RZ/V2H, RZ/V2N

AWO Example Program Startup Guide

Introduction

This material shows how to set up and invoke AWO Example Program for RZ/V2H and RZ/V2N.

Target Device

RZ/V2H, RZ/V2N

Contents

1. Specifications 2
1.1 Deliverables 2
2. Proven Environment 2
3. AWO Example Program Setup for RZ/V2H 2
3.1 Setup of Cortex-A55 software related stuff 2
3.2 Note for integration 3
3.3 Deployment of CA55 Build Artifacts 4
4. AWO Example Program Setup for RZ/V2N 6
4.1 Setup of Cortex-A55 software related stuff 6
4.2 Deployment of CA55 Build Artifacts 7
4.3 Setup of CM33 software related stuff 9
4.4 AWO Example Program Invocation 11
5. AWO Example Program Invocation (Suspend-to-RAM Unsupported) 14
6. AWO Example Program Invocation (Suspend-to-RAM Supported) 15
7. Assignment of peripherals 16

1. Specifications

1.1 Deliverables

Table 1-1. Deliverables of RZ/V AWO Example Project

Deliverables	File name	Description
RZ/V2H Cortex®-M33 AWO example project	freertos_w_awo_rzv2h_evk_example.zip	AWO example project for RZ/V2H.
RZ/V2N Cortex®-M33 AWO example project	freertos_w_awo_rzv2n_evk_example.zip	AWO example project for RZ/V2N.
RZ/V AWO Example Program Start-up Guide	r01an7723ej0420-rzv2h-rzv2n-awo-example-program-startup-guide.pdf	This material.

Note: The RZ/V2H AWO example is developed using the Cortex-M33 (CM33) cold-boot environment, which requires the use of J-Link DLL version 7.96e. Therefore, you must update your installed J-Link software to this specific version to ensure correct operation when debugging the AWO example. In contrast, the RZ/V2N AWO example does not have this restriction and can be debugged using J-Link DLL version 9.44.

For more details regarding the development and debugging environment, please refer to Chapter 7 of the RZV Getting Started with Flexible Software Package.

2. Proven Environment

Environments	Contents and versions
Integrated Development Environment	e2 studio 2026-07 or later
JTAG Emulator	Segger J-Link 7.96 Segger J-Link 9.44
Dependent Software	• RZ/V2H AI SDK v6.00 • RZ/V2N AI SDK v6.30 • RZ Flexible Software Package (FSP) v4.2.0

3. AWO Example Program Setup for RZ/V2H

3.1 Setup of Cortex-A55 software related stuff

1. Carry out Step 1, Step 2, and 1-8 of Step 3 stated in **How to build RZ/V2H AI SDK Source Code** showcased at AI Applications and AI SDK of RZ/V series.
2. Download Multi-OS Package (r01an8260ej0420-rz-multi-os-pkg.zip) to a working directory and run the commands stated below:

```
$ cd ${YOCTO_WORK}
$ unzip <Multi-OS Dir>/r01an8260ej0420-rz-multi-os-pkg.zip
$ tar zxvf r01an8260ej0420-rz-multi-os-pkg/meta-rz-features_multi-os_v4.2.0.tar.gz
```

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

3. Uncomment the following lines in **meta-rz-features/meta-rz-multi-os/meta-rzv/meta-rzv2h/conf/layer.conf**.

```
#MACHINE_FEATURES:append = " RZ_SRAM_REGION_ACCESS"
#MACHINE_FEATURES:append = " RZ_REMOTEPROC"
MACHINE_FEATURES:append = " RZ_CM33_COLDBOOT"
#MACHINE_FEATURES:append = " RZ_CM33_FIRMWARE_LOAD"
#MACHINE_FEATURES:append = " RZ_CA55_CPU_CLOCKUP"
```

Be sure NOT to uncomment the above-mentioned 2nd, 4th and 5th lines when CM33 cold boot support is enabled.

4. Add the layer for Multi-OS Package, simply run the helper script and select the **meta-rzv2h** layer from the list.

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

5. Continue to carry out the remaining procedures stated in **Step 3** of **How to build RZ/V2H AI SDK Source Code**.

   Note: The AI SDK includes a setting that disables network source fetching (`0000-AI_SDK-settings.patch`). For the Multi-OS Package, network access is required to fetch the OpenAMP sources. Please update the following entry in **conf/local.conf**:

```
BB_NO_NETWORK = "0"
```

3.2 Note for integration

By default, all clock-related configuration is handled on the Linux (main core) side. If the system design is changed such that clocks for specific drivers are configured by the RTOS (sub core) side, the implementation must be customized to ensure these settings do not conflict with Linux's clock initialization sequence.

The peripherals which are NOT enabled enter Module Standby Mode after Linux kernel is booted up. That means the peripherals used on sub core (CM33) side might stop working at that time. To avoid such a situation, Multi-OS Package incorporates the patch below:

- 0002-Updated-GTM-channels-to-support-RTOS.patch (RZ/V2H)
- 0002-Set-OSTM-for-MCPU-and-RCPU.patch (RZ/V2N)

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/r9a09g057-cpg.c b/drivers/clk/renesas/r9a09g057-cpg.c
index 8535afb0d7cd..dff5458deb43 100644
--- a/drivers/clk/renesas/r9a09g057-cpg.c
+++ b/drivers/clk/renesas/r9a09g057-cpg.c
@@ -336,21 +336,21 @@ static const struct rzv2h_mod_clk r9a09g057_mod_clks[] __initconst = {
                  BUS_MSTOP(6, BIT(1))),
   DEF_MOD("rcpu_cmtw3_clkm",    CLK_PLLCLN_DIV32, 4, 2, 2, 2,
                  BUS_MSTOP(6, BIT(2))),
- DEF_MOD("gtm_0_pclk",         CLK_PLLCM33_DIV16, 4, 3, 2, 3,
+ DEF_MOD_CRITICAL("gtm_0_pclk",       CLK_PLLCM33_DIV16, 4, 3, 2, 3,
                  BUS_MSTOP(5, BIT(10))),
- DEF_MOD("gtm_1_pclk",         CLK_PLLCM33_DIV16, 4, 4, 2, 4,
+ DEF_MOD_CRITICAL("gtm_1_pclk",       CLK_PLLCM33_DIV16, 4, 4, 2, 4,
                  BUS_MSTOP(5, BIT(11))),
   DEF_MOD("gtm_2_pclk",         CLK_PLLCLN_DIV16, 4, 5, 2, 5,
                  BUS_MSTOP(2, BIT(13))),
   DEF_MOD("gtm_3_pclk",         CLK_PLLCLN_DIV16, 4, 6, 2, 6,
                  BUS_MSTOP(2, BIT(14))),
- DEF_MOD("gtm_4_pclk",         CLK_PLLCLN_DIV16, 4, 7, 2, 7,
+ DEF_MOD_CRITICAL("gtm_4_pclk",       CLK_PLLCLN_DIV16, 4, 7, 2, 7,
                  BUS_MSTOP(11, BIT(13))),
- DEF_MOD("gtm_5_pclk",         CLK_PLLCLN_DIV16, 4, 8, 2, 8,
+ DEF_MOD_CRITICAL("gtm_5_pclk",       CLK_PLLCLN_DIV16, 4, 8, 2, 8,
                  BUS_MSTOP(11, BIT(14))),
- DEF_MOD("gtm_6_pclk",         CLK_PLLCLN_DIV16, 4, 9, 2, 9,
+ DEF_MOD_CRITICAL("gtm_6_pclk",       CLK_PLLCLN_DIV16, 4, 9, 2, 9,
                  BUS_MSTOP(11, BIT(15))),
- DEF_MOD("gtm_7_pclk",         CLK_PLLCLN_DIV16, 4, 10, 2, 10,
+ DEF_MOD_CRITICAL("gtm_7_pclk",       CLK_PLLCLN_DIV16, 4, 10, 2, 10,
                  BUS_MSTOP(12, BIT(0))),
   DEF_MOD("wdt_0_clkp",         CLK_PLLCM33_DIV16, 4, 11, 2, 11,
                  BUS_MSTOP(3, BIT(10))),
```

Re placing the DEF_MOD of the target clock with a DEF_MOD_CRITICAL patch prevents the Linux BSP from changing the module to standby mode.

3.3 Deployment of CA55 Build Artifacts

1. Connect CN12 of RZ/V2H EVK with Host PC and established serial port connection.
2. Configure DSW1-4 and DSW1-5 of RZ/V2H EVK as OFF and ON respectively to specify boot mode as SCIF download mode.
3. Turn on RZ/V2H EVK. Then, the following message is shown on your terminal:

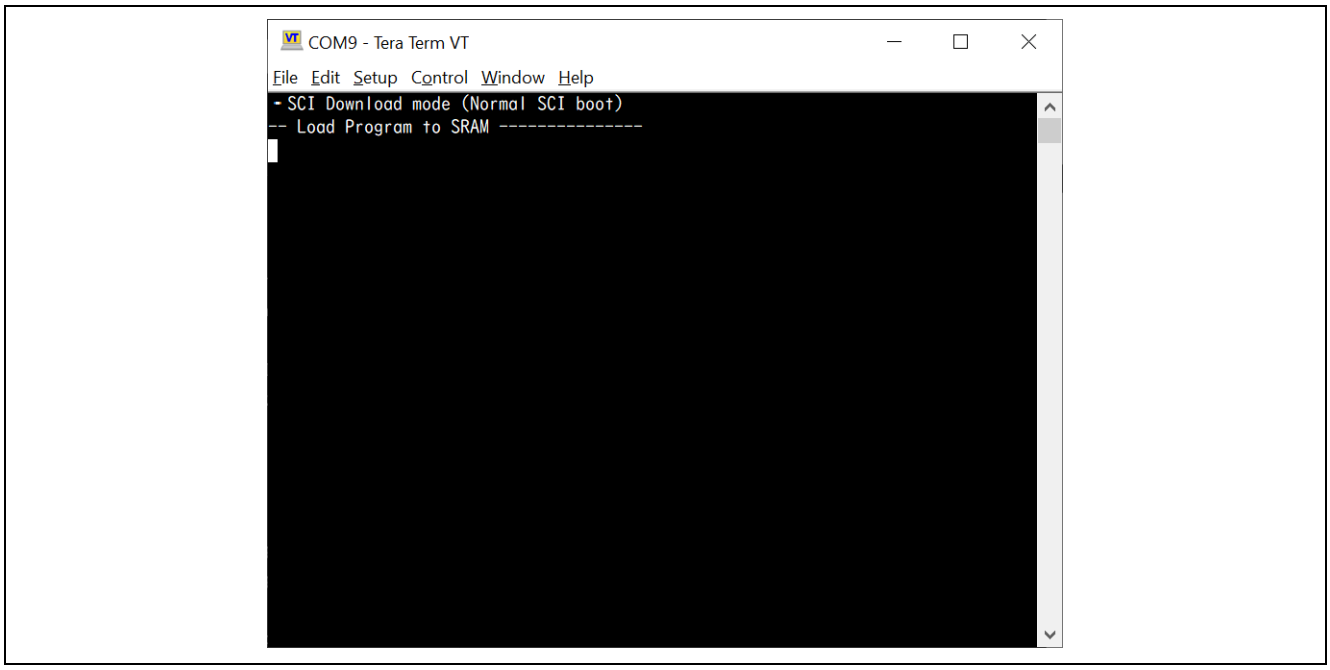


Figure 3-1. SCIF Download mode

4. Send **Flash_Writer_SCIF_RZV2H_DEV_INTERNAL_MEMORY.mot** to RZ/V2H EVK via terminal software. If it's successfully transferred, the following message is shown on your terminal:

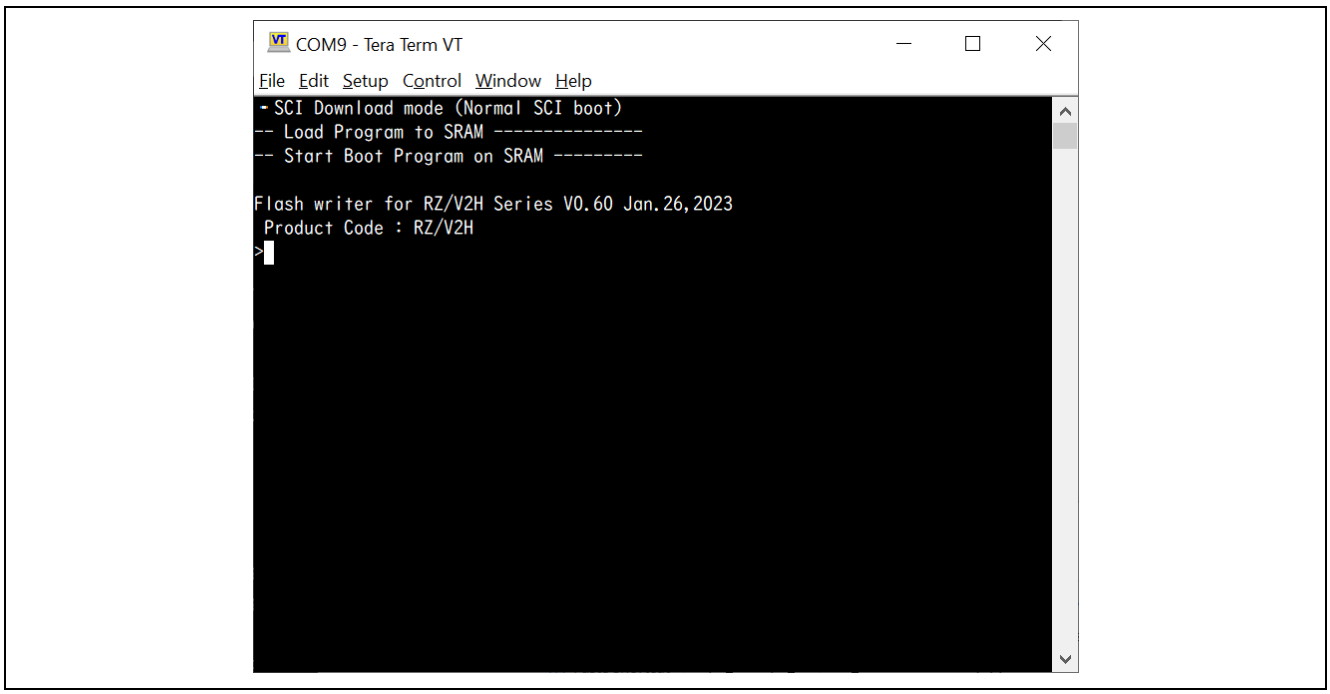


Figure 3-2. Flash Writer invocation

5. Program **bl2_bp_spi-rzv2h-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.
======= Qspi  Save Information  =================
SpiFlashMemory Stat Address : H'00100000
SpiFlashMemory End Address  : H'00136D17
============================================================
```

6. Program **fip-rzv2h-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.
======= Qspi  Save Information  =================
SpiFlashMemory Stat Address : H'00280000
SpiFlashMemory End Address  : H'0033C2BE
============================================================
```

4. AWO Example Program Setup for RZ/V2N

4.1 Setup of Cortex-A55 software related stuff

1. Carry out Step 1, Step 2, and 1-8 of Step 3 stated in **How to build RZ/V2N AI SDK Source Code** showcased at AI Applications and AI SDK of RZ/V series.
2. Download Multi-OS Package (r01an8260ej0420-rz-multi-os-pkg.zip) to a working directory and run the commands stated below:

```
$ cd ${YOCTO_WORK}
$ unzip <Multi-OS Dir>/r01an8260ej0420-rz-multi-os-pkg.zip
$ tar zxvf r01an8260ej0420-rz-multi-os-pkg/meta-rz-features_multi-os_v4.2.0.tar.gz
```

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

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

```
#MACHINE_FEATURES:append = " RZ_SRAM_REGION_ACCESS"
#MACHINE_FEATURES:append = " RZ_REMOTEPROC"
MACHINE_FEATURES:append = " RZ_CM33_COLDBOOT"
#MACHINE_FEATURES:append = " RZ_CM33_FIRMWARE_LOAD"
#MACHINE_FEATURES:append = " RZ_CA55_CPU_CLOCKUP"
```

Be sure NOT to uncomment the above-mentioned 2nd, 4th and 5th lines when CM33 cold boot support is enabled.

4. Add the layer for Multi-OS Package, simply run the helper script and select the **meta-rzv2n** layer from the list.

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

5. (Optional) To enable the AWO example in an S2R-supported environment, add the following line to **meta-renesas/meta-rz-bsp/recipes-bsp/trusted-firmware-a/ trusted-firmware-a_2.10.bb**.

```
EXTRA_OEMAKE:append:rzv2n-family = " PLAT_SYSTEM_SUSPEND=1"
```

6. Continue to carry out the remaining procedures stated in **Step 3** of **How to build RZ/V2N AI SDK Source Code**.

   Note: The AI SDK includes a setting that disables network source fetching (`0000-AI_SDK-settings.patch`). For the Multi-OS Package, network access is required to fetch the OpenAMP sources. Please update the following entry in **conf/local.conf**:

```
BB_NO_NETWORK = "0"
```

4.2 Deployment of CA55 Build Artifacts

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:

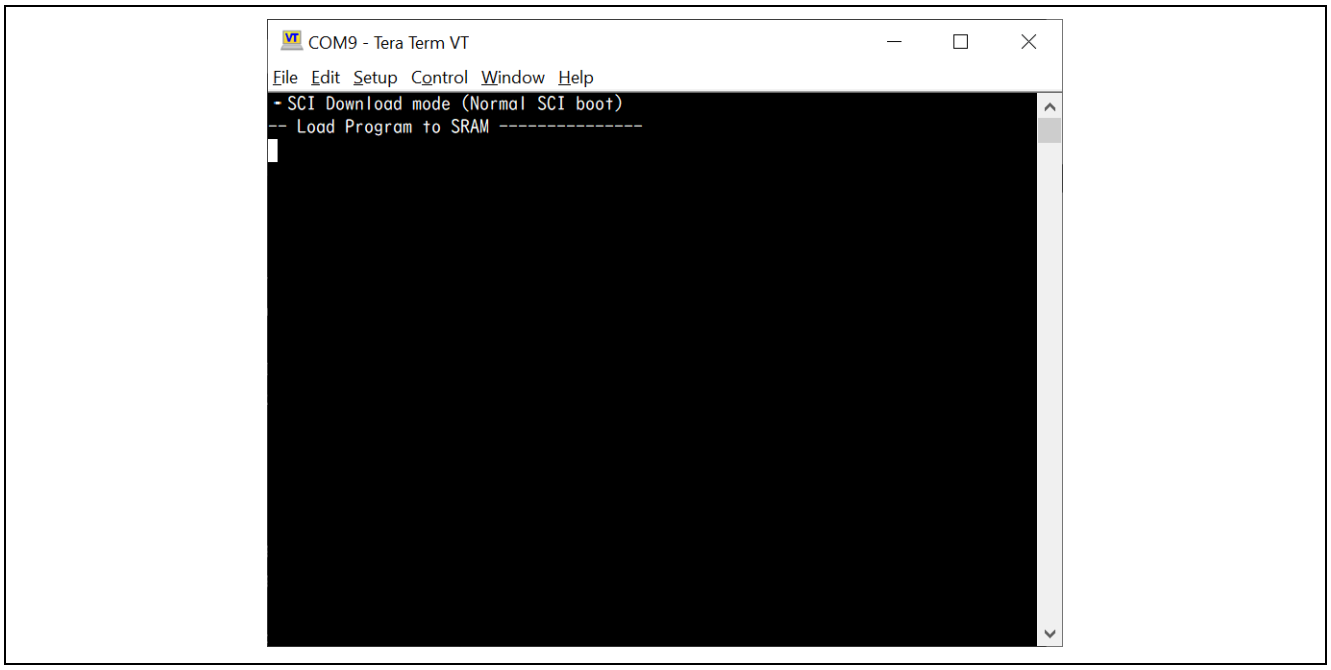


Figure 4-1. SCIF Download mode

4. Send **Flash_Writer_SCIF_RZV2N_DEV_LPDDR4X.mot** to RZ/V2N EVK via terminal software. If it's successfully transferred, the following message is shown on your terminal:

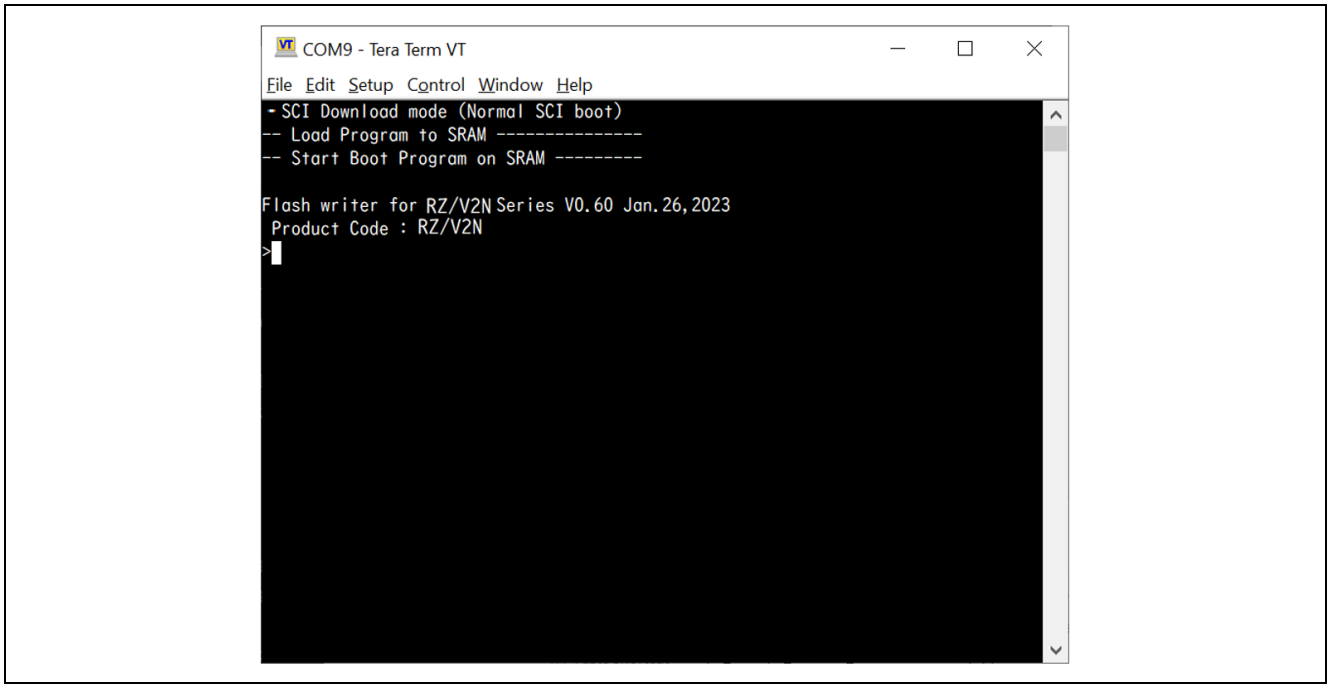


Figure 4-2. Flash Writer invocation

5. Program **bl2_bp_spi-rzv2n-evk.srec** with Flash Writer as shown below:

```
xls2
===== Qspi writing of RZ/G3S Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
 Micron : MT25QU512
Program Top Address & 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.
======= Qspi  Save Information  =================
 SpiFlashMemory Stat Address : H'00100000
 SpiFlashMemory End Address  : H'0011E81F
============================================================
```

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

```
xls2
===== Qspi writing of RZ/G3S Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
 Micron : MT25QU512
Program Top Address & 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.
======= Qspi  Save Information  =================
 SpiFlashMemory Stat Address : H'00280000
 SpiFlashMemory End Address  : H'0033ACFE
============================================================
```

4.3 Setup of CM33 software related stuff

1. Extract **r01an8260ej0420-rz-multi-os-pkg.zip** on your development PC.
2. Extract either of **freertos_w_awo_rzv2h_evk_example.zip** or **freertos_w_awo_rzv2n_evk_example.zip** included in **r01an8260ej0420-rz-multi-os-pkg**.
3. Invoke e^2 studio 2026-07 and click **File** > **Import**.
4. Double-click **General** and select **Existing Projects into Workspace** as shown in Figure 4-3:

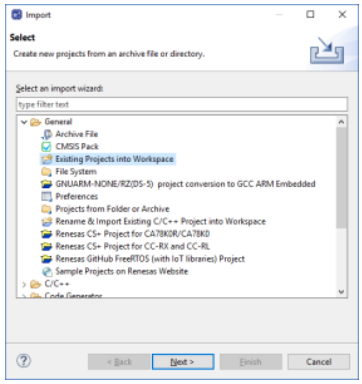


Figure 4-3. Import sample project (1)

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

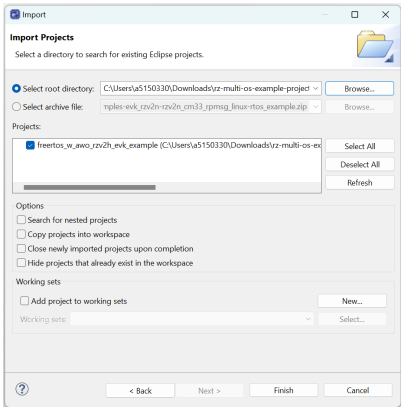


Figure 4-4. Import sample project (2)

6. Open **configurator.xml** in the project and choose **BSP** tab.
7. Configure **Launch CA55(core0)** as Enabled. Also, enabled **Clock up for CA55** if you would like to configure operational frequency of CA55 as 1.8GHz.
8. Click **Generate Project Content** to reflect the changes to your project.

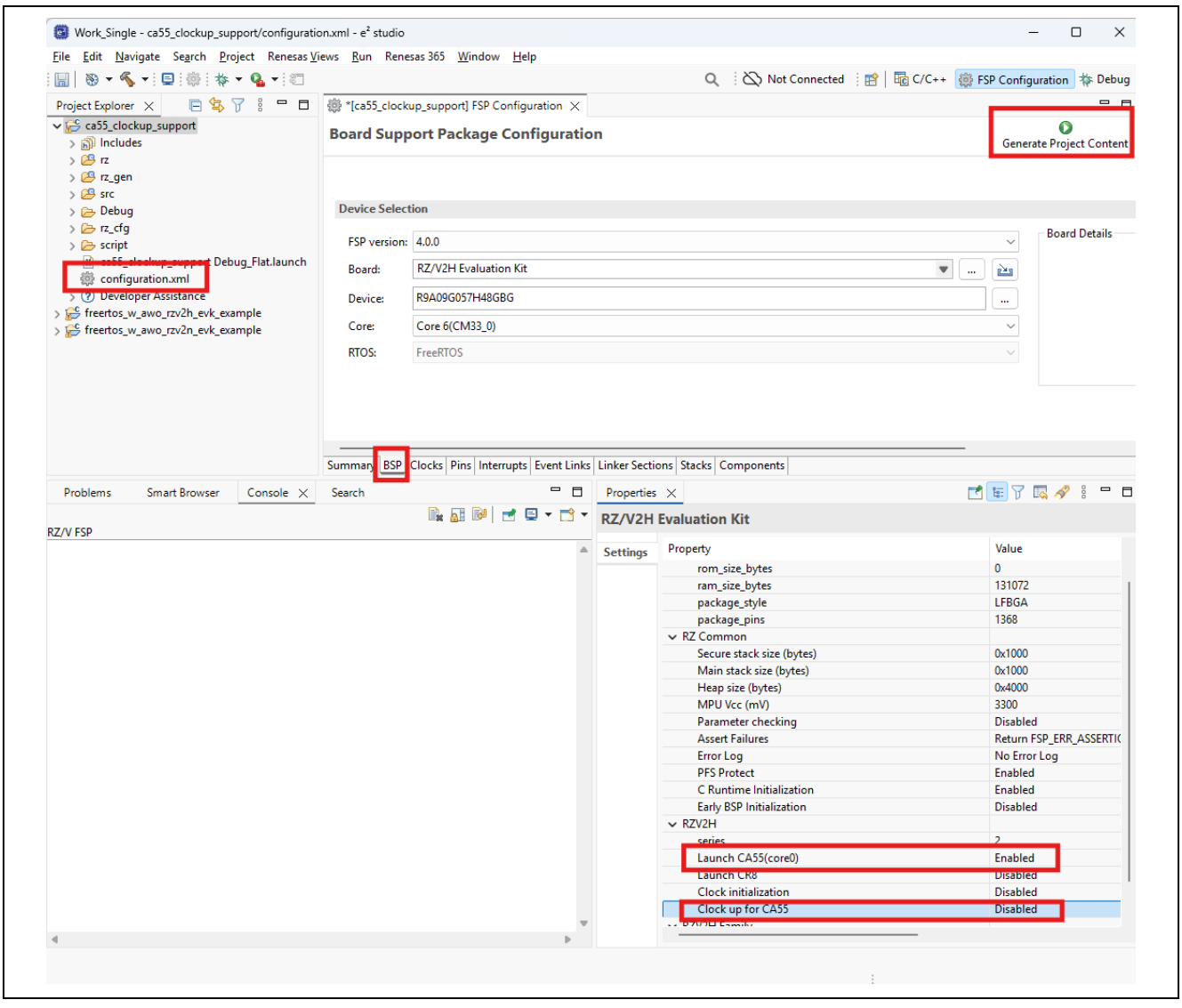


Figure 4-5. CM33 project setting for CM33 cold boot

9. Build the project from **Choose Project** > **Build Project**.
10. 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.
    - freertos_w_awo_<cpu>_evk_example.elf

    Hereafter, <cpu> should be either rzv2h or rzv2n.

4.4 AWO Example Program Invocation

1. Click **Run** > **Debug Configurations...** , expand **Renesas GDB Hardware Debugging** and choose **freertos_w_awo_<cpu>_evk_example Debug_Flat**.

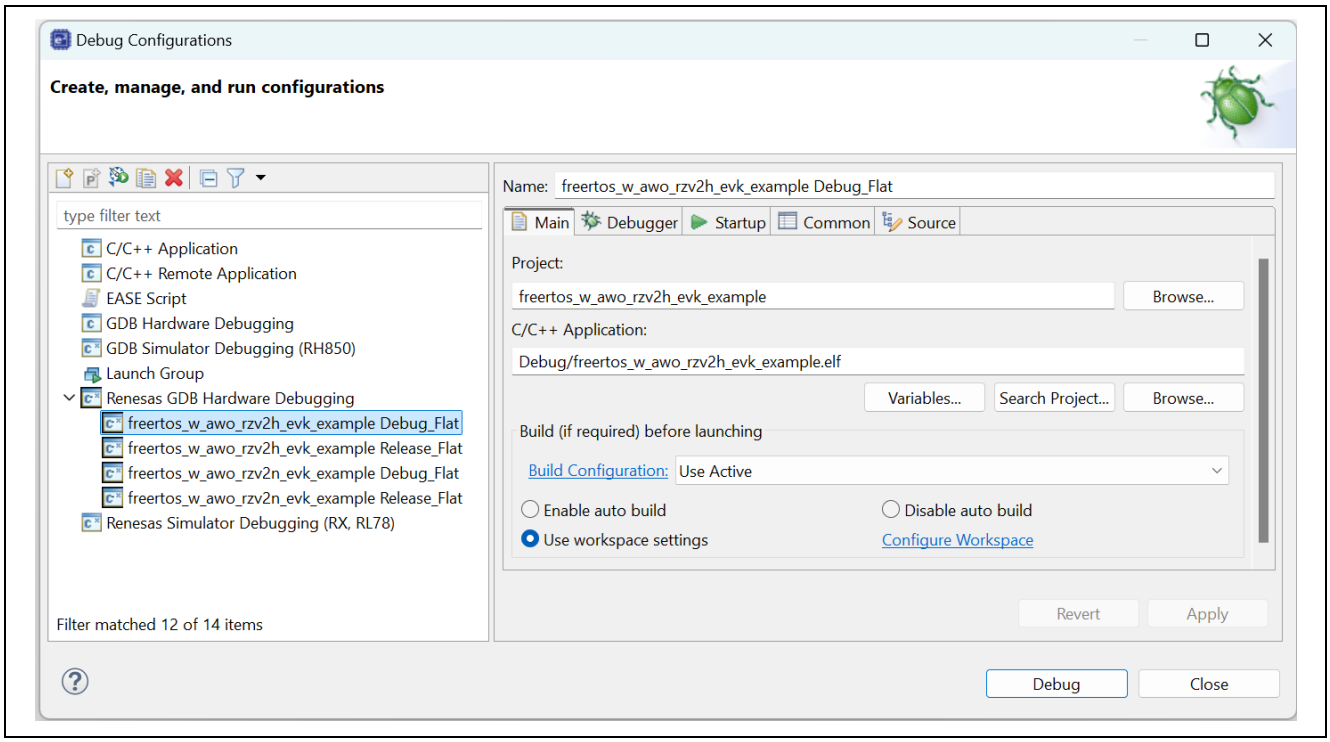


Figure 4-6. Debug Configuration Launch (1)

2. Click **Debug** button as shown below:

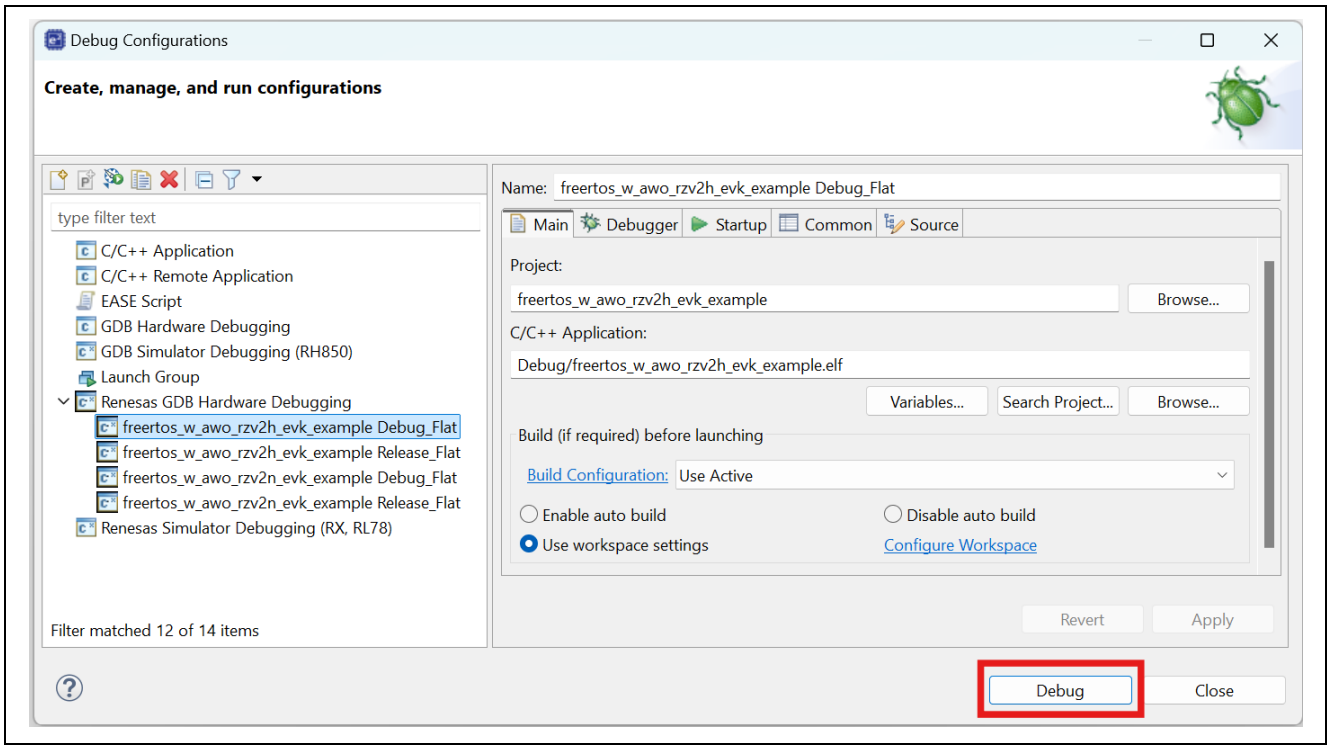


Figure 4-7. Debug Perspective Launch (2)

Note that DSW1-4, 5 and 7 of RZ/V2H EVK or RZ/V2N EVK must be specified as OFF, OFF, ON, respectively beforehand.

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

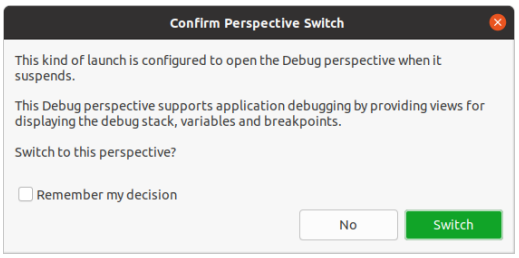


Figure 4-8. Debug Perspective Launch (3)

4. When **Debug Perspective** is opened, Program Counter (PC) should be located as shown in Figure 4-9. Then, continue the program to push the button shown in Figure 4-9.

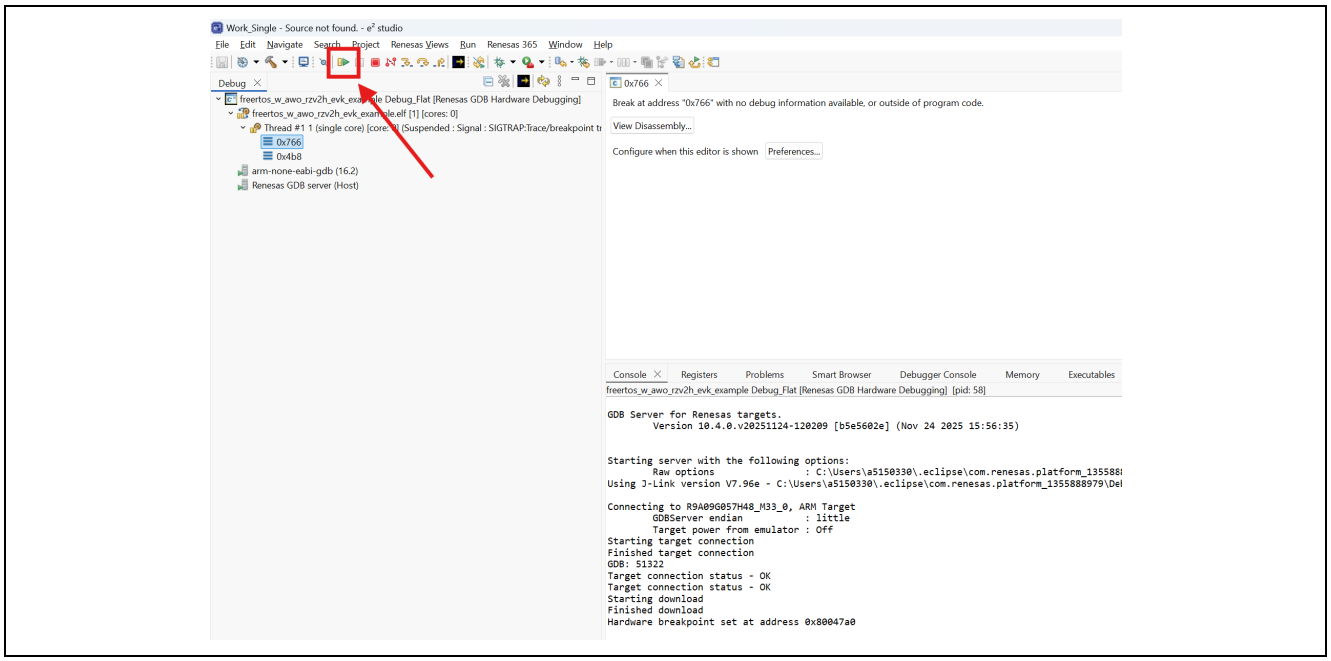


Figure 4-9. How to start to AWO Example Program (1)

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

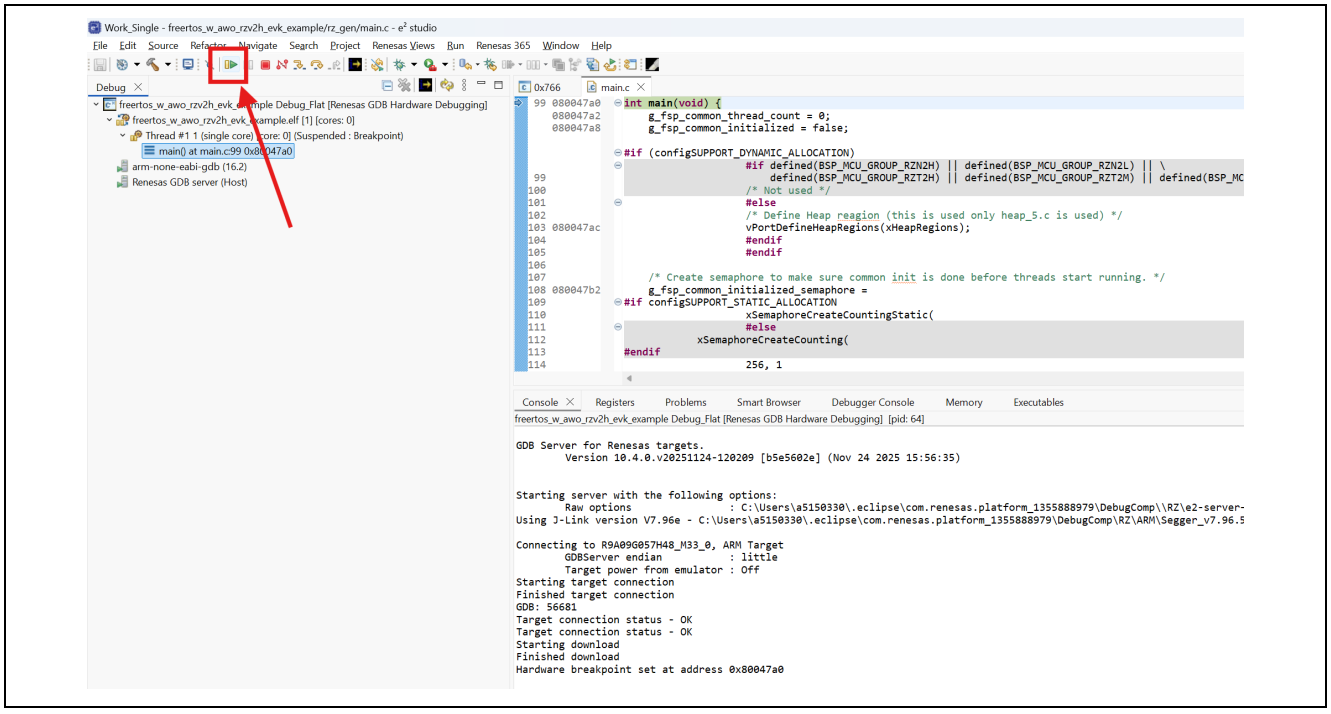


Figure 4-10. How to start to AWO Example Program (2)

CM33 AWO example program now starts, loads CA55 build artifacts and kick CA55.

5. AWO Example Program Invocation (Suspend-to-RAM Unsupported)

This chapter describes how AWO Example Program works on RZ/V2H EVK and RZ/V2N EVK ver1.

1. Make sure configuration on **DSW1** as below:

DSW1:

	1	2	3	4	5	6	7	8
ON			■				■	
OFF	■	■		■	■	■		■

2. Boot up Linux kernel.
3. Login as **root**.

```
rzv2h-evk login: root
```

4. Load **AWO Notifier** which notifies CM33 that CA55 becomes ready to enter AWO.

```
root@rzv2h-evk:~# insmod /home/root/awo-notifier.ko
```

5. Issue shutdown command as shown below:

```
root@rzv2h-evk:~# shutdown -h now
```

While CA55 is being shut down, AWO Notifier fires an external interrupt.

5. Once CM33 AWO example project detects the external interrupt, RZ/V2H or RZ/V2N is configured as AWO. Then, the following message is shown on your console:

```
CM33: AWO request accept.
Hit any key to go to ALLON mode.
```

6. When hitting any key on your development PC, CM33 AWO example project configures RZ/V2H or RZ/V2N as ALLON, load BL2 of TrustedFirmware-A to internal SRAM and kick CA55 for restart.

```
CM33: Set GreenPAK to ALLON
```

6. AWO Example Program Invocation (Suspend-to-RAM Supported)

This chapter describes how AWO Example Program works on RZ/V2N EVK ver2.

1. Make sure configuration on **DSW1** and **DSW2** as below:

DSW1:

	1	2	3	4	5	6	7	8
ON			■				■	
OFF	■	■		■	■	■		■

DSW2:

	1	2
ON	■	■
OFF		

2. Boot up Linux kernel login as **root**.

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

3. Invoke the commands below on Linux console to move Linux to Suspend mode (AWO mode).

```
root@rzv2n-evk:~# echo 0 > /sys/module/printk/parameters/console_suspend
root@rzv2n-evk:~# echo deep > /sys/power/mem_sleep
root@rzv2n-evk:~# echo mem > /sys/power/state
```

4. When Linux successfully moves to S2R, you should see the following display on Linux console:

```
[   45.342134] PM: suspend entry (deep)
[   45.345952] Filesystems sync: 0.000 seconds
[   45.351363] Freezing user space processes
[   45.356947] Freezing user space processes completed (elapsed 0.001 seconds)
[   45.363911] OOM killer disabled.
[   45.367133] Freezing remaining freezable tasks
[   45.372905] Freezing remaining freezable tasks completed (elapsed 0.001
seconds)
[   45.380289] printk: Suspending console(s) (use no_console_suspend to debug)
CM33: AWO request accept.
Hit any key to go to ALLON mode.
```

5. When hitting any key on your development PC, CM33 AWO example project configures RZ/V2N as ALLON, load BL2 of TrustedFirmware-A to internal SRAM and kick CA55 for restart. Then the Linux will resume from TrustedFirmwareF-A:

```
CM33: Set GreenPAK to ALLON
```

7. Assignment of peripherals

Table 7.1 shows the expected assignment of peripherals to sub core (CM33/CR8) on this example project while all other devices remain assigned to the Linux (main core) side.

Table 7.1 Peripherals assignment on RZ/V Multi-OS Example Project

Peripherals	CPU		Remarks
	Main core (CA55)	Sub core (CM33/CR8)	
GTM	ch2-3	ch0, ch1, ch4-ch7	0002-Updated-GTM-channels-to-support-RTOS.patch (RZ/V2H) 0000-change-ostm-using-for-Linux.patch (RZV2N) 0002-Set-OSTM-for-MCPU-and-RCPU.patch (RZ/V2N)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar 11, 2025	-	First edition.
1.10	May 22, 2025	2	Updated 2. Proven Environment.
1.11	Jun.13.2025	-	Update Multi-OS Package version to 3.1.1.
1.20	Aug.29.2025	-	Update Multi-OS Package version to 3.2.0.
1.30	Jan.13.2026	-	Update Multi-OS Package version to 3.3.0.
4.00	Mar.31.2026	-	Update Multi-OS Package version to 4.0.0
4.10	Jun.30.2026	-	Update Multi-OS Package version to 4.1.0
4.20	Sep.03.2026	-	Update Multi-OS Package version to 4.2.0

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/.

© 2026 Renesas Electronics Corporation. All rights reserved.