RZ/G3S

AWO Example Project Start-up Guide

Introduction

This application note describes how to set up and execute RZ/G3S AWO Example Project.

Target Device

RZ/G3S

Contents

1. Specifications

1.1 Deliverables

Table 1-1. Deliverables of AWO Example Project

Deliverables	File name	Description
RZ/G3S AWO Patch for RZ/G Verified Linux Package (VLP)	• 0000-add-support-AWO-multi-os-pkg.patch • 0002-enable-AWO-on-TFA.patch • 0003-disable-TFA-AWO-M33boot-synchronous.patch	Patch for AWO support on RZ/G VLP v3.0.7
RZ/G3S AWO Example Project	freertos_w_awo_rzg3s_evk_ep.zip	CM33 project file for e2studio
RZ/G3S AWO Example Project Start-up Guide	r01an7396ej0201_rzg3s_awo_example_project_start-up_guide.pdf	This material.

2. Proven Environment

Table 2-1. Proven Environment of AWO Example Project

Item	Contents
Integration Development Environment	e^2 studio 2025-07 or later
RZ/G VLP	v3.0.7 and v4.0.1
RZ/G Flexible Software Package (FSP)	V3.1.0

3. AWO Example Project Setup

3.1 RZ/G FSP Setup

Please refer to Getting Started with RZ/G Flexible Software Package.

3.2 RZ/G VLP Setup

3.2.1 Using RZ/G VLP v3.0.7

This section described how to integrate AWO related stuff to RZ/G VLP v3.0.7.

1. Follow the procedure from the beginning of **2.2 Building Images** to **(3) Add layers** of **SMARC EVK of RZ/G3S Linux Start-up Guide**.

2. Download Multi-OS Feature Package (r01an5869ej0310-rzg-multi-os-pkg.zip) to your working directory and run the commands stated below:

```
$ cd ~/rzg_vlp_<pkg ver>
$ unzip <Multi-OS download dir>/r01an5869ej0310-rzg-multi-os-pkg.zip
$ tar zxvf r01an5869ej0310-rzg-multi-os-pkg/meta-rz-features_multi-os_v3.1.0.tar.gz
```

3. Apply the patch files of Table1-1 as follows:

```
$ cd ~/rzg_vlp_<pkg_ver>
$ patch -p1 < ./r01an5869ej0310-rzg-multi-os-pkg/0000-add-support-AWO-multi-os-pkg.patch
$ patch -p1 < ./r01an5869ej0310-rzg-multi-os-pkg/0002-enable-AWO-on-TFA.patch
$ patch -p1 < ./r01an5869ej0310-rzg-multi-os-pkg/0003-disable-TFA-AWO-M33boot-synchronous.patch
```

Note: 0001-bl2-cm33-coldboot-support.patch must not be applied in this AWO environment.

4. Configure the settings to start executing the CM33 program from CA55.
   Edit the following files with a text editor.

   ```
   ~/rzg_vlp_<pkg_ver>/meta-renesas/meta-rzg3s/recipes-bsp/trusted-firmware-a/trusted-firmware-a.bbappend
   ```

   Then add the following red part to EXTRA_FLAGS_smarc-RZG3S

   ```
   EXTRA_FLAGS_smarc-rzg3s = "BOARD=smarc PLAT_SYSTEM_SUSPEND=awo PLAT_M33_BOOT_SUPPORT=1"
   ```

5. Add the layer for Multi-OS Package

   ```
   $ cd ~/rzg_vlp_<pkg_ver>/build
   $ bitbake-layers add-layer ../meta-rz-features/meta-rz-multi-os/meta-rzg3s-vlp3
   ```

6. Continue to set up VLP by following **(4) - (5) of 2.2 Building images** in **SMARC EVK of RZ/G3S Linux Start-up Guide**.

3.2.2 Using RZ MPU VLP v4.0.1

This section described how to integrate AWO related stuff to RZ MPU VLP v4.0.1.

1. Follow the procedure from the beginning of **2.2 Building Images** to **(3) Add layers** of **SMARC EVK of RZ/G3S Linux Start-up Guide**.

2. Download Multi-OS Feature Package (r01an5869ej0310-rzg-multi-os-pkg.zip) to your working directory and run the commands stated below:

   ```
   $ cd ~/rzg_vlp_<pkg ver>
   $ unzip <Multi-OS download dir>/r01an5869ej0310-rzg-multi-os-pkg.zip
   $ tar zxvf r01an5869ej0310-rzg-multi-os-pkg/meta-rz-features_multi-os_v3.1.0.tar.gz
   ```

3. Uncomment the following line in **meta-rz-features/meta-rz-multi-os/meta-rzg3s-vlp4/conf/layer.conf** for enabling remoteproc support:

   ```
   #MACHINE_FEATURES:append = " RZG3S_CM33_REMOTEPROC"
   #MACHINE_FEATURES:append = " RZG3S_CM33_BOOT_IN_BL2"
   #MACHINE_FEATURES:append = " RZG3S_CM33_COLDBOOT"
   MACHINE_FEATURES:append = " RZG3S_AWO_SUPPORT"
   ```

4. Add the layer for Multi-OS Package

   ```
   $ cd ~/rzg_vlp_<pkg_ver>/build
   $ bitbake-layers add-layer ../meta-rz-features/meta-rz-multi-os/meta-rzg3s-vlp4
   ```

5. Continue to set up VLP by following **(4) - (5) of 2.2 Building images** in **SMARC EVK of RZ/G3S Linux Start-up Guide**.

3.3 How to build AWO Example Project

Here is the procedure to build AWO Example Project:

1. Deploy and boot up Linux by following 3. Preparing the SD Card, 4. Reference Board Setting and 5. Booting and Running Linux of **SMARC EVK of RZ/G3S Linux Start-up Guide**.
2. Extract **freertos_w_awo_rzg3s_evk_ep.zip** on your development PC.
3. Launch e^2 studio and click **File** > **Import**.

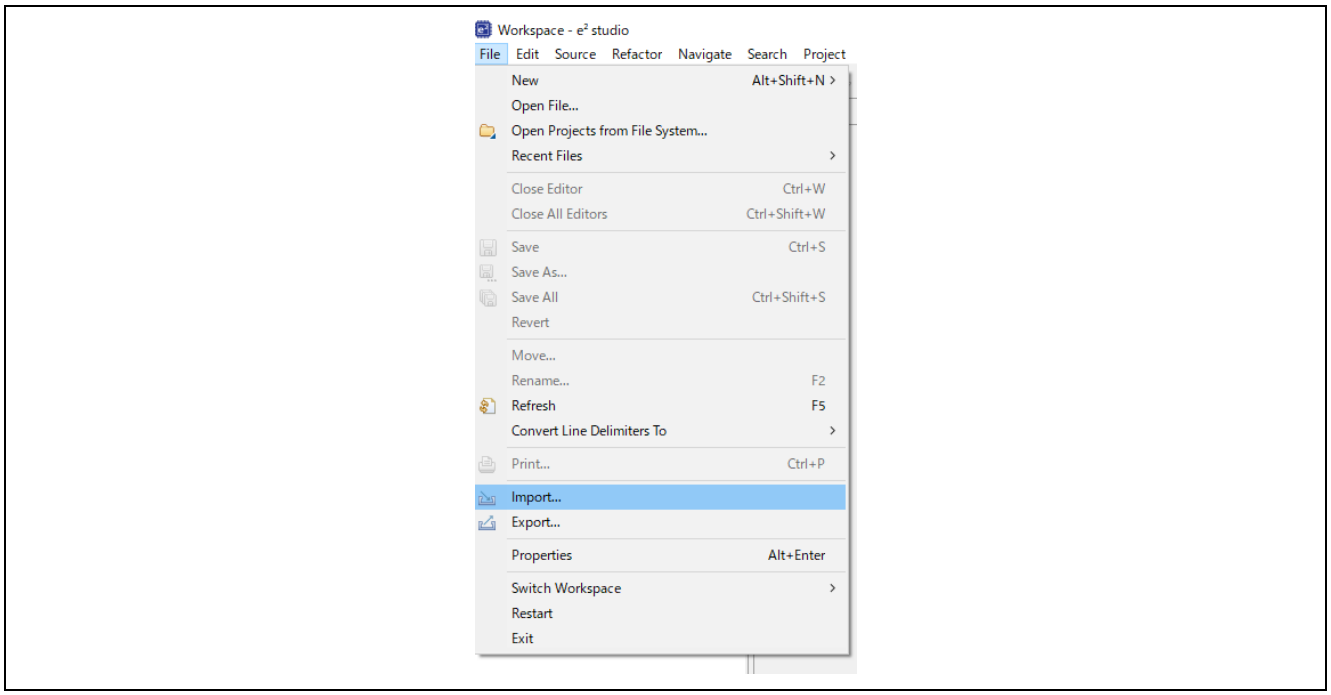

Figure 3.1 Import of CM33 AWO Example Project (1)

4. Select Existing Projects into Workspace and click Next >.

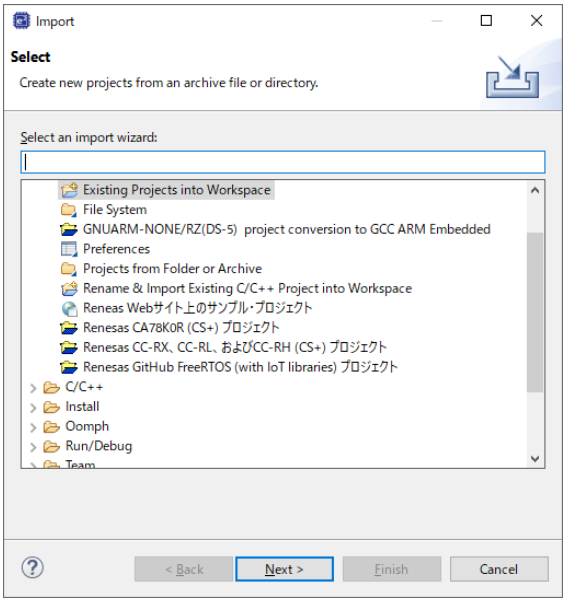

Figure 3.2 Import of CM33 AWO Example Project (2)

5. Input the path to the directory where **freertos_w_awo_rzg3s_evk_ep** project was extracted and click **Finish**.

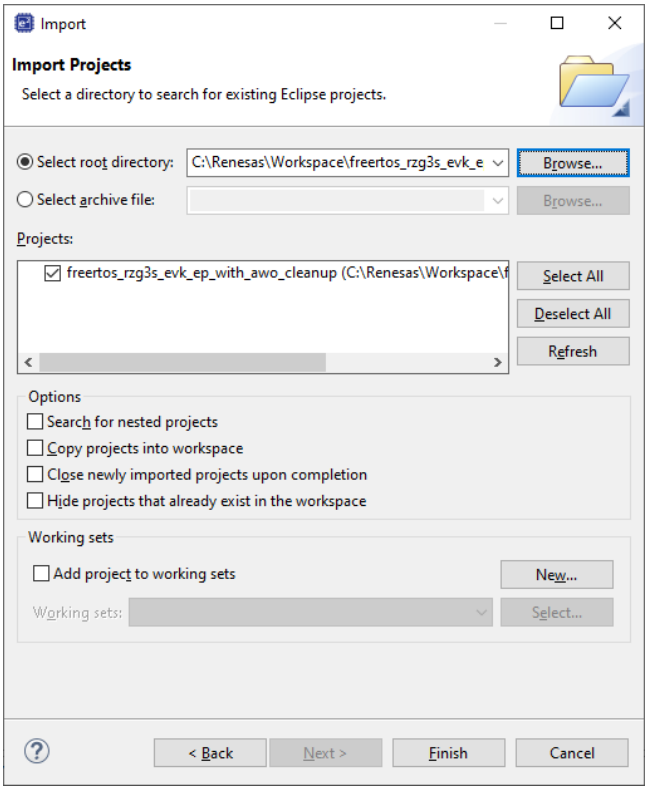

Figure 3.3 Import of CM33 AWO Example Project (3)

6. Build **freertos_w_awo_rzg3s_evk_ep** project from **Project** > **Build All** or **Build Project**.

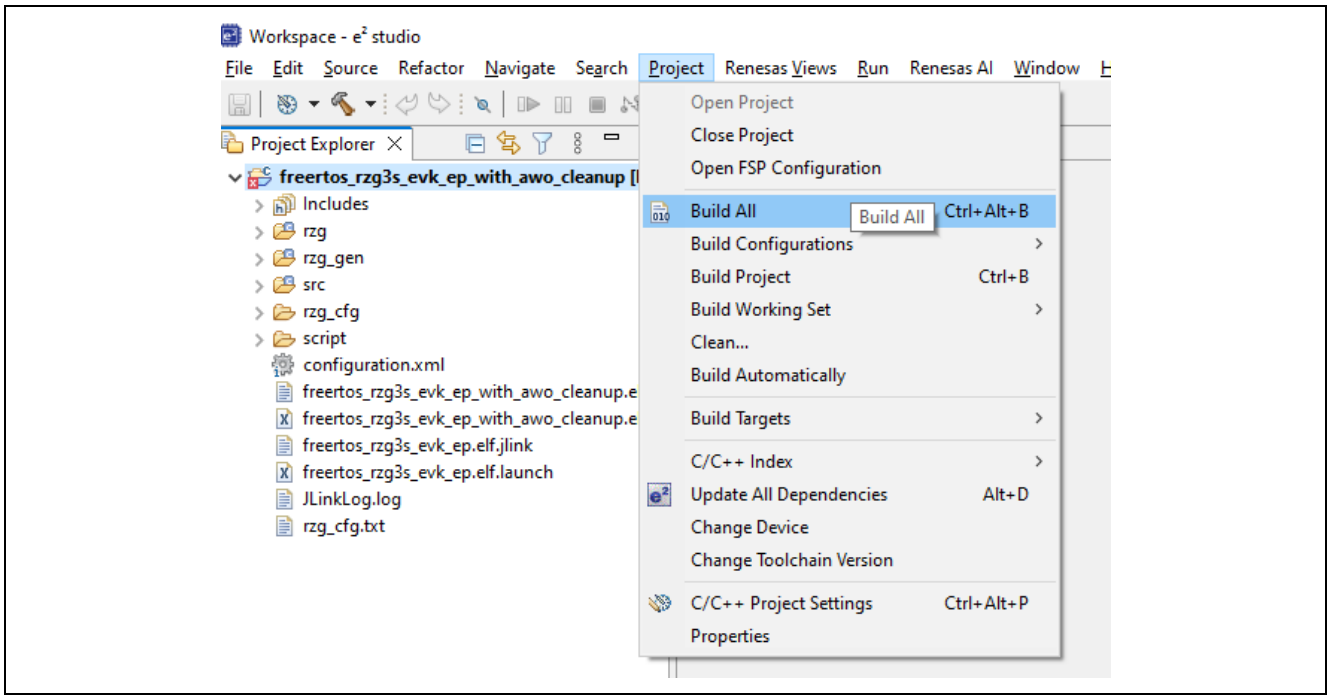

Figure 3.4 Build CM33 AWO Example Project

7. If the build is successfully completed, you can see the build artifact in **Debug** or **Release** directory as stated below:

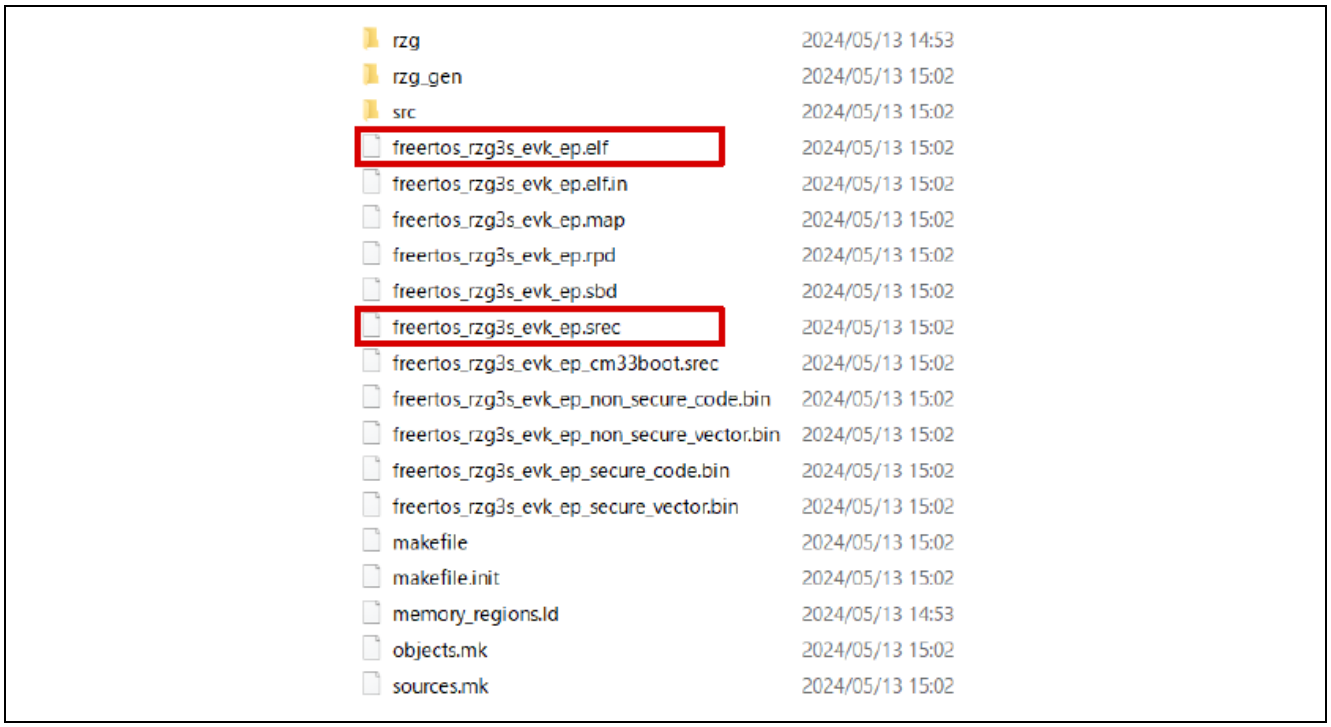

Figure 3.5 Build Artifact of CM33 AWO Example Project

3.4 How to deploy AWO Example Project

This section describes how to deploy the AWO example program built in the previous chapter to the board.

To write to the SPI flash on the board, use the Flash Writer in the RZ/G VLP Linux package.

3.4.1 For QSPI boot (1.8V)

1. Follow the procedure from the beginning of **4 Reference Board Setting** to **4.4 Write the Bootloader of RZ/G3S Linux Start-up Guide**.

2. Program **freertos_w_rzg3s_evk_ep.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'80200000

===== Please Input Qspi Save Address ===
  Please Input : H'200000
please send ! ('.' & CR stop load)
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
======= Qspi  Save Information  =================
SpiFlashMemory Stat Address : H'00200000
SpiFlashMemory End Address  : H'0020D16E
============================================================
```

3. Continue to set up board by following **4.5 Change Back to Normal Boot Mode** in **SMARC EVK of RZ/G3S Linux Start-up Guide**.

3.4.2 For eMMC boot (1.8V)

1. Follow the procedure from the beginning of **8.2 How to boot from eMMC** to **Send the data of "fip-smarc-rzg3s.srec"** in **8.2.1 Writing Bootloader for eMMC Boot**.

2. Program **freertos_w_rzg3s_evk_ep.srec** with Flash Writer as shown below:

```
> EM_W
EM_W Start --------------
-----------------------------------------------------------
Please select,eMMC Partition Area.
 0:User Partition Area    : 62160896 Kbytes
  eMMC Sector Cnt : H'0 - H'0768FFFF
 1:Boot Partition 1       : 32256 Kbytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
 2:Boot Partition 2       : 32256 Kbytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
-----------------------------------------------------------
  Select area(0-2)>1
-- Boot Partition 1 Program ---------------------------
Please Input Start Address in sector :1000
Please Input Program Start Address : 80200000
Work RAM (H'00020000-H'000FFFFF) Clear....
please send ! ('.' & CR stop load)
```

3. Continue to set up board by following **8.2.1 Writing Bootloader for eMMC Boot**.

4. AWO Example Program Invocation

This chapter describes how AWO Example Program works.

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

```
smarc-rzg3s login: root
```

3. Invoke the commands below on Linux console to move Linux to Suspend to RAM (S2R):

```
root@smarc-rzg3s:~# echo deep > /sys/power/mem_sleep
root@smarc-rzg3s:~# echo mem > /sys/power/state
```

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

```
[ 1082.105386] PM: suspend entry (deep)
[ 1082.109183] Filesystems sync: 0.000 seconds
[ 1082.122622] Freezing user space processes ... (elapsed 0.001 seconds) done.
[ 1082.131266] OOM killer disabled.
[ 1082.134496] Freezing remaining freezable tasks ... (elapsed 0.001 seconds) done.
[ 1082.143134] printk: Suspending console(s) (use no_console_suspendto debug)

CM33:Init PMIC for AWO mode
CM33:AWO Mode

Hit any key to go to ALLON mode.
```

RZ/G3S now moves to AWO, and only CM33 AWO Example Project can work. (Note)

Note: On Linux console, Line Feed (LF) should be specified as New Line Code.

5. When typing any key on Linux Console, RZ/G3S starts to move to ALLON, and Linux should be resumed as shown below:

```
Hit any key to go to ALLON mode.
a
CM33:Set GreenPAKto ALLON
NOTICE: BL2: v2.7(release):2.7.0/g3s_1.0.0_rc4
NOTICE: BL2: Built : 11:52:53, Feb 292024
NOTICE: BL2: Booting BL31
[ 60.710450] ehci-platform 11e30100.usb: port 1 resume error -110
[ 60.836140] usbusb2-port1: device 2-1 not suspended yet
[ 60.880506] Disabling non-boot CPUs ...
[ 60.899495] Microchip KSZ9131 Gigabit PHY 11c30000.ethernet-ffffffff:07: attached PHY driver
[Microchip KSZ9131 Gigabit PHY] (mii_bus:phy_addr=11c30000.ethernet-ffffffff:07, irq=137)
[ 60.920867] Microchip KSZ9131 Gigabit PHY 11c40000.ethernet-ffffffff:07: attached PHY driver
[Microchip KSZ9131 Gigabit PHY] (mii_bus:phy_addr=11c40000.ethernet-ffffffff:07, irq=138)
[ 61.017924] usbusb3: root hub lost power or was reset
[ 61.018018] usbusb1: root hub lost power or was reset
[ 61.105918] usbusb4: root hub lost power or was reset
[ 61.106020] usbusb2: root hub lost power or was reset
[ 61.309991] OOM killer enabled.
[ 61.313131] Restarting tasks ...
[ 61.313647] usb2-1: USB disconnect, device number 2
[ 61.334271] done.
[ 61.352163] PM: suspend exit
[ 61.619922] usb2-1: new high-speed USB device number 3 using ehci-platform
[ 61.788714] hub 2-1:1.0: USB hub found
[ 61.798400] hub 2-1:1.0: 4 ports detected
root@smarc-rzg3s:~#
```

5. Sequence Diagram of AWO Example Project

5.1 Brief Sequence of AWO Example Project

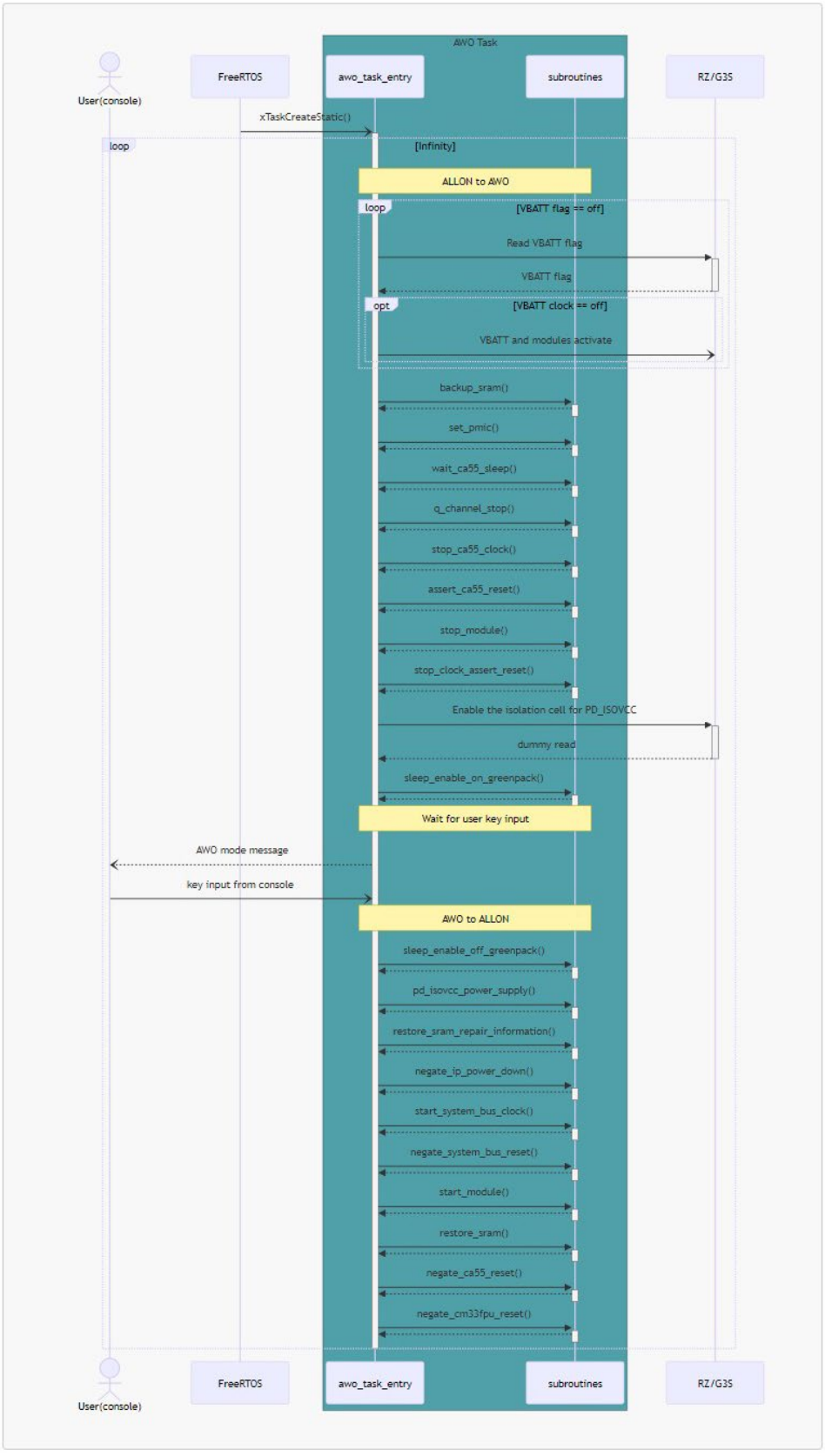

Figure 1. Brief Sequence of AWO Example Project

5.2 Suspend Sequence of TF-A

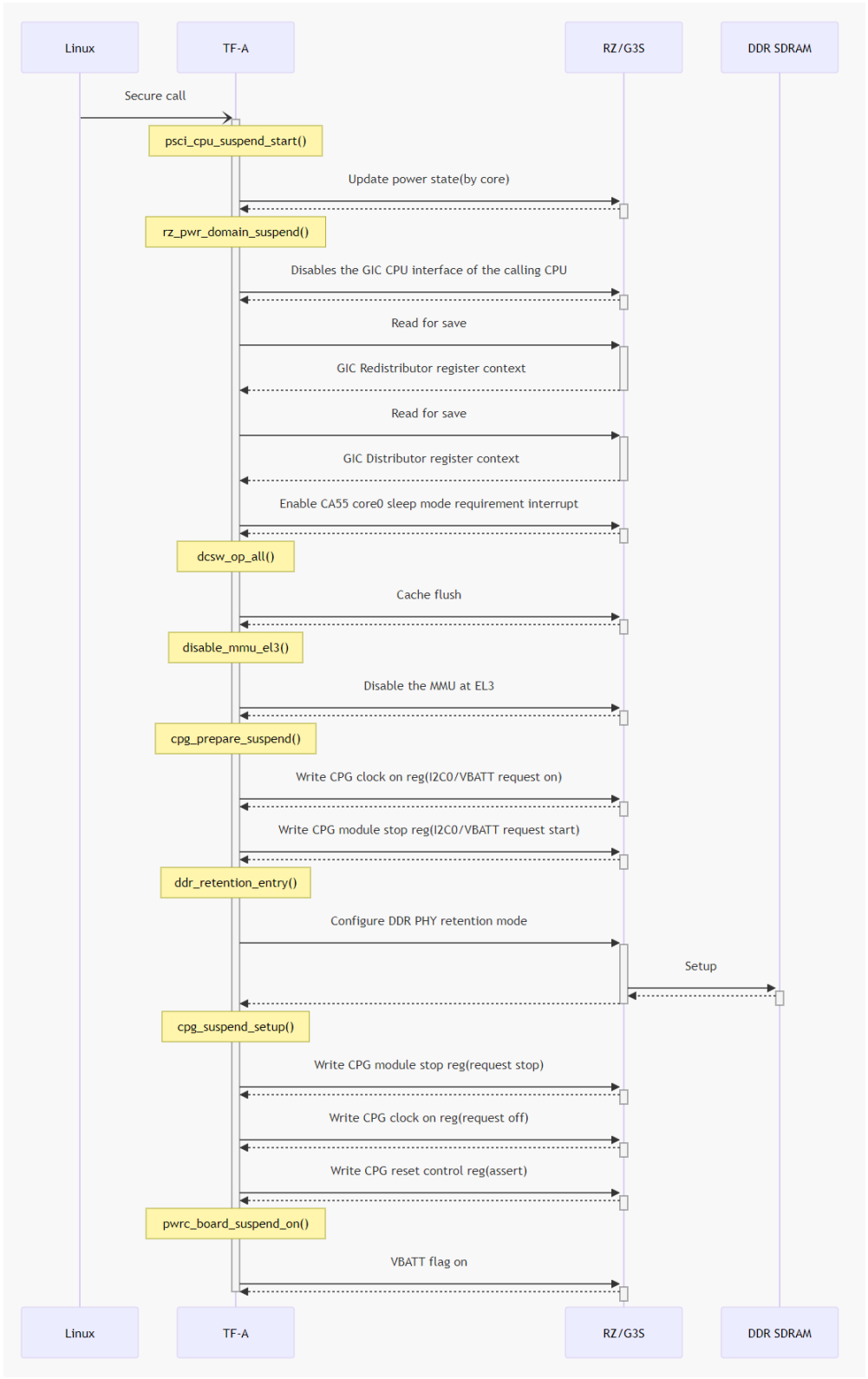

Figure 2. TF-A Suspend Sequence

5.3 Resume Sequence of TF-A

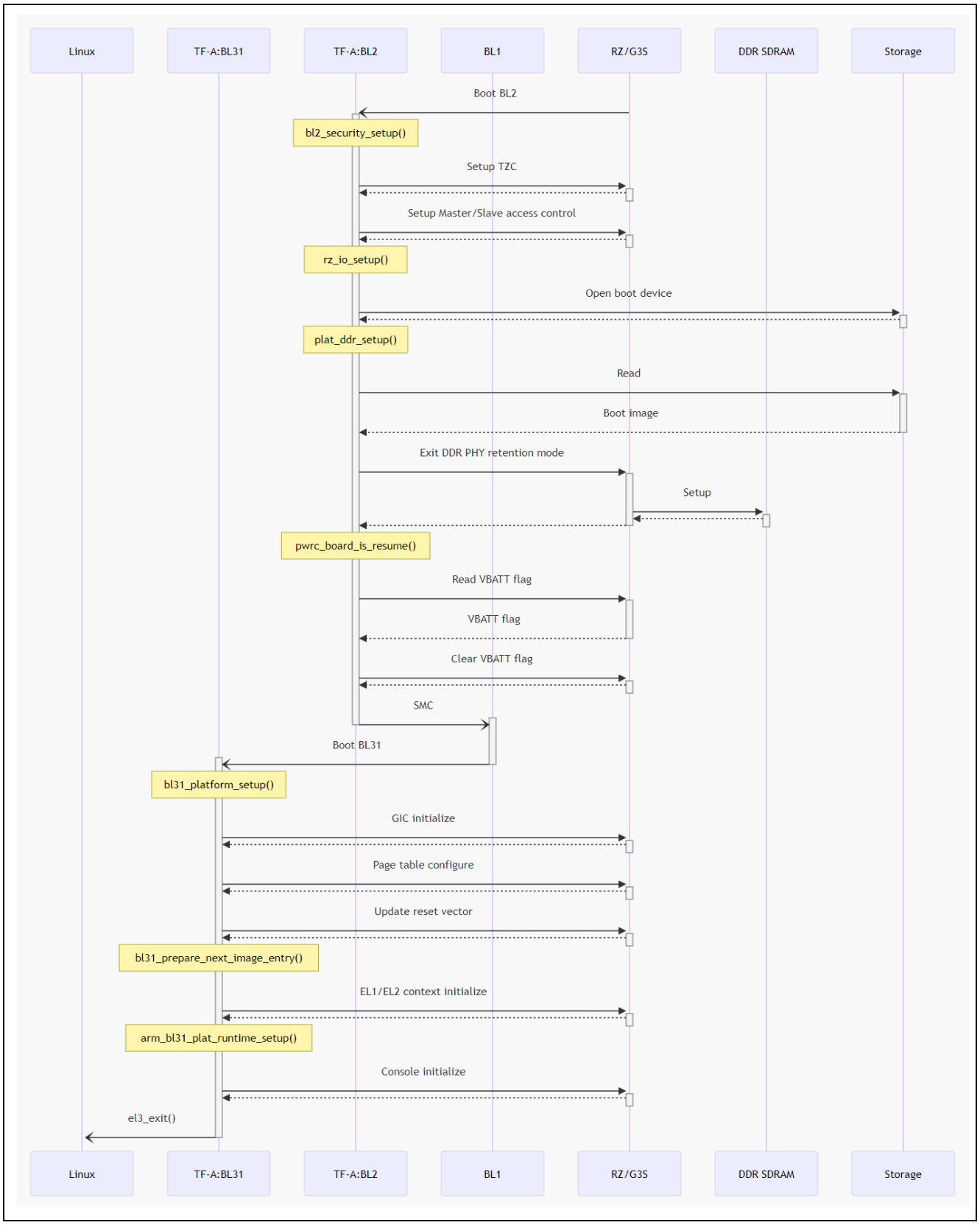

Figure 3. TF-A Suspend Sequence

6. Assignment of peripherals for AWO mode

On RZ/G3S, only the peripherals which belong to PD_VCC domain can continue to work under AWO mode. For details, please refer to 41. Low Power Consumption in RZ/G3S Group User's Manual. Table 5.1 shows the expected assignment of peripherals to CA55 and CM33 on this example project.

Table 5.1 Peripherals assignment on AWO Example Project

Peripherals	CPU		Remarks
	CA55	CM33	
ADC	X		
CANFD		X	
DMA (Non Secure)	X		
DMA (Secure)		X	
GPT	X		
I2C	X		
I3C	X		
MTU3	X		
POE	X		
POEG	X		
SCI	X		
SCIF	ch0, ch2-5	ch1	Assumed use case of SCIF ch1 on CM33 is to get Pmod USBUART to be worked for the console dedicated to CM33.
SPDIF	X		
RSPI	ch1-4	ch0	Assumed use case of RSPI ch0 on CM33 is to get Pmod SF3 to be worked on RZ/G3S Smarc EVK.
SSI	X		
TSU	X		
WDT	ch0	ch1, ch2	
GTM	ch4-7	ch0-3	
xSPI	X		

7. Function Reference of AWO Example Project

7.1.1 awo_task_entry

```
void awo_task_entry (void *pvParameters)
```

- **Parameters**
  - pvParameters
    Pointer to the parameter passed to AWO task.

- **Returns**
  None

- **Description**
  This function is the entry function of AWO task.
  Here is the overview of processing flow:
  1. Set up timer.
  2. Wait until Arm® Cortex®-A55 (hereinafter referred to as CA55) Linux enters Suspend-to-RAM.
  3. Configure RZ/G3S and PMIC as AWO.
  4. Wait until key input to console is issued.
  5. Configure RZ/G3S and PMIC as ALLON.
  6. Return to 2.

7.1.2 backup_sram

```
static void backup_sram(uint32_t *sys_ca55_cfg_rval_back,
                        uint32_t *sys_ca55_cfg_rvah_back)
```

- **Parameters**
  - sys_ca55_cfg_rval_back
    Pointer to the buffer where lower 32-bit of reset vector base address is stored.
  - sys_ca55_cfg_rvah_back
    Pointer to the buffer where upper 32-bit of reset vector base address is stored.

- **Returns**
  None

- **Description**
  This function first copies the contents in SRAM ACPU0 to the dedicated area in SRAM MPU1. Then, CA55 reset vector address should be copied to the buffer specified by the parameters.
  The reset vector base address will be restored when transiting from AWO to ALLON mode for the resume of CA55 Linux.

7.1.3 set_pmic

```
static void set_pmic(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Configure Power Management IC (PMIC) RAA215300A2GNP#HA3 mounted on RZ/G3S SMARC EVK specific to AWO mode.

7.1.4 wait_ca55_sleep

```
static void wait_ca55_sleep(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**

Issue the request to CA55 Linux to transit to sleep state and wait until the transition is completed.

7.1.5 q_channel_stop

```
static void q_channel_stop(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Stop clock supply to peripheral clock domain and bus bridge via Q-Channel.

7.1.6 stop_ca55_clock

```
static void stop_ca55_clock(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Stop the clock supply to CA55.

7.1.7 assert_ca55_reset

```
static void assert_ca55_reset(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Carry out reset assertion of CA55.

7.1.8 stop_module

```
static void stop_module(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Transit all the modules assigned to the registers listed below to Module Stop State:
  - MSTOP Register ACPU (CPG_BUS_ACPU_MSTOP)
  - MSTOP Register PERI_COM (CPG_BUS_PERI_COM_MSTOP)
  - MSTOP Register PERI_DDR (CPG_BUS_PERI_DDR_MSTOP)
  - MSTOP Register TZCDDR (CPG_BUS_TZCDDR_MSTOP)
  - MSTOP Register MHU (CPG_MHU_MSTOP)
  - Power Down MSTOP Register (CPG_PWRDN_MSTOP)

7.1.9 stop_clock_assert_reset

```
static void stop_clock_assert_reset(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Stop the clock supply assigned to the registers listed below:
  - Clock Control Register AXI_ACPU_BUS (CPG_CLKON_AXI_ACPU_BUS)
  - Clock Control Register AXI_COM_BUS (CPG_CLKON_AXI_COM_BUS)
  - Clock Control Register PERI_COM (CPG_CLKON_PERI_COM)
  - Clock Control Register PERI_DDR (CPG_CLKON_PERI_DDR)
  - Clock Control Register AXI_TZCDDR (CPG_CLKON_AXI_TZCDDR)
  - Clock Control Register Cortex-M33 (hereinafter referred to as CM33) (CPG_CLKON_CM33)
  - Power Down IP Register 1 (CPG_PWRDN_IP1)
  - Power Down IP Register 2 (CPG_PWRDN_IP2)

  Configure the unit clock associated with the register below as Power Down mode:
  - Power Down CLKON Register (CPG_PWRDN_CLKON)

  Cofigure the reset pin associated with the register below as Power Down mode:
  - Power Down RST Register (CPG_PWRDN_RST)

  Assert the reset signal associated with the registers listed below:
  - Reset Control Register AXI_ACPU_BUS (CPG_RST_AXI_ACPU_BUS)
  - Reset Control Register AXI_COM_BUS (CPG_RST_AXI_COM_BUS)
  - Reset Control Register PERI_COM (CPG_RST_PERI_COM)
  - Reset Control Register AXI_TZCDDR (CPG_RST_AXI_TZCDDR)

  Turn off the USB Region Power by configuring USB PWRRDY Register (SYS_USB_PWRRDY).
  Assert PCI_ARESETN reset signal with SYS_PCIE_RST_RSM_B and wait until it's actually asserted by monitoring Reset Monitor Register PCI (CPG_RSTMON_PCI register.

7.1.10 sleep_enable_on_greenpack

```
static void sleep_enable_on_greenpack(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Configure Sleep Enable of GreePAK (SLG7RN46131) connected with RIIC ch1 as ON.

7.1.11 sleep_enable_off_greenpack

```
static void sleep_enable_off_greenpack(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Configure Sleep Enable of GreePAK (SLG7RN46131) connected with RIIC ch1 as OFF.

7.1.12 wait_pd_isovcc_stable

```
static void wait_pd_isovcc_stable(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Wait for the power supply for PD_ISOVCC from PMIC. In the current sample program, 5 msec wait is inserted as an example. Then, DDR is turned on.

7.1.13 pd_isovcc_power_supply

```
static void pd_isovcc_power_supply(void)
```

- Parameters
  None

- **Returns**
  None

- **Description**
  Disable the isolation cell for PD_ISOVCC.

7.1.14 restore_sram_repair_information

```
static void restore_sram_repair_information(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Restore SRAM repair information.

7.1.15 negate_ip_power_down

```
static void negate_ip_power_down(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Negate Power Down mode of IPs assigned to PD_ISOVCC region.

7.1.16 start_system_bus_clock

```
static void start_system_bus_clock(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Start the clock supply to System Bus and Peripherals assigned to PD_ISOVCC region.

7.1.17 negate_system_bus_reset

```
static void negate_system_bus_reset(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Negate reset signal of System Bus and Peripherals assigned to PD_ISOVCC region.

7.1.18 start_module

```
static void start_module(void)
```

- **Parameters**
  None

- **Returns**
  None

- **Description**
  Transmit Peripherals assigned to PD_ISOVCC from Module Stop State to Normal Operation state.

7.1.19 restore_sram

```
static void restore_sram(volatile uint32_t *dummy_read)
```

- **Parameters**
  - dummy_read
    Pointer to the buffer where SRAM ACPU0 data to be backed up by backup_sram function is stored.

- **Returns**
  None

- **Description**
  Restore SRAM ACPU0 data that was backed up by backup_sram function. Be sure to call this function after backup_sram is invoked. Otherwise, the data copied to SRAM_ACPU0 should be unpredictable.

7.1.20 negate_ca55_reset

```
static void negate_ca55_reset(uint32_t sys_ca55_cfg_rval_back,
                              uint32_t sys_ca55_cfg_rvah_back)
```

- **Parameters**
  - sys_ca55_cfg_rval_back
    Lower 32-bit of reset vector address to be backed up by backup_sram function.
  - sys_ca55_cfg_rval_back
    Upper 32-bit of reset vector address to be backed up by backup_sram function.

- **Returns**
  None

- **Description**
  Start the clock supply to CA55 and then carry out the reset release of CA55.

8. Appendix

8.1 Debugging CM33 AWO Example Project from e2studio

This chapter describes how to debug an AWO example project from e2studio.

If you want to debug from e2studio, you need to rebuild TF-A. Follow the steps below to recreate the TF-A.

1. If you build RZ/G VLP v3.0.7 with skipping Step 4 of "**3.2.2 Using RZ/G VLP v.3.07**"
   In other words, **trusted-firmware-a.bbappend** will be built in the following state:

```
EXTRA_FLAGS_smarc-rzg3s = "BOARD=smarc PLAT_SYSTEM_SUSPEND=awo"
```

   Note: For RZ/G VLP v.4.0.1, skip this step because it will automatically applied.

2. Write the rebuilt TF-A to the board according to **3.4 How to deploy AWO Example Project.**

Once you have completed the above preparations, open the AWO project in e2studio and start debugging by following the steps:

1. Configure the debugger for **freertos_w_awo_rzg3s_evk_ep** project from **Run** > **Debug Configurations...**.

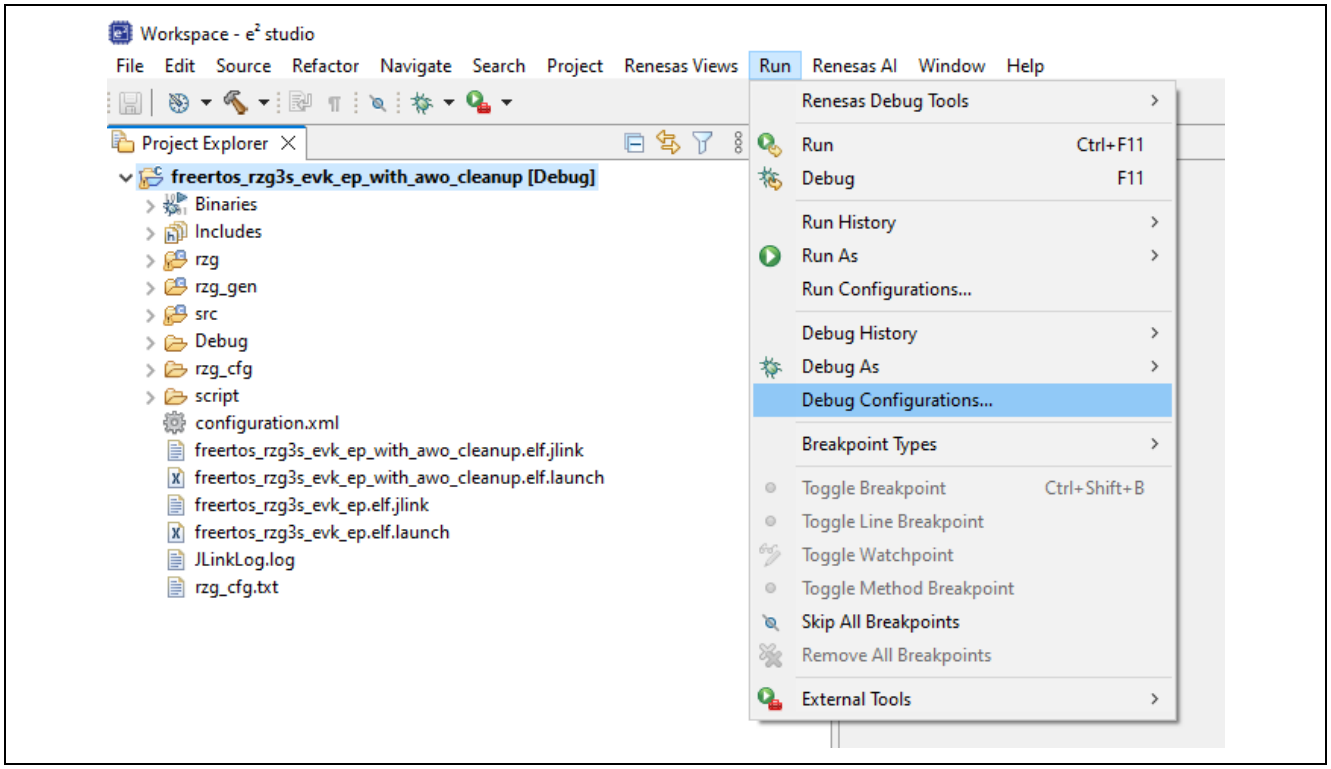

Figure 8.1 Debug Configuration of CM33 AWO Example Project (1)

2. Extract **Renesas DGB Hardware Debugging**, choose **freertos_w_awo_rzg3s_evk_ep.elf** and click **Debug**.

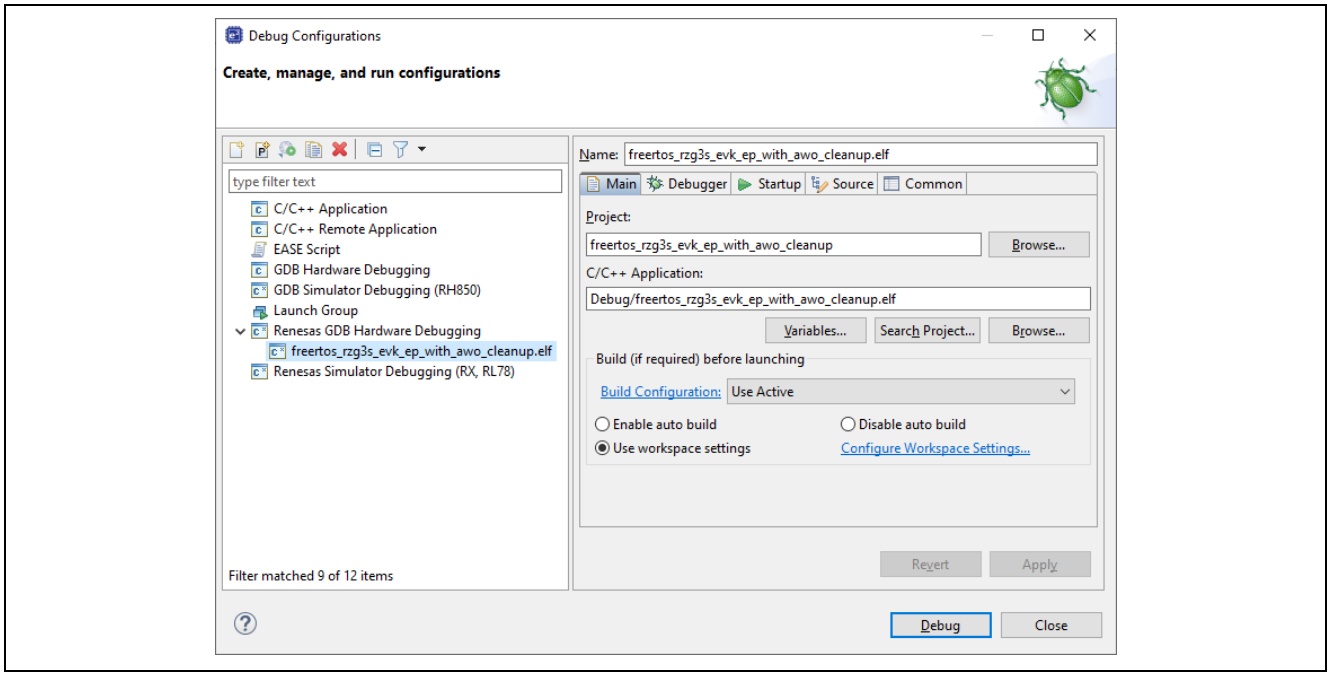

Figure 8.2 Debug Configuration of CM33 AWO Example Project (2)

3. Now the load module of CM33 AWO Example Project has been loaded and Program Counter (PC) should indicate the top of Warm_Reset_S function. Then, click **Run** > **Run** for continuing the invocation.

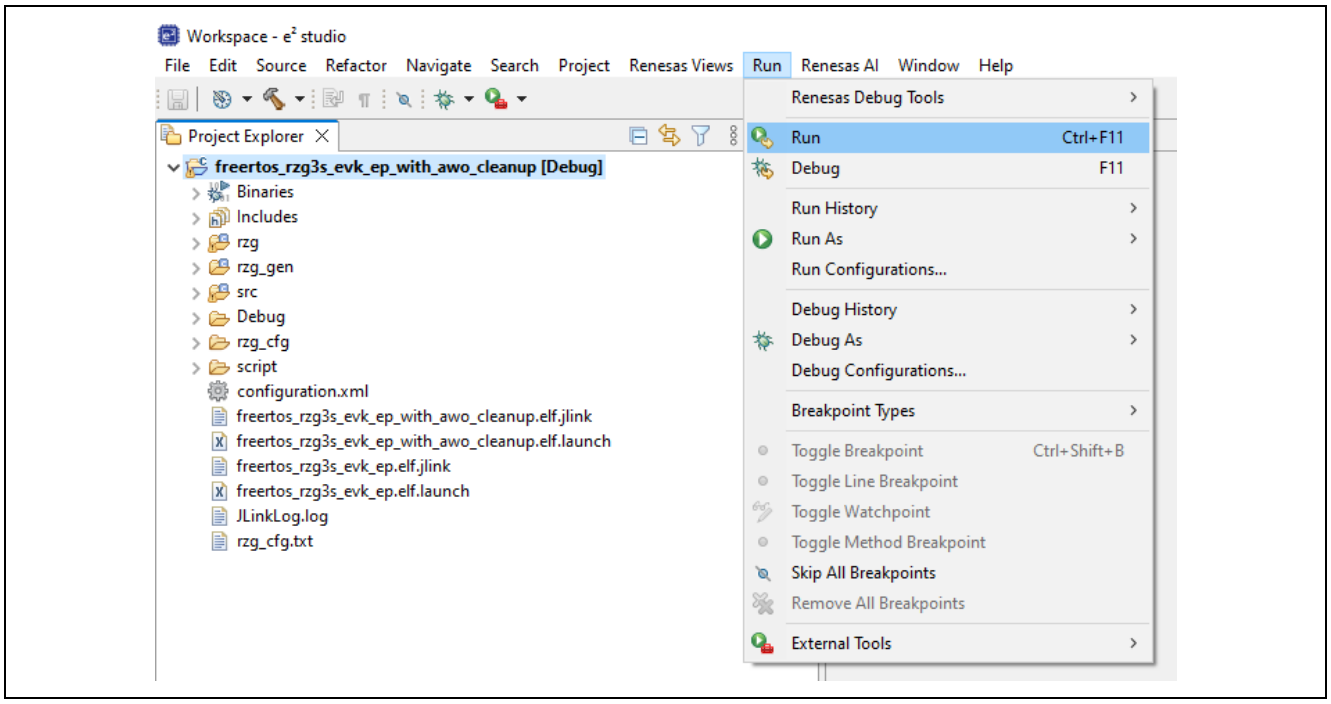

Figure 8.3 Run CM33 AWO Example Project (1)

4. Program should stop at the top of main function. Click Run > Run again to continue. Now, CM33 AWO Example Program should be working.

5. Log in to Linux as **root** user.

8.2 Restrictions

This AWO Example environment has the following restrictions regarding the boot mode.

- CM33 coldboot is not supported. Use this environment with CA55 coldboot.
- eSD boot is not supported.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.31.24	-	1st revision issued.
1.01	Sep.30.24	3	Updated the deliverables stated in Table 1-1.
		3	Updated 3.2 RZ/G VLP Setup in accordance with the update of deliverables.
1.02	Apr.25.25	3	Updated the deliverables stated in Table 1-1.
		3-10	Added a process to apply RZ/G Multi-OS Package to the setup procedure.
		26	Added boot mode restriction.
2.00	Jul.22.25	3	Updated the deliverables stated in Table 1-1.
		3	Updated 3.2 RZ/G VLP Setup in accordance with the update of deliverables.
		7-8	Updated the address specified for the SerialFlashWriter to match the latest version.
2.01	Dec.26.25	4	Added instructions for running multiOS with VLP v4.0.1.

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 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/.