User's Manual

RZ/V Verified Linux Package

Software Manual for RZ/V2M

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).

Renesas Electronics
www.renesas.com

Rev.1.40 Jul 2023

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

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/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2023 Renesas Electronics Corporation. All rights reserved.

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 provides users with an understanding of the information to develop software for the RZ/VM2 microprocessor. It is intended for users to develop software incorporating the RZ/V2M microprocessor. To use this document, you need basic knowledge of software development and Linux systems.

This document comprises the following:

- An overview of RZ/V Verified Linux Package for RZ/V2M
- Software in this package
- Boot loader

About the restrictions of this package, refer to the release note.

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.

All trademarks and registered trademarks are the property of their respective owners.

・Linux is a registered trademark or trademark of Linus Torvalds.

・Yocto Project is a registered trademark or trademark of Linux Foundation.

・U-boot is a registered trademark or trademark of DENX Software Engineering.

Trademark or registered trademark marks (®, ™) may be omitted in this document.

The following documents apply to RZ/V Verified Linux Package.

Document Type	Description	Document Title
Release note	Description of release information of RZ/V Verified Linux Package. The restriction may be described in this document.	RZ/V Verified Linux Package Release Note
User's manual for Software	Description of RZ/V Verified Linux Package instruction.	This document.
Usage guide	Guide how to use the RZ/V2M Linux package.	RZ/V Verified Linux Package Start-Up Guide for RZ/V2M.

2. List of Abbreviations and Acronyms

Abbreviation	Full form
BSP	Board support package
CUI	Character user interface
CSI	Clock serial interface
DMA	Direct memory access
EHCI	Enhanced Host Controller Interface
eMMC	Embedded multimedia card
GUI	Graphical user interface
GPIO	General peripheral input output
I2C	Inter-integrated circuit
OHCI	Open Host Controller Interface
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
SDK	Software development kit
SDXC	SDXC SD extended capacity
TCP	Transmission Control Protocol
UART	Universal asynchronous receiver/transmitter
UDP	User Datagram Protocol
USB	Universal serial bus
V4L2	Video for Linux 2
xHCI	Extensible Host Controller Interface

Table of Contents

1. Overview 1
1.1 Overview of RZ/V2M Linux Package 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 22
2.4.12 Timer 23
2.4.13 Watchdog Timer 24
2.5 V4L2 25
2.5.1 API 25
2.5.2 Software Interface 25
2.6 Yocto environment, root File System and SDK 26

3. U-Boot 27
3.1 Supported Functions 27

1. Overview

Renesas provides the RZ/V Verified Linux Package (Here after RZ/V VLP (RZ/V2M)) 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.

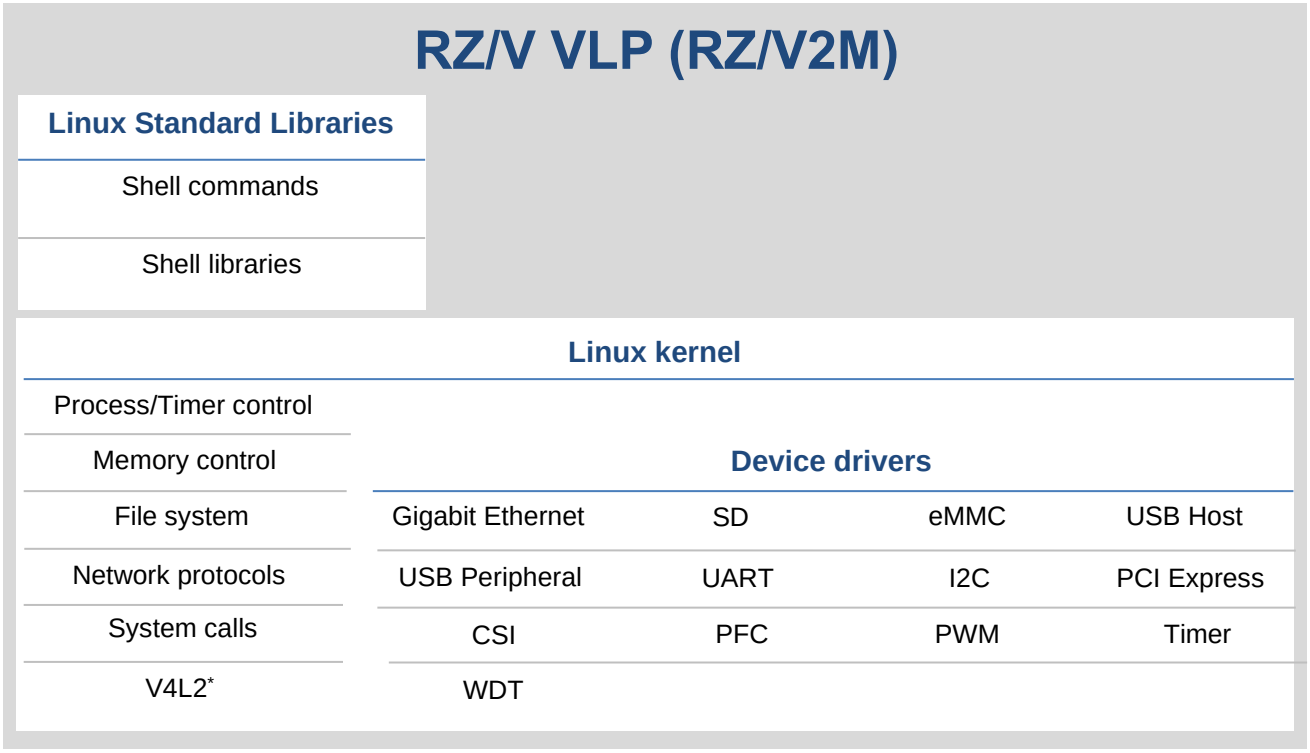

Figure 1-1. RZ/V VLP (RZ/V2M)

Note: In this package, V4L2 provides only functions required for using a USB camera or MIPI camera.

1.1 Overview of RZ/V2M Linux Package

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/V2M Linux Package 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

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/V2M 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/V2M 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/V2M
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/V2M to perform reading/writing and DMA transfer to an SD card.

2.4.2.1 Hardware Configuration

Table 2-3. Hardware configuration (SD)

RZ/V2M
2ch* (SDI0, SDI1)

Note: RZ/V2M Evaluation Board Kit has only 1ch (SDI0) for SD.

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*1
SDIO Card	Supported*3
Transfer Modes	1bit, 4bit
Bus Speed Mode*2	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*3
SPI Mode	Unsupported
SD Mechanical Write Protect Switch	Supported*3
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/V2M 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*

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/V2M 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)

RZ/V2M
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/V2M 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/V2M
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/V2M 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/V2M
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/V2M 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/V2M
1ch (UART0)

2.4.6.2 Description of Functionality

This module provides the following functions. *1

- 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*2
  - Bit width for transfer: 7 or 8 bit*2
- Hardware Flow control

Notes: *1. 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. RZ/V2M Evaluation Board Kit does not support the 2bit of stop bits and the 7bit of bit width for transfer.

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

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/V2M to transmit data between the slave devices on the I2C bus.

2.4.7.1 Hardware Configuration

Table 2-17. Hardware configuration (I2C)

RZ/V2M
2ch (I2C0, I2C2)

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 I2C2: /dev/i2c-2

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/V2M and performs data communication. This module complies with PCIe Gen2.

2.4.8.1 Hardware Configuration

Table 2-22. Hardware configuration (PCIe)

RZ/V2M
2 lane (PCI)

Note: PIC Express cannot be used on the RZ/V2M Evaluation Board Kit.

2.4.8.2 Description of Functionality

This module provides the following function.

- Supported configuration: Root complex
- 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: PCI0: /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/V2M and performs serial data transmission.

2.4.9.1 Hardware Configuration

Table 2-24. Hardware configuration (CSI)

RZ/V2M
1ch (CSI4)

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: CSI4: /dev/spidev0.0

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/V2M 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/V2M are shown in the table below. The shared pins setting initially in this BSP are described in bold. Refer to the “RZ/V2M 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	NAND	eMMC
NACEN	**P00_08**	NACEN	-
NAREN	**P00_09**	NAREN	-
NAALE	**P00_12**	NAALE	-
NARBN	**P00_13**	NARBN	-

Table 2-28. IO Group PORT01

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	PWM	External interruption
PM8	P01_08	**PM8**	INEXINT16
PM9	P01_09	**PM9**	INEXINT17
PM10	P01_10	**PM10**	INEXINT18
PM11	P01_11	**PM11**	INEXINT19
PM12	P01_12	**PM12**	INEXINT20
PM13	P01_13	**PM13**	INEXINT21
PM14	P01_14	**PM14**	INEXINT22

Table 2-29. IO Group PORT02

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	—	External interruption
INEXINT4	**P02_04**	—	INEXINT4
INEXINT5	**P02_05**	—	INEXINT5
INEXINT6	**P02_06**	—	INEXINT6
INEXINT7	**P02_07**	—	INEXINT7

Table 2-30. IO Group PORT03

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	CSI0, CSI1, CSI2, CSI3	UART0, UART1, IIC2, IIC3, CSI3
CSTXD0	P03_00	CSTXD0	**UATX0**
CSRXD0	P03_01	CSRXD0	**UARX0**
CSSCLK0	P03_02	CSSCLK0	**UACTS0N**
CSCS0	P03_03	CSCS0	**UARTS0N**
CSTXD2	P03_08	CSTXD2	**I2SDA2**
CSRXD2	P03_09	CSRXD2	**I2SCL2**

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	**CSTXD4**	CSRXD4
CSRXD4	P04_01	**CSRXD4**	-
CSSCLK4	P04_02	**CSSCLK4**	-
CSCS4	P04_03	**CSCS4**	-

Table 2-32. IO Group PORT06

External pin	Dual-purpose pin0	Dual-purpose pin1	Dual-purpose pin2
	GPIO	-	-
P0608	**P06_08**	-	-
P0609	**P06_09**	-	-
P0610	**P06_10**	-	-
P0611	**P06_11**	-	-

Table 2-33. IO Group PORT09

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

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 2-34. 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/V2M and outputs a PWM signal.

2.4.11.1 Hardware Configuration

Table 2-35. Hardware configuration (PWM)

RZ/V2M
7ch (PWM8-PWM14)

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 2-36. Device file (PWM)

Device file	Summary
/sys/class/pwm/pwmchip*/pwm0	The default configuration sets as bellow: PWM8: /sys/class/pwm/pwmchip0/pwm0 PWM9: /sys/class/pwm/pwmchip1/pwm0 PWM10: /sys/class/pwm/pwmchip2/pwm0 PWM11: /sys/class/pwm/pwmchip3/pwm0 PWM12: /sys/class/pwm/pwmchip4/pwm0 PWM13: /sys/class/pwm/pwmchip5/pwm0 PWM14: /sys/class/pwm/pwmchip6/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/V2M.

2.4.12.1 Hardware Configuration

Table 2-37. Hardware configuration (TIM)

RZ/V2M
16ch (TIM8-TIM23)

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 2-38. Device file (TIM)

Device file	Summary
/dev/hw_tim*	The default configuration sets as bellow: TIM8: /dev/hw_tim0 TIM9: /dev/hw_tim1 TIM10: /dev/hw_tim2 TIM11: /dev/hw_tim3 TIM12: /dev/hw_tim4 TIM13: /dev/hw_tim5 TIM14: /dev/hw_tim6 TIM15: /dev/hw_tim7 TIM16: /dev/hw_tim8 TIM17: /dev/hw_tim9 TIM18: /dev/hw_tim10 TIM19: /dev/hw_tim11 TIM20: /dev/hw_tim12 TIM21: /dev/hw_tim13 TIM22: /dev/hw_tim14 TIM23: /dev/hw_tim15

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/V2M.

2.4.13.1 Hardware Configuration

Table 2-39. Hardware configuration (WDT)

RZ/V2M
1ch (WDT)

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 2-40. Device file (WDT)

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

2.4.13.4 Device Driver Specification

Refer to the following documentation for details.

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

2.5 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.

Note: If you use a MIPI camera, please use the RZ/V2M ISP Support Package.

2.5.1 API

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

Table 2-41. 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 2-42. 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 “bitbike”) in this Linux package on Linux PC, it can be generated the Linux Kernel binaries to boot Linux for RZ/V2M 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 bitbike, 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 2-43. RZ/V2M Linux Package configuration

Configuration name	Summary
core-image-bsp	An image supporting for the RZ/V2M 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 in RZ/V2M 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/V2M

Rev.	Date	Description	
		Page	Summary
1.00	Jun.30.2021	—	First edition issued.
1.10	Oct.8.2021	17	2.4.8 PCI Express Add the notice V1.1.0 does not support PCI Express.
		25	2.5 V4L2 Add supported status when using MIPI cameras.
1.20	Jan 28. 2022	7, 14	Changed the name of the target board to “RZ/V2M Evaluation Board Kit”.
		12	2.4.5.2 Description of Functionality (USB Peripheral) Changed the number of supported endpoints.
		16	Table 2-21. Device file (I2C) Changed the device node for I2C2.
		17	2.4.8 PCI Express Deleted the description for the limitation in the previous version.
		17	2.4.8.2 Description of Functionality Corrected the supported type of interrupts.
		18	Table 2-25. Supported functions (CSI) Add the supported maximum frequency in the master mode.
1.30	Oct 14, 2022	-	Update the document number and version for V1.3.0.
1.40	Jul 31, 2023	-	Changed the document title.

RZ/V Verified Linux Package Software Manual for RZ/V2M

Publication Date: Rev.1.40 Jul.31.2023

Published by: Renesas Electronics Corporation

RZ/V Verified Linux Package Software Manual for RZ/V2M

Renesas Electronics Corporation

R01US0512EJ0140