

# RZ/V Verified Linux Package

ソフトウェアマニュアル for RZ/V2MA

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。  
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
  5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
  7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
  8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
  13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違っていると、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

# このマニュアルの使い方

## 1. 目的と対象者

このマニュアルは、RZ/V2MA 上で動作するソフトウェアの開発に必要な情報をユーザに理解していただくためのマニュアルです。RZ/V2MA を用いたアプリケーションおよびシステムを設計・開発するユーザを対象にしています。このマニュアルを使用するには、ソフトウェア開発、および Linux に関する基本的な知識が必要です。

このマニュアルは、下記の項目で構成されています。

- RZ/V VLP (RZ/V2M)パッケージの概要
- パッケージ内のソフトウェア
- ブートローダ

制限事項については、本パッケージのリリースノートをご参照ください。

本プロセッサは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

すべての商標および登録商標は、それぞれの所有者に帰属します。

- Linux は、Linus Torvalds の米国またはその他の国における登録商標あるいは商標です。
- Arm と Arm Cortex は、Arm Limited (またはその子会社) の EU またはその他の国における登録商標です。
- Yocto Project は、Linux Foundation の米国またはその他の国における登録商標です。
- U-boot は、DENX Software Engineering の登録商標あるいは商標です。

本書では、商標または登録商標のマーク(®、™)を省略して記載する場合がございます。

## 2. 略語および略称の説明

略語／略称	正式名称
BSP	Board support package
CUI	Character user interface
CSI	Clock serial interface
DMA	Direct memory access
eMMC	Embedded multimedia card
GUI	Graphical user interface
GPIO	General peripheral input output
I2C	Inter-integrated circuit
IP	Internet Protocol
MMC	Multimedia card
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
TSU	Temperature Sensor Unit
UART	Universal asynchronous receiver/transmitter
UDP	User Datagram Protocol
USB	Universal serial bus
V4L2	Video for Linux 2
xHCI	Extensible Host Controller Interface

# 目次

1. 概要.....	1
1.1 RZ/V VLP (RZ/V2M)概要.....	2
2. RZ/V VLP (RZ/V2M)主要機能 .....	3
2.1 Linuxカーネル.....	3
2.1.1 Linuxシステムコール.....	3
2.1.2 プロセス/タイマ管理 .....	3
2.1.3 メモリ管理.....	3
2.1.4 ファイルシステム.....	3
2.1.5 デバイスドライバ.....	3
2.1.6 ネットワークプロトコル.....	3
2.2 Linux Standard Libraries .....	4
2.3 Linux Shell .....	5
2.4 デバイスドライバ.....	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 ソフトウェアインタフェース.....	30
2.6 Yocto環境、rootファイルシステム、SDKIについて .....	31

3. U-Boot.....	32
3.1 機能 .....	32

# RZ/V Verified Linux Package ソフトウェアマニュアル for RZ/V2MA

## ルネサスマイクロコンピュータ

### 1. 概要

RZ/V2MA のソフトウェアは、統合型 Linux パッケージとして提供されます。

Linux システムは、Linux kernel、Linux Standard Libraries、Linux shell により構成されています。

本 Linux パッケージ（以下、RZ/V VLP (RZ/V2M)）は、ルネサスが動作確認とバージョン管理を行っています。

RZ/V VLP (RZ/V2M)				
<b>Linux Standard Libraries</b>				
Shell commands				
Shell libraries				
<b>Linux kernel</b>				
Process/Timer control				
Memory control		<b>Device drivers</b>		
File system	Gigabit Ethernet	SD	eMMC	USB Host
Network protocols	USB Peripheral	UART	I2C	PCI Express
System calls	CSI	PFC	PWM	Timer
V4L2 <sup>*1</sup>	WDT	TSU	DMA	VCD <sup>*2</sup>

図 1.1 RZ/V VLP (RZ/V2M)

【注 1】 V4L2 は、USB カメラおよび MIPI カメラ接続時に必要な機能のみを提供しております。

【注 2】 Video Codec Package にて別提供となっています。



## 1.1 RZ/V VLP (RZ/V2M)概要

本パッケージに含まれる主なソフトウェアを以下の表に示します。

各ソフトウェアのバージョンについては、「RZ/V Verified Linux Package Release Note」（以下、リリースノート）をご参照ください。

表 1.1 RZ/V VLP (RZ/V2M)ソフトウェアコンポーネント

コンポーネント	ソフトウェア名	概要	章番号
Linux kernel	CIP SLTS Linux kernel	CIP プロジェクトが提唱する Linux カーネル	2.1
Linux standard libraries	busybox	UNIX ユーティリティを小さな実行ファイルにまとめたプログラム	2.2
	binutils	バイナリツールセット	2.2
	glibc	GNU 標準 C ライブラリ	2.2
	openssl	TLS/SSL プロトコルを実装した暗号ライブラリ	2.2
	上記以外の 各種ソフトウェア	Yocto Project が提供するビルドシステム	2.6
Video	V4L2	Linux OS 上で動画を扱う API の総称	2.5

本パッケージに含まれるデバイスドライバを以下に示します。

表 1.2 RZ/V VLP (RZ/V2M) デバイスドライバー一覧

機能名称	概要	章番号
Gigabit Ethernet	Ethernet フレームを送受信します。	2.4.1
SD	SD カードのリード/ライトを行います。	2.4.2
eMMC	eMMC のリード/ライトを行います。	2.4.3
USB Host	USB Host 機能として、USB デバイス間でデータを送受信します。	2.4.4
USB Peripheral	USB Peripheral 機能として、USB デバイス間でデータを送受信します。	2.4.5
UART	調歩同期方式によるシリアル通信を行います。	2.4.6
I2C	I2C バス上に接続されているデバイスとのデータを送受信します。	2.4.7
PCI Express	PCIe 通信を行います。	2.4.8
CSI	クロック同期式シリアル通信を行います。	2.4.9
PFC	LSI 外部端子の兼用機能の切り替え、GPIO ピンの設定を行います。	2.4.10
PWM	PWM 信号を出力します。	2.4.11
TIMER	インターバルタイマの設定を行います。	2.4.12
Watchdog Timer	ウォッチドッグタイマの設定を行います。	2.4.13
TSU	Temperature sensor unit が出力した LSI 温度を取得します。	2.4.14
DMA	RZ/V2MA に搭載している DMA Controller を制御します。	2.4.15

## 2. RZ/V VLP (RZ/V2M)主要機能

本章では、本パッケージに含まれるソフトウェアの主な機能について説明します。

### 2.1 Linux カーネル

Linux カーネルとは、UNIX ライクの OS です。Linux カーネルは、アプリケーションが存在するユーザー空間とは異なる、カーネル空間に存在します。

RZ/V VLP (RZ/V2M)では、Civil Infrastructure Platform(CIP) Project が管理している CIP SLTS Linux Kernel を提供します。

Linux カーネルは、以下に示す複数のコンポーネントで構成されています。

#### 2.1.1 Linux システムコール

Linuxシステムコールとは、ユーザー空間とLinuxカーネルを繋ぐインタフェースを指します。アプリケーションから下記の各コンポーネントを操作する時は、このLinuxシステムコールを介して行います。

#### 2.1.2 プロセス/タイマ管理

プロセスとは、動作中のプログラムのことを指します。Linuxではプロセスのことをスレッドと呼びます。

Linuxカーネルでは複数のスレッドに対して、1つのスレッドを一定時間動かしたら次のスレッドに移る、という管理を行っています。スレッドの動作時間は、ハードウェアタイマによる割り込みで制御されます。

#### 2.1.3 メモリ管理

Linuxカーネルでは、プロセスごとにメモリを割り当てています。その際、メモリは「仮想メモリ空間」と呼ばれる、仮想的な領域から割り当てられます。

#### 2.1.4 ファイルシステム

通常、OSはデータを操作する際、そのデータをファイルとして扱います。ファイルシステムとは、ファイルを抽象化し、管理するためのシステムを指します。

Linuxでは、データの他に、デバイス（マウス、キーボードなど）もファイルとして扱います。この仮想的なファイルシステムをVFS(Virtual File System)と呼びます。VFSにより、Linuxはデータやデバイスに対して、ファイルへのアクセスという統一されたインタフェースで操作することが可能となっています。

#### 2.1.5 デバイスドライバ

デバイスドライバとは、特定のハードウェアデバイスを使用するためのソフトウェアモジュールを指します。デバイスドライバはアーキテクチャーに依存しない部分と依存する部分に分かれており、前者はBSP (Board Support Package)と呼ばれます。それぞれのソースコードは、前者は”./linux/drivers”以下に、後者は”./linux/arch”以下に格納されています。

本Linuxパッケージで提供しているデバイスドライバの詳細については、2.4章を参照してください。

#### 2.1.6 ネットワークプロトコル

Linux カーネルはネットワーク機能として、IP(Internet Protocol)、TCP(Transmission Control Protocol)、UDP(User Datagram Protocol)といった複数のプロトコルに対応しています。

## 2.2 Linux Standard Libraries

ライブラリとは、使用頻度の高い機能をまとめ、他のプログラムからも利用できるようにしたものです。Linux ライブラリは、静的ライブラリ (Static Library) と共有ライブラリ (Shared Library) の2つをサポートしています。

静的ライブラリは、プログラム作成時に実行ファイル内に組み込まれるライブラリを指します。拡張子は「\*.a」です。静的ライブラリには実行速度が早い、一度コンパイルすれば動作が変わらないというメリットがあります。しかしその反面、使用するプログラム全てにライブラリが組み込まれるため、プログラムサイズが大きくなるというデメリットがあります。

共有ライブラリは、プログラム実行時にロードされるライブラリを指します。拡張子は「\*.so」ですが、その後ピリオドとバージョン番号が付加されます。共有ライブラリは1つのライブラリに対して複数のプログラムがアクセスするため、プログラムサイズが小さくなるというメリットがあります。しかしその反面、ライブラリが更新されると、そのライブラリを使用している全ての機能に対して、ライブラリ使用時に影響が出るというデメリットがあります。

- リンクとライブラリの関係性について

プログラムからライブラリを呼び出すことをリンクと呼びます。リンクには、スタティックリンク (Static Link) とダイナミックリンク (Dynamic Link) という2つの種類が存在します。

スタティックリンクは、プログラム作成する際、予めライブラリをプログラム本体に組み込んでおく方法です。このリンク方法で組み込まれたライブラリが静的ライブラリとなります。

ダイナミックリンクは、プログラム実行時にライブラリを呼び出す方法です。このリンク方法で呼び出されるライブラリが共有ライブラリとなります。

RZ/V VLP (RZ/V2M)内のLinux Standard Librariesは、CIP Coreをベースにしています。CIP Coreに含まれていないソフトウェアコンポーネントについては、Yocto Projectから提供されているソフトウェアコンポーネントを使用しています。

CIP Coreの提供するソフトウェアは以下のとおりです。各ソフトウェアのバージョンはリリースノートまたはコンポーネントリストをご参照ください。

- glibc
- binutils \*ルートファイルシステムには含まれていません
- busybox
- openssl

## 2.3 Linux Shell

Linux Shell とは、Linux を実際に操作するためのインタフェースと、その橋渡しを行うプログラムを指します。

ユーザーがLinuxを操作する際には、CUI（Character User Interface：キャラクタユーザーインタフェース）による文字列入力や、GUI（Graphical User Interface：グラフィカルユーザーインタフェース）による操作によって、Linux OSのコアであるLinuxカーネルを操作します。

しかし、Linuxカーネル自身はCUIやGUIによる操作を直接理解することが出来ません。また、Linuxカーネルの処理結果も、そのままではユーザーが解釈可能な形で出力されません。

そのため、ユーザーとLinuxカーネルの間に立ち、それぞれに解釈可能な形に変換するための橋渡し役とも言えるプログラムが必要になります。そのプログラムがLinux Shellです。

Linux Shellにはいくつか種類がありますが、RZ/V VLP (RZ/V2M)では” bash” を使用しています。

## 2.4 デバイスドライバ

本パッケージでは、RZ/V2MA に搭載されたハードウェア機能や、ボード上の各種ハードウェアデバイスを制御するデバイスドライバを提供しています。本章では、本パッケージに含まれる主要なデバイスドライバを説明いたします。

### 2.4.1 Gigabit Ethernet

本モジュールは、RZ/V2MA に搭載されている Ether MAC インタフェースを制御して、Linux 上にて Ethernet フレームデータの送受信を行います。

#### 2.4.1.1 ハードウェア構成概要

表 2.1 ハードウェア構成 (Gigabit Ethernet)

RZ/V2MA
1ch (ETH0)

#### 2.4.1.2 提供機能

本モジュールは、以下の機能を提供します。

- IEEE802.3 に準拠
- Ethernet フレームの送受信に対応
- 1000Mbps, 100Mbps の全二重通信をサポート\*

【注】 半二重転送は非対応です。

#### 2.4.1.3 ソフトウェアインタフェース

本モジュールの機能は、Linux の標準的なネットワークインタフェースとして提供されます。以下に示すネットワークインタフェースを指定して、Linux の各種ネットワーク機能を使用することができます。

表 2.2 ネットワークインタフェース (Gigabit Ethernet)

ネットワークインタフェース	概要
eth0	デフォルトのコンフィギュレーションでは eth0 に設定されます。

#### 2.4.1.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.2 SD

本モジュールは、RZ/V2MAに搭載されているSD/MMC Host Controllerを制御し、SDカードへのリード/ライトやDMA転送などを行います。

### 2.4.2.1 ハードウェア機能概要

表 2.3 ハードウェア構成 (SD)

RZ/V2MA
2ch* (SDIO, SDI1)

【注】 RZ/V2MA Evaluation Board Kit に搭載されている SD スロットは SDIO のみになります。

### 2.4.2.2 提供機能

本モジュールは、下記の機能に対応します。

表 2.4 機能一覧表 (SD)

機能	対応状況
SD メモリカード	SD, SDHC, SDXC <sup>*1</sup>
SDIO カード	対応 <sup>*3</sup>
転送モード	1bit, 4bit
バススピードモード <sup>*2</sup>	Default Speed (DS), High Speed (HS), UHS-I
DMA 機能	内部 DMAC 対応
カードパワー制御	対応
カード検出(CD 検出)	対応
カード検出(DAT3 検出)	非対応
ライトプロテクト	対応 <sup>*3</sup>
SPI モード	非対応
SD Mechanical Write Protect Switch	対応 <sup>*3</sup>
SD クロック制御	対応
CPRM Security	非対応
CMD52 During Data Transfer(C52PUB)	非対応
Data Transfer Abort (IOABT)	非対応
Read Wait (RWREQ)	非対応
SDIO Wake Up	非対応
SDIO Suspend/Resume	非対応

【注】 \*1 標準 BSP のファイルシステムは exFAT 非対応であるため、exFAT でフォーマットされた SDXC メモリカードはマウントできません。

\*2 DDR50 は非対応です。

\*3 RZ/V2MA Evaluation Board Kit では非対応の機能です。

また、本機能に関連する規格を以下の表に示します。

表 2.5 SD 規格関連資料

資料番号	発行	タイトル	版数	日付
—	SD Card Association	SD Specifications Part 1 Physical Layer Specification Simplified Specification	4.10	Jan. 22, 2013
—	SD Card Association	SD Specifications Part E1 SDIO Simplified Specification	3.00	Feb. 25, 2011

### 2.4.2.3 ソフトウェアインタフェース

本モジュールの機能は、Linux の標準的なデバイスインタフェースとして提供されます。以下に示すデバイスファイルを指定して、Linux 上で SD デバイスを使用することができます。

表 2.6 デバイスファイル (SD)

デバイスファイル	概要
/dev/mmcblkN*	デフォルトのコンフィギュレーションでは次のように設定されます。 SDI0: /dev/mmcblkN* SDI1: /dev/mmcblkN*

【注】 システムによっては N の値は異なります。(例: /dev/mmcblk0)

### 2.4.2.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

### 2.4.3 eMMC

本モジュールは、RZ/V2MAに搭載されているSD/MMC Host Controllerを制御し、eMMCメモリへのリード/ライトやDMA転送などを行います。

#### 2.4.3.1 ハードウェア機能概要

表 2.7 ハードウェア構成 (eMMC)

RZ/V2MA
1ch (eMMC)

#### 2.4.3.2 提供機能

本モジュールは、下記の機能に対応します。

表 2.8 機能一覧表 (eMMC)

機能	対応状況
eMMC デバイスアクセス	対応
転送モード	1, 4, 8bit
バススピードモード*	High-Speed, HS200, Backward compatible
DMA 転送	内部 DMAC 対応
ストリーム転送	非対応
オープンエンデッド・マルチブロック転送	非対応

【注】 High Speed DDR, HS400 は非対応です。

また、本機能に関連する規格を以下の表に示します。

表 2.9 eMMC 規格関連資料

資料番号	発行	タイトル	版数	日付
JESD 84-B51	JEDEC STANDARD	EMBEDDED MULTI-MEDIA CARD (eMMC), ELECTRICAL STANDARD (5.1)	5.1	2013.09



### 2.4.3.3 ソフトウェアインタフェース

本モジュールの機能は、Linux の標準的なデバイスインタフェースとして提供されます。以下に示すデバイスファイルを指定して、Linux 上で eMMC メモリを使用することができます。

表 2.10 デバイスファイル (eMMC)

デバイスファイル	概要
/dev/mmcblkN*	デフォルトのコンフィギュレーションでは次のように設定されます。 eMMC: /dev/mmcblkN*

【注】 N の値はシステムによって異なります。(例: /dev/mmcblk0)

### 2.4.3.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.4 USB Host

本モジュールは、RZ/V2MA に搭載されている USB ホストコントローラを制御し、USB デバイス間とのデータ送信、受信を行います。本モジュールは、USB3.1 Gen1 規格に準拠しています。

### 2.4.4.1 ハードウェア機能概要

表 2.11 ハードウェア構成 (USB Host)

RZ/V2MA
1ch <sup>*</sup> (USB)

【注】 USB Host と USB Peripheral は排他での使用となります。

### 2.4.4.2 提供機能

本モジュールは、以下の機能を提供します。

- Super-Speed (5Gbps) /High-Speed (480Mbps) /Full-Speed (12Mbps) /Low-Speed (1.5Mbps) の転送をサポート
- Isochronous/Interrupt/Control/Bulk の全転送タイプに対応  
Isochronous/Interrupt High-band 転送に対応
- eXtensible Host Controller Interface (xHCI) Specification for USB 準拠

本モジュールは、USB On-The-Go に対応しています。

本機能に関連する規格を以下に示します。

- 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 ソフトウェアインタフェース

本モジュールは USB デバイスを接続することにより、以下の表のデバイスファイルを介して、Linux の標準的なインタフェースによって USB デバイスを制御することができます。

これらの対応する USB デバイスは、カーネルコンフィグレーションにより有効化することが可能です。

表 2.12 デバイスファイル (USB Host)

デバイスファイル	概要
/dev/input/eventX <sup>*</sup>	キーボードに対応するデバイスファイルです。
/dev/input/mouseY <sup>*</sup>	マウスに対応するデバイスファイルです。
/dev/sdZ <sup>*</sup>	USB ストレージに対応するデバイスファイルです。

【注】 システムによっては X, Y, Z の値は異なります。(例: /dev/input/event0)

### 2.4.4.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.5 USB Peripheral

本モジュールは、RZ/V2MA に搭載されている USB ペリフェラルコントローラを制御し、USB デバイス間とのデータ送信、受信を行います。本モジュールは、USB3.1 Gen1 規格に準拠しています。

### 2.4.5.1 ハードウェア機能概要

表 2.13 ハードウェア構成 (USB Peripheral)

RZ/V2MA
1ch <sup>*</sup> (USB)

【注】 USB Host と USB Peripheral は排他での使用となります。

### 2.4.5.2 提供機能

本モジュールは、以下の機能を提供します。

- Super-Speed (5Gbps) /High-Speed (480Mbps) /Full-Speed (12Mbps) の転送をサポート
- Interrupt/Control/Bulk の転送タイプに対応
- 各 PIPE は以下の設定が可能

#### 【USB3.1】

- エンドポイント番号 : 0~8
- 転送タイプ : Control (PIPE0 のみ) 、Bulk、Interrupt
- 転送方向 : IN/OUT
  - IN : ep1(PIPE1)、ep3(PIPE3)、ep5(PIPE5)、ep7(PIPE7)
  - OUT : ep2(PIPE2)、ep4(PIPE4)、ep6(PIPE6)、ep8(PIPE8)
- 最大パケットサイズ : PIPE0 は 512byte、その他は 1024byte (\*高帯域は非対応)

#### 【USB2.0】

- エンドポイント番号 : 0~8
- 転送タイプ : Control (PIPE0 のみ) 、Bulk、Interrupt
- 転送方向 : IN/OUT
  - IN : ep1(PIPE1)、ep3(PIPE3)、ep5(PIPE5)、ep7(PIPE7)
  - OUT : ep2(PIPE2)、ep4(PIPE4)、ep6(PIPE6)、ep8(PIPE8)
- 最大パケットサイズ : PIPE0 は 64byte、その他は 1024byte (\*高帯域は非対応)

本モジュールは、USB On-The-Go に対応しています。

本機能に関連する規格を以下に示します。

- 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 ソフトウェアインタフェース

本モジュールは USB ペリフェラルとしての機能を提供するソフトウェアを実現する Linux の標準的なインタフェースを提供します。

以下の表ではデバイスファイルの例を示します。対応する USB デバイスは、カーネルコンフィグレーションにより有効化することが可能です。

表 2.14 デバイスファイル (USB Peripheral)

デバイスファイル	概要
/dev/ttyGS0	USB ポート上でのシリアル通信に使用するデバイスファイルです。

### 2.4.5.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.6 UART

本モジュールは、RZ/V2MA に搭載されている UART を制御することで、RS232C によるデータ送受信、通信設定制御などの機能を提供します。

### 2.4.6.1 ハードウェア構成概要

表 2.15 ハードウェア構成 (UART)

RZ/V2MA
2ch (UART0, UART1)

### 2.4.6.2 提供機能

本モジュールは、以下の機能を提供します。<sup>\*1</sup>

- RS232C によるデータ送受信
- 通信設定の制御
  - ・ ボーレート (9600, 19200, 38400, 57600, 115200bps)
  - ・ パリティビット (なし/ODD/EVEN)
  - ・ ストップビット (1bit, 2bit)
  - ・ 転送ビット長 (7bit, 8bit)

【注】 <sup>\*1</sup> 本 Linux システムにおいて Ch0 は コンソール接続用の UART として標準的に使用します。

Linux 起動直後は、下記のとおりを設定されます。

ボーレート: 115200bps、パリティビット:なし、ストップビット:1bit、転送ビット長:8bit

### 2.4.6.3 ソフトウェアインタフェース

本モジュールの機能は、Linux の標準的なデバイスインタフェースとして提供されます。以下に示すデバイスファイルを指定して、Linux 上で UART による通信を行います。

表 2.16 デバイスファイル (UART)

デバイスファイル	概要
/dev/ttyS*	デフォルトのコンフィギュレーションでは次のように設定されます。 UART0: /dev/ttyS0 UART1: /dev/ttyS1

### 2.4.6.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.7 I2C

本モジュールは、RZ/V2MA に搭載されている I2C インタフェースを制御し、I2C バスに接続されているスレーブデバイスとの間でデータを送受信します。

### 2.4.7.1 ハードウェア構成概要

表 2.17 ハードウェア構成 (I2C)

RZ/V2MA
4ch (I2C0-I2C3)

### 2.4.7.2 提供機能

本モジュールは、下記の機能に対応します。

表 2.18 機能一覧表 (I2C)

機能	対応状況
マスタモード	対応
スレーブモード	非対応
シリアルクロック (SCL)	"High"レベル幅、"Low"レベル幅を設定可能

対応する I2C 転送速度を以下の表に示します。

表 2.19 転送速度 (I2C)

インタフェースモード	実転送速度	対応状況
Standard mode (100KHz)	最大 100Kbit/s	対応
Fast mode (400KHz)	最大 400Kbit/s	対応
High Speed mode (3.4 MHz)	最大 3400Kbit/s	非対応

本機能に関連する規格を以下の表に示します。

表 2.20 I2C 規格関連資料

資料番号	発行	タイトル	版数	日付
—	Philips Semiconductors	THE I2C-BUS SPECIFICATION	2.1	2000.01

### 2.4.7.3 ソフトウェアインタフェース

本モジュールの機能は、Linux の標準的なインタフェースとして提供されます。以下に示すデバイスファイルを指定して、Linux 上で I2C による通信を行います。

表 2.21 デバイスファイル (I2C)

デバイスファイル	概要
/dev/i2c-*	デフォルトのコンフィギュレーションでは次のように設定されます。 I2C0: /dev/i2c-0 I2C1: /dev/i2c-1 I2C2: /dev/i2c-2 I2C3: /dev/i2c-3

### 2.4.7.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.8 PCI Express

本モジュールは、RZ/V2MA に搭載されている PCI Express インタフェースを制御し、通信を行います。本モジュールは、PCI Express Gen2 に準拠しています。

### 2.4.8.1 ハードウェア構成概要

表 2.22 ハードウェア構成 (PCIe)

RZ/V2MA
2 lane (PCI)

### 2.4.8.2 提供機能

本モジュールは、以下の機能を提供します。

- 対応コンフィグレーション : root complex, endpoint
- 転送レート : 5GT/s
- コンフィグレーション空間のリード/ライト
- 起動時 I/O(ホスト)のリード/ライト
- 起動時メモリ(ホスト)のリード/ライト
- 起動時メモリ(スレーブ)のリード/ライト
- 割り込み機能 : INTx、MSI

本機能に関連する規格を以下に示します。

- PCI Express Base Specification 4.0

### 2.4.8.3 ソフトウェアインタフェース

本モジュールは、以下に示すデバイスファイルを指定して、Linux 上で PCIe 通信をします。

表 2.23 デバイスファイル (PCIe)

デバイスファイル	概要
/sys/class/pci_bus	デフォルトのコンフィギュレーションでは次のように設定されます。 PCI: /sys/class/pci_bus

### 2.4.8.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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



## 2.4.9 CSI

本モジュールは、RZ/V2MA に搭載されている 3 線式シリアルインターフェース（CSI）を制御し、通信を行います。

### 2.4.9.1 ハードウェア構成概要

表 2.24 ハードウェア構成（CSI）

RZ/V2MA
6ch (CSI0-CSI5)

### 2.4.9.2 提供機能

本モジュールは、以下の機能を提供します。

表 2.25 機能一覧表（CSI）

機能	対応状況
動作モード	マスタ受信専用モード、マスタ送受信モード、スレーブ受信専用モード、スレーブ送受信モード
シリアルデータ長	8, 16bit
先頭ビット	MSB/LSB *デフォルトの設定は MSB
データ転送方法	割込み転送方式
シリアルクロック	(マスタモード) 最大 48MHz、2-32766 分周 (スレーブモード) 最大 24MHz
送信/受信の FIFO	16bit (8bit + 8bit) × 16 段
シリアルデータ間のウェイト時間	マスタモードで設定可能

### 2.4.9.3 ソフトウェアインタフェース

本モジュールの機能は、以下に示すデバイスファイルを指定して、Linux 上で CSI による通信を行います。

表 2.26 デバイスファイル（CSI）

デバイスファイル	概要
/dev/spidevN.0*	デフォルトのコンフィギュレーションでは次のように設定されます。 CSIN: /dev/spidevN.0* ※ Nは使用するCSIのチャンネル数によって異なります。(N=0~5)

### 2.4.9.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.10 PFC

本モジュールは、RZ/V2MA に搭載されている Pin Function Controller を制御し、LSI 外部端子の兼用機能の切り替え、汎用入出力ポート(GPIO)機能の設定を行います。

### 2.4.10.1 ハードウェア構成概要

#### (1) 対応ピン

GPIO として使用できる端子の一覧を以下に示します。本 Linux BSP にて、初期設定している兼用端子については太字で記載しています。各兼用端子の詳細につきましては、RZ/V2MA ハードウェアマニュアルをご参照ください。

表 2.27 IO グループ PORT00

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	eMMC
P0008	<b>P00_08</b>	—	—
P0009	<b>P00_09</b>	—	—
P0012	<b>P00_12</b>	—	—
P0013	<b>P00_13</b>	—	—

表 2.28 IO グループ PORT01

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	PWM	外部割り込み
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

表 2.29 IO グループ PORT02

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
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>

表 2.30 IO グループ PORT03

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2	兼用端子 3
	GPIO	CSI0, CSI1, CSI2, CSI3	UART0, UART1, IIC2, IIC3, CSI3	CSI0, CSI1, CSI2
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>	—	—

表 2.31 IO グループ PORT04

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	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>	—

表 2.32 IO グループ PORT05

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	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>

表 2.33 IO グループ PORT06

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	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>	—	—

表 2.34 IO グループ PORT07

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	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>	—	—

表 2.35 IO グループ PORT08

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	SDIO	—
SD0CMD	P08_00	<b>SD0CMD</b>	—
SD0CLK	P08_01	<b>SD0CLK</b>	—
SD0DAT0	P08_02	<b>SD0DAT0</b>	—
SD0DAT1	P08_03	<b>SD0DAT1</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>	—

表 2.36 IO グループ PORT09

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	SDI1	外部割り込み
SD1CMD	P09_00	<b>SD1CMD</b>	—
SD1CLK	P09_01	<b>SD1CLK</b>	—
SD1DAT0	P09_02	<b>SD1DAT0</b>	—
SD1DAT1	P09_03	<b>SD1DAT1</b>	—
SD1DAT2	P09_04	<b>SD1DAT2</b>	—
SD1DAT3	P09_05	<b>SD1DAT3</b>	—
SD1WP	P09_06	<b>SD1WP</b>	INEXINT24
SD1CD	P09_07	<b>SD1CD</b>	INEXINT25

表 2.37 IO グループ PORT10

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
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>	—	—

表 2.38 IO グループ PORT11

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
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>	—	—

表 2.39 IO グループ PORT12

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
P1200	<b>P12_00</b>	—	INEXINT30
P1201	<b>P12_01</b>	—	INEXINT31
P1202	<b>P12_02</b>	—	INEXINT32
P1203	<b>P12_03</b>	—	INEXINT33

表 2.40 IO グループ PORT13

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
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

表 2.41 IO グループ PORT14

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	外部割り込み
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>	—	—

表 2.42 IO グループ PORT17

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	—
P1700	<b>P17_00</b>	—	—

表 2.43 IO グループ PORT20

外部端子名	兼用端子 0	兼用端子 1	兼用端子 2
	GPIO	—	—
P2000	<b>P20_00</b>	—	—
P2001	<b>P20_01</b>	—	—
P2002	<b>P20_02</b>	—	—

### 2.4.10.2 提供機能

本モジュールは、以下の機能を提供します。

- GPIO の制御
  - GPIO ピンの Input/Output 切り替え
  - 入力ピンの High/Low 状態を読み出し
  - 出力ピンの High/Low を設定
- LSI 兼用端子の切り替え
- 兼用端子の Pull-up/Down 制御
- LSI 端子のドライブ能力切り替え
- LSI 兼用端子のスルーレート切り替え
- 外部割り込み信号のマスク制御
- 外部割り込み信号の極性反転
  - 外部割り込み信号が Low アクティブ時、極性の反転を指定

### 2.4.10.3 ソフトウェアインタフェース

本モジュールでは、以下に示す GPIO sysfs インタフェースを指定して、Linux 上で GPIO を設定、制御します。

表 2.44 デバイスファイル (PFC)

ファイル	概要
/sys/class/gpio/gpioN*	デフォルトのコンフィギュレーションでは次のように設定されます。 gpioN: /sys/class/gpio/gpioN*

【注】 N は各 GPIO ピン番号を示します。

### 2.4.10.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.11 PWM

本モジュールは、RZ/V2MA に搭載されているパルス幅変調タイマ（PWM）を設定し、PWM 信号を出力します。

### 2.4.11.1 ハードウェア構成概要

表 2.45 ハードウェア構成（PWM）

RZ/V2MA
16ch (PWM0-PWM15)

### 2.4.11.2 提供機能

本モジュールは、以下の機能を提供します。

- ハイレベル幅の PWM 信号を連続出力
- PWM\_CLK=48MHz の場合、PWM 信号は下記の設定が可能
  - ・ 周期: 2.083 $\mu$ s~0.416s(2.4 Hz~480 kHz)
  - ・ 最小分解能: 21ns
  - ・ デューティ比: 0%~100%
- PWM 信号のレベル（Low/High）を反転

### 2.4.11.3 ソフトウェアインタフェース

本モジュールは、以下に示す PWM sysfs インタフェースを指定して、Linux 上で PWM 信号の設定、出力をします。

表 2.46 デバイスファイル（PWM）

デバイスファイル	概要
/sys/class/pwm/pwmchip*/pwm0	デフォルトのコンフィギュレーションでは次のように設定されます。 PWMN: /sys/class/pwm/pwmchipN/pwm0 (N=0~15) (例) PWM0: /sys/class/pwm/pwmchip0/pwm0 PWM1: /sys/class/pwm/pwmchip1/pwm0 ... PWM15: /sys/class/pwm/pwmchip15/pwm0

### 2.4.11.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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



## 2.4.12 TIMER

本モジュールは、RZ/V2MA に搭載されているタイマ（TIM）にて、インターバルタイマの設定を行います。

### 2.4.12.1 ハードウェア構成概要

表 2.47 ハードウェア構成（TIM）

RZ/V2MA
32ch (TIM0-TIM31)

### 2.4.12.2 提供機能

本モジュールは、以下の機能を提供します。

- 32bit のカウンタにて、インターバルタイマモードまたはフリーランモードを指定しタイマを動作

### 2.4.12.3 ソフトウェアインタフェース

本モジュールは、以下に示すインタフェースを指定して、Linux 上でタイマ信号の設定、出力をします。

表 2.48 デバイスファイル（TIM）

デバイスファイル	概要
/dev/hw_tim*	デフォルトのコンフィギュレーションでは次のように設定されます。 TIMN: /dev/hw_timN (N=0~31) (例) TIM0: /dev/hw_tim0 TIM1: /dev/hw_tim1 ... TIM31: /dev/hw_tim31

### 2.4.12.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

### 2.4.13 Watchdog Timer

本モジュールは、RZ/V2MA に搭載されているウォッチドッグタイマ（WDT）の設定を行います。

#### 2.4.13.1 ハードウェア構成概要

表 2.49 ハードウェア構成（WDT）

RZ/V2MA
2ch* (WDT0, WDT1)

【注】 WDT0 は CA53 Core0、WDT1 は CA53 Core1 に対応しています。

#### 2.4.13.2 提供機能

本モジュールは、以下の機能を提供します。

- WDT\_CLK(48MHz)で駆動される 32bit のカウンタにて、ウォッチドッグタイマを設定
- 周期カウンタは 1s から 89s の範囲で設定可能
- 指定した周期で割り込み信号が発生後、WDT カウンタがオーバーフローする前に、WDT をクリア

#### 2.4.13.3 ソフトウェアインタフェース

本モジュールは、以下に示すインタフェースを指定して、Linux 上でウォッチドッグタイマ信号の設定、出力をします。

表 2.50 デバイスファイル（WDT）

デバイスファイル	概要
/dev/watchdog*	デフォルトのコンフィギュレーションでは次のように設定されます。 WDT0: /dev/watchdog0 WDT1: /dev/watchdog1

#### 2.4.13.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.14 TSU

本モジュールは、RZ/V2MA に搭載されている Temperature sensor unit (TSU) の設定と Temperature sensor から出力された LSI の温度の取得を行います。

### 2.4.14.1 ハードウェア構成概要

表 2.51 ハードウェア構成 (TSU)

RZ/V2MA
1ch* (※TSU0, TSU1)

### 2.4.14.2 提供機能

本モジュールは、以下の機能を提供します。

- LSI の温度を取得\*
  - ※TSU0, TSU1 が取得した LSI 温度のうち、温度が高い方を選択
- LSI の温度が設定した閾値を超えた場合、システムをシャットダウンする

### 2.4.14.3 ソフトウェアインタフェース

本モジュールは、以下に示すインタフェースを指定して、TSU の制御を行います。

表 2.52 デバイスファイル (TSU)

デバイスファイル	概要
/sys/class/thermal/thermal_zone0	デフォルトのコンフィギュレーションは以下のように設定されます。 TSU: /sys/class/thermal/thermal_zone0

### 2.4.14.4 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.4.15 DMA

本モジュールは、RZ/V2MA に搭載されている DMA Controller を制御します。

### 2.4.15.1 ハードウェア構成概要

表 2.53 ハードウェア構成 (DMA)

RZ/V2MA
16ch

### 2.4.15.2 提供機能

本モジュールは、以下の機能を提供いたします。

- DMA Engine フレームワークによる DMA デバイスドライバモジュール管理
- DMA 各チャンネルの制御
- メモリ - メモリ間の DMA 転送制御
- DMA 転送が完了時に登録されているコールバック関数を呼び出す
- ディスクリプタ転送制御

### 2.4.15.3 デバイスドライバ仕様書

本モジュールの詳細は、以下のドキュメントをご参照ください。

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

## 2.5 V4L2

V4L2 (Video for Linux 2) は、Linux OS 上で動画を扱うための機能を提供する API の総称です。本パッケージの V4L2 は、USB カメラおよび MIPI カメラから動画取り込みを行うために必要な機能のみ提供いたします。

カメラからの動画取り込みでは、カメラデバイスに対応するデバイスファイル(/dev/video0 等)を介してカメラを制御し、取り込んだフレームデータをユーザー空間上にマッピングして、アプリケーションプログラムから使用する事ができます。

### 2.5.1 API

RZ/V VLP (RZ/V2M)で動作を確認した V4L2 API を以下の表に示します。

表 2.54 V4L2 にて使用するシステムコール

システムコール	概要
open	V4L2 デバイスファイルをオープンします。
close	V4L2 デバイスファイルをクローズします。
ioctl	オープンしたデバイスファイルに対して V4L2 API を発行します。
select	フレームがキャプチャされたことを確認します。
mmap	V4L2 から取得したバッファをユーザー空間にマッピングします。
munmap	マッピングされたバッファを解放します。
ioctl(VIDIOC_S_FMT)	データフォーマットを指定します。
ioctl(VIDIOC_REQBUFS)	バッファを要求します。
ioctl(VIDIOC_QUERYBUF)	バッファの情報を取得します。
ioctl(VIDIOC_QBUF)	バッファを取得します。
ioctl(VIDIOC_DQBUF)	バッファを解放します。
ioctl(VIDIOC_STREAMON)	ストリーミングを開始します。
ioctl(VIDIOC_STREAMOFF)	ストリーミングを停止します。
ioctl(VIDIOC_QUERYCAP)	デバイスの機能を問い合わせます。

【注】 システムコール “ioctl” に続く括弧内の文字列は、“ioctl” 発行時の第二引数として指定する定数名です。

### 2.5.2 ソフトウェアインタフェース

V4L2 を使用する場合は、以下に示すインタフェースを指定します。

表 2.55 デバイスファイル (V4L2)

デバイスファイル	概要
/dev/videoN	カメラモジュールに対応するデバイスファイルです。 ※N はデバイスの接続数に依存します。 ※MIPI カメラの場合、/dev/video0 で固定となり、動的に変更しません。

## 2.6 Yocto 環境、root ファイルシステム、SDK について

RZ/V VLP (RZ/V2M)は Yocto Project をベースに開発されています。Linux PC 上で本 Linux パッケージに含まれる Yocto 環境をビルド (bitbake) することにより、RZ/V2MA ボード上で Linux 起動に使用する Linux kernel のバイナリ や Linux 起動時にマウントするファイルシステムイメージ (root ファイルシステム) を生成することができます。また所定の方法で bitbake することにより、Linux PC にインストール可能なクロスコンパイル環境 (SDK) のインストール用ファイルを生成することもできます。

【注】 \*1 本パッケージがベースとした Yocto Project のバージョンについては、リリースノートをご参照ください。

\*2 bitbake や SDK の作成方法については、スタートアップガイドをご参照ください。

上記 root ファイルシステム及び SDK は、AArch64 バイナリとライブラリで構成されています。また bitbake 時に AArch32 ライブラリを有効にすることで、AArch64 に加えて AArch32 バイナリとライブラリをサポートし、multilib に対応します。(AArch32 バイナリやライブラリはユーザーによる追加設定がない限り、root ファイルシステム及び SDK には含まれません。)

なお、上記 root ファイルシステムには、GPLv3+ソフトウェアコンポーネントを含みません。オプションでの提供となります。

本 Linux パッケージは以下の表に示した構成をサポートします。

表 2.56 RZ/V VLP (RZ/V2M)構成

イメージ	概要
core-image-bsp	RZ/V2MA の基本的なデバイスドライバに対応したイメージです。
core-image-minimal	Yocto Project より提供されている core-image-minimal をベースとしています。

## 3. U-Boot

本章では、RZ/V VLP (RZ/V2M)に含まれる U-Boot について説明します。

### 3.1 機能

U-Boot は、以下の機能を提供します。

- チップ、及びボード固有のハードウェア設定
- ブートデバイス (SD, Ethernet) から Linux カーネルのロードと起動
- U-Boot 環境変数<sup>\*</sup>を eMMC へ保存
- シリアルコンソールを使用したブートコマンドの実行
- eMMC への書き込み<sup>\*</sup>

【注】 U-Boot 環境変数および eMMC への書き込み方法については、"RZ/V Verified Linux Package Start-Up Guide for RZ/V2MA"をご参照ください。

改訂記録	RZ/V Verified Linux Package ソフトウェアマニュアル for RZ/V2MA
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2022/09/16	—	初版改訂
1.10	2023/01/13	—	本 Linux Package 改版に伴い、本ドキュメントのリビジョン番号を変更
1.20	2023/07/31	—	ドキュメントタイトルを変更
		1, 17	PCI Express Endpoint 未対応である注意書きを削除



---

RZ/V Verified Linux Package ソフトウェアマニュアル for RZ/V2MA

発行年月日 2023年7月31日 Rev.1.20

発行 ルネサス エレクトロニクス株式会社  
〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

---

RZ/V Verified Linux Package ソフトウェアマニュアル  
for RZ/V2MA



ルネサスエレクトロニクス株式会社

R01US0577JJ01120