

Linux Interface Specification Device Driver MTU3a

User's Manual: Software

RZ/G2L Group, RZ/V2L 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.1.10 May 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. 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/Five Group User's Manual: Hardware	---
User's manual for Software	Description of MTU3a Linux interface Specification	Linux interface Specification – MTU3a	This User's manual
	Description of POE3 Linux interface Specification	Linux Interface Specification POE3	---
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
MTU3a	Multi-Function Timer Pulse Unit 3
PWM	Pulse Width Modulation
Counter	Counter interface

Table of Contents

Corporate Headquarters	1
Contact information	1
Trademarks	1
1. Purpose and Target Readers	2
2. Notation of Numbers and Symbols	3
3. Register Notation.....	3
4. List of Abbreviations and Acronyms	4
1. Overview	1
1.1 Overview.....	1
1.2 Function	1
1.3 Reference	1
1.3.1 Standard.....	1
1.3.2 Related documents	1
1.4 Restrictions	1
2. Terminology.....	2
3. Operating Environment.....	3
3.1 Hardware Environment.....	3
3.2 Module Configuration.....	4
4. External Interface.....	6
4.1 Device modes.....	6
4.2 Linux interface.....	7
4.2.1 Clocksource and clockevent interface	7
4.2.2 PWM interface	8
4.2.2.1 PWM mode 1	8
4.2.2.2 Complementary PWM.....	9
4.2.3 Counter interface.....	10
4.2.3.1 16-bit phase counting Counter interface	10
4.2.3.2 32-bit phase counting counter interface	11
4.3 Definitions	14
4.3.1 Device information in Device Tree	14
4.3.2 PWM modes configuration	16
4.3.2.1 PWM mode 1 configuration.....	16
4.3.2.2 Complementary PWM configuration.....	17
4.3.3 Phase counting configuration in Device Tree.....	18
4.3.3.1 16-bit phase counting Device Tree setting	18
4.3.3.2 32-bit phase counting Device Tree setting	19
4.3.4 Priority order of MTU3 devicetree properties	20
5. Integration	21
5.1 Directory Configuration.....	21
5.2 Integration Procedure.....	21
5.3 Option Setting	21

5.3.1 Module Parameters..... 21

5.3.2 Kernel Parameters 21

1. Overview

1.1 Overview

This manual explains the driver module that controls the MTU3a module for RZ/G2L Group, RZ/V2L Group, RZ/FIVE Group.

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 and RZ/G2LC.

1.2 Function

This driver module supports the following functions:

- A clocksource using free-running counter.
- A clockevent using periodic counter.
- Normal PWM output.
- Complementary PWM output.
- 16-bit phase counting.
- 32-bit phase counting.

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
MTU3a	Multi-Function Timer Pulse Unit 3
PWM	Pulse width modulation
Counter	Counter interface

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/V2L, RZ/G2UL, RZ/G2LC and RZ/FIVE)

Name	Version	Manufacture
RZ/G2L Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZ/G2LC Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZ/G2UL Evaluation Board Kit	SMARC module: v01 Carrier board: v03	Renesas Electronics Europe GmbH
RZ/V2L 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

3.2 Module Configuration

The following figure shows the configuration of this module.

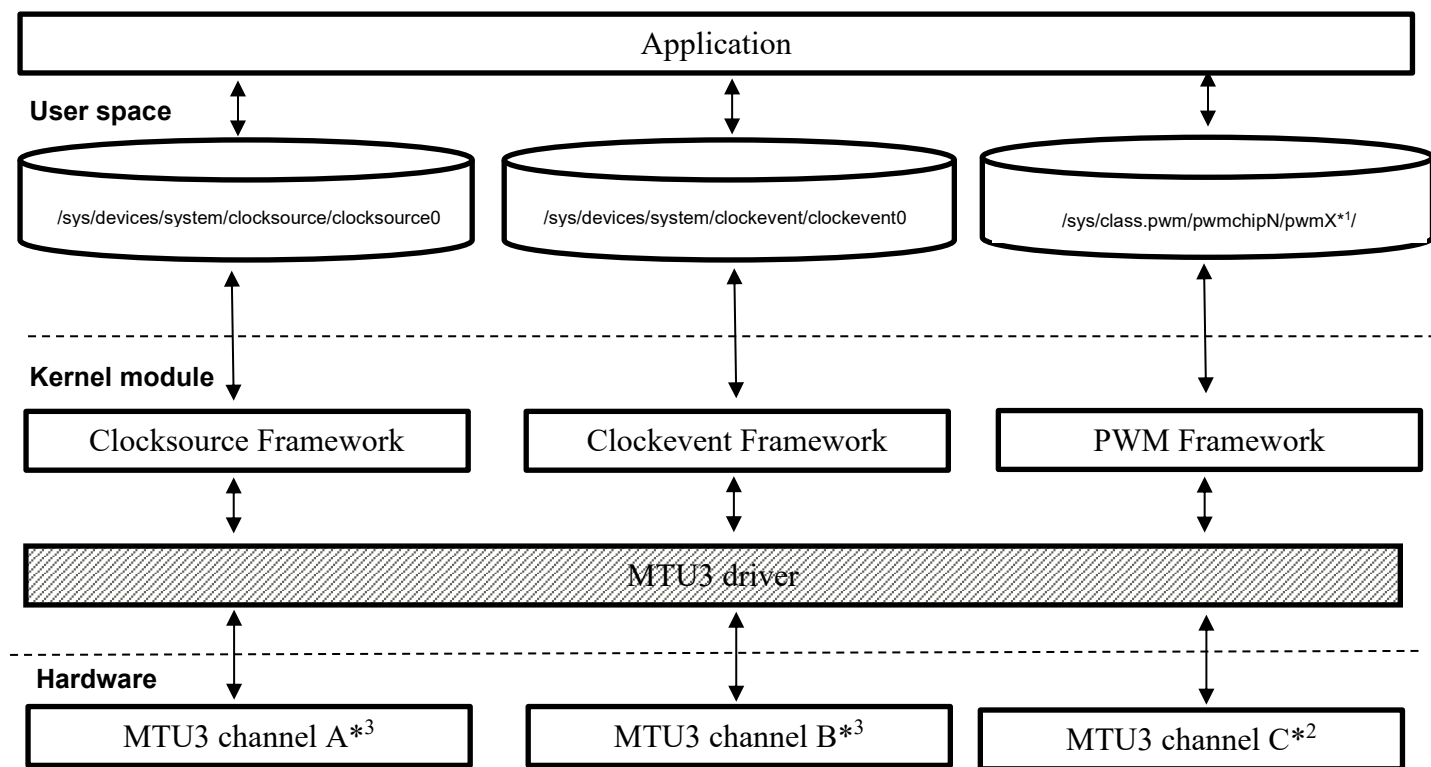


Figure 3-1 MTU3a Driver Module configuration (RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L and RZ/FIVE)

Note 1: There're many pwm devices in 1 pwmchip used as PWM controller of MTU3 device to control PWM output signals of corresponding MTU3 channels.

Note 2: MTU3 channels can be configured to PWM modes. Details are described in

Note 3: MTU3 channel 8 will be automatically selected for clocksource. Clockevent is currently disabled.

16-bit and 32-bit phase counting have been supported for MTU3 channel1 and channel2. Here is the expansion of MTU3 module configuration including counter interface.

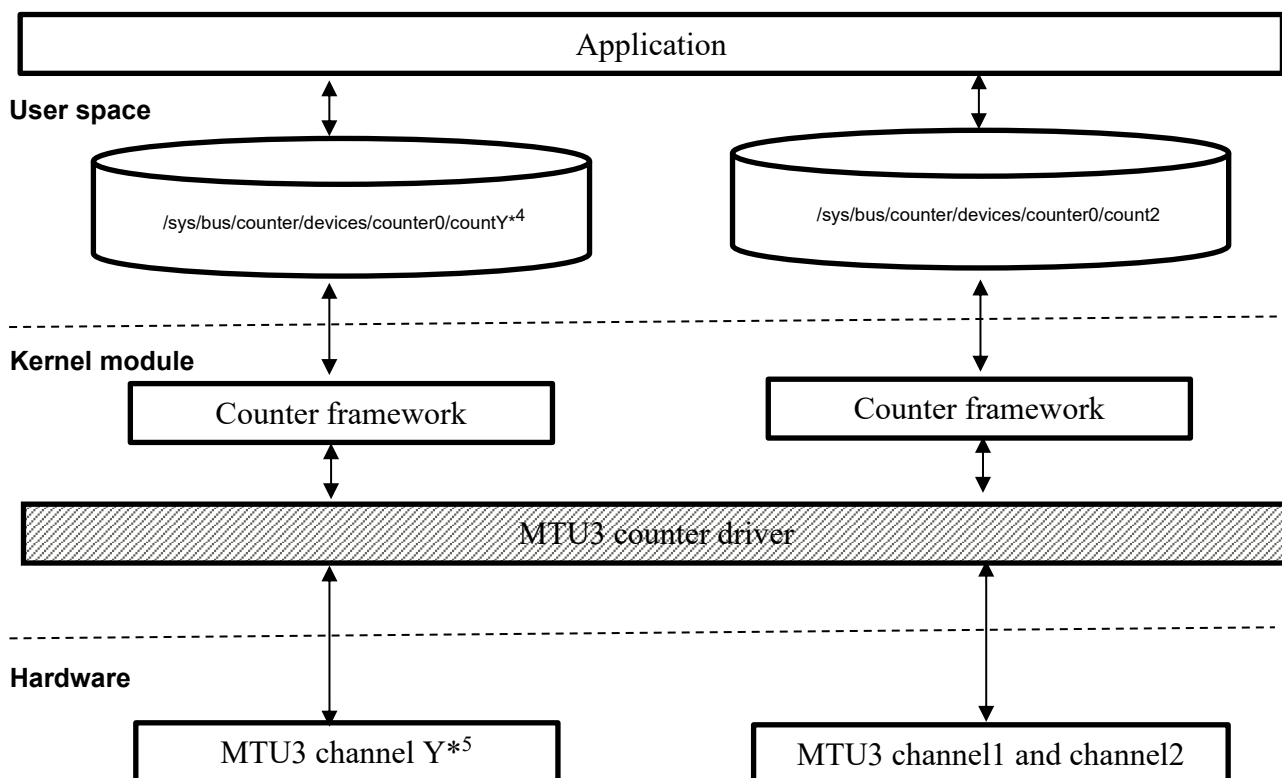


Figure 3-3 MTU3a Driver Module configuration (RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L and RZ/FIVE)
(cont)

Note 4: Y could be 0 or 1 showing which MTU3 channel is being used for 16-bit phase counting function.

4. External Interface

The supported external interface of this module is explained.

4.1 Device modes

MTU3 support upto 9 multi-function timer channels from MTU3 channel 0 to channel 8. User can use it for clocksource, clockevent and PWM output signals at a same time by device tree configuration and sysfs interface.

Table 4.1 MTU3a device node (RZ/G2L and RZ/V2L)

MTU3 channels	Device Node
MTUx	/sys/devices/system/clocksource/clocksource0
MTUy	/sys/devices/system/clockevent/clockevent0
MTUz	/sys/class/pwm/pwmchipA/pwmB
MTUw* ¹	/sys/bus/counter/devices/counter0/countw* ²

Note 1:

- w could 0 showing 16-bit phase counter value channel on MTU1
- w could 1 showing 16-bit phase counter value channel on MTU2
- w could 2 showing 32-bit phase counter value channel by cascading MTU1 and MTU2 channels.

4.2 Linux interface

4.2.1 Clocksource and clockevent interface

Clocksource devices will automatically be registered for MTU3 channel 8.

This picture shows an example to use clocksource on userspace.

```
To check available devices for clocksource function.  
# cat /sys/devices/system/clocksource/clocksource0/available_clocksource  
arch_sys_counter timer@12801400 rz-mtu3-clk.0  
  
To set MTU3 for current clocksource  
# echo rz-mtu3-clk.0 > /sys/devices/system/clocksource/clocksource0/current_clocksource  
  
To Check current clocksource.  
#cat /sys/devices/system/clocksource/clocksource0/current_clocksource  
rz-mtu3-clk.0
```

Figure 4-1 Example of MTU3 clocksource usage

4.2.2 PWM interface

4.2.2.1 PWM mode 1

The external interface of this function is based on Linux. The interface for operating PWM from user space is PWM sysfs. PWM devices can be flexibly configured in sysfs. For details of the configuration, see **Figure 4-2 Example of MTU3's PWM sysfs usage** and **4.3.2.1 PWM mode 1 configuration**

After completing setting in devicetree, user can control operation of PWM output signals via `/sys/class/pwm/pwmchipX1/pwmY2`

Note 1: There's only 1 pwmchip for whole MTU3, and X index of pwmchip is the order number when probing pwmchips.

Note 2: pwmY are PWM devices which contain attributes described in **Table 4-2 Supported PWM sysfs device attributes**. User can directly control or adjust output signals via those parameters. Quantity of usable pwmY depends on amount of PWM devices set successfully by parsing device tree configuration.

Table 4.2 Supported PWM sysfs device attributes

PWM Sysfs Interface	Description	Support status	Notes
period	Set the PWM signal period in nanoseconds	Yes	-
duty_cycle	Sets the PWM signal duty cycle in nanoseconds.	Yes	Need to set period before it is set.
enable	Enable/ disable the PWM signal.	Yes	Set period and duty_cycle, before set it 1 to enable. To disable, write it to 0.
polarity	Invert output signal	Yes	Only officially support for PWM mode1, does not take effect with complementary PWM.

To make PWM interface usage of MTU3 clear, we look at below example.

Figure 4-2 Example of MTU3's PWM sysfs usage

```

We have 12 channels (pwm0-pwm11) for PWM mode1 and 4 channels (pwm12-pwm15) for Complementary PWM mode. To check whether there's pwmchip of MTU3, type.
# cat /sys/kernel/debug/pwm
Expect to see below returned message containing base address of MTU3
platform/pwm-rz-mtu3.0, 16 PWM devices
pwm-0 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
pwm-1 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
.....
pwm-15 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
pwmchip1 is PWM controller of MTU3, and pwm4 and pwm5 is PWM device of channel 3. Here is an example of using a PWM device (MTIOC3A).
To export pwm4's attributes to control its operation.
# echo 4 > /sys/class/pwm/pwmchip1/export
To set PWM signal whose period is 800us and duty is 400us.
# echo 800000 > /sys/class/pwm/pwmchip1/pwm4/period
# echo 400000 > /sys/class/pwm/pwmchip1/pwm4/duty_cycle
To output signal in MTIOC3A pins.
# echo 1 > /sys/class/pwm/pwmchip1/pwm4/enable
To change polarity of PWM signal.
# echo inversed > /sys/class/pwm/pwmchip1/pwm4/polarity
Do the same to control pwm4.

```


4.2.2.2 Complementary PWM

PWM complementary of MTU3 can be controlled via PWM sysfs of Linux, PWM devices can be flexibly configured in sysfs. For details of the configuration, see **Figure 4-3 Example of MTU3's complementary PWM sysfs usage** and **section 4.3.2.2 Complementary PWM configuration**.

User can control operation of PWM output signals via `/sys/class/pwm/pwmchipX1/pwmY2`

Note 1: There's only 1 pwmchip for whole MTU3, and X index of pwmchip is the order number when probing pwmchips.

Note 2: pwmY are PWM devices which contain attributes described in **Table 4-2 Supported PWM sysfs device attributes**. User can directly control or adjust output signals via those parameters. Quantity of usable pwmY depends on amount of PWM devices set successfully by parsing device tree configuration.

Besides, deadtimes of complementary PWM can be set via attributes **mtu34_pwm_deadtime** and **mtu67_pwm_deadtime** correspondingly which are located at `sys/class/pwm/pwmchipX1/device/`.

We have 12 channels (pwm0-pwm11) for PWM mode1 and 4 channels (pwm12-pwm15) for Complementary PWM mode

Let's assume that we have MTU3 and MTU4 assigned PWM complementary to output signals at MTIOC3B and MTIOC3D. To check whether there's pwmchip of MTU3, type.

```
# cat /sys/kernel/debug/pwm
```

Expect to see below returned message containing base address of MTU3 and **pwm12** contain output signals at MTIOC3B and MTIOC3D.

platform/pwm-rz-mtu3.0, 16 PWM devices

```
pwm-0 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
```

```
.....
```

```
pwm-12 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
```

```
pwm-13 ((null)) ): period: 0 ns duty: 0 ns polarity: normal
```

Pwmchip1 is PWM controller of MTU3, and pwm12 is PWM device of MTU3 complementary. Here is an example of using a PWM device.

To export pwm12's attributes to control its operation.

```
# echo 12 > /sys/class/pwm/pwmchip1/export
```

To set PWM signal whose period is 800us and duty is 400us.

```
# echo 800000 > /sys/class/pwm/pwmchip1/pwm12/period
```

```
# echo 400000 > /sys/class/pwm/pwmchip1/pwm12/duty_cycle
```

To output signal in MTIOC3B and MTIOC3D pins.

```
# echo 1 > /sys/class/pwm/pwmchip1/pwm12/enable
```

To add or change deadtimes of PWM signal. Set deadtime value via attribute **mtu34_pwm_deadtime**.

```
# echo 100000 > /sys/class/pwm/pwmchip1/device/mtu34_pwm_deadtime
```

Figure 4-3 Example of MTU3's complementary PWM sysfs usage

4.2.3 Counter interface

4.2.3.1 16-bit phase counting Counter interface

16-bit phase counting has been supported for channel 1 (MTU1) and channel 2 (MTU2). In this function, MTU1 and MTU2 will count edges of MTLCK pins. There're 5 basic modes supporting for phase counting; for operation with each mode, please refer hardware manual.

To use this function, user must add setting in MTU3 devicetree node first. For details of the definition, see **section 4.3.3.1 16-bit phase counting Device Tree setting**.

After probing driver successfully, MTU3 counter devices' attributes will be created at:

/sys/bus/counter/devices/counter0/countY*¹

Note 1: Y could be 0 or 1 showing us which MTU3 channel is being used for this function. There's a series of attributes described as below table.

Table 4.3 Counter attributes of MTU3 of 16 bit phase counting

Counter attribute	Description
enable	Enable counter. 1 is working, 0 is disabled.
count	Setting timer counter
ceiling	Output compare or input capture
cascade_counts_enable	MTU1/MTU2 Combination Longword Access Control. 1 is 32-bit access is enabled, 0 is 16-bit access is enabled
direction	Count Direction Flag. 1 is TCNT counts up "forward", 0 is TCNT counts down "backward"
external_input_phase_clock_select	External Input Phase Clock Select Selects the external clock pin for phase counting mode. 0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.

Here is an example of using both MTU1 and MTU2 for A/B counting.

Connect MTCLKA and MTCLKB to A pulse and B pulse of an incremental encoder.
Phase counting mode 1 is used as default in initialization

Disable cascade counts, write 0 to **cascade_counts_enable**

```
# echo 0 > /sys/bus/counter/devices/counter0/cascade_counts_enable
```

Setting timer counter, write 0 to **count**

```
# echo 0 > /sys/bus/counter/devices/counter0/count0/count
```

Setting ceiling for counter

```
# echo 50 > /sys/bus/counter/devices/counter0/count0/ceiling
```

Enable counter, write 1 to **enable**

```
# echo 1 > /sys/bus/counter/devices/counter0/count0/enable
```

Read **count**, to get counter value and see the changes when rotate encoder.

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
0
```

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
10
```

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
36
```

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
42
```

Counter will count from 0 if total value is over max value.

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
13
```

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

```
5
```

Figure 4-4 16-bit counter of MTU3 example

4.2.3.2 32-bit phase counting counter interface

As 16-bit phase counting mode, we supported counter interface to interact with 32-bit cascade phase counter which is combined from MTU1 and MTU2. In this mode, counter will count edges of pairs of MTCLK pins as A/B-phases and can be cleared by capturing triggered signals from MTIOC1A pin as Z-phase.

To use this function, user must add setting in MTU3 devicetree node first. For details of the definition, see **section 4.2.3.2 32-bit phase counting Device Tree setting**.

Table 4.4 Counter attributes of MTU3 32-bit phase counting

Counter attribute	Description
enable	Enable counter. 1 is working, 0 is disabled.
count	Setting timer counter
ceiling	Output compare or input capture
cascade_counts_enable	MTU1/MTU2 Combination Longword Access Control. 1 is 32-bit access is enabled, 0 is 16-bit access is enabled
direction	Count Direction Flag. 1 is TCNT counts up "forward", 0 is TCNT counts down "backward"
external_input_phase_clock_select	External Input Phase Clock Select Selects the external clock pin for phase counting mode. 0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.

Operation of 32-bit mode is mostly as same as 16-bit, but there's some change due to hardware difference.

- 32-bit phase counting has larger counter (32-bit compared to 16-bit), so the available signed value is from -2147483648 to 2147483647.
- 32-bit phase counting can work with A/B/Z phase from encoder. Z-phase is used to clear counter to 0 by capturing signal from MTIOC1A pin. However, counter clear by compare match cannot work along with Z-phase clearing, so max value cannot be set.

Here is an example of using both MTU1 and MTU2 for A/B counting.

Connect MTCLKA and MTCLKB to A pulse and B pulse of an incremental encoder.
Phase counting mode 1 is used as default in initialization

Select Phase clock

```
# echo "MTCLKA-MTCLKB" > /sys/bus/counter/devices/counter0/external_input_phase_clock_select
```

Enable cascade counts, write 1 to **cascade_counts_enable**

```
# echo 1 > /sys/bus/counter/devices/counter0/cascade_counts_enable
```

Setting timer counter, write 0 to **count**

```
# echo 0 > /sys/bus/counter/devices/counter0/count0/count
```

Setting ceiling for counter

```
# echo 12456786 > /sys/bus/counter/devices/counter0/count0/ceiling
```

Enable counter, write 1 to **enable**

```
# echo 1 > /sys/bus/counter/devices/counter0/count0/enable
```

Read **count**, to get counter value and see the changes when rotate encoder.

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

0

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

10

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

36

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

42

Counter will count from 0 if total value is over max value.

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

13

```
# cat /sys/bus/counter/devices/counter0/count0/count
```

5

Figure 4-5 32-bit counter of MTU3 example

4.3 Definitions

4.3.1 Device information in Device Tree

MTU3a device properties are shown as below.

```
mtu3: timer@10001200 {
    compatible = "renesas,r9a07g044-mtu3",
        "renesas,rz-mtu3";
    reg = <0 0x10001200 0 0xb00>;
    interrupts = <GIC_SPI 170 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 171 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 172 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 173 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 174 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 175 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 176 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 177 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 178 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 179 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 180 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 181 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 182 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 183 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 184 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 185 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 186 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 187 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 188 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 189 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 190 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 191 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 192 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 193 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 194 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 195 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 196 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 197 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 198 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 199 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 200 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 201 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 202 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 203 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 204 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 205 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 206 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 207 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 208 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 209 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 210 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 211 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 212 IRQ_TYPE_EDGE_RISING>,
        <GIC_SPI 213 IRQ_TYPE_EDGE_RISING>;
}
```

```

        interrupt-names = "tgia0", "tgib0", "tgic0", "tgid0",
                           "tciv0", "tgie0", "tgif0",
                           "tgia1", "tgib1", "tciv1", "tciu1",
                           "tgia2", "tgib2", "tciv2", "tciu2",
                           "tgia3", "tgib3", "tgic3", "tgid3",
                           "tciv3",
                           "tgia4", "tgib4", "tgic4", "tgid4",
                           "tciv4",
                           "tgiu5", "tgiv5", "tgiw5",
                           "tgia6", "tgib6", "tgic6", "tgid6",
                           "tciv6",
                           "tgia7", "tgib7", "tgic7", "tgid7",
                           "tciv7",
                           "tgia8", "tgib8", "tgic8", "tgid8",
                           "tciv8", "tciu8";
    clocks = <cpug CPG_MOD R9A07G044_MTU_X_MCK_MTU3>;
    power-domains = <&cpug>;
    resets = <cpug R9A07G044_MTU_X_PRESET_MTU3>;
    #pwm-cells = <2>;
};

```

Figure 4-6 Configuration example of MTU3's required properties

Note: All of the information in above device tree is used for both RZ/G2L, RZ/G2LC and RZ/V2L except R9A07G044_MTU_X_MCK_MTU3 is used for RZ/G2L, RZ/G2LC and R9A07G054_MTU_X_MCK_MTU3 is used for RZ/V2L.

Required properties:

- compatible:
must be "renesas,rz-mtu3".
- reg:
need two resources, first one mapped to register base address, second one mapped to extension resource registers base address.
- interrupts:
interrupt specifier for MTU3a
- interrupt-names:
mapped with interrupts and must be in the order with interrupts number.
- clocks, resets:
phandle + clock/reset specifier pairs
- power-domains:
specify the power-domain for MTU3a

4.3.2 PWM modes configuration

Currently, we support PWM mode 1 and PWM complementary mode through device tree and sysfs configuration. To enable MTU3a PWM on PMOD0:

* Disable PMOD1_SER0 by setting "#define PMOD1_SER0 0"

* Enable PMOD_MTU3 by setting "#define PMOD_MTU3 1"

4.3.2.1 PWM mode 1 configuration

PWM mode 1 export PWM signals in MTIOCxA or MTIOCxC pins of channel x. Follow below format to assign

Table 4-5 List of MTU3 channels and pins supporting PWM mode 1 (RZ/G2L, RZG2LC, RZ/V2L)

MTU3 channel	MTU3 pins	PWM index on sysfs
MTU0	MTIOC0A	0
	MTIOC0C	1
MTU1	MTIOC1A	2
MTU2	MTIOC2A	3
MTU3	MTIOC3A	4
	MTIOC3C	5
MTU4	MTIOC4A	6
	MTIOC4C	7
MTU6	MTIOC6A	8
	MTIOC6C	9
MTU7	MTIOC7A	10
	MTIOC7C	11

To take fully effect, PWM definition should be used along with pinctrl.

Here is an example to define MTIOC0A, MTIOC2A, MTIOC3C, MTIOC7C in device tree to use as PWM output signals.

Figure 4-7 Example of pwm_mode1 configuration in device tree.

```
&pinctrl {
    mtu3_pins: mtu3 {
        pinmux =
            <RZG2L_PORT_PINMUX(0, 0, 4)>, /* MTIOC0A */
            <RZG2L_PORT_PINMUX(3, 0, 4)>, /* MTIOC2A */
            <RZG2L_PORT_PINMUX(44, 2, 4)>, /* MTIOC3C */
            <RZG2L_PORT_PINMUX(42, 3, 5)>; /* MTIOC7C */
    };
};

...

&mtu3 {
    pinctrl-0 = <&mtu3_pins>;
    pinctrl-names = "default";
    status = "okay";
};
```


4.3.2.2 Complementary PWM configuration

MTU3 support complementary PWM from pwm12 to pwm15 but need to use pairs of MTU3's channels. Follow below format to assign PWM mode 1 of MTU3's channels. To enable MTU3a Complementary PWM on PMOD0:

* Disable PMOD1_SER0 by setting "#define PMOD1_SER0 0"

* Enable PMOD_MTU3 by setting "#define PMOD_MTU3 1"

Table 4-6 List of MTU3 channels and pins supporting complementary PWM (RZ/G2L, RZG2LC, RZ/G2UL, RZ/V2L and RZ/FIVE)

MTU3 channel pair	MTU3 pins	Index in sysfs
MTU3 and MTU4	MTIOC3B and MTIOC3D, MTIOC4A and MTIOC4C	12
	MTIOC3B and MTIOC3D, MTIOC4B and MTIOC4D	13
MTU6 and MTU7	MTIOC6B and MTIOC6D, MTIOC7A and MTIOC7C	14
	MTIOC6B and MTIOC6D, MTIOC7B and MTIOC7D	15

To take fully effect, PWM definition should be used along with pinctrl setting.

Here is an example to define MTIOC3B with MTIOC3D whose deadtime is 100000 ns.

Figure 4-8 Example of complementary pwm mode configuration in device tree

```
&pinctrl {
    mtu3_pins: mtu3 {
        pinmux =
            <RZG2L_PORT_PINMUX(44, 1, 4)>, /* MTIOC3B */
            <RZG2L_PORT_PINMUX(44, 3, 4)>, /* MTIOC3D */
    };
};
...
&mtu3 {
    pinctrl-0 = <&mtu3_pins>;
    pinctrl-names = "default";
    status = "okay";
};
```

4.3.3 Phase counting configuration in Device Tree

4.3.3.1 16-bit phase counting Device Tree setting

In MTU3, 16-bit phase counting mode only support for channel 1 (MTU1) and channel 2 (MTU2). To enable this mode, follow below format in file r9a07g044l2-smarc.dts. **PMOD_MTU3 1**

Here is an example of setting channel1 (MTU1) and channel2 (MTU2) for 16-bit phase counting to count input signals from MTCLKA, MTLCKB

```
&pinctrl {
    mtu3_pins: mtu3 {
        mtu3-ext-clk-input-pin {
            pinmux = <RZG2L_PORT_PINMUX(48, 0, 4)>, /* MTCLKA */
                  <RZG2L_PORT_PINMUX(48, 1, 4)>; /* MTLCKB */
        };
    };
};
...
&mtu3 {
    pinctrl-0 = <&mtu3_pins>;
    pinctrl-names = "default";
    status = "okay";
};
```

Figure 4-9 Example of 16-bit phase counting mode configuration in device tree

4.3.3.2 32-bit phase counting Device Tree setting

32-bit cascade phase counting of MTU3 is created by combining channel1 (MTU1) and channel2 (MTU2).

To enable this mode, follow below format in file r9a07g044l2-smarc.dts **PMOD_MTU3 1**

In this mode, can be enable **MTU3_COUNTER_Z_PHASE_SIGNAL 1** if want to test Z phase

Here is an example of setting channel1 (MTU1) and channel2 (MTU2) for 32-bit phase counting to count input signals from MTCLKA, MTCLKB and Z phase.

```
&pinctrl {
    mtu3_pins: mtu3 {
        mtu3-ext-clk-input-pin {
            pinmux = <RZG2L_PORT_PINMUX(48, 0, 4)>, /* MTCLKA */
                  <RZG2L_PORT_PINMUX(48, 1, 4)>; /* MTCLKB */
        };
    };
};
...
&mtu3 {
    pinctrl-0 = <&mtu3_pins>;
    pinctrl-names = "default";
    status = "okay";
};

#ifdef MTU3_COUNTER_Z_PHASE_SIGNAL
/* SDHI cd pin is muxed with counter Z phase signal */
&sdhi1 {
    status = "disabled";
};
#endif /* MTU3_COUNTER_Z_PHASE_SIGNAL */
```

Figure 4-10 Example of 32-bit phase counting mode configuration in device tree

4.3.4 Priority order of MTU3 devicetree properties

When a MTU3 channel is set more than 1 function, the priority order of MTU3 properties decides which function will be assigned for this channel.

Below table describes properties and their privileges; the smaller privilege the higher priority of optional MTU3 devicetree properties.

Table 4-7 Priority order of MTU3 devicetree properties

Privilege	Devicetree property	Description
1	pwm_mode1	PWM mode 1 can be configured for MTU3 from channel0 to channel7.
2	pwm_complementary	Complementary PWM can be configured for pairs of MTU3 channels3/4 and channels6/7.
3	Not set	Clocksource will automatically be selected for channel 8.

5.Integration

5.1 Directory Configuration

The directory configuration is shown below.

—— drivers/mfd/ —— rz-mtu3.c : MTU3a Driver source file core

Figure 5-1 Directory multi-function configuration

—— drivers/clocksource/ —— rz_mtu3_clk.c : MTU3a Driver clocksource, clockevent

Figure 5-2 Directory clocksource and clockevent configuration

—— drivers/pwm/ —— pwm-rz-mtu3.c : MTU3a Driver PWM source file

Figure 5-3 Directory PWM function configuration.

—— drivers/counter/ —— rz-mtu3-cnt.c : MTU3a Driver counter source file

Figure 5-4 Directory counter function configuration.

5.2 Integration Procedure

To enable the function of this module, make the following setting with Kernel Configuration.

```
Platform selection --->
    [*] Renesas SoC Platforms
```

Figure 5-5 Kernel configuration

5.3 Option Setting

5.3.1 Module Parameters

There are no module parameters.

5.3.2 Kernel Parameters

There are no kernel parameters.

Revision History	Linux Interface Specification Device Driver MTU3a User's Manual: Software
------------------	--

Rev.	Date	Description	
		Page	Summary
0.50	Apr. 30, 2021	—	First Edition issued
1.0	Jul. 15, 2021	—	No modification, keep version to keep consistent with other documents.
1.1	Sep. 15, 2021	—	Merge RZ/G2L driver manual with RZ/V2L.
1.2	Feb. 15, 2022	—	Add PWM support information and update information of RZ/G2LC, RZ/G2UL.
1.3	Mar. 31, 2022	—	Update information.
1.4	May. 31, 2022	—	Remove redundant information. Add 16-bit phase counting support and update new information.
1.5	Jun. 24, 2022	—	<ul style="list-style-type: none"> - Remove phase-counting information because it's not officially supported in this release. - Update information of RZ/Five.
1.6	Sep. 15, 2022	—	Update new information of PWM and phase counting usage.
1.7	Dec. 15, 2022	—	Update version of document.
1.8	Mar. 15, 2023	—	No modification, keep version to keep consistent with other documents.
1.9	Mar. 31, 2024	5-21	<ul style="list-style-type: none"> - Update guideline PWM interface and Clocksource, Clockevent interface - Use the Counter interface for support 16-bit phase counting, 32-bit phase counting, Z-phase counting instead of industrial I/O subsystem. Modify the test run of the MTU3a counter to match the Counter interface. - Update configuration setting in device tree for MTU3a.
1.10	May. 30, 2025	1	- Add MPU information support for both kernel versions v5.10 and v6.1.

Linux Interface Specification Device Driver MTU3a
User's Manual: Software

Publication Date: Rev.1.10 May. 30, 2025

Published by: Renesas Electronics Corporation

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



Renesas Electronics Corporation