

# RZ/V Verified Linux Package

## Software Manual for RZ/V2MA

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.5.0-1 October 2020)

## 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 document is designed to provide the user with an understanding of the software development environment for RZ/V2MA Group processors. It is intended for users developing software incorporating the processors. Basic knowledge of software development and Linux systems is necessary 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.

## 2. List of Abbreviations and Acronyms

Abbreviation	Full form
BSP	Board support package
CSI	Clock serial interface
DMA	Direct memory access
eMMC	Embedded multi-media card
GPIO	General purpose input/output interface
GPL	GNU General Public License
I2C	Inter-integrated circuit
LGPL	GNU Lesser General Public License
PCI	Peripheral component interconnect
PCIe	PCI Express
PFC	Pin function controller
PWM	Pulse width modulation
SDHC	SD high capacity
SDHI	SD card host interface H/W module
SDIO	SD input/output
SDXC	SD extended capacity
TSU	Thermal sensor unit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

### 3. Conventions

Command line runs on Linux host PC will be shown as below:

```
$ echo "This is command line run on x86-64 Linux PC"
```

Command line run on target board will be shown as below:

```
# echo "This is command line run on ARM board"
```

File content will be shown as below:

```
<$WORK/a script>  
#!/bin/bash  
echo "This is content in a file"
```

# Table of Contents

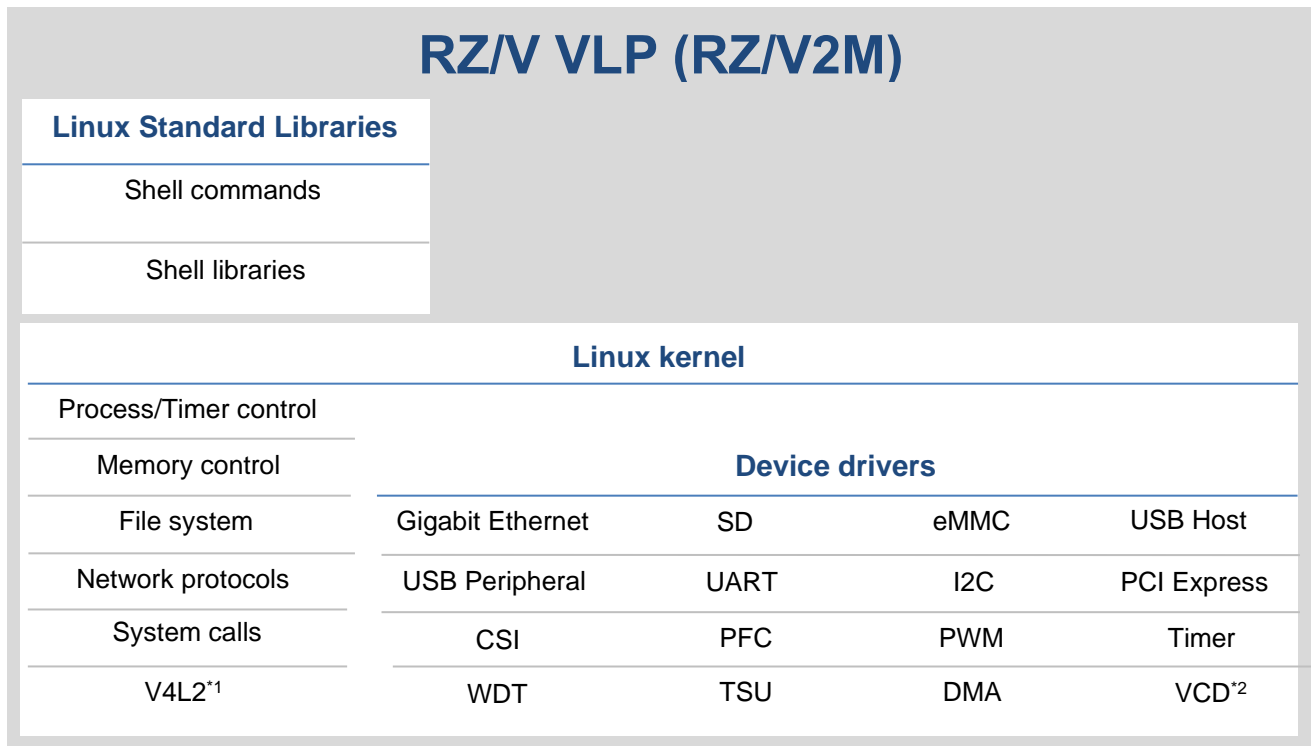
1.	Overview.....	1
1.1	Overview of RZ/V VLP (RZ/V2M) .....	2
2.	RZ/V VLP (RZ/V2M) Main Functions .....	3
2.1	Linux Kernel .....	3
2.1.1	Linux System Call.....	3
2.1.2	Process/Timer Control .....	3
2.1.3	Memory Control .....	3
2.1.4	File System .....	3
2.1.5	Device Driver .....	3
2.1.6	Network Protocols.....	3
2.2	Linux Standard Libraries .....	4
2.3	Linux Shell.....	5
2.4	Device Drivers.....	6
2.4.1	Gigabit Ethernet.....	6
2.4.2	SD.....	7
2.4.3	eMMC .....	9
2.4.4	USB Host .....	11
2.4.5	USB Peripheral .....	12
2.4.6	UART .....	14
2.4.7	I2C .....	15
2.4.8	PCI Express.....	17
2.4.9	CSI.....	18
2.4.10	PFC.....	19
2.4.11	PWM .....	25
2.4.12	Timer.....	26
2.4.13	Watchdog Timer .....	27
2.4.14	TSU.....	28
2.4.15	DMA.....	29
2.5	V4L2.....	30
2.5.1	API .....	30
2.5.2	Software Interface.....	30
2.6	Yocto environment, root File System and SDK .....	31
3.	U-Boot.....	32
3.1	Supported Functions.....	32

# 1. Overview

Renesas provides the RZ/V Verified Linux Package (hereafter as “RZ/V VLP (RZ/V2MA)”) software as the integrated Linux package.

The Linux system in this package consists of a Linux kernel, Linux Standard Library, and Linux shell.

This Linux package has checked the operation and managed this version by Renesas.



**Figure 1-1. RZ/V VLP (RZ/V2M)**

Note1: V4L2 in this package provides functions required for using a USB/MIPI camera.

Note2: Provided in the RZ/V2MA Video Codec Package.

## 1.1 Overview of RZ/V VLP (RZ/V2M)

The main software in this package is as follows.

Refer to "RZ/V Verified Linux Package Release Note" (hereafter as "release note") for each version of the software.

**Table 1-1. RZ/V VLP (RZ/V2M) software components**

Component	Software name	Summary	Section
Linux Kernel	CIP SLTS Linux kernel	Linux kernel proposed by CIP project.	2.1
Linux Standard Libraries	busybox	A Program that combines UNIX utilities into a small execute file.	2.2
	binutils	Binary tool sets.	2.2
	glibc	GNU standard C library.	2.2
	openssl	Cryptographic library with the TLS/SSL protocol.	2.2
	(Various software other than above)	Building system provided by Yocto Project.	2.6
Video	V4L2	The name of a popular API used in the handling of moving image data under the Linux OS.	2.5

The following table shows the device drivers in this package.

**Table 1-2. Device drivers in RZ/V VLP (RZ/V2M)**

Driver	Summary	Section
Gigabit Ethernet	Transmit/Receive Ethernet frames.	2.4.1
SD	Read/Write SD cards.	2.4.2
eMMC	Read/Write eMMC.	2.4.3
USB Host	Transmit/Receive data between USB devices as USB Host function.	2.4.4
USB Peripheral	Transmit/Receive data between USB devices as USB Peripheral function.	2.4.5
UART	Serial communication by UART.	2.4.6
I2C	Transmit/Receive data between connected devices on the I2C bus.	2.4.7
PCI Express	Communicate by PCI express.	2.4.8
CSI	Clock synchronism serial transmission.	2.4.9
PFC	Switch the dual function of LSI external pins and configure the setting of GPIO pins.	2.4.10
PWM	Output PWM signal.	2.4.11
Timer	Configure the setting of the interval timer.	2.4.12
Watchdog Timer	Configure the setting of the watchdog timer.	2.4.13
TSU	Get the LSI temperature output from the temperature sensor.	2.4.14
DMA	Control the DMA controller on the RZ/V2MA.	2.4.15



## 2. RZ/V VLP (RZ/V2M) Main Functions

This chapter explains the major functions of the software included in this package.

### 2.1 Linux Kernel

The Linux kernel is the core of the UNIX-like Linux OS. The Linux kernel is in its own kernel space, instead of in the user space which holds applications.

The RZ/V VLP (RZ/V2M) provides CIP SLTS Linux kernel which is managed by Civil Infrastructure Platform (CIP) Project.

The Linux kernel consists of the several components described below.

#### 2.1.1 Linux System Call

A Linux system call is the interface between the user space and the Linux kernel. It allows applications to handle the components described below.

#### 2.1.2 Process/Timer Control

A process is a running program. In Linux, processes have a one-to-one relation with threads, so there is no distinction between the two.

The Linux kernel controls multiple threads by running respectively for a specified time and then running the next. The time over which a thread run is controlled by interrupts generated by a hardware timer.

#### 2.1.3 Memory Control

The Linux kernel allocates memory for each process. At this time, the kernel allocates memory from a virtual area called the virtual memory space.

#### 2.1.4 File System

OSs generally handles data as files. A File system is a system for abstracting and controlling files.

Linux handles not only data but also certain devices (such as mouse and keyboard devices) as files. This file system is referred to as the virtual file system (VFS). The VFS enables Linux to handle data and devices through a common interface for access to files.

#### 2.1.5 Device Driver

Device drivers are software modules to operate the hardware devices. Device drivers are divided into two types according to whether they are independent of or dependent on the machine's architecture. The former is called the board support package (BSP). Besides, the source code for those which are independent of the architecture is under `./linux/drivers`, and that for the others (depending) under `./linux/arch`.

Refer to section 2.4 for details on the device drivers in this Linux package.

#### 2.1.6 Network Protocols

The Linux kernel supports multiple protocols, including the Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) as network functions.

## 2.2 Linux Standard Libraries

A library is a collection of often-used functions that can be used by other computer programs. The two types of library Linux supports are static and shared libraries.

Static libraries, with “.a” as the filename extension, are included in executable files when programs are created. Static libraries have merits in their higher speed of execution and that operation will not change once a program has been compiled. On the other hand, their drawback is that including static libraries in all programs leads to the programs being bigger.

Shared libraries, with “.so” followed by a period and the version number as the filename extension, are only loaded when a program is executed. Shared libraries have the merit that access by multiple programs to a shared library leads to the programs being smaller. On the other hand, one drawback is that updating a library can affect all functions for which the library is in use.

- Relationship between links and libraries

The means for a program to call the functions of libraries is referred to as linking. There are two types of linking, static and dynamic.

In static linking, the functions of a library are built into the main body of the program in advance when a program is created. Static libraries are those for which building is with the use of static linking.

In dynamic linking, the functions of a library are called during the execution of a program. Shared libraries are those in which linking takes the form of setting up such calls.

Linux standard libraries in this package is based on CIP Core. Software components not included in CIP Core are used that is provided by Yocto Project.

CIP Core provides the following software. Please refer to the release note or component list about each software version.

- glibc
- binutils \*Unincluded in the root file system.
- Busybox
- openssl

## 2.3 Linux Shell

Linux shell is a program bridging the interfaces for operating the actual Linux operation.

In running processes under Linux, users enter character strings through a shell, which provides a character user interface (CUI), or use a graphical user interface (GUI). These handles processing by the Linux kernel, which serves as the core of the Linux OS.

However, how processing by CUI or GUI relates to the Linux kernel itself cannot be directly understood. In addition, most results of processing by the Linux kernel are output informs that users cannot understand without intermediary interpretation.

For these reasons, an appropriate program is required to mediate between users and the Linux kernel and to convert data to suit the respective forms in which they can understand the data. The Linux shell is that program.

There are various types of Linux shell. The RZ/V VLP (RZ/V2M) uses the Bourne Again Shell (bash).

## 2.4 Device Drivers

This package provides device drivers for controlling the hardware functions on RZ/V2MA and various devices on the board. This subsection describes the main device drivers in this package.

### 2.4.1 Gigabit Ethernet

This module controls the Ethernet MAC interface on RZ/V2MA to handle the transfer/receive of Ethernet frame data on Linux.

#### 2.4.1.1 Hardware Configuration

**Table 2-1. Hardware configuration (Gigabit Ethernet)**

RZ/V2MA
1ch (ETH0)

#### 2.4.1.2 Function

This module provides the following functions.

- Comply with IEEE802.3
- Transmission and reception of Ethernet frame
- Transfer rate: 100 Mbps, 1000 Mbps
- Transmission mode: Full-duplex mode

#### 2.4.1.3 Software Interface

The function of this module is provided as a standard Linux network interface. It can be used various network functions of Linux by specifying the following network interface.

**Table 2-2. Network interface (Gigabit Ethernet)**

Network interface	Summary
eth0	The default configuration sets to eth0.

#### 2.4.1.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver Gigabit Ethernet User's Manual: Software

## 2.4.2 SD

This module controls the SD/MMC Host Controller on the RZ/V2MA to perform reading/writing and DMA transfer to an SD card.

### 2.4.2.1 Hardware Configuration

**Table 2-3. Hardware configuration (SD)**

<b>RZ/V2MA</b>
2ch* (SDI0, SDI1)

Note: RZ/V2MA Evaluation Board Kit has only one connector for SDI0.

### 2.4.2.2 Description of Functionality

This module supports the following functions.

**Table 2-4. Supported functions (SD)**

Function	Support status
SD memory Card	SD, SDHC, SDXC <sup>*1</sup>
SDIO Card	Supported <sup>*3</sup>
Transfer Modes	1bit, 4bit
Bus Speed Mode <sup>*2</sup>	Default Speed (DS), High Speed (HS), UHS-I
DMA Transfer	Supported the internal DMA
Card Power Control	Supported
Card Detection (CD)	Supported
Card Detection (DAT3)	Unsupported
Write Protection	Supported <sup>*3</sup>
SPI Mode	Unsupported
SD Mechanical Write Protect Switch	Supported <sup>*3</sup>
SD Clock Control	Supported
CPRM Security	Unsupported
CMD52 During Data Transfer(C52PUB)	Unsupported
Data Transfer Abort (IOABT)	Unsupported
Read Wait (RWREQ)	Unsupported
SDIO Wake Up	Unsupported
SDIO Suspend/Resume	Unsupported

Notes: \*1. SDXC memory cards that are formatted with the exFAT cannot be mounted because BSP standard file system does not support the exFAT.

\*2. DDR50 is unsupported.

\*3. RZ/V2MA Evaluation Board Kit does not support this function.

The following table shows the list of SD standard documents related to this module.

**Table 2-5. SD standard document**

Document No.	Publication	Document title	Revision	Date
-	SD Card Association	SD Specifications Part 1 Physical Layer Specification Simplified Specification	4.01	Jan. 22, 2013
-	SD Card Association	SD Specifications Part E1 SDIO Simplified Specification	3.00	Feb. 25, 2011

### 2.4.2.3 Software Interface

The function of this module is provided as a standard Linux device interface. It can be used the SD function on Linux by specifying the following device file.

**Table 2-6. Device file (SD)**

Device file	Summary
/dev/mmcblkN*	The default configuration sets as bellow: SDI0: /dev/mmcblkN* SDI1: /dev/mmcblkN*

Note: The numerical value N might be different according to the system. (ex: /dev/mmcblk0)

### 2.4.2.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver SD/eMMC User's Manual: Software

### 2.4.3 eMMC

This module controls the SD/MMC Host Controller on RZ/V2MA to read from or write into an eMMC memory and to proceed with DMA transfer.

#### 2.4.3.1 Hardware Configuration

**Table 2-7. Hardware configuration (eMMC)**

<b>RZ/V2MA</b>
1ch (eMMC)

#### 2.4.3.2 Description of Functionality

This module supports the following functions.

**Table 2-8. Supported functions (eMMC)**

Function	Support status
eMMC device access	Supported
Transfer modes	1,4,8bit
Bus speed modes*	High-Speed, HS200, Backward compatible
DMA transfer	Supported internal DMA
Stream transfer	Unsupported
Open-ended / Multi block transfer	Unsupported

Note: High Speed DDR and HS400 are unsupported.

The following table shows the list of eMMC standard documents related to this module.

**Table 2-9. eMMC standard document**

Document No.	Publication	Document Title	Revision	Date
JESD84-B51	JEDEC STANDARD	EMBEDDED MULTI-MEDIA CARD (e•MMC), ELECTRICAL STANDARD (5.1)	5.1	2013.09

### 2.4.3.3 Software Interface

The function of this module is provided as a standard Linux device interface. It can be used the eMMC function on Linux by specifying the following device file.

**Table 2-10. Device file (eMMC)**

Device file	Summary
/dev/mmcblkN*	The default configuration sets as bellow: eMMC: /dev/mmcblkN*

Note: The numerical value N might be different according to the system. (ex: /dev/mmcblk0)

### 2.4.3.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver SD/eMMC User's Manual: Software



## 2.4.4 USB Host

This module controls the USB host controller on RZ/V2MA to transfer data to/from USB devices. This module complies with the USB3.1 Gen1 standard.

### 2.4.4.1 Hardware Configuration

**Table 2-11. Hardware configuration (USB Host)**

RZ/V2MA
1ch* (USB)

Note: USB Host is used exclusivity with USB Peripheral.

### 2.4.4.2 Description of Functionality

This module provides the following functions.

- Transfer: Super-Speed(5Gbps)/High-Speed (480Mbps)/Full-Speed\*1 (12Mbps)/Low-Speed\*1 (1.5Mbps)
- Transfer types: Isochronous/Interrupt\*1/Control/Bulk
- High bandwidth transfer: Isochronous/Interrupt
- Comply with eXtensible Host Controller Interface (xHCI) Specification for USB

The module supports the USB On-The-Go.

The following list shows the USB standard documents related to this module.

- Universal Serial Bus 3.1 Specification Revision 1.0 and ECNs approved through June 27, 2017
- Universal Serial Bus Specification Revision 2.0

### 2.4.4.3 Software Interface

The function of this module is provided as a standard Linux device interface. After connecting to a USB device, USB devices can be controlled on Linux by specifying the following device file.

USB devices can be enabled through kernel configuration.

**Table 2-12. Device file (USB Host)**

Device file	Summary
/dev/input/eventX*	Device files compatible with keyboard.
/dev/input/mouseY*	Device files compatible with mouse.
/dev/sdZ*	Device files compatible with USB storage.

Note: The numerical value X, Y, and Z might be different according to the system. (ex: /dev/input/event0)

### 2.4.4.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver USB Host User's Manual: Software

## 2.4.5 USB Peripheral

This module controls the USB peripheral controller on RZ/V2MA to transfer data to/from USB devices. This module complies with USB3.1 Gen1 standard.

### 2.4.5.1 Hardware Configuration

**Table 2-13. Hardware configuration (USB Peripheral)**

RZ/V2MA
1ch* (USB)

Note: USB Peripheral is used exclusivity with USB Host.

### 2.4.5.2 Description of Functionality

This module provides the following functions.

- Transfer: Super-Speed (5Gbps)/High-Speed (480Mbps)/Full-Speed (12Mbps)
- Transfer type: Interrupt/Control/Bulk
- Each PIPE can be set below.
  - [USB 3.1]
    - Number of endpoints: 0 to 8
    - Transfer type: Control (PIPE0 only), Bulk, Interrupt
    - Transfer direction: IN/OUT
      - IN: ep1(PIPE1), ep3(PIPE3), ep5(PIPE5), ep7(PIPE7)
      - OUT: ep2(PIPE2), ep4(PIPE4), ep6(PIPE6), ep8(PIPE8)
    - Maximum packet size: 512byte (PIPE0), 1024byte (others) \*Unsupported for high bandwidth
  - [USB 2.0]
    - Number of endpoints: 0 to 8
    - Transfer type: Control (PIPE0 only), Bulk, Interrupt
    - Transfer direction: IN/OUT
      - IN: ep1(PIPE1), ep3(PIPE3), ep5(PIPE5), ep7(PIPE7)
      - OUT: ep2(PIPE2), ep4(PIPE4), ep6(PIPE6), ep8(PIPE8)
    - Maximum packet size: 64byte (PIPE0), 1024byte (others) \*Unsupported for high bandwidth

This module supports the USB On-The-Go.

The following list shows the USB standard documents related to this module.

- Universal Serial Bus 3.1 Specification Revision 1.0 and ECNs approved through June 27, 2017
- Universal Serial Bus Specification Revision 2.0

### 2.4.5.3 Software Interface

This module provides a standard Linux interface for implementing the function as USB peripheral.

The following table shows an example of a device file. The supported USB devices can be enabled through kernel configuration.

**Table 2-14. Device file (USB Peripheral)**

Device file	Summary
/dev/ttyGS0	Device files for serial communication on a USB port.

### 2.4.5.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver USB Peripheral User's Manual: Software

## 2.4.6 UART

This module controls UART on RZ/V2MA to transfer data over RS232C links and control the settings for communications.

### 2.4.6.1 Hardware Configuration

**Table 2-15. Hardware configuration (UART)**

RZ/V2MA
2ch (UART0, UART1)

### 2.4.6.2 Description of Functionality

This module provides the following functions. \*

- Data transfer over RS232C links
- Control of the settings for communications
  - The following baud rate are supported: 9600, 19200, 38400, 57600, and 115200 bps
  - Parity bit: no parity/odd/even
  - Stop bits: 1 or 2 bit
  - Bit width for transfer: 7 or 8 bit
- Hardware Flow control

Note: This Linux system uses ch0 standardly as the UART for console connection.

The following settings are executed after starting up Linux immediately:

Baud rate: 115200bps, Parity bit: None, Stop bit: 1bit, Bid width for transfer: 8bit

### 2.4.6.3 Software Interface

The function of this module is provided as a standard Linux device interface. It can be used for UART communication on Linux by specifying the following device file.

**Table 2-16. Device file (UART)**

Device file	Summary
/dev/ttyS*	The default configuration sets as bellow: UART0: /dev/ttyS0 UART1: /dev/ttyS1

### 2.4.6.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver UART User's Manual: Software

## 2.4.7 I2C

This module controls the I2C interface on RZ/V2MA to transmit data between the slave devices on the I2C bus.

### 2.4.7.1 Hardware Configuration

**Table 2-17. Hardware configuration (I2C)**

<b>RZ/V2MA</b>
4ch (I2C0~I2C3)

### 2.4.7.2 Description of Functionality

This module provides the following functions.

**Table 2-18. Supported functions (I2C)**

Function	Support status
Master mode	Supported
Slave mode	Unsupported
Serial clock (SCL)	Settable for "High" level width and "Low" level width

This module supports I2C transfer speed as follows.

**Table 2-19. Transfer speed (I2C)**

Interface mode	Real transfer speed	Support status
Standard mode (100KHz)	Max 100Kbit/s	Supported.
Fast mode (400KHz)	Max 400Kbit/s	Supported.
High speed mode (3.4MHz)	Max 3400Kbit/s	Unsupported.

This module complies with the following standard.

**Table 2-20. I2C standard document**

Document No.	Publication	Document title	Revision	Date
-	Philips Semiconductors	THE I2C-BUS SPECIFICATION	2.1	2000.01

### 2.4.7.3 Software Interface

The function of this module is provided as a standard Linux interface. I2C communication can be executed on Linux by specifying the following device file.

**Table 2-21. Device file (I2C)**

Device file	Summary
/dev/i2c-*	The default configuration sets as bellow: I2C0: /dev/i2c-0 I2C1: /dev/i2c-1 I2C2: /dev/i2c-2 I2C3: /dev/i2c-3

### 2.4.7.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver I2C User's Manual: Software

## 2.4.8 PCI Express

This module controls PCI Express (PCIe) interface on RZ/V2MA and performs data communication. This module complies with PCIe Gen2.

### 2.4.8.1 Hardware Configuration

**Table 2-22. Hardware configuration (PCIe)**

RZ/V2MA
2 lane (PCI)

Note: PCI Express cannot be used on the RZ/V2MA Evaluation Board Kit.

### 2.4.8.2 Description of Functionality

This module provides the following function.

- Supported configuration: Root complex, Endpoints
- Transfer rate: 5GT/s
- Host initiated IO (legacy) reads and writes
- Host initiated memory reads and writes
- Slave initiated memory reads and writes
- Interrupt: INTx, MSI

This module complies with the following standard.

- PCI Express Base Specification 4.0

### 2.4.8.3 Software Interface

PCI Express communication is executed on Linux by specifying the following device file.

**Table 2-23. Device file (PCIe)**

Device file	Summary
/sys/class/pci_bus	The default configuration sets as bellow: PCI: /sys/class/pci_bus

### 2.4.8.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver PCI Express User's Manual: Software

## 2.4.9 CSI

This module controls CSI interface on RZ/V2MA and performs serial data transmission.

### 2.4.9.1 Hardware Configuration

**Table 2-24. Hardware configuration (CSI)**

<b>RZ/V2MA</b>
6ch (CSI0~CSI5)

### 2.4.9.2 Description of Functionality

This module provides the following function.

**Table 2-25. Supported functions (CSI)**

Function	Support status
Data transfer mode	Master reception only mode, Master transmission/reception mode, Slave reception only mode, Slave transmission/reception mode
Serial data length	8, 16bit
Initial bit	MSB/LSB * Default setting is MSB.
Data transfer method	Interrupt
Serial clock	(Master mode) Max 48 MHz, divided from 2 to 32,766 (Slave mode) Max 24 MHz
Transmit/Receive FIFO	16bit (8bit + 8bit) * 16 stage
Waiting time between serial data	Settable in master mode

### 2.4.9.3 Software Interface

CSI communication is executed on Linux by specifying the following device file.

**Table 2-26. Device file (CSI)**

Device file	Summary
/dev/spidev*.0	The default configuration sets as bellow: CSIN: /dev/spidevN.0* * N is changed depending on the number of using CSI channels. (N=0~5)

### 2.4.9.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver CSI User's Manual: Software



## 2.4.10 PFC

This module controls the pin function controller (PFC) on RZ/V2MA to switch the shared function of LSI external pins and set general purpose input/output interface (GPIO).

### 2.4.10.1 Hardware Configuration

#### (1) Supported pin

Available GPIO pins on RZ/V2MA are shown in the table below. The shared pins setting initially in this BSP are described in bold. Refer to the “RZ/V2MA Hardware Manual” for the detail of each shared pin.

**Table 2-27. IO Group PORT00**

External pin	Dual-purpose pin0	Dual-purpose pin 1	Dual-purpose pin 2
	GPIO	-	eMMC
P0008	<b>P00_08</b>	-	-
P0009	<b>P00_09</b>	-	-
P0012	<b>P00_12</b>	-	-
P0013	<b>P00_13</b>	-	-

**Table 2-28. IO Group PORT01**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	PWM	External interruption
PM0	P01_00	<b>PM0</b>	INEXINT8
PM1	P01_01	<b>PM1</b>	INEXINT9
PM2	P01_02	<b>PM2</b>	INEXINT10
PM3	P01_03	<b>PM3</b>	INEXINT11
PM4	P01_04	<b>PM4</b>	INEXINT12
PM5	P01_05	<b>PM5</b>	INEXINT13
PM6	P01_06	<b>PM6</b>	INEXINT14
PM7	P01_07	<b>PM7</b>	INEXINT15
PM8	P01_08	<b>PM8</b>	INEXINT16
PM9	P01_09	<b>PM9</b>	INEXINT17
PM10	P01_10	<b>PM10</b>	INEXINT18
PM11	P01_11	<b>PM11</b>	INEXINT19
PM12	P01_12	<b>PM12</b>	INEXINT20
PM13	P01_13	<b>PM13</b>	INEXINT21
PM14	P01_14	<b>PM14</b>	INEXINT22
PM15	P01_15	<b>PM15</b>	INEXINT23

**Table 2-29. IO Group PORT02**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	External interruption
INEXINT0	P02_00	-	<b>INEXINT0</b>
INEXINT1	P02_01	-	<b>INEXINT1</b>
INEXINT2	P02_02	-	<b>INEXINT2</b>
INEXINT3	P02_03	-	<b>INEXINT3</b>
INEXINT4	P02_04	-	<b>INEXINT4</b>
INEXINT5	P02_05	-	<b>INEXINT5</b>
INEXINT6	P02_06	-	<b>INEXINT6</b>
INEXINT7	P02_07	-	<b>INEXINT7</b>

**Table 2-30. IO Group PORT03**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2	Dual-purpose pin3
	GPIO	CSI0, CSI1, CSI2, CSI3	UART0, UART1, IIC2, IIC3, CSI3	CSI0, CSI1, CSI2, CSI3
CSTXD0	P03_00	CSTXD0	<b>UATX0</b>	CSRXD0
CSRXD0	P03_01	CSRXD0	<b>UARX0</b>	-
CSSCLK0	P03_02	CSSCLK0	<b>UACTS0N</b>	-
CSCS0	P03_03	CSCS0	<b>UARTS0N</b>	-
CSTXD1	P03_04	CSTXD1	<b>UATX1</b>	CSRXD1
CSRXD1	P03_05	CSRXD1	<b>UARX1</b>	-
CSSCLK1	P03_06	CSSCLK1	<b>UACTS1N</b>	-
CSCS1	P03_07	CSCS1	<b>UARTS1N</b>	-
CSTXD2	P03_08	<b>CSTXD2</b>	I2SDA2	CSRXD2
CSRXD2	P03_09	<b>CSRXD2</b>	I2SCL2	-
CSSCLK2	P03_10	<b>CSSCLK2</b>	I2SDA3	-
CSCS2	P03_11	<b>CSCS2</b>	I2SCL3	-
CSTXD3	P03_12	<b>CSTXD3</b>	CSRXD3	-
CSRXD3	P03_13	<b>CSRXD3</b>	-	-
CSSCLK3	P03_14	<b>CSSCLK3</b>	-	-
CSCS3	P03_15	<b>CSCS3</b>	-	-

**Table 2-31. IO Group PORT04**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	CSI4, CSI5	CSI4, CSI5
CSTXD4	P04_00	<b>CSTXD4</b>	CSRXD4
CSRXD4	P04_01	<b>CSRXD4</b>	-
CSSCLK4	P04_02	<b>CSSCLK4</b>	-
CSCS4	P04_03	<b>CSCS4</b>	-
CSTXD5	P04_04	<b>CSTXD5</b>	CSRXD5
CSRXD5	P04_05	<b>CSRXD5</b>	-
CSSCLK5	P04_06	<b>CSSCLK5</b>	-
CSCS5	P04_07	<b>CSCS5</b>	-

**Table 2-32. IO Group PORT05**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	IIC0, IIC1
I2SDA0	P05_00	-	<b>I2SDA0</b>
I2SCL0	P05_01	-	<b>I2SCL0</b>
I2SDA1	P05_02	-	<b>I2SDA1</b>
I2SCL1	P05_03	-	<b>I2SCL1</b>

**Table 2-33. IO Group PORT06**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	-
P0600	<b>P06_00</b>	-	-
P0601	<b>P06_01</b>	-	-
P0602	<b>P06_02</b>	-	-
P0603	<b>P06_03</b>	-	-
P0604	<b>P06_04</b>	-	-
P0605	<b>P06_05</b>	-	-
P0606	<b>P06_06</b>	-	-
P0607	<b>P06_07</b>	-	-
P0608	<b>P06_08</b>	-	-
P0609	<b>P06_09</b>	-	-
P0610	<b>P06_10</b>	-	-
P0611	<b>P06_11</b>	-	-

**Table 2-34. IO Group PORT07**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	-
P0700	<b>P07_00</b>	-	-
P0701	<b>P07_01</b>	-	-
P0702	<b>P07_02</b>	-	-
P0703	<b>P07_03</b>	-	-
P0704	<b>P07_04</b>	-	-
P0705	<b>P07_05</b>	-	-

**Table 2-35. IO Group PORT08**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	SDIO	-
SD0CMD	P08_00	<b>SD0CMD</b>	-
SD0CLK	P08_01	<b>SD0CLK</b>	-
SD0DAT0	P08_02	<b>SD0DAT0</b>	-
SD0DAT0	P08_03	<b>SD0DAT0</b>	-
SD0DAT2	P08_04	<b>SD0DAT2</b>	-
SD0DAT3	P08_05	<b>SD0DAT3</b>	-
SD0WP	P08_06	<b>SD0WP</b>	-
SD0CD	P08_07	<b>SD0CD</b>	-

Table 2-36. IO Group PORT09

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	SDI1	External interruption
SD1FCMD	P09_00	<b>SD1FCMD</b>	-
SD1FCLK	P09_01	<b>SD1FCLK</b>	-
SD1FDAT0	P09_02	<b>SD1FDAT0</b>	-
SD1FDAT1	P09_03	<b>SD1FDAT1</b>	-
SD1FDAT2	P09_04	<b>SD1FDAT2</b>	-
SD1FDAT3	P09_05	<b>SD1FDAT3</b>	-
SD1FWP	P09_06	<b>SD1FWP</b>	INEXINT24
SD1FCD	P09_07	<b>SD1FCD</b>	INEXINT25

Table 2-37. IO Group PORT10

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	SDI1	External interruption
P1000	<b>P10_00</b>	-	—
P1001	<b>P10_01</b>	-	—
P1002	<b>P10_02</b>	-	—
P1003	<b>P10_03</b>	-	—
P1004	<b>P10_04</b>	-	—
P1005	<b>P10_05</b>	-	—
P1006	<b>P10_06</b>	-	INEXINT26
P1007	<b>P10_07</b>	-	INEXINT27
P1008	<b>P10_08</b>	-	—

Table 2-38. IO Group PORT11

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	External interruption
P1100	<b>P11_00</b>	-	-
P1101	<b>P11_01</b>	-	-
P1102	<b>P11_02</b>	-	-
P1103	<b>P11_03</b>	-	-
P1104	<b>P11_04</b>	-	-
P1105	<b>P11_05</b>	-	-
P1106	<b>P11_06</b>	-	INEXINT28
P1107	<b>P11_07</b>	-	INEXINT29
P1108	<b>P11_08</b>	-	-

Table 2-39. IO Group PORT12

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	External interruption
P1200	<b>P12_00</b>	-	INEXINT30
P1201	<b>P12_01</b>	-	INEXINT31
P1202	<b>P12_02</b>	-	INEXINT32
P1203	<b>P12_03</b>	-	INEXINT33

**Table 2-40. IO Group PORT13**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	External interruption
P1300	<b>P13_00</b>	-	-
P1301	<b>P13_01</b>	-	-
P1302	<b>P13_02</b>	-	-
P1303	<b>P13_03</b>	-	-
P1304	<b>P13_04</b>	-	-
P1305	<b>P13_05</b>	-	-
P1306	<b>P13_06</b>	-	-
P1307	<b>P13_07</b>	-	-
P1308	<b>P13_08</b>	-	-
P1309	<b>P13_09</b>	-	INEXINT34
P1310	<b>P13_10</b>	-	INEXINT35
P1311	<b>P13_11</b>	-	INEXINT36

**Table 2-41. IO Group PORT14**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	External interruption
P1400	<b>P14_00</b>	-	-
P1401	<b>P14_01</b>	-	-
P1402	<b>P14_02</b>	-	INEXINT37
P1403	<b>P14_03</b>	-	-
P1404	<b>P14_04</b>	-	-
P1405	<b>P14_05</b>	-	-
P1406	<b>P14_06</b>	-	INEXINT38
P1407	<b>P14_07</b>	-	-

**Table 2-42. IO Group PORT17**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	-
P1700	<b>P17_00</b>	-	-

**Table 2-43. IO Group PORT20**

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	-
P2000	<b>P20_00</b>	-	-
P2001	<b>P20_01</b>	-	-
P2002	<b>P20_02</b>	-	-

### 2.4.10.2 Description of Functionality

This module provides the following function.

- Control GPIO.
  - Select input/output of GPIO pin.
  - Read a state of high/low in an input pin.
  - Set high/low in an output pin.
- Switch between dual-purpose pins.
- Control Pull-up/down of a dual-purpose pin.
- Switch drive capability of LSI pin.
- Switch slew-rate of a dual-purpose pin.
- Mask control for an external interruption signal.
- Inverse the polarity of an external interruption signal.
  - Specify the inverse of polarity when the external interruption signal is low-active.

### 2.4.10.3 Software Interface

This module function is executed on Linux by specifying the following device file.

**Table 2-44. Device file (PFC)**

Device file	Summary
/sys/class/gpio/gpioN*	The default configuration sets as bellow: gpioN: /sys/class/gpio/gpioN*

Note: N indicates each pin number.

### 2.4.10.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver PFC User's Manual: Software

## 2.4.11 PWM

This module sets pulse width modulation timer (PWM) on RZ/V2MA and outputs a PWM signal.

### 2.4.11.1 Hardware Configuration

**Table 2-45. Hardware configuration (PWM)**

<b>RZ/V2MA</b>
16ch (PWM0-PWM15)

### 2.4.11.2 Description of Functionality

This module provides the following function.

- Output continuous high-level width of signal\*
- When PWM\_CLK = 48MHz, PWM signal is settable as follows:
  - Period: 2.083μs - 0.416s (2.4 Hz -> 480 kHz)
  - Minimum resolution: 21ns
  - Duty rate: 0% - 100%
- Invert the PWM signal level (Low/High)

### 2.4.11.3 Software Interface

This module sets and outputs a PWM signal on Linux by specifying the following PWM sysfs interfaces.

**Table 2-46. Device file (PWM)**

Device file	Summary
/sys/class/pwm/pwmchip*/pwm0	The default configuration sets as below: PWMN: /sys/class/pwm/pwmchipN/pwm0 (N=0~15) (e.g.) PWM0: /sys/class/pwm/pwmchip0/pwm0 PWM1: /sys/class/pwm/pwmchip1/pwm0 .... PWM15: /sys/class/pwm/pwmchip15/pwm0

### 2.4.11.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver PWM User's Manual: Software

### 2.4.12 Timer

This module configures the setting of interval timer by the timer (TIM) on RZ/V2MA.

#### 2.4.12.1 Hardware Configuration

**Table 2-47. Hardware configuration (TIM)**

<b>RZ/V2MA</b>
32ch (TIM0-TIM31)

#### 2.4.12.2 Description of Functionality

This module provides the following function.

- Run a timer with the 32-bit counter by specifying the timer mode (interval or free-running).

#### 2.4.12.3 Software Interface

This module sets and outputs a timer signal on Linux by specifying the following interfaces.

**Table 2-48. Device file (TIM)**

Device file	Summary
/dev/hw_tim*	The default configuration sets as bellow: TIMN: /dev/hw_timN (N=0~31) (e.g) TIM0: /dev/hw_tim0 TIM1: /dev/hw_tim1 ... TIM31: /dev/hw_tim31

#### 2.4.12.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver Timer User's Manual: Software



### 2.4.13 Watchdog Timer

This module configures the setting of watchdog timer (WDT) on RZ/V2MA.

#### 2.4.13.1 Hardware Configuration

**Table 2-49. Hardware configuration (WDT)**

<b>RZ/V2MA</b>
2ch* (WDT0, WDT1)

Note: WDT0 and WDT1 are respectively for CA53 core 0 and core 1.

#### 2.4.13.2 Description of Functionality

This module provides the following function.

- Set the watchdog timer with a 32-bit counter driven by WDT\_CLK (48 MHz).
- The value of the periodic counter is settable from 1s to 89s.
- The interrupt occurs with the specified period. WDT is cleared before the WDT counter overflows.

#### 2.4.13.3 Software Interface

This module sets and outputs a timer signal on Linux by specifying the following interfaces.

**Table 2-50. Device file (WDT)**

Device file	Summary
/dev/watchdog*	The default configuration sets as bellow: WDT0: /dev/watchdog0 WDT1: /dev/watchdog1

#### 2.4.13.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver WDT User's Manual: Software

## 2.4.14 TSU

This module configures the setting of the temperature sensor unit (TSU) on RZ/V2MA and gets the temperature of LSI output from this sensor.

### 2.4.14.1 Hardware Configuration

**Table 2-51. Hardware configuration (TSU)**

RZ/V2MA
1ch* (TSU0, TSU1) *

### 2.4.14.2 Description of Functionality

This module provides the following functions.

- Get the LSI temperature. \*  
Note: Comparing the temperature from TSU0 and TSU1, the higher value is the LSI temperature.
- Shut down the system when the LSI temperature exceeds the prespecified threshold.

### 2.4.14.3 Software Interface

The function of this module is provided as a standard Linux thermal sysfs interface.

**Table 2-52. Device file (TSU)**

Device file	Summary
/sys/class/thermal/thermal_zone0	The default configuration sets as bellow:

### 2.4.14.4 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver TSU User's Manual: Software

## 2.4.15 DMA

This module controls the DMA controller on the RZ/V2MA.

### 2.4.15.1 Hardware Configuration

**Table 2-53. Hardware configuration (DMA)**

<b>RZ/V2MA</b>
16ch

### 2.4.15.2 Description of Functionality

This module provides the following functions.

- DMA device driver module management with DMA Engine framework.
- Control each channel of support DMAC.
- Call the registered callback function when completing the DMA transfer.
- Control the data transfer between memory to memory.
- Control the descriptor transfer.

### 2.4.15.3 Device Driver Specification

Refer to the following documentation for details.

- Linux Interface Specification Device Driver DMA User's Manual: Software

## 2.5 V4L2

Video for Linux 2 (V4L2) is a name of popular API. V4L2 handles moving image data on Linux OS, such as importing video from the camera. In this package, V4L2 provides only the required functions to import moving images from a USB camera or MIPI camera.

To capture video data from a camera, V4L2 controls the camera by specifying device files (e.g. /dev/video0) that corresponds to the camera device. The captured frame data can be used by application programs after mapping the data into user land.

### 2.5.1 API

The following table shows V4L2 APIs that are checked the operation in the RZ/V VLP (RZ/V2M).

**Table 2-54. System calls using in V4L2**

System call	Summary
open	Open a V4L2 device file.
close	Close a V4L2 device file.
ioctl	Issue V4L2 API for the opened device file.
select	Check if DQBUF can be executed.
mmap	Map a buffer in user space.
munmap	Unmap a buffer in user space.
ioctl(VIDIOC_S_FMT)	Specify the data format.
ioctl(VIDIOC_REQBUFS)	Request a buffer area.
ioctl(VIDIOC_QUERYBUF)	Query the status of a buffer.
ioctl(VIDIOC_QBUF)	Register a buffer.
ioctl(VIDIOC_DQBUF)	Release a buffer.
ioctl(VIDIOC_STREAMON)	Start streaming.
ioctl(VIDIOC_STREAMOFF)	Stop streaming.
ioctl(VIDIOC_QUERYCAP)	Inquire about the device capabilities.

Note: The string in parentheses after system call "ioctl" is the constant name to specify as the second argument when "ioctl" is issued.

### 2.5.2 Software Interface

Specify the following interface when using V4L2.

**Table 2-55. Device file (V4L2)**

Device file	Summary
/dev/videoN	This device file corresponds to the camera module. * N depends on the number of connecting devices. * For a MIPI camera, the device file is fixed as /dev/video0 and not changed dynamically.

## 2.6 Yocto environment, root File System and SDK

The RZ/V VLP (RZ/V2M) is developed based on Yocto Project. By building the Yocto environment (hereafter called “bitbake”) in this Linux package on Linux PC, it can be generated the Linux Kernel binaries to boot Linux for RZ/V2MA board and the file system image that mounts at Linux booting (hereafter called “root file system”). Using the prescribed method of bitbake also enables to generate the files to install the cross-compiler environment (hereafter called “SDK”) that can be installed on the Linux PC. Refer to the release note about the Yocto Project version that this Linux package is based on.

Notes: \*1. Refer to the release note for the version of the Yocto Project this package is based on.

\*2. Refer to the start-up guide on how to bitbake and make SDK.

The above root file system and SDK are composed by AArch64 binaries and libraries. Furthermore, by enabling the AArch32 libraries at bitbake, in addition to AArch64, it can support not only AArch32 binaries and libraries but also multilib. (Any AArch32 binaries and libraries would not be included in the root file system and SDK without additional user’s setting.)

The root file system in this package does not include GPLv3+ software components, it may be provided as an option.

This Linux package supports the configurations listed in the table below.

**Table 2-56. RZ/V VLP (RZ/V2M) configuration**

Configuration name	Summary
core-image-bsp	An image supporting for the RZ/V2MA device drivers.
core-image-minimal	Based on core-image-minimal provided by Yocto Project.

## 3. U-Boot

This chapter describes about the U-Boot included in the RZ/V VLP (RZ/V2M).

### 3.1 Supported Functions

U-boot for RZ/V2MA provides the following functions.

- Chip and board-specific hardware configuration.
- Load and boot the Linux kernel from a boot device (SD).
- Store the U-Boot environmental variables into eMMC.
- Executing boot commands using the serial console.
- Write into eMMC.

Note: Refer to the start-up guide for the method of how to write into eMMC and supported U-Boot environmental variables.

Revision History	RZ/V Verified Linux Package Software Manual for RZ/V2MA
------------------	---

Rev.	Date	Description	
		Page	Summary
1.00	Sep.16.2022	—	First edition issued.
1.10	Jan.13.2023	—	Updated the revision number.
1.20	Jul.31.2023	—	Changed the document title.
		1, 17	Removed notes indicating that the PCIe Express endpoint is not supported.

---

RZ/V Verified Linux Package Software Manual for RZ/V2MA

Publication Date: Rev.1.20 Jul.31.2023

Published by: Renesas Electronics Corporation

---



# RZ/V Verified Linux Package Software Manual for RZ/V2MA

