R-Car Series, 3rd Generation

Power Management

要旨

[Purpose]

車載ユースケースでは従来からエンジンキーON時の高速起動と発熱への対策が求められてきました。この２つの要求事項に対して、弊社ではR-Car Series, 3rd Generationからサポートした電力制御関連機能の利用がソリューションの１つとして有効であると考えております。本ドキュメントでは電力制御関連機能の具体的な活用方法について事例を用いて説明します。

[Readers]

本ドキュメント読者は、下記の一般的な知識を持っていることを想定しています。

* Engineering, logic circuits, microcontrollers, and Linux.
* The functionality of the multiple processor cores of R-Car H3, R-Car M3-W, R-Car M3-W+, R-Car M3-N and R-Car E3 products.
* The electrical specifications of the multiple processor cores of R-Car H3, R-Car M3-W, R-Car M3-W+, R-Car M3-N and R-Car E3 products.
* The functions of the BSP drivers for R-Car H3, R-Car M3-W, R-Car M3-W+, R-Car M3-N and R-Car E3 products.

[Note]

電源制御関連機能についてはRENESAS\_RCH3M3M3NE3\_PowerManagement\_UMEに動作概要、ユーザーインタフェース、及びパラメータなど、記載されているため、本ドキュメントでは取り扱いません。

本ドキュメントは弊社が提供するYocto v3.7.0以降を対象として記述しております。

ターゲットデバイス

・R-Car H3

・R-Car M3-W

・R-Car M3-W+

・R-Car M3-N

・R-Car E3

　各ターゲットデバイスの電力制御関連機能の対応は、RENESAS\_RCH3M3M3NE3\_PowerManagement\_UMEを参照してください。

目次

[1. 電力制御関連機能のソフトウェア仕様 3](#_Toc536614068)

[2. 要求事項と電力制御関連機能との関係 4](#_Toc536614069)

[3. エンジンキーONで高速起動を実現 5](#_Toc536614070)

[3.1 要求事項の概要 5](#_Toc536614071)

[3.1.1 背景 5](#_Toc536614072)

[3.1.2 実現方法 6](#_Toc536614073)

[3.2 System Suspend to RAMの概要 7](#_Toc536614074)

[3.3 System Suspend to RAMのSWフロー 8](#_Toc536614075)

[3.4 System Suspend to RAMの制御事例 10](#_Toc536614076)

[4. 車載端末の発熱抑止 11](#_Toc536614077)

[4.1 要求事項の概要 11](#_Toc536614078)

[4.1.1 背景 11](#_Toc536614079)

[4.1.2 実現方法 12](#_Toc536614080)

[4.2 Thermal Managementの概要 13](#_Toc536614081)

[4.3 パラメータ設定事例 14](#_Toc536614082)

[5. 電力制御関連機能の設定事例 17](#_Toc536614083)

[5.1 Typical mode設定事例 18](#_Toc536614084)

[5.2 High performance mode設定事例 18](#_Toc536614085)

[5.3 Low power mode設定事例 18](#_Toc536614086)

[Appendix 19](#_Toc536614087)

[A1. 概要 19](#_Toc536614088)

[A2. CPU online/offlineに要する処理時間 20](#_Toc536614089)

[A3. DVFS/DFS使用時の注意点 21](#_Toc536614090)

[A3.1 DFS制御の処理フローと処理時間 22](#_Toc536614091)

[A3.2 DVFS制御の処理フローと処理時間 23](#_Toc536614092)

[A4. System Suspend to RAMの復帰時間の注意点 24](#_Toc536614093)

# 電力制御関連機能のソフトウェア仕様

電力制御関連機能の活用の前段として、本章では各機能のソフトウェア仕様の概要を説明します。

表 1‑1　電力制御関連機能のソフトウェア概要

|  |  |
| --- | --- |
| 機能名 | ソフトウェア仕様の概要説明 |
| CPU Hotplug | CPU Hotplugは、システム稼働中にユーザーがCPUのonline/offline制御をする機能です。offlineにされたCPUはKernel管理下から外れ、電源がshutdownされます。 |
| CPU Idle | CPU Idleは、次のタイマー割り込みまでの時間を基に、KernelがCPUをSleep modeもしくはCore Standby modeへ動的に遷移させる機能です。次のタイマー割り込みまでの時間がながければ、より省電効果の高いCore Standby modeへ遷移します。 |
| CPU Freq | CPU Freqは、CPUの動作周波数、及び電圧を変更する機能です。 |
| Runtime PM | Runtime PMは、各デバイスのクロックと電源ドメインを一括で管理し、各デバイスの使用状況に応じて、クロック/電源ドメインのOn/Off制御を行う機能です。 |
| System Suspend to RAM | System Suspend to RAMは、動作中の状態を保持したままシステムを停止させ、その後、前回動作中の状態で復帰することできる機能です。また、System Suspend to RAM中にバックアップ電源を除き、PMICから出力されるすべての電源系統を停止することで暗電流の削減も可能となります。 |
| Intelligent power allocation (IPA) | IPAは、SoCのジャンクション温度を監視し、SoCの限界耐熱温度が許容量を超えないために、CPU Freqを用いてパフォーマンスを調整する機能です。 |
| Emergency shutdown (EMS) | EMSは、SoCのジャンクション温度を監視し、SoCの限界耐熱温度が許容量を超えないために、CPU Hotplugを用いてCPUの電源をshutdownさせる機能です。 |
| Capacity Aware migration Strategy  (CAS) | CASは Cortex-A57とCortex-A53が同時に動作する環境下において、性能を重視したタスク配置を行います。 |

# 要求事項と電力制御関連機能との関係

本章では２つの要求事項(「エンジンキーONで高速起動を実現」と「車載端末の発熱抑止」)と電力制御関連機能の関連を示し、実現方法の詳細については3章と4章で示します。

**CPU core frequency control**

**Power state coordination interface (PSCI)**

電力制御関連機能

SoC依存機能

**Secure Layer**

**Kernel Layer**

**CPU Freq**

**PMIC driver**

**Thermal control driver**

**CPU**

**Hotplug**

**Runtime**

**PM**

**CPU Idle**

**System Suspend to RAM**

**Clock**

**Framework**

**CPU Core**

**Power Switch**

**IPA**

**Power domain**

**control**

**EMS**

**エンジンキーONで**

**高速起動を実現**

**車載端末の発熱抑止**

要求事項

図 2‑1　要求事項と電力制御関連機能の関連性

# エンジンキーONで高速起動を実現

## 要求事項の概要

### 背景

近年、車載情報端末は従来のカーナビ、カーオーディオといった枠組みを超え、エンターテイメントの充実や情報通信など多様なニーズが求められています。さらに、効率性・利便性・快適性等のストレスフリーなサービス対する要望も増してきております。そのうちの一例として、エンジンキーONですぐにナビゲーションや各種ボタン操作ができることは、ユーザーに快適性の向上に繋がります。特に、前回エンジンキーOFF時に中断していた音楽や動画をすぐに続きから再生できることは、いまやユーザーは当たり前のように求めていることであり、エンジンキーON時の高速起動は不可欠と言えます。

### 実現方法

エンジンキーON時の高速起動を実現するには、システム起動時で処理時間の長い処理(DRAMメモリの展開、ドライバの初期化、Applicationの初期化)を短縮する必要があります。短縮の手法の１つとして、エンジンキーOFF時に動作中の状態を保持したままシステムを停止させ、エンジンキーON時に前回動作中の状態で復帰することで処理時間の長い処理をスキップすることができ、これを実現できるのがSystem Suspend to RAMとなります。また、System Suspend to RAM中はDRAM内に前回動作中の状態を保持するためDRAMの電源はONにする必要がありますが、これにより車載バッテリーが枯渇した場合はSystem Suspend to RAMで復帰できなくなります。そのため、System Suspend to RAM時はバッテリー消費を抑える(暗電流削減)ために最小限の電源系統のみを使用する設計も同時に必要となります。

弊社では、System Suspend to RAMと暗電流対策の２つを標準サポートとして提供致します。

## System Suspend to RAMの概要

弊社がサポートするSystem Suspend to RAMは、動作中の状態を保持したままシステムを停止させ、その後、前回動作中の状態で復帰する機能を実現します。また、System Suspend to RAM中にバックアップ電源を除き、PMICから出力されるすべての電源系統を停止することで暗電流の削減も対策しております。図 3‑1に示すのは暗電流削減時の状態遷移イメージです。(System Suspend to RAMのソフトウェアフローは3.3章を参照ください)

DDR0/

DDR1

VD33, VD18, etc.

VD33, VD18, etc.

DVFS, VDD

DVFS, VDD

I2C\_SCL

I2C\_SDA

I2C\_SCL

I2C\_SDA

BKUP\_TRG

BKUP\_TRG

BKUP\_CTRL

BKUP\_CTRL

PRESETB

PRESETB

DDR０C/DDR1C

DDR０C/DDR1C

RSTB

RSTB

OFF

ON

Separate Devices

SW23

PMIC

DRAM

DDR IO interface

R-Car H3

Separate Devices

SW23

PMIC

DRAM

DDR IO interface

R-Car H3

Normal operation

System Suspend To RAM

電源OFF

電源ON

DDR0/

DDR1

図 3‑1　System Suspend to RAMにおける電源状態

## System Suspend to RAMのSWフロー

System Suspend to RAMのソフトウェアフローを図 3‑2に示します。

ARM Trusted Firmware

ARM Trusted Firmware

App

Drivers & Kernel

Drivers & Kernel

App

* + **Store playback point by App**
  + **Stop data transfer and close Drivers by App**
  + **Re-start playback point by App**
  + **Initialize, re-detect devices and open Drivers**

**Detect Suspend trigger**

Detect Resume trigger

**Call sysfs-IF(\*1)**

Framework start to work

All process wakeup

🡪 Power Manager start

Suspend

Resume

Suspend processing：

- Backup registers by ARM Trusted Firmware

Resume processing：

- Restore registers by ARM Trusted Firmware

All process freezing

🡪 Power Manager stop

Suspend processing：

- If needs, backup register by driver

- If needs, stop clock for device

Resume processing：

- If needs, restore register by driver

- If needs, supply clock for device

- If needs, initialize driver

(\*1) “Suspend to RAM”要求を発行する前にSuspendへ遷移可能な状態にすること。

図 3‑2　System Suspend to RAMのソフトウェアフロー

System Suspend to RAMは、Suspend時はデータの退避やデバイスの停止等を行い、Resume時はデータのリストアやデバイスの再開等を行います。これにより動作中の状態を保持したままシステムを停止させ、その後、前回動作中の状態で復帰することを実現しております。通常のシステム起動に比べて処理時間の長い処理(DRAMメモリの展開、ドライバの初期化、Applicationの初期化)をスキップすることができるため、復帰後にすぐにユーザー操作が可能となります。

## System Suspend to RAMの制御事例

System Suspend to RAM利用時に、動作中のアプリを継続動作させるユースケースの場合は、お客様にて

Suspend時に動作中のアプリを中断し、Resume時にアプリを再開する必要があります。

一例としてVideo playback動作中にSystem Suspend to RAMを実行するユースケースを紹介します。

図 3‑3に示す通り、動画再生アプリはGstreamerを使用してVideo playbackを行います。また、各アプリの管理やSystem Suspend to RAMの実行はシステムマネージャが行います。

Suspend時は現在再生中の状態(再生位置)を保持し、Resume時には保持した再生中の状態から再開することにより、Video playbackが継続して動作し続けることを実現しています。

(Video playbackの詳細は” RENESAS\_RCH3M3M3NE3\_GStreamer\_UsersManual\_UME”を参照下さい)

User

Linux

HW

動画再生アプリ

システムマネージャ

GStreamer

ドライバ

動画再生

動画再生

中断

再生中

再開

画像更新

Suspend

画像更新

再生中

\*1

\*2

\*3

\*4

\*5

\*6

Display表示

Display表示

\*7

図 3‑3　 Video playback動作中のSystem Suspend to RAMユースケース

表 3‑1　 Video playbackの詳細動作

|  |  |  |
| --- | --- | --- |
| Notes | 概要 | 処理内容 |
| \*1 | 動画再生開始要求 | 動画再生アプリからGStreamerへ動画再生を要求する。 |
| \*2 | アプリ中断要求 | システムマネージャから動画再生アプリへ中断を要求する。 |
| \*3 | 動画再生終了要求 | 動画再生アプリからGStreamerへ再生停止を要求する。  （動画再生アプリにて再生位置を保持する） |
| \*4 | System Suspend to RAM実行 | System Suspend to RAMを実行し、suspend状態に移行する。 |
| \*5 | Resume要求 | Suspendから復帰する。 |
| \*6 | アプリ再開要求 | システムマネージャから動画再生アプリへ再開を要求する。 |
| \*7 | 動画再生再開要求 | 動画再生アプリからGStreamerへ再生位置を指定して、動画再生を要求する。 |

# 車載端末の発熱抑止

## 要求事項の概要

### 背景

近年、車載情報端末は従来のカーナビ、カーオーディオといった枠組みを超え、エンターテイメントの充実や情報通信など多様なニーズに合わせて高機能且つ高性能なシステムへと進化し続けています。その一方で、性能向上による消費電力の増加は、車載情報端末で使用されるSoCやDRAMなどの耐熱規格を超える膨大な発熱を引き起こし、熱対策・熱設計の難易度は高まっています。また、発熱は半導体の製品寿命や故障率に大きく影響を与えることから、製品ベンダは、ファンや放熱器など多大なコストを費やし対策しているのが現状です。そのため、ソフトウェアで発熱抑止をすることで、ファンや放熱器などのボムコストを抑えることが強く求められています。

### 実現方法

発熱を抑止する場合、従来からソフトウェアで実現できる手法としては消費電力を削減することが一般的です。R-Car Series, 3rd GenerationからはKernel層でSoCの温度上昇に応じて自動で消費電力を削減する機能をサポートしています(Thermal Management)。本章ではこのThermal Managementの概要と事例について説明します。

## Thermal Managementの概要

* Thermal Managementは、高温環境下において発熱を抑止するために消費電力をコントロールすることができます。

(詳細は”RENESAS\_RCH3M3M3NE3\_PowerManagement\_UM”を参照下さい)

表 4‑1　Thermal Managementのソフトウェア概要

|  |  |
| --- | --- |
| 機能名 | ソフトウェア仕様の概要説明 |
| IPA | IPAは、SoCのジャンクション温度を監視し、SoCの限界耐熱温度が許容量を超えないために、CPU Freqを用いてパフォーマンスを調整する機能です。 |
| EMS | EMSは、SoCのジャンクション温度を監視し、SoCの限界耐熱温度が許容量を超えないために、CPU Hotplugを用いてCPUの電源をshutdownさせる機能です。 |

* 消費電力をコントロールする際、動作周波数/電圧、及びCortex-A57の一部コアの電源遮断を行うため、性能面では大幅な低下が見込まれます。つまり、これは動作中のApplicationのレスポンスが悪くなり、ユーザーへの快適性の低下につながる可能性があります。Thermal Managementはあくまで耐熱規格を超える発熱から製品を守るためのバックアッププランになります。ユーザーへの快適性を維持する場合は、お客様にて高温環境下に至る前に不要なApplicationを停止するなど消費電力を下げる施策を実施頂くことを推奨します。
* Thermal Managementは制御に必要なパラメータ設定があり環境に依存します。パラメータの設定方法は4.3章を参照ください。

## パラメータ設定事例

* Thermal managementのパラメータは、お客様の環境に依存し設定する必要があります。
* 弊社ではSalvator-Xを用いて表 4‑2の考え方の基でパラメータを設定しました。

表 4‑2　パラメータ設定事例

|  |  |  |  |
| --- | --- | --- | --- |
| # | パラメータ | 考え方 | 弊社 設定値 |
| ① | IPA開始/停止温度 | ②よりも低く、且つお客様環境におけるIdle状態(低負荷)でのTjよりも高く設定して下さい。Idle状態でのTjよりも低く設定した場合、常にIPAが動作することとなります。 | 90℃ |
| ② | IPAターゲット温度 | ③よりも低く設定して下さい。また、お客様の想定ユースケースにおける最大性能を満たすポイントを設定下さい。 | 100℃ |
| ③ | EMS開始温度 | ⑤よりも低く設定して下さい。なお、お客様環境の放熱対策に依存しEMS開始(CPUコアが4つ->1つ)直後、④を下回る可能性があり、その場合負荷状況に応じて③と④を繰り返します。そのため、③と④の設定はお客様環境における方熱対策効果を十分考慮した上で設定して下さい。 | 110℃ |
| ④ | EMS停止温度 | ③よりも低く設定して下さい。なお、お客様環境の放熱対策に依存しEMS停止(CPUコアが1つ->4つ)直後、③を上回る可能性があり、その場合負荷状況に応じて③と④を繰り返します。そのため、③と④の設定はお客様環境における方熱対策効果を十分考慮した上で設定して下さい。 | 95℃ |
| ⑤ | System shutdown発動温度 | SoCの限界耐熱温度の要件に従い設定して下さい。 | 120℃ |

* お客様環境における放熱対策を考慮した上で、パラメータを設定しないと図 4‑1のグラフに示すように開始と停止を繰り返すことになります。

|  | 温度Tj[℃] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | EMS開始 |  |  |  | EMS開始 |  |  |  |  |  |
|  | EMS開始温度 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | EMS開始 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | EMS停止温度 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | EMS停止 |  |  | EMS停止 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Core数 | 4 | | | | | | 1 | | 4 | | 1 | | 4 | | 1 | |  |
|  | EMS制御 | － | | | | | | EMS開始 | | EMS停止 | | EMS開始 | | EMS停止 | | EMS開始 | |  |
|  |  |  | | | |  | |  | |  |  |  |  |  |  |  |  |  |

図 4‑1　不適切なパラメータ設定による温度推移

* 弊社設定パラメータとSalvator-Xにおける高負荷と中負荷での制御イメージを図 4‑2に示します。

|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | 温度Tj[℃] |  |  |  | *IPA動作期間* |  |  | EMS動作期間 |  |  |  |  |  |  | *IPA動作期間* |  |  |
|  | SS発動温度：120℃ | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | EMS開始温度：110℃ | |  |  |  |  |  | EMS開始 |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  | 負荷低下(外部要因) |  |  |  |  |  |  |  |
| IPAターゲット温度：100℃ |  | |  |  |  |  |  |  |  |  |  |  |  | Tjが飽和温度に達した |  |  |  |  |  |
|  | EMS停止温度：95℃ | |  |  |  |  |  | Tjが飽和温度に達した |  |  |  |  |  |  |  |  |  |  |  |
| IPA開始/停止温度：90℃ |  | |  |  |  |  |  |  |  |  |  |  |  | EMS停止 |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  | IPAスタート |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 負荷 | | | 高負荷 | | 高負荷 | | 高負荷 | | | | 中負荷 | | 中負荷 | | | |  |  |
|  | 発熱制御 | IPA制御 | | 無効 | | 有効 | | 無効 | | | | 無効 | | 有効 | | | |  |  |
|  | EMS制御 | | 無効 | | 無効 | | 有効 | | | | 有効 | | 無効 | | | |  |  |
|  | CPU  (Cortex-A57) | コア数 | | 4 | | *4* | | 1 | | | | 1 | | *4* | | | |  |  |
|  | 動作周波数 | | 1.7GHz | | *0.5～1.7GHz* | | 0.5GHz | | | | 0.5GHz | | *0.5～1.7GHz* | | | |  |  |
|  | GPU | Shading | | 3 | | *3* | | 1 | | | | 1 | | *3* | | | |  |  |
|  | 動作周波数 | | 0.6GHz | | *0.6GHz* | | 0.6GHz | | | | 0.6GHz | | *0.6GHz* | | | |  |  |
|  | (\*) 高負荷：CPU及びGPUの負荷が90%-100% | | | | | | | | | | | | | | | | |  |  |
|  | 中負荷：CPU及びGPUの負荷が50%程度 | | | | | | | | | | | | | | | | |  |  |
|  |  | |  | | | |  | |  | |  |  |  |  |  |  |  |  |  |

図 4‑2　高負荷・中負荷における温度遷移イメージ

# 電力制御関連機能の設定事例

電力制御関連機能の設定は、お客様のユースケースに強く依存することから、お客様にて適切に設定して頂く必要があります。下記は電力制御関連機能の設定の参考事例として、表 5‑1に示す3つのモードを紹介します。

なお、各モードの設定事例は、R-Car H3を使用した場合の設定例となります。

表 5‑1　電力制御モードの設定事例

|  |  |
| --- | --- |
| モード | 概要 |
| Typical mode | 電力制御関連機能を有効にしたまま、標準スペックの処理性能を維持して動作させる場合に使用するモードです。  (R-Car Series, 3rd Generation BSPのデフォルト設定) |
| High performance mode | システム全体を最大性能で動作させる場合に使用するモードです。  但し、消費電力／発熱が増加するので注意が必要です。 |
| Low power mode | システム全体を低消費電力で動作させる場合に使用するモードです。  但し、システム全体の処理性能は低下するため注意が必要です。 |

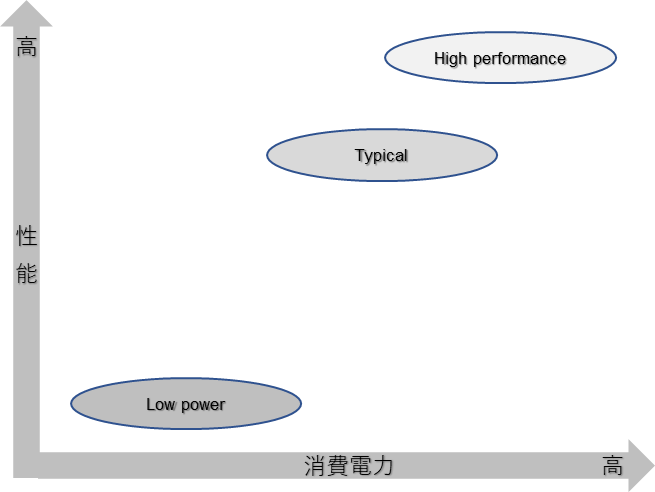


図 5‑1　 電力制御関連機能のモード設定による性能と消費電力

## Typical mode設定事例

表 5‑2　Typical mode設定事例

|  |  |  |  |
| --- | --- | --- | --- |
|  | 機能 | 概要 | コマンド |
| 1 | CPU Freq | 周波数を指定する（1.5GHz） | 設定不要（標準BSPデフォルト設定） |
| 2 | CPU Idle | CPU Idleを有効にする | 設定不要（標準BSPデフォルト設定） |
| 3 | CPU Hotplug | CPU使用個数を最大にする | 設定不要（標準BSPデフォルト設定） |

## High performance mode設定事例

表 5‑3　 High performance mode設定事例

|  |  |  |  |
| --- | --- | --- | --- |
|  | 機能 | 概要 | コマンド |
| 1 | CPU Freq | 周波数を指定する（1.7GHz） | $ echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  $ echo 1 > /sys/devices/system/cpu/cpufreq/boost  $ echo 1700000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_setspeed |
| 2 | CPU Idle | CPU Idleを無効にする | $ echo 1 > /sys/devices/system/cpu/cpu0/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu1/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu2/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu3/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu4/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu5/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu6/cpuidle/state1/disable  $ echo 1 > /sys/devices/system/cpu/cpu7/cpuidle/state1/disable |
| 3 | CPU Hotplug | CPU使用個数を最大にする | 設定不要（標準BSPデフォルト設定） |

## Low power mode設定事例

表 5‑4　Low power mode設定事例

|  |  |  |  |
| --- | --- | --- | --- |
|  | 機能 | 概要 | コマンド |
| 1 | CPU Freq | 周波数を指定する（500MHz） | $ echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  $ echo 500000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_setspeed |
| 2 | CPU Idle | CPU Idleを有効にする | 設定不要（標準BSPデフォルト設定） |
| 3 | CPU Hotplug | CPU1-3をofflineにする | $ echo 0 > /sys/devices/system/cpu/cpu1/online  $ echo 0 > /sys/devices/system/cpu/cpu2/online  $ echo 0 > /sys/devices/system/cpu/cpu3/online |

# Appendix

# 概要

本章ではAppendixでは、「エンジンキーONで高速起動を実現」と「車載端末の発熱抑止」を実現するために用いた電力制御関連機能であるCPU HotplugとCPU Freqを個別に使用する場合の注意事項や処理時間等について説明します。なお、記載した値は測定環境や測定条件により差が生じる可能性があり、参考値扱いとなります。

# CPU online/offlineに要する処理時間

CPU Hotplugの使用用途としては、低消費電力や発熱対策のために一部のCPUをofflineにする場合と、一部のCPUがofflineの状態から処理性能が必要となりCPUをonlineにする場合の２つのケースが考えられます。後者のケースは、CPU onlineに掛かる時間だけタイムラグが発生するため、性能が要求されるユースケースで使用する場合は注意ください。以下にCPU Hotplugによる各処理時間を示します。

* **処理時間**
* **CPU online : 36ms**
* **CPU offline : 53ms**
* 測定環境
* SW : Yocto v3.7.0(Kernel BSP 3.6.2)
* HW : R-Car H3 Ver.3.0 (Salvator-XS)
* 測定条件
* 起動後、アプリを動作させないidle状態で測定
* ターゲットCPUは任意のCortex-A57
* 測定ポイント
* cpu\_up/cpu\_down関数の処理時間
* 測定方法
* ftraceにて計測

# DVFS/DFS使用時の注意点

DVFS/DFS制御は、変更する周波数帯によって処理時間が異なります。また、PLL0の逓倍率変更を伴う周波数変更を行った場合は、ＨＷ仕様により変更処理中にCPUが最大で50us低速動作をします。そのため、us単位の性能が要求される場合(例：SDアクセス)や50us以内でレスポンスを期待する場合(例：割り込み)では、 PLL0の逓倍率変更を伴う周波数変更を実施しないよう注意願います。表 A3‑1にPLL0の逓倍率変更が発生する条件を示します。

表 A3‑1　DVFS/DFS設定テーブル

Normal（\*2）

Boost (\*1)

DFS制御の範囲

DVFS制御の範囲

|  |  |  |  |
| --- | --- | --- | --- |
| Cortex-A57  Frequency | Voltage | CPU Divider | PLL0 |
| 1.7 GHz | 0.96 V | 32/32 | 100逓倍 |
| 1.6 GHz | 0.90 V | 32/32 | 96逓倍 |
| 1.5 GHz | 0.82 V | 32/32 | 90逓倍 |
| 1.0 GHz | 0.82 V | 21/32 | 90逓倍 |
| 0.5 GHz | 0.82 V | 11/32 | 90逓倍 |

(\*)R-Car H3 AVS0の周波数テーブルを例として説明しています

(\*1) Boost←→のNormalの切替え、及びBoost有効下での周波数変更でPLL0の逓倍率変更が行われます。

(\*2) Normal範囲内はPLL0の逓倍が等しいため、本範囲内の周波数変更時は50usの低速動作は発生しません。

表 A3‑1で示した条件に該当する操作は以下の２つとなります。  
(詳細は”RENESAS\_RCH3M3M3NE3\_PowerManagement\_UME”を参照下さい)

* sysfs経由のboostの有効/無効の切り替え設定
* boost有効時の動作周波数変更(静的変更、及び動的変更)

また、DVFS/DFSの処理の差分についてはA3.1章、A3.2章を参照ください。

## DFS制御の処理フローと処理時間

DFS制御の処理フローを例として図 A3‑1に示します。

Clock framework

cpufreq framework

regulator

framework

i2c-dvfs

PMIC

Z clock

PLL0

周波数変更設定

分周変更

14us

33us

SW

HW

DFSは分周変更のみで実現できます。そのため処理時間は33us程度となります。

本結果は以下の条件下で測定したもので、環境や条件によっては数値が異なる可能性があります。

＜測定環境＞

・SW : Yocto v3.7.0(Kernel BSP 3.6.2)

・SoC : R-Car H3 Ver.3.0(Salvator-XS)

・PMIC : bd9571mwv

・測定条件 : 起動後、アプリ動作未のidle状態

・測定ポイント : 周波数変更設定(clk\_set\_rate)

図 A3‑1　DFS制御シーケンス (0.5GHz->1.5GHz)

## DVFS制御の処理フローと処理時間

DVFS制御の処理フローを図 A3‑2に示します。

673us

123us

逓倍率変更

分周変更

813us

周波数変更設定

電圧変更設定

SW

HW

Clock framework

cpufreq framework

regulator

framework

i2c-dvfs

PMIC

Z clock

PLL0

PMIC情報取得

read modify write

read x 2 times

PMIC設定変更

BoostモードはPLL0の逓倍率を変更するため、上記の123usの内、PLLの発振安定待ち時間(最大50us)が必要です。

本区間は割り込み禁止ではないため別の処理が動作することは可能

その間はPLL0からの出力クロックを使用できないため、Cortex-A57へEXTALが供給されて低速動作します。

本結果は以下の条件下で測定したもので、環境や条件によっては数値が異なる可能性があります。

＜測定環境＞

・SW : Yocto v3.7.0(Kernel BSP 3.6.2)

・SoC : R-Car H3 Ver.3.0(Salvator-XS)

・PMIC : bd9571mwv

・測定条件 : 起動後、アプリ動作未のidle状態

・測定ポイント :電圧変更設定(\_set\_opp\_voltage)、周波数変更設定(clk\_set\_rate)

図 A3‑2　DVFS制御シーケンス(0.5GHz->1.7GHz)

# System Suspend to RAMの復帰時間の注意点

System Suspend to RAM実行時、USBデバイスの接続状況に応じてSuspendからの復帰時間が遅延する場合があります。具体的なサンプル事例を表 A4‑1に示します。

表 A4‑1　USBデバイスの接続状況毎の復帰時間

|  |  |  |  |
| --- | --- | --- | --- |
| 接続端子 | 接続方法 | USBデバイス | 復帰時間(\*1)(\*2) |
| USB 2.0 | 直接接続 | USBマウス x 1 | 1391 ms |
| USB 2.0 | HUB経由で接続 | USBマウス x 1 | 2266 ms |
| USB 3.0 | 直接接続 | USBマウス x 1 | 1653 ms |
| USB 3.0 | HUB経由で接続 | USBマウス x 1 | 2362 ms |
| 無し | 無し | 無し | 1391 ms |

(\*1) Yocto v3.15.0で測定したサンプル結果の１つであり、お客様環境では数値が若干異なる場合が御座います。

(\*2) 復帰時間は、ARM Trusted Firmwareの先頭からKernel層のresume処理が完了するまでの時間を測定

これらの復帰時間の遅延はLinux KernelのUSBフレームワーク実装に起因しています。本フレームワークではサポートしている全てのUSBドライバを考慮したリセットシーケンスが実装されているため、弊社USBドライバでは不要なWait処理も実行されます。また、弊社BSPではLinux KernelのSoC依存部外であるフレームワーク実装に対して個別修正は行わないポリシーとしておりますので、同様にUSBフレームワークも個別修正は行いません。

本問題は以下の方法で回避することが可能です。

* System Suspend to RAMの実行前後でUSB 2.0ホストコントロールドライバのunload/reloadを実行

しかしながら、ユーザーアプリケーション側の対策となるため、お客様側でご対応頂く必要が御座います。

詳細手順を以下に示します。

Step1 : USB 2.0ホストコントロールドライバのモジュール化と組み込み

(1) Linux Kernel Configurationの変更

arch/arm64/configs/defconfig

CONFIG\_USB\_EHCI\_HCD=m

CONFIG\_USB\_OHCI\_HCD=m

CONFIG\_PHY\_RCAR\_GEN3\_USB2=m

(2) Linux KernelとModuleのリビルド

$ make defconfig

$ make Image

$ make modules

(3) Linux KernelとModuleのインストール

Linux:

arch/arm64/Image を差し替える

Module:

$ make modules\_install INSTALL\_MOD\_PATH=<rootfs\_dir>

Step2 : USB 2.0ホストコントロールドライバモジュールのunload/reload

(1) System Suspend to RAM実行前に対象モジュールをunload

$ rmmod -f ohci-platform

$ rmmod -f ohci-hcd

$ rmmod -f ehci-platform

$ rmmod -f ehci-hcd

$ rmmod -f phy\_rcar\_gen3\_usb2

(2) System Suspend to RAM実行後に対象モジュールをreload

$ modprobe phy\_rcar\_gen3\_usb2

$ modprobe ehci-hcd

$ modprobe ehci-platform

$ modprobe ohci-hcd

$ modprobe ohci-platform

注意事項：本手順は以下の条件に該当する場合には適用できません。

- ユーザ側がUSBデバイスを使用中でSuspend実行前にドライバのrmmodが実施できない場合

- 該当USBデバイスがHID devicesの場合(mouse/keyboard)

改訂記録

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | 発行日 | 改訂内容 | |
| ページ | ポイント |
| 1.00 | 2017.04 | － | 新規作成 |
| 1.01 | 2017.07 | P1,4,10-11,21 | ・本文の記述を更新 |
| P12,21 | ・本文の誤記訂正 |
| P2 | ・表を更新(表1-1) |
| P4,10,17-18 | ・図を更新(図2-1,3-3,4-1,4-2) |
| P7 | ・図を訂正(図3-1) |
| 1.02 | 2017.11 | P1 | ・ターゲットデバイスとReadersにR-Car M3-Nを追加 |
| P1,P15,P21 | ・参照ドキュメントのファイル名を更新 |
| 1.03 | 2018.06 | P1 | ・ターゲットデバイスとReadersにR-Car E3を追加 |
| P1,P13,P21 | ・参照ドキュメントのファイル名を更新 |
| P3 | ・表を更新(表1-1) |
| P10 | ・3.4 System Suspend to RAMの制御事例を追加 |
| P17 | ・5. 電力制御関連機能の設定事例を追加 |
| 1.04 | 2019.02 | P24,P25 | ・A.4 System Suspend to RAMの復帰時間の注意点を追加 |
| P1 | ・ターゲットデバイスとReadersにR-Car M3-W+を追加 |

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、VIL（Max.）からVIH（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、VIL（Max.）からVIH（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うと、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、ノイズ幅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。

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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。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、本資料に記載された製品デ－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、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。

5. 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も、当社は一切その責任を負いません。

6. 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。

7. 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。

8. 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。

9. 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。

10. 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。

11. 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。

12. 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.4.0-1 2017.11)

|  |  |  |
| --- | --- | --- |
| 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](https://www.renesas.com/) |  |  |
| 商標について |  |  |
| Arm、CortexはArm Limitedの登録商標または商標です。  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。 |  |  |



変更内容〔ルネサス内部向け〕

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | 発行日 | 改訂内容 | |
| ページ | ポイント |
| 0.5 | Oct. 27th, 2016 |  | ・カラー図をモノクロに変更  - パワポP.8, 11, 12, 13, 25, 26  ・パワポ図(編集不可能な絵)をワードで作成  - パワポP.19, 20  ・文言の変更  - パワポP.11「以下」→「図3-1」  - パワポP.12「以下」→「図3-2」  - パワポP.13「下記」→「表3-1」  - パワポP.15「以下」→「表3-3」  - パワポP.19「右のグラフ」→「図4-1のグラフ」  - パワポP.19「下記」→「表4-2の」  - パワポP.20「右図」→「図4-2」  - パワポP.24「以下」→「表A3-1」  - パワポP.24「上記」→「表A3-1」  - パワポP.25「以下」→「図A3-1」  - パワポP.26「以下」→「図A3-2」  ・注記マークを (\*) のスタイルに統一  - ワードP.17  - パワポP.23,24 ※  ・文言の削除  - パワポP. 23「36ms」 |
| 1.00 | 2017.04 | － | ・Rev0.51をベースとしてRev1.00とする  ・ファイル名変更  ・ターゲットデバイス名を記載  ・ドキュメント管理番号、発行年月を記載  ・ご注意書きのページのCOPYRIGHTを2017年版に変更 |
| 1.01 | 2017.07 | － | ・ヘッダーに「CONFIDENTIAL」を付加 |
| P1,4,10-11,  21 | 翻訳からのフィードバックを反映 |
| ・本文の記述を更新 |
| P12,21 | ・本文の誤記訂正 |
| P2 | ・表を更新(表1-1) |
| P4,10,17-18 | ・図を更新(図2-1,3-3,4-1,4-2) |
| P7 | ・図を訂正(図3-1) |
| 1.02 | 2017.11 | P1 | ・ターゲットデバイスとReadersにR-Car M3-Nを追加 |
| P1,P15,P21 | ・参照ドキュメントのファイル名を更新  (RENESAS\_RCH3M3M3N\_PowerManagement\_UME) |
| 1.03 | 2018.06 | P1 | ・ターゲットデバイスとReadersにR-Car E3を追加 |
| P1,P21,P23,  P24 | ・Yoctoのバージョンを3.7.0に修正 |
| P1,P13,P21 | ・参照ドキュメントのファイル名を更新  (RENESAS\_RCH3M3M3NE3\_PowerManagement\_UME) |
| P3 | ・表を更新(表1-1)  CASの説明追加 |
| P10 | ・3.4 System Suspend to RAMの制御事例を新規追加 |
| P17 | ・5. 電力制御関連機能の設定事例を新規追加 |
| P20,P22,P23 | ・処理時間、測定ポイント、測定方法を更新 |
| 1.04 | 2019.01 | P1 | ・Yoctoのバージョンを3.7.0以降に修正 |
| P24,P25 | ・A.4 System Suspend to RAMの復帰時間の注意点を追加 |
| P1 | ・ターゲットデバイスとReadersにR-Car M3-W+を追加 |