

# Linux Interface Specification Device Driver I3C

User's Manual: Software

RZ/V2N Group and RZ/G3E Group

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. 1.01 Jul 2025)

## Corporate Headquarters

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

[www.renesas.com](http://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/](http://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. A basic knowledge of electric circuits, logical circuits, and MPUs is necessary in order to use this manual. The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

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/V2N group and RZ/G3E 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/V2N Group User's Manual: Hardware	---
		RZ/G3E Group User's Manual: Hardware	---
User's manual for Software	Description of I3C Linux Interface Specification	Linux Interface Specification Device Driver I3C	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
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
Hi-Z	High Impedance
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
PLL	Phase Locked Loop

# Table of Contents

1. Overview.....	1
1.1 Overview .....	1
1.2 Function .....	1
1.2.1 Driver Function .....	1
1.2.2 Transfer Speed.....	1
1.2.3 Connected Device .....	2
1.3 Reference .....	3
1.3.1 Standards .....	3
1.3.2 Related Documents .....	3
1.4 Restrictions .....	4
2. Terminology.....	5
3. Operating Environment.....	6
3.1 Hardware Environment .....	6
3.2 Module Configuration .....	7
3.3 State Transition Diagram .....	8
4. External Interface.....	9
4.1 Device Node .....	9
4.2 External Function .....	9
4.3 Structure.....	11
4.3.1 I2C structure.....	11
4.3.2 I3C structure.....	12
Table 4.3-1 struct i3c_priv_xfer.....	12
Table 4.3-2 struct i3c_device_info .....	12
Table 4.3-3 struct i3c_bus.....	12
4.4 Global Variables and Constants .....	13
4.4.1 Global Variables .....	13
4.4.2 Global Constants.....	13
4.5 Definitions .....	14
4.5.1 Device information in Device Tree .....	14
5. Integration.....	18
5.1 Directory Configuration .....	18
5.2 Integration Procedure .....	18
5.3 Option Setting .....	19
5.3.1 Module Parameters .....	19
5.3.1.1 Device Tree .....	19
5.3.2 Kernel Parameters .....	21
5.3.3 Device tree bindings.....	21

# 1. Overview

## 1.1 Overview

This manual explains the Linux I3C device driver in RZ/V2N Group and RZ/G3E Group.

## 1.2 Function

This module transmits/receives data to/from a device connected to the I3C interface on RZ/V2N and RZ/G3E.

### 1.2.1 Driver Function

The following table lists the functions of this module.

**Table 1.2-1 Driver Function**

function	support status (RZ/V2N and RZ/G3E)
Number of channels	1
Master mode	Support
Slave mode	Not support
I2C Operation	Support
Common Command Code (CCC)	Support
In-band Interrupt (IBI)	Not support
Hot-join support	Not support
SDR message	Support
DDR message	Not support

### 1.2.2 Transfer Speed

The transfer rate is up to 3 Mbps in I3C mode and Fast-mode (Fm) or Fast-mode plus (Fm+) in I2C mode.

**Table 1.2-2 Transfer speed**

Interface mode	Real transfer speed	Support
I3C Speed (3Mbps)	3Mbit/s	yes
I2C Standard mode (100kps)	100Kbit/s	yes
I2C Fast mode (400kps)	400Kbit/s	yes
I2C Fast mode Plus (1Mbps)	1Mbit/s	yes
I2C High Speed (3.4MHz)	3.4Mbit/s	no



### 1.2.3 Connected Device

The RZ/V2N Evaluation Kit and RZ/G3E SMARC Evaluation Board Kit does not come with on-board I3C devices connected to the I3C bus.

Instead, it offers an output I3C connector (CN32 on RZ/V2N and CN1 on RZ/G3E) which can be used as an interface to external I3C/I2C devices.

On the RZ/V2N Evaluation Kit, there is no dedicated I3C connector. However, I3C interface is accessible through the pins available on the PMOD CN6 header (pin 3 for SCL and pin 4 for SDA).

On the RZ/G3E Evaluation Kit there is a dedicated I3C connector in CN1 (Breakout board JTAG etc) with pin 1 for I3C-VDD, pin 3 for SCL and pin 4 for SDA. Breakout board JTAG also has SW1 to pull-up for I3C SCL and SDA pin.

**Table 1.2-3 Connector for RZ/V2N and RZ/G3E**

Channel	Connector
I3C0	I3C

## 1.3 Reference

### 1.3.1 Standards

The following table shows the standard that this module complies with.

**Table 1.3-1 Standard**

Reference Number	Issue	Title	Edition	Date
-	NXP Semiconductors	THE I2C-BUS SPECIFICATION	-	-
-	SBS Implementers Forum	System Management Bus (SMBus) Specification	Version 2.0	Aug. 03, 2000

### 1.3.2 Related Documents

The following table lists the document related to this module.

**Table 1.3-2 Related documents**

Number	Issue	Title	Edition	Date
-	Renesas Electronics	Linux Interface Specification Device Driver I2C	-	-

## **1.4 Restrictions**

There is no restriction in this module.

## 2. Terminology

The following table shows the terminologies related to this module.

**Table 2.1-1 Terminology**

Terms	Explanation
I2C	Inter-Integrated Circuit
I3C	Improved Inter-Integrated Circuit
CCC	Common Command Code
IBI	In-band Interrupt

## 3. Operating Environment

### 3.1 Hardware Environment

The following table lists the hardware needed to use this module.

**Table 3.1-1 Hardware specification**

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

## 3.2 Module Configuration

Figure 3-1 shows the configuration of I3C module on RZ/V2N.

Figure 3-2 shows the configuration of I3C module on RZ/G3E.

You can control the connected I2C device using the /dev/i2c-2 on RZ/V2N.

Note: Unlike I2C, there is no I3C device interface from user space.

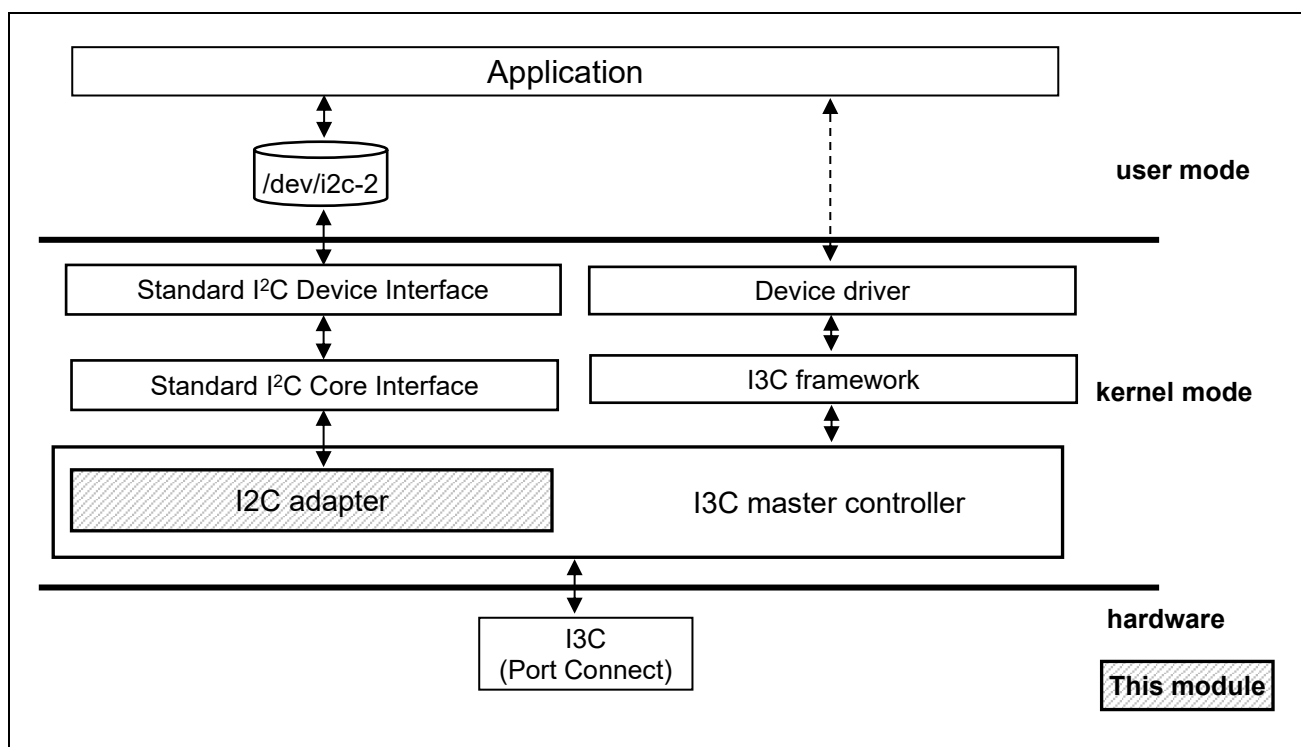


Figure 3-1 I3C Driver Module configuration on RZ/V2N

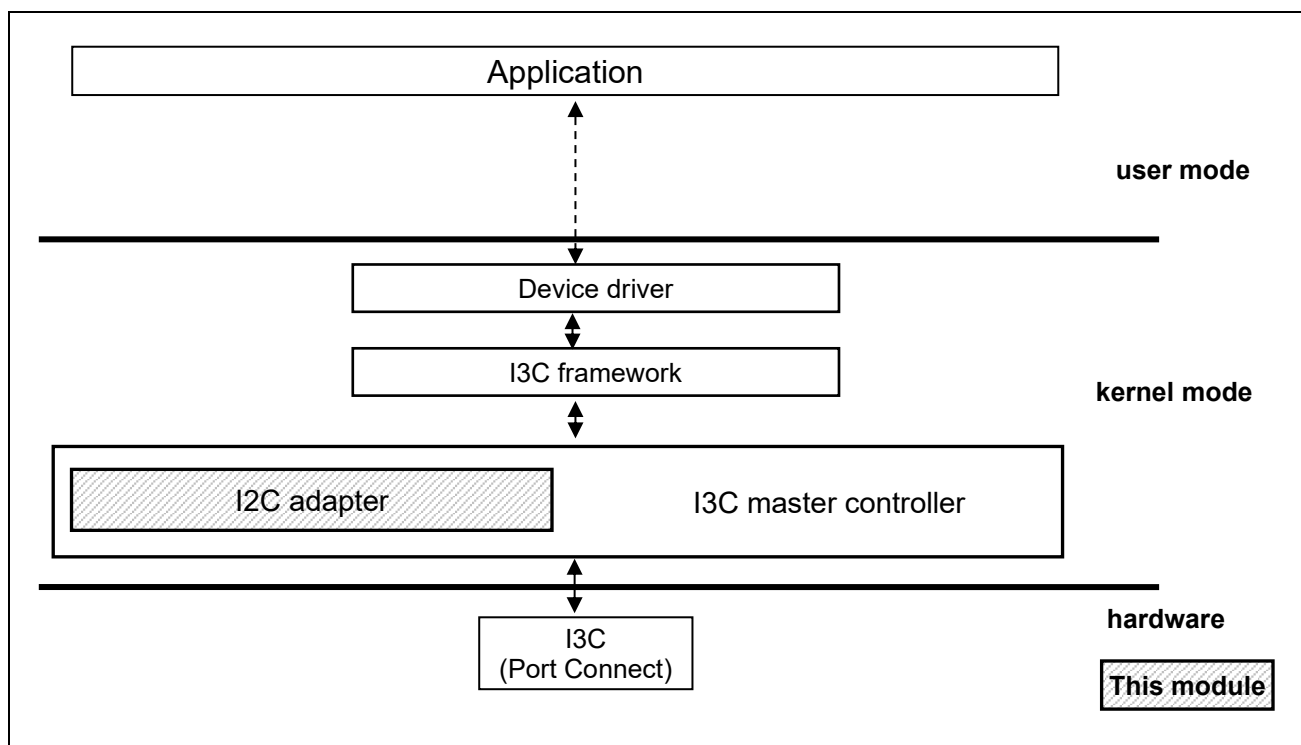


Figure 3-2 I3C Driver Module configuration on RZ/G3E

### 3.3 State Transition Diagram

There is no state transition diagram for this module.

## 4. External Interface

### 4.1 Device Node

The following table shows the device node of this module.

**Table 4.1-1: I2C device node**

Channel	Device node	Major number	Minor number
I2C2	/dev/i2c-2	89	2

### 4.2 External Function

This section explains in the following format about the functions this module supplies.

[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.
[Error number]	Explains the error number(s) of the function.
[Feature]	Explains the features of the function.
[Remark]	Explains points to be noted when using the function.



The following table lists the interface functions in this module, and Standard I2C core Interface and I3C master interface.

**Table 4.2-1 System calls for i2c**

Function name	Description	File
i2cdev_open	Open I2C.	drivers/i2c/i2c-dev.c
i2cdev_release	Close I2C.	drivers/i2c/i2c-dev.c
i2cdev_read	Read I2C (8bit data is received).	drivers/i2c/i2c-dev.c
i2cdev_write	Write I2C (8bit data is sent).	drivers/i2c/i2c-dev.c
i2cdev_ioctl	Setting some specific operations of I2C	drivers/i2c/i2c-dev.c

**Table 4.2-2 Standard I2C and I3C device interface**

Function name	Description	File
renesas_i3c_master_i2c_xfers	Transfer I2C necessary information	drivers/i3c/master/renesas-i3c-master.c
i3c_tx_isr	Use TX interrupt (Transmit interrupt enable)	drivers/i3c/master/renesas-i3c-master.c
i3c_rx_isr	Use RX interrupt (Receive interrupt enable)	drivers/i3c/master/renesas-i3c-master.c
i3c_tend_isr	Determine the transfer is completed	drivers/i3c/master/renesas-i3c-master.c
i3c_stop_isr	Stop the Interrupt	drivers/i3c/master/renesas-i3c-master.c
i3c_resp_isr	Read Response Descriptor to check status	drivers/i3c/master/renesas-i3c-master.c
renesas_i3c_master_daa	Do a DAA (Dynamic Address Assignment) procedure	drivers/i3c/master/renesas-i3c-master.c
renesas_i3c_master_send_ccc_cmd	Send a CCC command	drivers/i3c/master/renesas-i3c-master.c
renesas_i3c_master_priv_xfers	Do private I3C SDR transfers	drivers/i3c/master/renesas-i3c-master.c
i2c_of_match_device	Match and connect the ID of Device	drivers/i2c/i2c-core-of.c
i2c_of_match_device_syssfs	Adding devices through the i2c interface	drivers/i2c/i2c-core-of.c
of_i2c_register_device	Register the device	drivers/i2c/i2c-core-of.c
of_i2c_register_devices	Register the child devices	drivers/i2c/i2c-core-of.c
of_i2c_get_board_info	Get information from board using	drivers/i2c/i2c-core-of.c
i2c_transfer	Execute a single or combined I2C message	drivers/i2c/i2c-core-base.c
i2c_get_functionality	Return the functionality.	include/linux/i2c.h

## **4.3 Structure**

The structure of this driver module is based on Linux v5.10

### **4.3.1 I2C structure**

Please refer “Linux Interface Specification Device Driver I2C” in “Related documents”

## 4.3.2 I3C structure

Table 4.3-1 struct i3c\_priv\_xfer

Structure name	Member		
	Type	Member name	Overview
i3c_priv_xfer	u8	rwn	the transfer direction
	u16	len	the transfer direction
	union { void *in; const void *out; }	data	input/output buffer
	enum i3c_error_code	err	I3C error code

Table 4.3-2 struct i3c\_device\_info

Structure name	Member		
	Type	Member name	Overview
i3c_device_info	u64	pid	Provisional ID
	u8	bcr	Bus Characteristic Register
	u8	dcr	Device Characteristic Register
	u8	static_addr	static/I2C address
	u8	dyn_addr	dynamic address
	u8	hdr_cap	supported HDR modes
	u8	max_read_ds	max read speed information
	u8	max_write_ds	max write speed information
	u8	max_ibi_len	max IBI payload length
	u32	max_read_turnaround	max read turn-around time in micro-seconds
	u16	max_read_len	max private SDR read length in bytes
	u16	max_write_len	max private SDR write length in bytes

Table 4.3-3 struct i3c\_bus

Structure name	Member		
	Type	Member name	Overview
i3c_bus	struct i3c_dev_desc *	cur_master	I3c master
	Int	Id	Bus ID
	unsigned long	addrslots[(((I2C_MAX_ADDR + 1) * 2) / BITS_PER_LONG)]	Address slots
	enum i3c_bus_mode	Mode	Bus mode
	struct { unsigned long i3c; unsigned long i2c; }	scl_rate	SCL signal rate for I3C and I2C mode
	struct { struct list_head i3c; struct list_head i2c; }	Devs	2 lists containing all I3C/I2C devices connected to the bus
	struct rw_semaphore	Lock	read/write lock on the bus

## 4.4 Global Variables and Constants

### 4.4.1 Global Variables

There are no global variables for this module.

### 4.4.2 Global Constants

The following table shows the global constants used by standard I2C core and I3C master.

**Table 4.4-1 List of Global constants**

Global Constant Name	Value	Remark
I2C_CLIENT_END	0xfffeU	-
I2C_CLIENT_PEC	0x04	-
I2C_CLIENT_TEN	0x10	-
I2C_CLIENT_WAKE	0x80	-
I2C_FUNC_SMBUS_QUICK	0x00010000	-
I2C_FUNC_SMBUS_READ_BYTE	0x00020000	-
I2C_MODULE_PREFIX	"i2c:"	-
I2C_M_RD	0x0001	-
I2C_M_RECV_LEN	0x0400	-
I2C_M_TEN	0x0010	-
I2C_NAME_SIZE	20	-
I2C_SMBUS_BYTE	1	-
I2C_SMBUS_BYTE_DATA	2	-
I2C_SMBUS_BLOCK_DATA	5	-
I2C_SMBUS_BLOCK_MAX	32	-
I2C_SMBUS_BLOCK_PROC_CALL	7	-
I2C_SMBUS_I2C_BLOCK_DATA	8	-
I2C_SMBUS_PROC_CALL	4	-
I2C_SMBUS_QUICK	0	-
I2C_SMBUS_READ	1	-
I2C_SMBUS_WRITE	0	-
I2C_SMBUS_WORD_DATA	3	-
I2C_RDWR	0x0707	-
I2C_FUNCS	0x0705	-
I2C_SLAVE	0x0703	-
I2C_SMBUS	0x0720	-
I3C_HOT_JOIN_ADDR	0x2	-
I3C_BROADCAST_ADDR	0x7e	-
I3C_BUS_MAX_DEVS	11	-
I3C_BUS_MAX_I3C_SCL_RATE	12900000	-
I3C_BUS_TYP_I3C_SCL_RATE	12500000	-
I3C_BUS_I2C_FM_PLUS_SCL_RATE	1000000	-
I3C_BUS_I2C_FM_SCL_RATE	400000	-
I3C_BUS_TLOW_OD_MIN_NS	200	-

## 4.5 Definitions

### 4.5.1 Device information in Device Tree

The RZ/V2N I3C device properties are showed below.

```
i3c: i3c-master@12400000 {
    compatible = "renesas,i3c-master";
    reg = <0 0x12400000 0 0x10000>;
    clocks = <&cpg CPG_MOD R9A09G056_I3C_PCLKRW>,
            <&cpg CPG_MOD R9A09G056_I3C_PCLK>,
            <&cpg CPG_MOD R9A09G056_I3C_TCLK>;
    clock-names = "pclk", "pclk", "tclk";
    interrupts = <GIC_SPI 674 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 675 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 676 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 677 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 678 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 679 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 680 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 681 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 682 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 683 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 684 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 685 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 686 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 687 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 688 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 689 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 690 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 691 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 692 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 693 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 694 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 695 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 696 IRQ_TYPE_LEVEL_HIGH>;
    interrupt-names = "ierr", "terr", "abort", "resp",
                     "cmd", "ibi", "rx", "tx", "rcv",
                     "hterr", "habort", "hresp", "hcmd",
                     "hrx", "htx", "st", "sp", "exit",
                     "tend", "nack", "al", "tmo", "wu";
    resets = <&cpg R9A09G056_I3C_PRESETN>,
            <&cpg R9A09G056_I3C_TRESETN>;
    reset-names = "presetn", "tresetn";
    #address-cells = <3>;
    #size-cells = <0>;
    i2c-scl-hz = <1000000>;
    i3c-scl-hz = <3000000>;
    status = "disabled";
};
```

Note: All the information in the above device tree is used for RZ/V2N.

The RZ/G3E I3C device properties are showed below

```
i3c: i3c-master@12400000 {
    compatible = "renesas,i3c-master";
    reg = <0 0x12400000 0 0x10000>;
    clocks = <&cpg CPG_MOD 0x90>,
            <&cpg CPG_MOD 0x91>,
            <&cpg CPG_MOD 0x92>;
    clock-names = "pclk", "pclk", "tclk";
    interrupts = <GIC_SPI 674 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 675 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 676 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 677 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 678 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 679 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 680 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 681 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 682 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 683 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 684 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 685 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 686 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 687 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 688 IRQ_TYPE_EDGE_RISING>,
                <GIC_SPI 689 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 690 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 691 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 692 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 693 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 694 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 695 IRQ_TYPE_LEVEL_HIGH>,
                <GIC_SPI 696 IRQ_TYPE_LEVEL_HIGH>;
    interrupt-names = "ierr", "terr", "abort", "resp",
                     "cmd", "ibi", "rx", "tx", "rcv",
                     "hterr", "habort", "hresp", "hcmd",
                     "hrx", "htx", "st", "sp", "exit",
                     "tend", "nack", "al", "tmo", "wu";
    resets = <&cpg 0x96>,
            <&cpg 0x97>;
    reset-names = "presetn", "tresetn";
    #address-cells = <3>;
    #size-cells = <0>;
    i2c-scl-hz = <1000000>;
    i3c-scl-hz = <3000000>;
    status = "disabled";
};
```

Note: All the information in the above device tree is used for RZ/G3E

The I3C required properties:

compatible:

Must be set to "renesas,i3c-master" for R9A09G056 (RZ/V2N), R9A09G047 (RZ/G3E).

reg:

Base address and length of the memory resource used by the I3C CH0.

clocks:

slot 1:

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 must be set to R9A09G056\_I3C\_PCLKRW (RZ/V2N), 0x90 (RZ/G3E).

slot 2:

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 must be set to R9A09G056\_I3C\_PCLK (RZ/V2N), 0x91 (RZ/G3E).

slot 3:

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 must be set to R9A09G056\_I3C\_TCLK (RZ/V2N), 0x92 (RZ/G3E).

interrupt:

<GIC_SPI 674 IRQ_TYPE_LEVEL_HIGH>	Non-recoverable internal error
<GIC_SPI 675 IRQ_TYPE_LEVEL_HIGH>	Normal Transfer Error
<GIC_SPI 676 IRQ_TYPE_LEVEL_HIGH>	Normal Transfer Abort
<GIC_SPI 677 IRQ_TYPE_LEVEL_HIGH>	Normal Response Status buffer full
<GIC_SPI 678 IRQ_TYPE_LEVEL_HIGH>	Normal Command buffer empty
<GIC_SPI 679 IRQ_TYPE_LEVEL_RISING >	Normal IBI Status buffer full
<GIC_SPI 680 IRQ_TYPE_LEVEL_RISING >	Normal Rx Data buffer full
<GIC_SPI 681 IRQ_TYPE_LEVEL_RISING >	Normal Tx Data buffer empty
<GIC_SPI 682 IRQ_TYPE_LEVEL_RISING >	Normal Receive Status buffer full
<GIC_SPI 683 IRQ_TYPE_LEVEL_RISING >	High Priority Transfer Error
<GIC_SPI 684 IRQ_TYPE_LEVEL_RISING >	High Priority Transfer Abort
<GIC_SPI 685 IRQ_TYPE_LEVEL_RISING >	High Priority Response Status buffer full
<GIC_SPI 686 IRQ_TYPE_LEVEL_RISING >	High Priority Command buffer empty
<GIC_SPI 687 IRQ_TYPE_LEVEL_RISING >	High Priority Rx Data buffer full
<GIC_SPI 688 IRQ_TYPE_LEVEL_RISING >	High Priority Tx Data buffer empty
<GIC_SPI 689 IRQ_TYPE_LEVEL_HIGH>	Start condition detection
<GIC_SPI 690 IRQ_TYPE_LEVEL_HIGH>	Stop condition detection
<GIC_SPI 691 IRQ_TYPE_LEVEL_HIGH>	HDR Exit Pattern detection
<GIC_SPI 692 IRQ_TYPE_LEVEL_HIGH>	Transmit end
<GIC_SPI 693 IRQ_TYPE_LEVEL_HIGH>	NACK detection
<GIC_SPI 694 IRQ_TYPE_LEVEL_HIGH>	Arbitration lost
<GIC_SPI 695 IRQ_TYPE_LEVEL_HIGH>	Timeout detection
<GIC_SPI 696 IRQ_TYPE_LEVEL_HIGH>	Wake-up condition detection

Interrupt-name:

Name of each interrupt source, driver get interrupt by interrupt-name

reset:

release reset state: R9A09G056\_I3C\_TRESETN and R9A09G056\_I3C\_PRESETN are used for R9A09G056 (RZ/V2N), 0x96 and 0x97 are used for R9A09G047 (RZ/G3E).

i2c-scl-hz:

Set 1000000 as default.

I3c-scl-hz:

Set 3000000 as default.

address-cells:

Must be set to 3.

Size-cells:

Must be set to 0



## 5. Integration

### 5.1 Directory Configuration

The directory configuration is shown below.

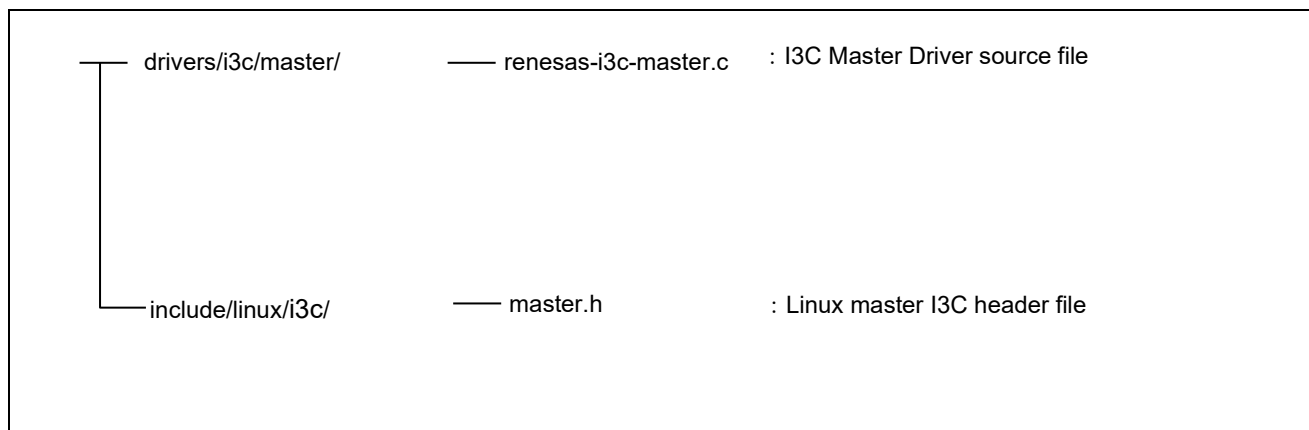


Figure 5-1 Directory configuration

### 5.2 Integration Procedure

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

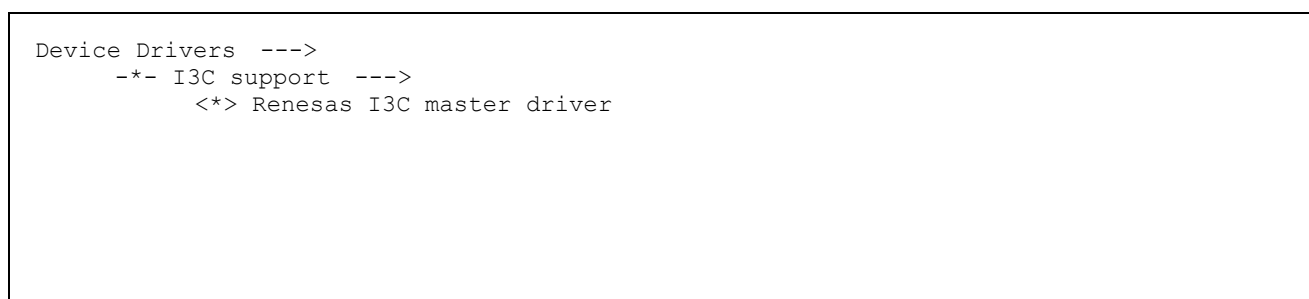


Figure 5-2 Kernel configuration

Figure 5-3 lists the kernel config symbols to support I3C for RZ/V2N and RZ/G3E.



Figure 5-3 Kernel config symbols

## 5.3 Option Setting

### 5.3.1 Module Parameters

#### 5.3.1.1 Device Tree

Enable node and devices setting in r9a09g056n44-evk.dts for RZ/V2N

```
&pinctrl {
    i3c_pins: i3c {
        pinmux = <RZG2L_PORT_PINMUX(2, 0, 1)>, /* I3C_SDA */
                <RZG2L_PORT_PINMUX(2, 1, 1)>; /* I3C_SCL */
    };
}

&i3c {
    pinctrl-0 = <&i3c_pins>;
    pinctrl-names = "default";
    status = "okay";

    /* I2C device. */
    nunchuk: nunchuk@52 {
        compatible = "nintendo,nunchuk";
        reg = <0x52 0x0 0x10>;
    };

    /* I3C device with a static I2C address. */
    thermal_sensor: sensor@68,39200144004 {
        reg = <0x68 0x392 0x144004>;
        assigned-address = <0xa>;
    };
};
```

Figure 5-4 Example of setting I2C and I3C devices on RZ/V2N

Enable node and devices setting in rzg3e-smarc-som.dtsi for RZ/G3E

```
&pinctrl {
    i3c_pins: i3c {
        pinmux = <RZV2H_PORT_PINMUX(2, 1, 2)>, /* I3C_SDA */
                <RZV2H_PORT_PINMUX(2, 0, 2)>; /* I3C_SCL */
    };
}

&i3c {
    pinctrl-0 = <&i3c_pins>;
    pinctrl-names = "default";
    status = "okay";

    /* I2C device.*/
    rtc@68 {
        compatible = "dlg,rtc";
        reg = <0x68 0 0x10>;
    };

    /* I3C device with a static I2C address. */
    lsm6dsm@6b,208006C100B {
        reg = <0x6b 0x208 0x6C100B>;
        assigned-address = <0xa>;
    };
};
```

**Figure 5-5 Example of setting I2C and I3C devices on RZ/G3E**

Note: I2C device must be defined to be recognized on the bus.

The required properties for:

I2C device:

reg: contains 3 cells

first cell : still encoding the I2C address. 10 bit addressing is not supported.

Devices with 10 bit address can't be properly passed through DEFSLVS command.

second cell: shall be 0

third cell: shall encode the I3C LVR (Legacy Virtual Register)

bit[31:8]: unused/ignored

bit[7:5]: I2C device index. Possible values

\* 0: I2C device has a 50 ns spike filter

\* 1: I2C device does not have a 50 ns spike filter but supports high frequency on SCL

\* 2: I2C device does not have a 50 ns spike filter and is not tolerant to high frequencies

\* 3-7: reserved

bit[4]: tell whether the device operates in FM (Fast Mode) or FM+ mode

\* 0: FM+ mode

\* 1: FM mode

bit[3:0]: device type

\* 0-15: reserved

I3C device:

reg: contains 3 cells

first cell : encodes the static I2C address. Should be 0 if the device does not have one (0 is not a valid I2C address).

second and third cells: should encode the ProvisionalID. The second cell contains the manufacturer ID left-shifted by 1. The third cell contains ORing of the part ID left-shifted by 16, the instance ID left-shifted by 12 and the extra information. This encoding is following the PID definition provided by the I3C specification.

assigned-address (Optional property): set specific dynamic-address for device. Without this property, the dynamic-address is set randomly.

### 5.3.2 Kernel Parameters

There are no module parameters.

### 5.3.3 Device tree bindings

Demonstration of I3C device tree settings can be found in Documentation/devicetree/bindings/i3c/i3c.txt

REVISION HISTORY	Linux Interface Specification Device Driver I3C User's Manual: Software
------------------	--

Rev.	Date	Description	
		Page	Summary
1.00	Jun 30, 2025	—	First Edition issued
1.01	Jul 22, 2025	—	Merge the document with RZ/G3E document

---

Linux Interface Specification Device Driver I3C  
User's Manual: Software

Publication Date: Rev. 1.01 Jul. 22, 2025

Published by: Renesas Electronics Corporation

---

RZ/V2N Group and RZ/G3E Group