

Linux Interface Specification Kernel Core

User's Manual: Software

RZ/G2L Group, RZ/V2L Group, RZ/V2N Group,
RZ/G3E Group and RZ/Five

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

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. 1.12 Jul 22, 2025)

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/.

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.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MPU. It is intended for users designing application systems incorporating the MPU.. It is intended for users developing software incorporating the processors. A basic knowledge of software development and Linux systems is necessary in order to use this document.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RZ/G2L Group, RZ/V2L Group, RZ/V2N Group, RZ/G3E Group and RZ/Five Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RZ/G2L Group User's Manual: Hardware	---
		RZ/V2L Group User's Manual: Hardware	---
		RZ/V2N Group User's Manual: Hardware	---
		RZ/Five Group User's Manual: Hardware	---
		RZ/G3E Group User's Manual: Hardware	---
User's manual for Software	Software specifications (basic function of Linux kernel, memory maps, interrupt, clock, pins mux, device tree)	Linux interface Specification - Kernel Core	This User's manual
Application Note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Electronics Web site.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

3. Register Notation

4. List of Abbreviations and Acronyms

Abbreviation	Full Form
CPG	<u>C</u> lock <u>P</u> ulse <u>G</u> enerator
MSSR	<u>M</u> odule <u>S</u> tandby, <u>S</u> oftware <u>R</u> eset
PFC	<u>P</u> in <u>F</u> unction <u>C</u> ontroller
GIC	<u>G</u> eneric <u>I</u> nterrupt <u>C</u> ontroller
Runtime PM	<u>R</u> untime <u>P</u> ower <u>M</u> anagement
PSCI	<u>P</u> ower <u>S</u> tate <u>C</u> oordinate <u>I</u> nterface
APMU	<u>A</u> dvanced <u>P</u> ower <u>M</u> anagement <u>U</u> nit for AP-System Core
CA55	<u>C</u> ortex A55 for AP-System Core
SPI	<u>S</u> hared <u>P</u> rocessor <u>I</u> nterrupts
PPI	<u>P</u> er <u>P</u> rocessor <u>I</u> nterrupts

Table of Contents

1. Overview.....	1
1.1 Overview	1
1.2 Function	1
1.3 Reference	2
1.3.1 Standard.....	2
1.3.2 Related documents	2
1.4 Restrictions	2
2. Terminology.....	3
3. Operating Environment.....	4
3.1 Hardware Environment	4
3.2 Module Configuration.....	5
4. Memory management.....	6
4.1 Physical memory.....	6
4.2 Memory map	6
5. Detail	10
6. External Interface.....	11
6.1 Device Tree.....	12
6.2 Interrupt Control	14
6.2.1 Interface specification	14
6.2.2 Definitions.....	14
6.2.2.1 Definitions of the interrupt	14
6.2.2.2 Get definitions of the interrupt	15
6.3 Pin Control	16
6.3.1 Interface specification	16
6.3.2 Definitions.....	16
6.4 Clock Control	16
6.4.1 Interface specification	16
6.4.2 Definitions.....	16
6.4.2.1 Definitions of the clock	16
6.4.2.2 Get definitions of the clock	19
6.5 Reset Control	20
6.5.1 Interface specification	20
6.5.2 Definitions.....	20
6.5.2.1 Definitions of the reset controller	20
6.5.2.2 Get definitions of the reset controller	21

1. Overview

1.1 Overview

This manual explains the part that added for the RZ/G2L Group, RZ/V2L Group, RZ/V2N, RZ/G3E and RZ/Five. Detailed explanation is skipped because the interface of this module is based on Linux.

Note: Currently, this device is supported in two kernel versions v5.10 and v6.1 with the information below:

- v5.10: RZ/G2L Group, RZ/V2L group and RZ/Five.
- v6.1: RZ/G2L, RZ/G2LC, RZ/V2N and RZ/G3E.

1.2 Function

This kernel supports the following functions.

- Single Processing / SMP
- I-Cache / D-Cache / L3 Cache
- Clock Pulse Generator (CPG)
- Pin Function Controller (PFC)
- Kernel System Timer
 - ARM architected timer: RZ/G2L Group, RZ/V2L, RZ/V2N and RZ/G3E.
 - RISC-V architected timer: RZ/Five.
- Device Tree
 - RZ/G2L: r9a07g044l2-smarc.dts
 - RZ/G2LC: r9a07g044c2-smarc.dts
 - RZ/G2UL: r9a07g043u11-smarc.dts
 - RZ/V2L: r9a07g054l2-smarc.dts
 - RZ/V2N: r9a09g056n44-evk.dts
 - RZ/Five: r9a07g043f01-smarc.dts
 - RZ/G3E: r9a09g047e57-smarc.dts

1.3 Reference

1.3.1 Standard

There is no reference document on standards.

1.3.2 Related documents

There is no document related to this kernel.

1.4 Restrictions

None.

2. Terminology

The following table shows the terminology related to this kernel.

Table 2-1 Terminology

Terms	Explanation
CPG	<u>C</u> lock <u>P</u> ulse <u>G</u> enerator
MSSR	<u>M</u> odule <u>S</u> tandby, <u>S</u> oftware <u>R</u> eset
PFC	<u>P</u> in <u>F</u> unction <u>C</u> ontroller
GIC	<u>G</u> eneric <u>I</u> nterrupt <u>C</u> ontroller
Runtime PM	<u>R</u> untime <u>P</u> ower <u>M</u> anagement
PSCI	<u>P</u> ower <u>S</u> tate <u>C</u> oordinate <u>I</u> nterface
APMU	<u>A</u> dvanced <u>P</u> ower <u>M</u> anagement <u>U</u> nit for AP-System Core
CA55	<u>C</u> ortex <u>A55</u> for AP-System Core
SPI	<u>S</u> hared <u>P</u> rocessor <u>I</u> nterrupts
PPI	<u>P</u> er <u>P</u> rocessor <u>I</u> nterrupts

3. Operating Environment

3.1 Hardware Environment

The following table shows the hardware needed to use this kernel.

Table 3-1 Hardware environment (RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L and RZ/Five)

Name	Version	Manufacturer
RZG2L Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZG2LC Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZG2UL Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZV2L Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZ/Five Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH

Table 3-2 Hardware environment (RZ/V2N, RZ/G3E)

Name	Product number
RZ/V2N Evaluation Board Kit	RTK0EF0186C03000BJ
RZ/G3E SMARC Evaluation Board Kit	RTK9947E57S01000BE

3.2 Module Configuration

The following figure shows the configuration of the Interrupt / Clock.

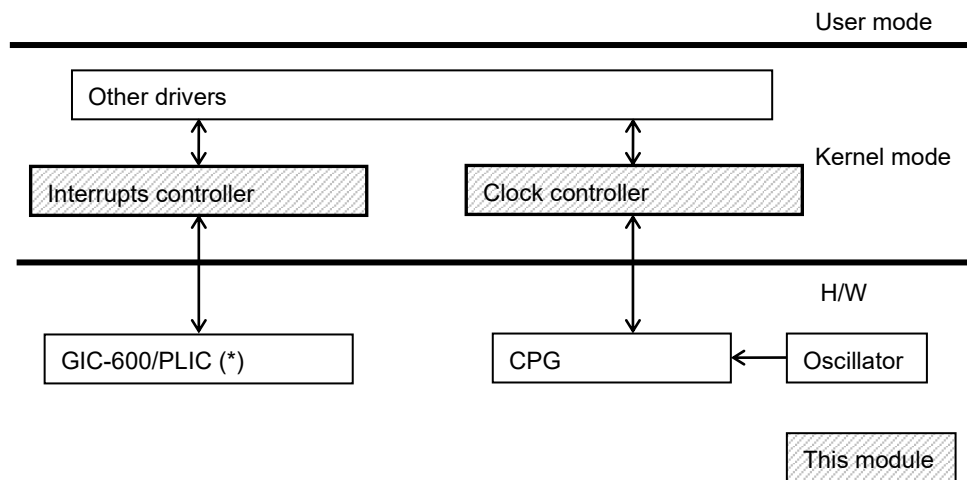


Figure 3-1 Interrupt / Clock Configuration

- (*) GIC-600 is Interrupt controller of RZ/G2L Series, RZ/V2L, RZ/V2N and RZ/G3E.
 PLIC is Interrupt controller of RZ/Five.

4. Memory management

4.1 Physical memory

This kernel manages the following memory.

Table 4-1 Physical memory (RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, RZ/V2N, and RZ/Five Evaluation Board Kit)

Kind of memory	size	Physical address
Main memory (RZ/G2L, RZ/V2L)	1922MB	0x00-48000000 ~ 0x00-BFFFFFFF
Main memory (RZ/G2LC)	896MB	0x00-48000000 ~ 0x00-7FFFFFFF
Main memory (RZ/G2UL)	896MB	0x00-48000000 ~ 0x00-7FFFFFFF
Main memory (RZ/V2N)	8064MB	0x00-48000000 ~ 0x02-40000000
Main memory (RZ/Five)	896MB	0x00-48000000 ~ 0x00-7FFFFFFF
Main memory (RZ/G3E)	3968MB	0x00-48000000 ~ 0x01-40000000

4.2 Memory map

Note:

- Kernel uses 4KB page size (VA_BITS=48) and 4 levels of translation tables.

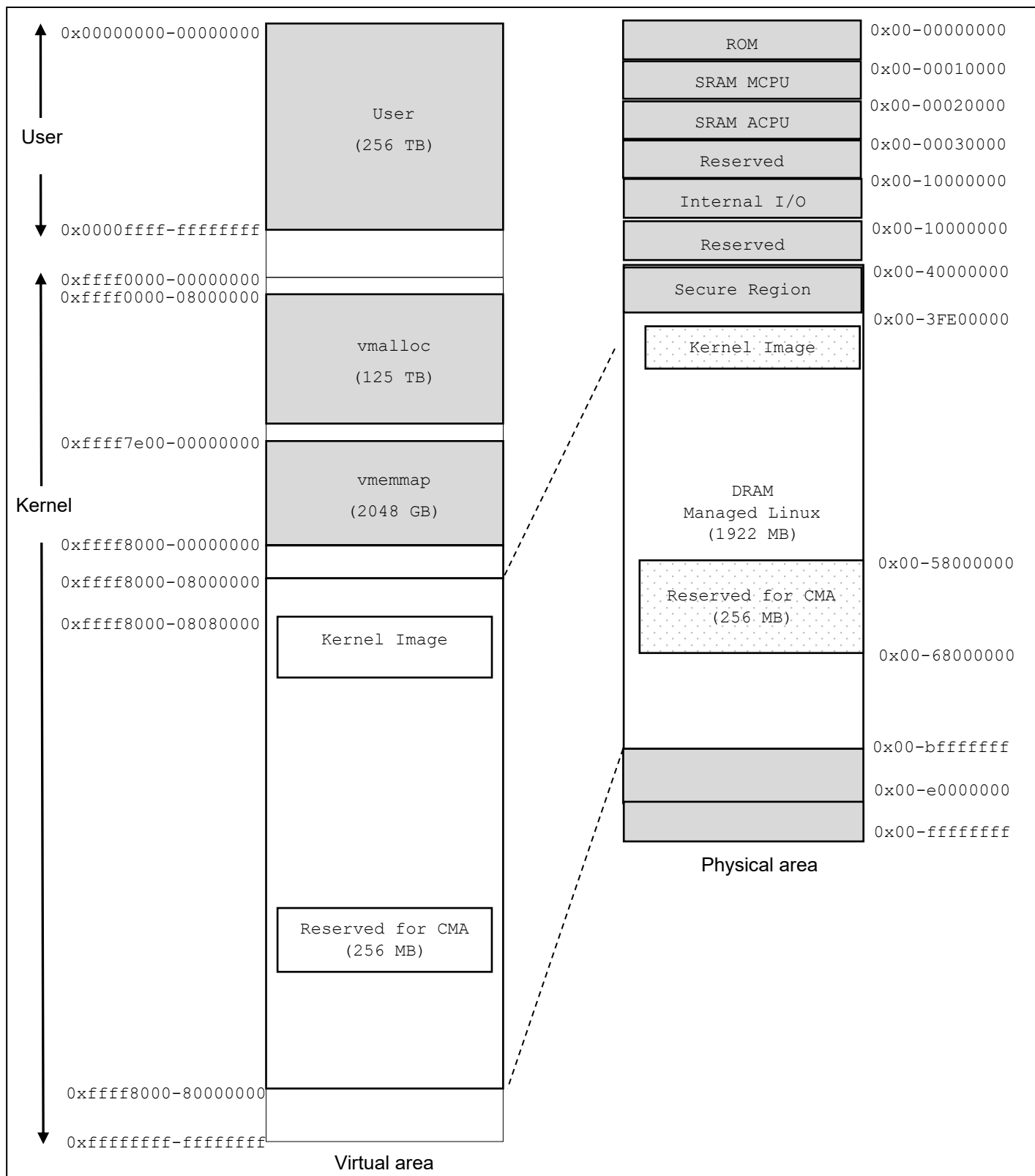


Figure 4-1 Memory map of kernel (RZ/G2L and RZ/V2L Evaluation Board Kit)

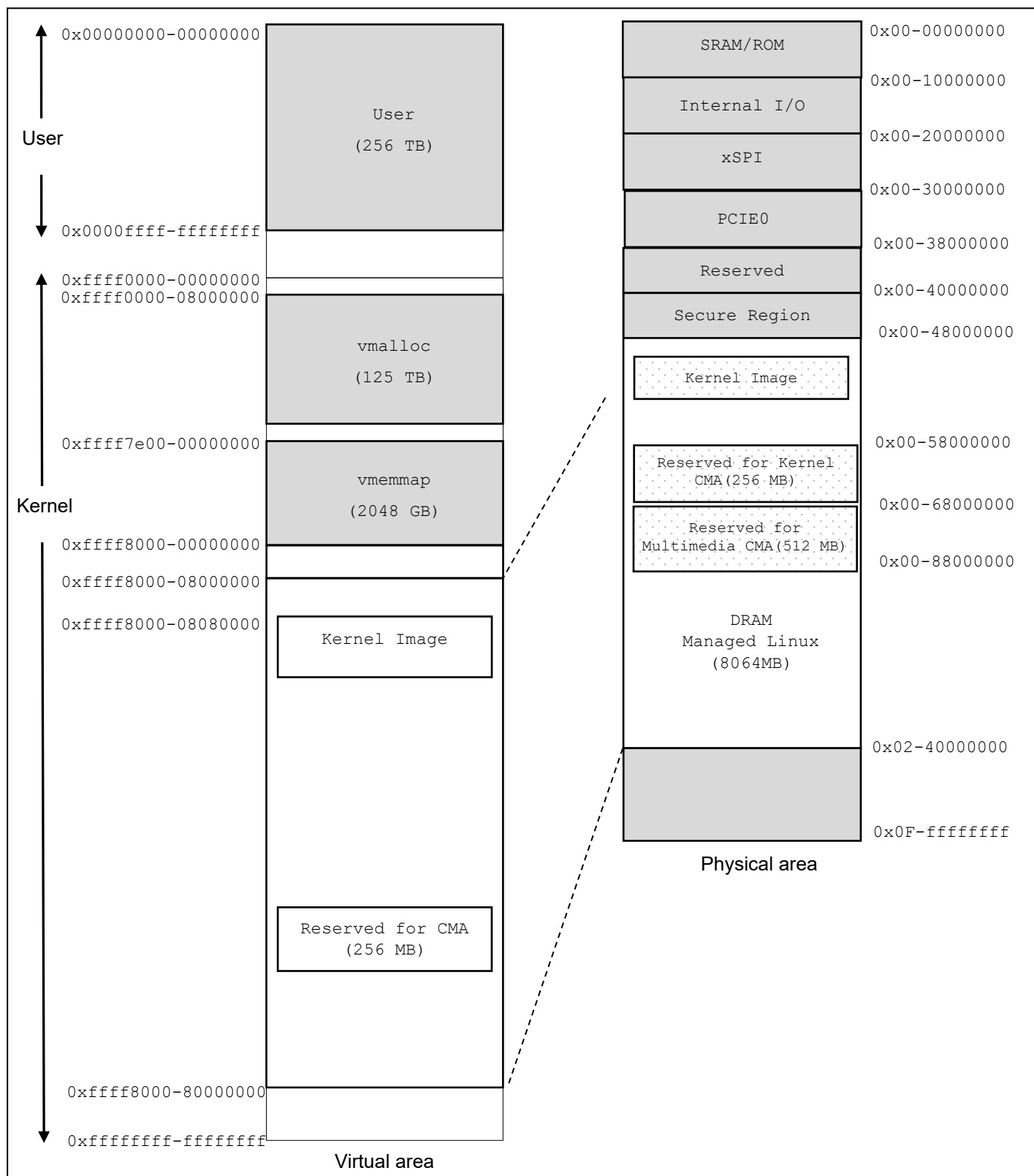


Figure 4-2. Memory map of kernel (RZ/V2N Evaluation Board Kit)

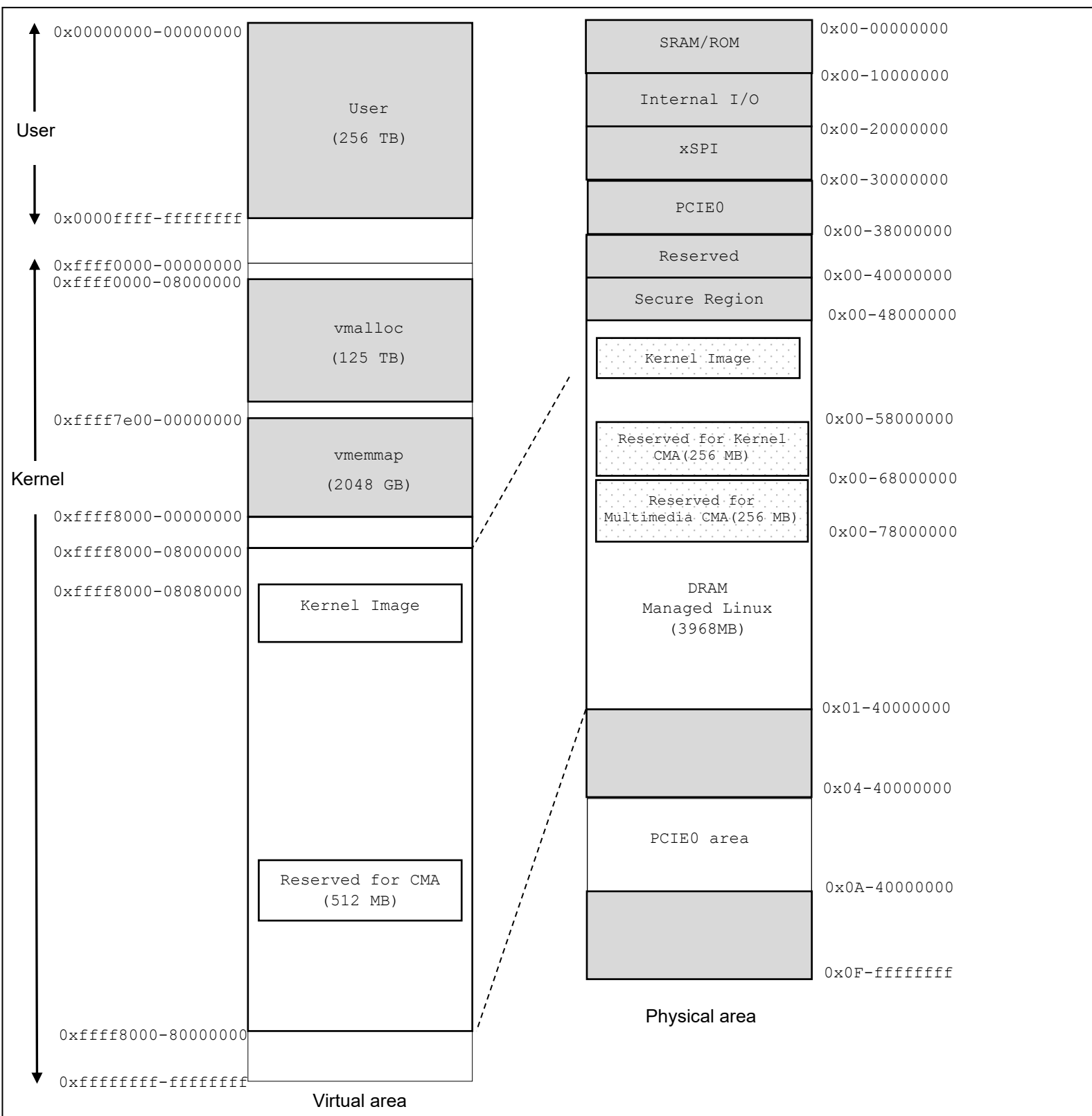


Figure 4-3. Memory map of kernel (RZ/G3E Evaluation Board)

5. Detail

This kernel controls the following devices on the RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, RZ/V2N and RZ/G3E System Evaluation Board.

- DDR4 -1600 2Gbyte (RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five Evaluation Board Kit)
- LPDDR4/LPDDR4X-3200 8Gbyte (RZ/V2N Evaluation Board Kit, RZ/G3E Evaluation Board Kit)

This kernel controls the following modules in RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, RZ/V2N and RZ/G3E.

- ARM Cortex-A55 MPCore 2 CPUs for RZ/G2L, RZ/G2LC, RZ/G2UL and RZ/V2L.
- ARM Cortex-A55 MPCore 4 CPUs for RZ/V2N, RZ/G3E.
- L3 Cache
- Generic Counter
- CPG (Clock Pulse Generator)
- GIC-600 (Interrupt Controller)
- PFC (Pin Function Controller)
- Serial Communication Interface

This kernel controls the following modules in RZ/Five

- RISC-V AndesCore™ AX45MP 64-bit CPU
- Level-1 and level-2 cache controllers with 64-byte cache line size
- CPG (Clock Pulse Generator)
- PLIC (Platform-Level Interrupt Controller)
- PFC (Pin Function Controller)
- Serial Communication Interface

6.External Interface

The functions provided by this kernel explain with reference to the following description format.

[Overview]	Presents an overview of a function.
[Function Name]	Explains the name of the function.
[Calling format]	Explains the format for calling the function.
[Argument]	Explains the argument(s) of the function.
[Return value]	Explains the return value(s) of the function.
[Feature]	Explains the features of the function.
[Remark]	Explains points to be noted when using the function.

6.1 Device Tree

The device tree is a data structure that describes hardware information.

By separating hardware-specific properties such as base address, interrupt number, pin control, and clock from the device driver to the device tree, the reusability of the device driver can be improved. Also, it is possible to use one Linux kernel image with multiple hardware.

For detailed information on Device Tree, refer to the document (Documentation/devicetree/) included in the Linux kernel source code, "Device Tree Specification" published on devicetree.org.

To customize the device driver supported by RZ/G2L, RZ/V2L, RZ/V2N and RZ/G3E Linux BSP using the device tree, refer to the user's manual of each device driver.

Refer to the Table 6-1 for Device Tree List, the defined Processor, Board, Memory Size, and Included Files dependencies.

Table 6-1 Device Tree List

Processor	LSI Version	Board	Board Revision	Memory Size	Device Tree	Included Files
RZ/G2L	1.0	RZ/G2L Evaluation Board Kit	SMARC module: v01 Carrier board: v03	2GB	<ul style="list-style-type: none"> r9a07g044l2-smarc.dts 	<ul style="list-style-type: none"> r9a07g044l2.dtsi rzg2l-smarc-som.dtsi rzg2l-smarc.dtsi
RZ/G2LC	1.0	RZ/G2LC Evaluation Board Kit	SMARC module: v01 Carrier board: v03	2GB	<ul style="list-style-type: none"> r9a07g044c2-smarc.dts 	<ul style="list-style-type: none"> r9a07g044c2.dtsi rzg2lc-smarc-som.dtsi rzg2lc-smarc.dtsi
RZ/G2UL	1.0	RZ/G2UL Evaluation Board Kit	SMARC module: v01 Carrier board: v03	1GB	<ul style="list-style-type: none"> r9a07g043u11-smarc.dts 	<ul style="list-style-type: none"> r9a07g043u.dtsi rzg2ul-smarc-som.dtsi rzg2ul-smarc.dtsi
RZ/V2L	1.0	RZ/V2L Evaluation Board Kit	SMARC module: v01 Carrier board: v03	2GB	<ul style="list-style-type: none"> r9a07g054l2-smarc.dts 	<ul style="list-style-type: none"> r9a07g054l2.dtsi rzg2l-smarc-som.dtsi rzg2l-smarc.dtsi
RZ/V2N	1.0	RZ/V2N Evaluation Kit	-	8GB	<ul style="list-style-type: none"> r9a09g056n44-evk.dts 	<ul style="list-style-type: none"> r9a09g056.dtsi
RZ/Five	1.0	RZ/Five Evaluation Board Kit	SMARC module: v01 Carrier board: v03	1GB	<ul style="list-style-type: none"> r9a07g043f01-smarc.dts 	<ul style="list-style-type: none"> r9a07g043f.dtsi rzfive-smarc-som.dtsi rzfive-smarc.dtsi
RZ/G3E	1.0	RZ/G3E SMARC Evaluation Board Kit	-	4GB	<ul style="list-style-type: none"> r9a09g047e57-smarc.dts 	<ul style="list-style-type: none"> r9a09g047.dtsi rzg3e-smarc-som.dtsi rzg3e-smarc.dtsi rzg3e-smarc-common.dtsi rzg3e-smarc-lvds.dtsi rzg3e-smarc-rgb.dtsi

Structure of device tree files for all platforms:

- **Carrier board + Module Board:**
 - RZ/G2L: r9a07g044l2-smarc.dts
 - RZ/ V2L: r9a07g054l2-smarc.dts
 - RZ/V2N: r9a09g056n44-evk.dts
 - RZ/G2LC: r9a07g044c2-smarc.dts
 - RZ/G2UL: r9a07g043u11-smarc.dts
 - RZ/Five: r9a07g043f01-smarc.dts.
 - RZ/G3E: r9a09g047e57-smarc.dts

6.2 Interrupt Control

6.2.1 Interface specification

Detailed explanation is skipped because the interrupt control functions such as the `request_irq` is based on Linux.

6.2.2 Definitions

This section explains the description form about interrupt.

6.2.2.1 Definitions of the interrupt

Definitions of interrupts are described in device tree. An example of device tree is as follows.

(1) Using the interrupt definitions

```
#include <dt-bindings/interrupt-controller/arm-gic.h>

i2c0: i2c@10058000 {
    interrupts = <GIC_SPI 350 IRQ_TYPE_LEVEL_HIGH>;
};
```

Figure 6-1 Example of device tree for interrupt definition

```
#include <dt-bindings/interrupt-controller/irq.h>

i2c0: i2c@10058000 {
    interrupts = <SOC_PERIPHERAL_IRQ(350) IRQ_TYPE_LEVEL_HIGH>;
};
```

Figure 6-2 Example of device tree for interrupt definition in RZ/Five

The format of a "interrupts" property is as follows.

- The 1st cell is a flag which indicates the type of interrupts.
0 (GIC_SPI) for SPI interrupts, 1 (GIC_PPI) for PPI interrupts.
- The 2nd cell contains the interrupt number for the interrupt type.
SPI interrupts are in the range [0-479]
PPI interrupts are in the range [0-15]
- The 3rd cell is the flags, encoded as follows:
1 (IRQ_TYPE_EDGE_RISING) = low-to-high edge triggered
4 (IRQ_TYPE_LEVEL_HIGH) = active high level-sensitive

6.2.2.2 Get definitions of the interrupt

The example which gets the definitions of the interrupt from a device tree is as follows.

```
static int riic_i2c_probe(struct platform_device *pdev) {  
    ...  
    res = platform_get_resource(pdev, IORESOURCE_IRQ, riic_irqs[i].res_num);  
    if (!res)  
        return -ENODEV;  
  
    ret = devm_request_irq(dev, res->start, riic_irqs[i].isr,  
                           0, riic_irqs[i].name, riic);  
  
    ...  
};
```

Figure 6-3 Example of getting interrupt from device tree

6.3 Pin Control

6.3.1 Interface specification

Detailed explanation is skipped because the pin control functions such as the `pinctrl_gpio_request` is based on Linux.

6.3.2 Definitions

Please refer "Chapter 5.4.3. Pin Control definition" of GPIO User Manual (r01us0488ej011x-rz-g). It described detail how to configure correct setting for pin control.

6.4 Clock Control

6.4.1 Interface specification

Detailed explanation is skipped because the clock control functions such as the `devm_clk_get` is based on Linux.

6.4.2 Definitions

6.4.2.1 Definitions of the clock

Definitions of the clock are described in device tree. Example of device tree setting is as following.

(1) Using the CPG core clock definitions

```
cpu0: cpu@0 {
    clocks = <&cpg CPG_CORE R9A07G044_CLK_I>;
};
```

Figure 6-4 Example of device tree for CPG core definitions

The format of a "clocks" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

The 2nd cell must be set to CPG_CORE.

The 3rd cell contains the identifier of CPG/MSSR driver.

Above example defines R9A07G044_CLK_I for I ϕ .

Valid values for identifier of the CPG/MSSR driver can be found in the struct `cpg_core_clk` data arrays corresponding to the SoC.

```
static const struct {
    struct cpg_core_clk common[56];
} core_clks __initconst = {
    ...
    /* Core output clk*/
    DEF_DIV("I", R9A07G044_CLK_I, CLK_PLL1, DIVPL1A, dtable_1_8)
    ...
};
```

Figure 6-5 struct `cpg_core_clk` data arrays (Refer to drivers/clock/renesas/ r9a07g044-cpg.c)

Note: All the information in the above device tree is used for RZ/G2L, also the same as RZ/V2L. To config for RZ/V2L, only replace r9a07g044 with r9a07g054.

For RZ/V2N, the CPG core clock definition is the same as RZ/G2L and the valid values for the identifier of CPG driver can be found in struct `cpg_core_clk` `r9a09g056_core_clks`.

```
static const struct cpg_core_clk r9a09g056_core_clks[] __initconst = {
    ...
    /* Core Clocks */
    DEF_FIXED("sys_0_pclk", R9A09G056_SYS_0_PCLK, CLK_QEXTAL, 1, 1),
    ...
};
```

Figure 6-6 struct `cpg_core_clk` data arrays (Refer to drivers/clock/renesas/ r9a09g056-cpg.c)

For RZ/G3E, the CPG core clock definition is the same as RZ/G2L and the valid values for the identifier of CPG driver can be found in struct `cpg_core_clk` `r9a09g047_core_clks`.

```
static const struct cpg_core_clk r9a09g047_core_clks[] __initconst = {
    ...
    /* Core Clocks */
    DEF_FIXED("sys_0_pclk", R9A09G047_SYS_0_PCLK, CLK_QEXTAL, 1, 1),
    ...
};
```

Figure 6-7 struct `cpg_core_clk` data arrays (Refer to drivers/clock/renesas/ r9a09g047-cpg.c)

(2) Using the CPG module clock definitions

- **RZ/G2L, RZ/G2LC, RZ/G2UL and RZ/V2L:**

```
i2c0: i2c@10058000{
    clocks = <&cpg CPG_MOD R9A07G044_I2C0_PCLK>;
    power-domains = <&cpg>;
};
```

Figure 6-8 Example of device tree for MSSR module clocks definitions

The format of a "clocks" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

The 2nd cell must be set to `CPG_MOD`.

The 3rd cell contains the identifier of CPG/MSSR driver.

Above example defines `R9A07G044_CLK_I2C0` for `I2C0`.

The format of a "power-domains" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

About this property, follow the Power Management User's Manual documentation in the environment where the Power Management is supported.

Valid values for identifier of the CPG/MSSR driver can be found in the struct `mssr_mod_clk` data arrays corresponding to the SoC.


```
static const struct {
    struct rzg2l_mod_clk common[85];
} mod_clks = {
    ...
    DEF_MOD("i2c0", R9A07G044_I2C0_PCLK, R9A07G044_CLK_P0, 0x580, 0),
    ...
};
```

Figure 6-9 struct mod_clks data arrays (Refer to drivers/clock/renesas/r9a07g044-cpg.c)

Note: All the information in the above device tree is used for RZ/G2L, also the same as RZ/V2L. To config for RZ/V2L, only replace r9a07g044 with r9a07g054.

- **RZ/V2N, RZ/G3E:**

```
i2c0: i2c@14400400{
    clocks = <&cpg CPG_MOD 0x94>;
    power-domains = <&cpg>;
};
```

Figure 6-10 Example of device tree for CPG module clocks definitions

The format of a "clocks" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

The 2nd cell must be set to CPG_MOD.

The 3rd cell contains the identifier of CPG driver. The identifier is calculated with the formula below:

Module identifier = ON_index * 16 + ON_bit

ON_index: is the m index of CLK_ON_m register that contains the module (m = 0 to 24).

ON_bit: is the bit index of the module in CLK_ON_m.

Above example is for I2C0 in RZ/V2N.

The format of a "power-domains" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

About this property, follow the Power Management User's Manual documentation in the environment where the Power Management is supported.

The variables for identifier formula of the CPG driver can be found in the struct rzv2h_mod_clk data arrays corresponding to the SoC.

```
#define DEF_MOD(_name, _parent, _onindex, _onbit, _monindex, _monbit) \
    DEF_MOD_BASE(_name, _parent, false, _onindex, _onbit, _monindex, _monbit)
```

Figure 6-11 DEF_MOD definition (Refer to drivers/clock/renesas/rzv2h-cpg.h)

```
Static const struct rzv2h_mod_clk r9a09g056_mod_clks[] __initconst = {
    ...
    DEF_MOD("riic_0_ckm", CLK_PLLCLN_DIV16, 9, 4, 4, 20),
    ...
};
```

Figure 6-12 struct rzv2h_mod_clk data arrays (Refer to drivers/clock/renesas/r9a09g056-cpg.c for RZ/V2N)

```
static const struct rzv2h_mod_clk r9a09g047_mod_clks[] __initconst = {  
    ...  
    DEF_MOD("riic_0_ckm", CLK_PLLCLN_DIV16, 9, 4, 4, 20),  
    ...  
};
```

Figure 6-13 struct rzv2h_mod_clk data arrays (Refer to drivers/clock/renesas/r9a09g047-cpg.c for RZ/G3E)

6.4.2.2 Get definitions of the clock

The example which gets the definitions of the clock from a device tree is as follows.

```
static int riic_i2c_probe(struct platform_device *pdev) {  
    struct device *dev = &pdev->dev;  
    ...  
    priv->clk = devm_clk_get(dev, NULL);  
    ...  
};
```

Figure 6-14 Example of getting clock from device tree

6.5 Reset Control

6.5.1 Interface specification

Detailed explanation is skipped because the reset control functions such as the `devm_reset_control` get is based on Linux.

6.5.2 Definitions

6.5.2.1 Definitions of the reset controller

- **RZ/G2L, RZ/G2LC, RZ/G2UL and RZ/V2L:**

Definitions of the reset controller are described in device tree. Example of device tree setting is as following.

```
i2c0: i2c@10058000 {
    ...
    resets = <&cpg R9A07G044_I2C0_MRST>;
    ...
};
```

Figure 6-15 Example of device tree for reset controller definitions

The format of a "resets" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

The 2nd cell contains the identifier of CPG/MSSR driver.

Above example defines `R9A07G044_I2C0_MRST` for `I2C0`.

Valid values for identifier of the CPG/MSSR driver can be found in the struct `rzg2l_reset` data arrays corresponding to the SoC.

```
static struct rzg2l_reset r9a07g044_resets[] = {
    ...
    DEF_RST(R9A07G044_I2C0_MRST, 0x880, 0), ...
};
```

Figure 6-16 struct `rzg2l_reset` data arrays (Refer to `drivers/clock/renesas/r9a07g044-cpg.c`)

Note: All the information in the above device tree is used for RZ/G2L, also the same as RZ/V2L. To config for RZ/V2L, only replace `r9a07g044` with `r9a07g054`.

- **RZ/V2N, RZ/G3E:**

Definitions of the reset controller are described in device tree. Example of device tree setting is as following.

```
i2c0: i2c@14400400{
    ...
    resets = <&cpg 0x98>;
    ...
};
```

Figure 6-17 Example of device tree for reset controller definitions

The format of a "resets" property is as follows:

The 1st cell is a node or label of CPG clock to be used.

The 2nd cell contains the identifier of CPG driver. The identifier is calculated with the formula below:

Module identifier = RST_index * 16 + RST_bit

RST_index: is the m index of CLK_RST_m register that contains the module (m = 0 to 17).

RST_bit: is the bit index of the module in CLK_RST_m.

Above example is for I2C0 in RZ/V2N.

The variables for identifier formula of the CPG driver can be found in the struct rzv2h_reset data arrays corresponding to the SoC.

```
#define DEF_RST(_resindex, _resbit, _monindex, _monbit) \
    DEF_RST_BASE(_resindex, _resbit, _monindex, _monbit)
```

Figure 6-18 DEF_RST definition (Refer to drivers/clock/renesas/rzv2h-cpg.h)

```
static const struct rzv2h_reset r9a09g056_resets[] __initconst = {
    ...
    DEF_RST(9, 8, 4, 9), /* RIIC_0_MRST */
    ...
};
```

Figure 6-19 struct rzv2h_reset data arrays (Refer to drivers/clock/renesas/r9a09g056-cpg.c for RZ/V2N)

```
static const struct rzv2h_reset r9a09g047_resets[] __initconst = {
    ...
    DEF_RST(9, 8, 4, 9), /* RIIC_0_MRST */
    ...
};
```

Figure 6-20 struct rzv2h_reset data arrays (Refer to drivers/clock/renesas/r9a09g047-cpg.c for RZ/G3E)

6.5.2.2 Get definitions of the reset controller

The example which gets the definitions of the clock from a device tree is as follows.

```
static int riic_i2c_probe(struct platform_device *pdev) {
    struct device *dev = &pdev->dev;
    ...
    riic->rstc = devm_reset_control_get_optional_exclusive(dev, NULL);
    ...
};
```

Figure 6-21 Example of getting clock from device tree

Revision History	Linux Interface Specification Kernel Core User's Manual: Software
------------------	--

Rev.	Date	Description	
		Page	Summary
0.50	Apr. 01, 2021	—	First Edition issued
1.0	Jul. 15, 2021	8, 10, 12	Add rzg2l-smarc.dtsi as included files of device trees Remove detail explanation of pinctrl control which can be referenced from GPIO document Remove detail explanation of clock/reset control which are based on Linux
1.1	Sep. 15, 2021	—	Merge RZ/G2L driver manual with RZ/V2L
1.2	Feb. 15, 2022	—	Add RZ/G2LC, RZ/G2UL device
1.3	Mar. 31, 2022	—	No modification, change version to keep consistent with other documents
1.4	May. 31, 2022	—	No modification, change version to keep consistent with other documents
1.5	Jun. 24, 2022	—	Add RZ/Five device
1.6	Sep. 15, 2022	—	No modification, change version to keep consistent with other documents
1.7	Dec. 15, 2022	1	Add all devicetree files for RZ/G2L Series, RZ/V2L and RZ/Five.
1.8	Mar. 15, 2023	—	No modification, change version to keep consistent with other documents
1.9	Mar.31, 2025	—	Update ARM, RISC-V architected timer Update main memory for RZ/G2UL Update name of included device tree files Add new RZ/FIVE interrupt example figure Replace RCAR I2C code to RIIC one Update clock macro
1.10	May.30, 2025	1	- Add MPU information support for both kernel versions v5.10 and v6.1.
1.11	Jun. 30, 2025	—	Add RZ/V2N information.
1.12	Jul. 22, 2025	1	Update: - Kernel v6.1, ARM architected timer, device tree for RZ/G3E
		2	Add RZ/G3E hardware environment
		5	Update GIC-600 for RZ/G3E
		6	Update Physical address for RZ/G3E Correct Physical address for RZ/V2N
		8	Correct DRAM Managed Linux Size
		9	Add memory map of kernel for RZ/G3E + Adjust Multimedia CMA (256 MB)
		10	Update: - LPDDR, Arm processor type for RZ/G3E
		12	Update: - Device tree list for RZ/G3E - Carrier board and Module Board device tree files.
		17	Add Figure 6-13: struct rzv2h_mod_clk data arrays for RZ/G3E Add Figure 6-7: CPG core clock definition for RZ/G3E
		19	Add Figure 6-13: struct rzv2h_mod_clk data arrays for RZ/G3E
		21	Add Figure 6-20: struct rzv2h_reset data arrays for RZ/G3E

Linux Interface Specification Kernel Core
User's Manual: Software

Publication Date: Rev. 1.12 Jul 22, 2025

Published by: Renesas Electronics Corporation

RZ/G2L Group, RZ/V2L Group, RZ/V2N Group,
RZ/G3E Group and RZ/Five



Renesas Electronics Corporation