤋
RZ/V2N	l Evaluation Kit					T 또 ア 교 《 8 오
	l Evaluation Kit			nico de la su la la sur en en la construcción de la construcción de la servición de la servic		
RZ/V2N	I Evaluation Kit Property					Value R9A09G056N48GBG
RZ/V2N	I Evaluation Kit Property × R9A09G056N48G8G part_number rom_size_bytes					Value R9A09G056N48G8G 0
RZ/V2N	I Evaluation Kit Property V R9A09G056N48GBG part_number rom_size_bytes ram_size_bytes					Value R9A09G056N48G8G 0 1572864
RZ/V2N	I Evaluation Kit Property V RA09005664486866 part_number rom_size_bytes ram_size_bytes package_styte					Value R9A096056N48G8G 0 1572864 LERGA
RZ/V2N	I Evaluation Kit Property V R9A096056N4868G part.number rom.size.bytes ram.size.bytes package_style package_pins					Value R9A09G056N48G8G 0 1572864
RZ/V2N	I Evaluation Kit Property V RA0950566448686 part,number rom,size,bytes package_style package_pins V RZ Common					Value R9A096056N4868G 0 1572864 LEBGA 840
RZ/V2N	I Evaluation Kit Property V R0A09G056N48GBG part_number rom_size_bytes ram_size_bytes package_sitel package_pins V RZ Common Secure stack size (bytes)					Value R9A09G056N48G8G 0 1572864 LFEGA 840 0x1000
RZ/V2N	I Evaluation Kit Property V RA09G0564448GBG part,number ram,size,bytes ram,size,bytes package,style package,pins V RZ Common Secure stack size (bytes) Main stack size (bytes)					Value R9A09G056N48G8G 0 1572864 LFBGA 840 0x1000 0x1000
RZ/V2N	I Evaluation Kit Property V RA09G056N48GBG part, number rom, size_bytes ram, size_bytes ram, size_bytes package_pins V R2 Common Secure stack size (bytes) Main stack size (bytes) Heep size (bytes)					Value R9A096056N48686 0 1572864 LERGA 840 0×1000 0×1000 0×1000 0×1000
RZ/V2N	I Evaluation Kit Property V R9A096056N486BG part,number rom,size_bytes package_pins V R2 Common Secure stack size (bytes) Main stack size (bytes) Heap size (bytes) MCU Vec (mV)					Value Value 0 1572864 LFEGA 840 0x1000 0x1000 0x0000 3300
RZ/V2N	Evaluation Kit     Property     VRA09G0566448G8G     part,number     ram,size_bytes     ram,size_bytes     package_style     package_style     package_style     VR2 Common     Secure stack size (bytes)     Main stack size (bytes)     Heap size (bytes)     MCU Vac (mV)     Parameter checking					Value R9A096056N48686 0 1572864 LEF6A 840 0 100 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
RZ/V2N	I Evaluation Kit Property VRA09G056N48G8G part,number rom,size_bytes ram,size_bytes package_pins VRZ Common Secure stack size (bytes) Main stack size (bytes) Heap size (bytes) MCU Vcc (mV) Parameter checking Assert Failures					Value R9A09G056N48G8G 0 1572864 LFEGA 840 0x1000 0x1000 0x0000 0x0000 03300 Disabled Return FSP_ERR_ASSERTION
RZ/V2N	Evaluation Kit Property Property Property Property Property Property Provide Provided Provid					Value           R3A095056N48GBG           0           1572854           LEBGA           840           0x1000           0x
RZ/V2N	I Evaluation Kit Property V RAD9G056N48G8G part, number rom, size_bytes ram, size_bytes ram, size_bytes package_pins V R2 Common Secure stack size (bytes) Main stack size (bytes) Mego size (bytes) MCU Vac (mV) Parameter checking Assert Failures Error Log PFS Protect					Value           R9A095056H48686           0           1572864           LF8GA           840           0x1000           0x4000           0x0000           3300           Disabled           Return FSP_ERR_ASSERTION           No Foro Log           Enabled
RZ/V2N	Evaluation Kit     Property     VRA09G0566448GBG     part.number     rom.size.bytes     ram.size.bytes     package.style     package.style     package.style     volume     Secure stack size (bytes)     Main stack size (bytes)     Meap size (bytes)     McU vcc (mV)     Parameter checking     Assert Falures     Error Log     PFS Protect     C Runtime Initialization					Value           R9A095056N48G8G           0           1572854           LFBGA           840           0x1000           0x
RZ/V2N	Evaluation Kit     Property     VRA09G056N48G8G     part,number     ram,size_bytes     ram,size_bytes     package_style     package_style     package_style     vR2 Common     Secure stack size (bytes)     Main stack size (bytes)     Meap size (bytes)     MCU Vac (mV)     Parameter checking     Assert Failures     Error Log     PFS Protect     C Runtime Initialization     Early BSP Initialization					Value           R9A095056H48686           0           1572864           LF8GA           840           0x1000           0x4000           0x0000           3300           Disabled           Return FSP_ERR_ASSERTION           No Foro Log           Enabled
RZ/V2N	Evaluation Kit     Property     VRA09G0566448GBG     part.number     rom.size.bytes     ram.size.bytes     package.style     package.style     package.style     volume     Secure stack size (bytes)     Main stack size (bytes)     Meap size (bytes)     McU vcc (mV)     Parameter checking     Assert Falures     Error Log     PFS Protect     C Runtime Initialization					Value           R9A095056N48G8G           0           1572854           LFBGA           840           0x1000           0x
RZ/V2N	Evaluation Kit     Property     VRXVS     PadogG0566448GBG     pat.rumber     rom.size.bytes     package.style     package.phs     RZ Common     Secure stack size (bytes)     Main stack size (bytes)     Meap size (bytes)     McU vcc (mV)     Parameter checking     Assert Falures     Error Log     PFS Protect     C Runtime Initialization     Entry BSP Initialization     Entry BSP Initialization     VRZVRN					Value           R9A09G056N48G8G           0           1572864           LFEGA           840           0x1000           0x4000           3300           Disabled           Enabled           Enabled           Disabled

Figure 5-1. CM33 project setting for CA55 1.8GHz support (CA55 coldboot)

6. Build the project from **Project > Build Project**.



	_					
Project Explorer × 🖻 🕸 🍸 🖇 🖻	[ca55_clockup_s	support] FSP Configuration $ imes$				
<ul> <li>Ca55_clockup_support</li> <li>Sinaries</li> </ul>	Board Supp	port Package Configuration	Generate Project Content			
> 🔊 Includes > 😕 rzv			^			
> 😂 rzv_gen > 😂 src	Device Select	Device Selection				
> 😓 Debug	FSP version:	3.1.0	✓ Board Details			
<ul> <li>&gt; &gt; rzv_cfg</li> <li>&gt; &gt;&gt; script</li> <li>: ca55_clockup_support Debug_Flat.launch</li> <li>: configuration.xml</li> <li>: Developer Assistance</li> </ul>	Board:	RZ/V2N Evaluation Kit 🗸 🛶	èч.			
	RTOS:	No RTOS				
	Device Selection         FSP version:       3.1.0         Board:       RZ/V2N Evaluation Kit         RZ/V2N Evaluation Kit          Device:       R9A09G056N48GBG         nl       Core:         Core:       Core 4(CM33_0)					
	<		>			
	Summary BSP	Clocks Pins Interrupts Event Links Linker S	ections Stacks Components			
Properties Problems Smart Browser	Summary BSP	Clocks Pins Interrupts Event Links Linker S				
DT Build Console [ca55_dockup_support] Building file:/rzv/fsp/src/bsp/mcu/all Building file:/rzv/fsp/src/bsp/cmsis/ Building target: ca55_clockup_support.elf arm-none-eabi-objcopy -0 srec "ca55_clock arm-none-eabi-sizeformat=berkeley "ca55_spression"	Console × L/bsp_sbrk.c L/bsp_select_irq. Device/RENESAS/Sc Device/RENESAS/Sc Device/RENESAS/Sc ard_init.c ard_init.c Device/RENESAS/Sc f (up_support.elf"	<pre></pre>				

Figure 5-2. Build CM33 project for CA55 1.8GHz support

#### 5.3 Deployment of Build Artifacts to RZ/V2N EVK

This section describes the procedure to program the build artifacts to RZ/V2N EVK.

- 1. Connect CN12 of RZ/V2N EVK with Host PC and established serial port connection.
- 2. Configure DSW1-4 and DSW1-5 of RZ/V2N EVK as OFF and ON respectively to specify boot mode as SCIF download mode.
- 3. Turn on RZ/V2N EVK. Then, the following message is shown on your terminal:



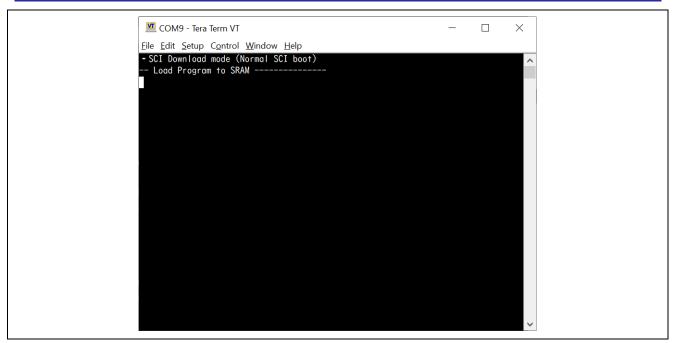


Figure 5-3. SCIF Download mode

4. Send **Flash\_Writer\_SCIF\_RZV2N\_DEV\_INTERNAL\_MEMORY.mot** to RZ/V2N EVK via terminal software. If it's successfully transferred, the following message is shown on your terminal:

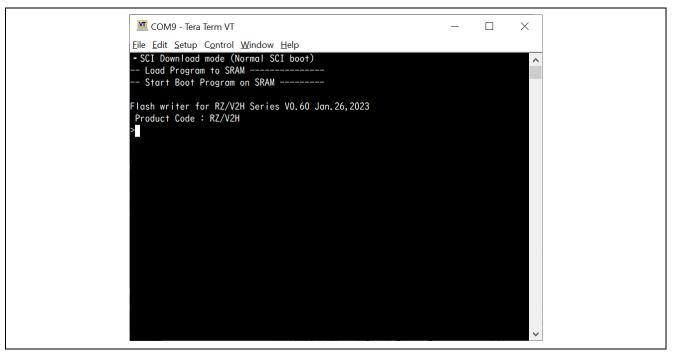


Figure 5-4. Flash Writer invocation

5. Program **bl2\_bp\_spi-rzv2n-evk.srec** with Flash Writer as shown below:



xls2 ===== Qspi writing of RZ/G2 Board Command ============= Load Program to Spiflash Writes to any of SPI address. Program size & Qspi Save Address ===== Please Input Program Top Address ========== Please Input : H'8101E00 ===== Please Input Qspi Save Address === Please Input : H'00000 please send ! ('.' & CR stop load) Erase SPI Flash memory... Erase Completed Write to SPI Flash memory. SpiFlashMemory Stat Address : H'00000000 SpiFlashMemory End Address : H'00037E57 \_\_\_\_\_

6. Program fip-rzv2n-evk.srec with Flash Writer as shown below:

xls2 Load Program to Spiflash Writes to any of SPI address. Program size & Qspi Save Address ===== Please Input Program Top Address ========== Please Input : H'00000 ===== Please Input Qspi Save Address === Please Input : H'60000 please send ! ('.' & CR stop load) Erase SPI Flash memory... Erase Completed Write to SPI Flash memory. SpiFlashMemory Stat Address : H'00060000 SpiFlashMemory End Address : H'0011C2BE \_\_\_\_\_



7. Program CM33 FW (S-record formatted one) with Flash Writer as shown below:

```
xls2
===== Qspi writing of RZ/G2 Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
Program size & Qspi Save Address
===== Please Input Program Top Address ==========
 Please Input : H'00000
===== Please Input Qspi Save Address ===
 Please Input : H'202000
please send ! ('.' & CR stop load)
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
SpiFlashMemory Stat Address : H'00202000
SpiFlashMemory End Address : H'002071F2
_____
```

#### 6. CM33 cold boot support

This chapter describes how CA55 and CM33 related stuff should be built for CM33 cold boot.

#### 6.1 Setup of CA55 related stuff

1. Uncomment the following lines in meta-rz-features/meta-rz-multi-os/meta-rzv2n/conf/layer.conf

```
MACHINE_FEATURES_append = " RZV2N_CM33_BOOT"
#MACHINE_FEATURES_append = " SRAM_REGION_ACCESS"
#MACHINE_FEATURES_append = " CM33_FIRMWARE_LOAD"
#MACHINE_FEATURES_append = " CA55_CPU_CLOCKUP"
```

Be sure to uncomment the above mentioned 3rd and 4th line when CM33 cold boot support is enabled.

2. Rebuild TrustedFirmware-A as shown below:

MACHINE=rzv2n-evk bitbake trusted-firmware-a -c cleansstate MACHINE=rzv2n-evk bitbake firmware-pack -c cleansstate MACHINE=rzv2n-evk bitbake core-image-weston

3. Deploy build artifacts to SD card by following <u>Renesas RZ/V AI | The best solution for starting your AI</u> <u>applications.</u>

#### 6.2 Deployment of CA55 Build Artifacts to RZ/V2N EVK

This section describes how to deploy CA55 Build Artifacts to RZ/V2N EVK. First, please carry out the same procedure as 1 to 4 stated in **5.3 Deployment of Build Artifacts to RZ/V2N EVK**. Then, follow the steps stated below:

1. Program **bl2\_bp\_spi-rzv2n-evk.srec** with Flash Writer as shown below:



```
xls2
===== Qspi writing of RZ/G2 Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
Program size & Qspi Save Address
===== Please Input Program Top Address ==========
 Please Input : H'8101E00
===== Please Input Qspi Save Address ===
 Please Input : H'100000
please send ! ('.' & CR stop load)
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
SpiFlashMemory Stat Address : H'00100000
SpiFlashMemory End Address : H'00136D17
_____
                                 _____
```

2. Program fip-rzv2n-evk.srec with Flash Writer as shown below:

```
xls2
===== Qspi writing of RZ/G2 Board Command =========
Load Program to Spiflash
Writes to any of SPI address.
Program size & Qspi Save Address
===== Please Input Program Top Address ==========
 Please Input : H'00000
===== Please Input Qspi Save Address ===
 Please Input : H'280000
please send ! ('.' & CR stop load)
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
SpiFlashMemory Stat Address : H'00280000
SpiFlashMemory End Address : H'0033C2BE
_____
```

#### 6.3 Setup of CM33 related stuff

- 1. Import or create RZ/V FSP project for CM33.
- 2. Open configurator.xml in the project and choose BSP tab.
- 3. (Optional) Configure Launch CA55(core0) and Clock up for CA55 as Enabled.
- 4. Click Generate Project Content to reflect the changes to your project.



#### RZ/V2N

Sealars E.	plorer X 🖪 😵 🍸 🖇 🗖 🗖	Mil Level elaste	up_support] FSP Configuration ×			
	udokup oppul					
> 6. 8		Board Supp	port Package Configuration		G	Senerate Project Conter
						veneret riejeer come
> 3 8						Restore Defau
> 😕 r						
> 😕 s		Device Selec	tion			
>00		FSP version:	210	~	Board Details	
>00		r ar version.				
	a <u>SS_clockup_supp</u> ort_Debug_Flat.launch	Board:	RZ/V2H Evaluation Kit ~	1		
	onfiguration.xml	Device:	R9A09G057H44G8G	-		
	rv_cfg.txt					
> ? (	eveloper Assistance	Core	Core 6(CM33_0)	~		
		RTOS:	No RTOS	4		
operties	× 問題 スマート・プラウザー Console	Summary BSP	Clocks Pins Interrupts Event Links Stacks	Compor	ients	
roperties Z/V2H	× 問題 スマート・ブラウザー Console Evaluation Kit	Summary BSP			ients	
Z/V2H	Evaluation Kit Property	Summary BSP		Compon	ents	
1	Evaluation Kit Property ~ R9A09G057H44G8G	Summary BSP		Value		
Z/V2H	Evaluation Kit Property ~ R9A09G057H44G8G part_number	Summary BSP		Value R9A09G05		
Z/V2H	Evaluation Kit Property ~ R9A096057H4468G part_number rom_size_bytes	Summary BSP		Value R9A09G05 0		
Z/V2H	Evaluation Kit Property V R9A096057H446BG part_number rom_size_bytes ram_size_bytes	Summary BSP		Value R9A09G05 0 131072		
Z/V2H	Evaluation Kit Property ~ R9A096057H4468G part_number rom_size_bytes	Summary (BSP)		Value R9A09G05 0		
Z/V2H	Evaluation Kit Property V R9A09G057H44GBG part_number rom_size_bytes ram_size_bytes package_style	Summary (BSP)		Value R9A09G05 0 131072 LFBGA		
Z/V2H	Evaluation Kit Property V R9A09G057H44GBG part_number rom_size_bytes ram_size_bytes package_style package_style package_pins V RZ Common Secure stack size (bytes)	Summary (BSP)		Value R9A09G05 0 131072 LFBGA 1368 Dx1000		
Z/V2H	Evaluation Kit Property	Summary (BSP)		Value R9A09G05 0 131072 LFBGA 1368 Dx 1000 Dx 1000		
Z/V2H	Evaluation Kit Property < R9A096057H44686 part_number rom_size_bytes ram_size_bytes package_style package_pins < RZ Common Secure stack size (bytes) Main stack size (bytes) Heap size (bytes)	Summary (BSP)		Value R9A09G051 0 131072 IFBGA 1368 0x1000 0x1000 0x4000		
Z/V2H	Evaluation Kit Property   Paperty  Package_style  Package_style  Package_style  Package_pins    RZ Common  Secure stack size (bytes)  Main stack size (bytes)  Meany size (bytes)	Summary (BSP)		Value R9A09G051 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300		
Z/V2H	Evaluation Kit  Property  R94096057H44686  part_number rom_size_bytes ram_size_bytes package_style package_pins  X2 Common Secure stack size (bytes) Main stack size (bytes) Heap size (bytes) MCU Vcc (mV) Parameter checking	Summary (SSP)		Value R9A09G05 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300 Disabled	7++44GBG	
Z/V2H	Evaluation Kit Property   Paperty  Package_style  Package_style  Package_style  Package_pins    RZ Common  Secure stack size (bytes)  Main stack size (bytes)  Meany size (bytes)	Summary (SSP)		Value R9A09G05: 0 131072 LFBGA 1368 0x1000 0x4000 3000 Disabled Return FSP	7H44GBG P_ERR_ASSERTION	
V2H	Evaluation Kit Property    R9A096057H44686  part_number rom_size_bytes ram_size_bytes package_style package_style Package_pins	Summary (SSP)		Value R9A09G05 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300 Disabled	7H44GBG P_ERR_ASSERTION	
V2H	Evaluation Kit  Property    R9A096057H4468BG  part_number  rom_size_bytes  ram_size_bytes  package_style  package_style  package_tyle  package	Summary (SSP)		Value R9A09G05 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300 Disabled Return FSP No Error Lo Enabled Enabled	7H44GBG P_ERR_ASSERTION	
/V2H	Evaluation Kit  Property  R9A096057H4468G part_number rom_size_bytes ram_size_bytes package_sityle package_pins  RZ Common Secure stack size (bytes) Main stack size (bytes) Meap size (bytes) MCU Vec (mV) Parameter checking Asset Failures Error Log PFS Protect C Runtime Initialization Early BSP Initialization Early BSP Initialization	Summary (SSP)		Value R9A09G05: 0 131072 LFBGA 1368 0x1000 0x4000 0x4000 0x4000 0x4000 Disabled Return FSP No Error L Enabled	7H44GBG P_ERR_ASSERTION	
/V2H	Evaluation Kit  Property  Property  Progety  Padog6057H4468G  padr_number rom_size_bytes ram_size_bytes package_style package_pins  RZ Common Secure stack size (bytes) Main stack size (bytes) Mein stack size (bytes) Mein stack size (bytes) MCU Vcc (mV) Parameter checking Asset Failures Error Log PrS Protect C Runtime Initialization Early BSP Initialization V RZV2H	Summary (SSP)		Value R9A09G05 0 131072 LFBGA 1368 0x1000 0x00000 0x00000 0x000000	7H44GBG P_ERR_ASSERTION	
/V2H	Evaluation Kit  Property  R9A096057H4468BG  part_number  rom_size_bytes  ram_size_bytes  package_style  package_style  package_tyle  package_t	Summary (SSP)		Value R9A09G05: 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300 Disabled Return FSP No Error Lo Enabled Disabled 2	7H44GBG P_ERR_ASSERTION	
/V2H	Evaluation Kit  Property  R9A096057H44686 part_number rom_size_bytes ram_size_bytes package_style package_pins  RZ Common Secure stack size (bytes) Main stack size (bytes) MGU Vec (mV) Parameter checking Assert Failures Error Log PFS Protect C Runtime Initialization Early BSP Initialization RXUV2H Series Launch CA55(core0)	Summary (SSP)		Value R9A09G05'0 131072 IF8GA 1368 0x1000 0x4000 3300 Disabled Return FSP No Error L Enabled Disabled Disabled 2 Enabled	7H44GBG P_ERR_ASSERTION	
V2H	Evaluation Kit  Property  R9A096057H4468BG  part_number  rom_size_bytes  ram_size_bytes  package_style  package_style  package_tyle  package_t	Summary (SSP)		Value R9A09G05: 0 131072 LFBGA 1368 0x1000 0x1000 0x4000 3300 Disabled Return FSP No Error Lo Enabled Disabled 2	7H44GBG P_ERR_ASSERTION	

Figure 5-5. CM33 project setting for CM33 cold boot

- 5. Build the project from **<u>Project</u> > Build Project**.
- 6. Click **Run > Debug Configurations...**, expand **Renesas GDB Hardware Debugging** and choose project name> Debug\_Flat.

📓 🗑 🕶 🍕 🕶 🎯 🛊 🕶 💁 🕶		Renesas Debug Tools	>	1	Create, manage, and run configurations
Project Explorer $ imes$	0	Run	Ctrl+F11	P	
😤 ca\$5_clockup_support [Debug]	核	Debug	F11	Ļ	
> 🖗 Binaries		Run History	>		
> 🔊 Includes > 😂 rzv	0	Run As			type filter text
> 😂 rzv > 😂 rzv_gen	~	Run Configurations	·		C C/C++ Application
> 😂 src				k	C C/C++ Remote Application
> 🗁 Debug		Debug History	>		EASE Script
> 🧁 rzv_cfg	林	Debug As	>		C GDB Hardware Debugging
> 🗁 script		Debug Configurations		1	GDB Simulator Debugging (RH850)
ca55_clockup_support Debug_Flat.launch     configuration.xml	9	External Tools	>		➡ Launch Group ✓ C <sup>3</sup> Renesas GDB Hardware Debugging
rzv_cfg.txt	_			1. I.	c <sup>*</sup> ca55_clockup_support Debug_Flat
<ul> <li>rzv_cfg.txt</li> <li>&gt; Oeveloper Assistance</li> </ul>			Core:	[	c=* ca55_clockup_support Debug_Flat c=* Renesas Simulator Debugging (RX, RL78)
			DTOC		

Figure 5-6. Debug Configuration Launch



7. Choose **Debugger > Connection Settings** and specify **Yes** to **Reset after download**.

Pre filter text     C/C++ Application     C/C++ Remote Application     E C/C++     Remote Application     EASE Script	Name: ca55_clockup_support Debug_Flat	•	
C/C++ Application     C/C++ Remote Application     EASE Script	Main 🌾 Debugger 🕨 Startup 🛄 Common 😜		
C/C++ Remote Application		Source	
EASE Script	Debug hardware: J-Link ARM V Target Device: F	R9A09G057H44GBG	
	ange bene		
C GDB Hardware Debugging	GDB Settings Connection Settings Debug Tool Set	tion and	
GDB Simulator Debugging (RH850)	✓ J-Link	ungs	•
Launch Group	Type	USB	vî I
Renesas GDB Hardware Debugging	J-Link Serial	(4 · · · )	
ca55_clockup_support Debug_Flat	Settings File	S{workspace_loc:/S{ProjName}}/S{LaunchConfigName}.jlink	
Renesas Simulator Debugging (RX, RL78)	Script File	stronspace_oes stroj tantej psteathereoring tantej jink	
	Log File		
	Low Power Handling		~
	✓ IP Connection		
	Connection Method	IP via LAN	~
	Host Name/IP Address[:port number]		
	Identifier		
	Tunnel Server		
	Port Number		
	Password		
	✓ Interface		
	Туре		~
	Speed (kHz)	15000	~
	✓ JTAG Scan Chain		
	Multiple Devices		~
	IRPre	0	
	DRPre	U	
	Connection	No	
	Register initialization Reset at the beginning of connection		<b>č</b>
	Reset at the end of connection		÷
	Reset before download		<b>~</b>
	Reset after download		÷.
	ID Code (Bytes)		
	Hold reset during connect		<b>v</b>
	Set CPSR(5bit) after download		~
	Prevent Releasing the Reset of the CM3 Core		~
	Secure Vector Address	&_Secure_Vectors	
	Non-secure Vector Address		
	Hot Plug		~
	Disconnection Mode	Continue	~ ~
	L		
		Revert Ap	ahi

Figure 5-7. Connection Settings

8. Choose Startup and change the Load type of Program Binary to Symbols only.

<ul> <li>Debug Configurations</li> <li>Create, manage, and run configurations</li> </ul>	
Image: Solution of the second seco	Name: Ca55_clockup_support Debug_Flat          Main       Debugger       Startup         Initialization Commands       Common         Reset and Delay (seconds):       3         Halt       Load image and symbols
	Filename     Load type     Offset (hex)     On connect       Program Binary [ca55     Symbols only     Yes

Figure 5-8. Startup Settings (1)



9. Click Add... > Workspace... , choose srec file and click OK.

				🕄 Add download module	ı ×
				Select a workspace resource	
Load image and symbols					^
Filename         Load type           Program Binary [ca55         Symbols only           Add download module         Add download module	Offset (hex) On Yes	connect		> > > src bin bin elf elf.in	
variables Search Project Workspace	File System		Add Edit Remove	rpd sbd	
OK	Cancel		Move up Move down	<ul> <li>compile_commands.json</li> <li>makefile</li> <li>makefile.init</li> <li>memory_regions.ld</li> </ul>	
				© OK Ca	ancel
☐ Main 参 Debugger ► S Initialization Commands	tartup 🔲 Common 🤤	Source			
Reset and Delay (second: Halt	): 3			Add download module	3
				Specify download module name: \${workspace_loc:\ca55_clockup_support\Debug\	-
Load image and symbols	Let we			Variables Search Project Workspace File S	System
Filename Program Binary (ca55_		Offset (hex) On connect Ves 0 Yes		OK Ca	ancel

Figure 5-8. Startup Settings (2)

10. Click **Debug** to launch Debug Perspective. Note that DSW1-1, 4, 5 and 7 should be specified as OFF, OFF, OFF, ON, respectively before the launch. Then, CM33 program starts, load CA55 build artifacts from QSPI Flash ROM and kick CA55.

For details on CM33 program invocation with e2studio, please see <u>Getting Started with RZ/V Flexible</u> <u>Software Package</u>.

#### 7. Reference Documents

- R01AN6240 RZ/V2L, RZ/V2H, RZ/V2N Getting Started with Flexible Software Package
- R12UZ0157 RZ/V2N Evaluation Board Kit (Secure type) Hardware Manual



## **Revision History**

		Description		
Rev.	Date	Page	Summary	
3.00	Mar.11.2025	-	Migrated to RZ/V2N as per Release Note of RZ/V Multi-OS Package v2.1.0.	
3.10	May.22.2025	-	Updated in align with RZ/V Multi-OS Package v3.1.0.	
3.11	Jun.13.2025	-	Update Multi-OS Package version to 3.1.1.	



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

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

#### 1. Precaution against Electrostatic Discharge (ESD)

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

#### 2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

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

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

#### Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 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.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
   Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

#### **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

#### Trademarks

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

#### **Contact information**

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