

## Renesas Synergy™ プラットフォーム

# DK-S3A7 向け Power Profiles v2 (PPV2) フレームワーク

R20AN0498JU0100  
Rev.1.00  
2018.07.13

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

## 要旨 (Introduction)

このアプリケーションプロジェクトを利用することで、ユーザは Power Profiles v2 (PPv2) フレームワークモジュール (Framework Module) を効率的に使用でき、Synergy MCU に搭載されている複数の電力制御モードと低消費電力モード (LPM) を効果的に使えるようになります。このドキュメントでは最初に、PPv2 の電力プロファイル設定 (power profile configuration) と API で抽象化された Synergy MCU の複数の電力制御モードと LPM に関する基本的な知識を紹介します。次に、アプリケーションで PPv2 を使用する場合の PPv2 フレームワークの動作フローと基本的な実装について説明します。最後に、電力制御モードと LPM モードの遷移を使用する PPv2 フレームワークの例を示します。

PPv2 フレームワークは PPv1 よりも、Synergy MCU の電力制御モードと LPM モードの制御性を向上しました。PPv2 フレームワークは、LPM v1 ドライバを使用する SSP (Synergy ソフトウェアパッケージ) フレームワークにおいて、Power Profiles v1 の全機能をサポートしています。ただし、Power Profiles v1 と PPv2 の各フレームワークに互換性はありません。同じプロジェクトで、Power Profiles v1 と PPv2 両方のフレームワークを使用することはできません。すべての新規プロジェクトで、アプリケーションに PPv2 フレームワークを使用することを推奨します。

本資料ではユーザが効率的に開発を始めることができるように、ユーザによるモジュールの追加、ターゲットアプリケーションのための正しい設定 (configuration)、参考として含まれているサンプルアプリケーションコードを利用したコード作成方法などを説明しています。モジュールのより高度な使用方法についての API に関する詳細な説明は、SSP ユーザーズマニュアル 1.3.0 またはそれ以降を参照してください。複雑な設計を行う際の情報源として、この SSP ユーザーズマニュアルは役立ちます。

## 必要とする環境 (Required Resources)

このアプリケーションノートで説明する手順を実行するには、以下のものがが必要です。

- Renesas DK-S3A7(v2.0) キット
- Micro USB ケーブル
- e<sup>2</sup> studio ISDE 5.4.0.023 またはそれ以降
- IAR Embedded Workbench® for Renesas Synergy™ 7.71.3 またはそれ以降 (IAR Systems)
- Synergy ソフトウェアパッケージ (SSP) v1.3.0 またはそれ以降
- Renesas Synergy Standalone Configurator (SSC) v5.4.0.023 またはそれ以降

必要な Renesas ソフトウェアは Renesas Synergy ギャラリー (<https://synergycastle.renesas.com>) からダウンロードできます。

## 前提条件 (Prerequisites)

このアプリケーションノートは、ユーザが Renesas e<sup>2</sup> studio ISDE と SSP を使用した経験があることを前提に説明しています。たとえば、このアプリケーションノートの手順を実行する前に、『SSP ユーザーズマニュアル』の手順に従い「Blinky」プロジェクトをビルドして実行する必要があります。この作業を実行すると、e<sup>2</sup> studio と SSP の使用方法を理解でき、さらにボードへのデバッグ接続が正常に機能していることを確認できます。

対象ユーザは、S1/S3/S5/S7 Synergy MCU シリーズと PPv2 を組み合わせて使用し、アプリケーションを開発される方々です。

## 目次

1.	Synergy Power Profiles の概要 (Synergy Power Profiles Overview) .....	5
1.1	Power Profile v2 フレームワーク (Power Profile v2 Framework Overview) .....	5
1.1.1	PPv2 フレームワークの機能 (PPv2 Framework Features) .....	5
1.2	Synergy MCU Power Profile の概要 (Synergy MCU Power Profile Overview) .....	5
1.2.1	Synergy MCU の電力制御モード (Synergy MCU Power Control Modes).....	5
1.2.2	Synergy MCU の低消費電力モード (LPM) (Synergy MCU Low Power Modes) .....	7
2.	Power Profile v2 フレームワークの動作概要 (Power Profile v2 Framework Operational Overview) .....	11
2.1	Power Profile v2 フレームワーク (Power Profile v2 Framework) .....	11
2.1.1	PPv2 フレームワークの実行プロファイル (PPv2 Framework Run Profile) .....	12
2.1.2	PPv2 フレームワークの低消費電力プロファイル (PPv2 Framework Low Power Profile) .....	13
2.2	PPv2 低消費電力モードの動作に関する注意 (PPv2 Low Power Modes Operational Notes) .....	13
2.3	PPv2 モジュールの制限 (PPv2 Module Limitations) .....	14
3.	Power Profile v2 モジュールの API の概要 (Power Profile v2 Module APIs Overview) .....	15
3.1	PPv2 フレームワーク API の関数 (PPv2 Framework API Function) .....	15
4.	アプリケーションに PPv2 フレームワークを包含する (Including the PPv2 Framework in an Application).....	18
4.1	ThreadX を使用して PPv2 フレームワークを包含する (Including the PPv2 Framework using ThreadX) .....	18
4.1.1	PPv2 の実行プロファイルモジュールを 1 つのスレッドに追加 (Adding PPv2 Run Profile Module into a Thread).....	18
4.1.2	PPv2 の低消費電力プロファイルモジュールをスレッドに追加 (Adding the PPv2 Low Power Profile Module into a Thread).....	21
4.2	ThreadX を使用せずに PPv2 フレームワークを包含する (Including PPv2 Framework without using ThreadX).....	21
4.2.1	PPv2 の実行プロファイルモジュールを HAL/共通スタックに追加 (Adding PPv2 Run Profile Module into the HAL/Common Stacks).....	22
4.2.2	PPv2 の低消費電力プロファイルモジュールを HAL/共通スタックに追加 (Adding PPv2 Low Power Module into the HAL/Common Stacks).....	22
5.	PPv2 フレームワークモジュールの設定 (Configuring PPv2 Framework Modules) .....	22
5.1	PPv2 フレームワークの実行プロファイルの設定 (Configuration of PPv2 Framework Run Profile)....	23
5.1.1	PPv2 フレームワークの実行プロファイルの I/O 設定 (I/O Configuration of PPv2	

---

Framework Run Profile).....	23
5.1.2 PPv2 フレームワークの実行プロファイルの CGC 設定 (CGC Configuration of PPv2 Framework Run Profile).....	24
5.2 PPv2 フレームワークの低消費電力プロファイルの設定 (Configuration of PPv2 Framework Low Power Profile) .....	27
5.2.1 LPM スリープモードの設定 (Configuration of the LPM Sleep Mode) .....	28
5.2.2 LPM スタンバイモードの設定 (Configuration of the LPM Standby Mode).....	29
5.2.3 LPM スヌーズモードの設定 (Configuration of the LPM Snooze Mode) .....	31
5.2.4 LPM ディープソフトウェアスタンバイモードの設定 (Configuration of the LPM Deep Software Standby Mode) .....	31
5.3 PPv2 フレームワークの共通モジュールの設定 (Configuration of the LPM Snooze Mode).....	34
6. PPv2 フレームワークのサンプルアプリケーション(PPv2 Framework Application Example)..	35
6.1 プロジェクトの概要 (Project Overview) .....	36
6.2 プロジェクトのソフトウェアアーキテクチャ (Project Software Architecture).....	38
6.3 プロジェクトの設定 .....	39
6.3.1 Power_Ctl_Mode_Thread (電力制御モードスレッド) の設定 (Configuration of the power_Ctl_Mode_Thread).....	39
6.3.2 LPM_Mode_Thread (LPM モードスレッド) の設定 .....	41
7. PPv2 フレームワークモジュールのサンプルアプリケーションの実行 (Running the PPv2 Framework Module Application Example).....	42
7.1 ボードの電源投入 (Powering up the Board) .....	42
7.2 デモの確認.....	43
8. 対象アプリケーションに合わせた PPv2 フレームワークモジュールのカスタマイズ.....	44
9. 参考資料.....	45

## 1. Synergy Power Profiles の概要 (Synergy Power Profiles Overview)

### 1.1 Power Profile v2 フレームワーク (Power Profile v2 Framework Overview)

Power Profiles v2 (PPv2) フレームワークは、Synergy Software Package (SSP) のコンポーネントであり、SSP ユーザが Synergy MCU グループの高度な省電力機能を設定して活用するのに役立ちます。PPv2 のモジュールを、ThreadX RTOS 対応のマルチスレッド内でインスタンス化し、SSP configurator を使用して想定する電力制御モードと LPM モードに設定した後、開発中のアプリケーションに挿入することができます。ただし、PPv2 モジュールを正しく設定するには、ユーザは Synergy MCU の各種の省電力機能について正しく理解する必要があります。

#### 1.1.1 PPv2 フレームワークの機能 (PPv2 Framework Features)

PPv2 がサポートしている機能のいくつかを以下に示します。

- カスタマイズ可能なクロックドメイン (clock domain) を使用して、MCU のさまざまな電力制御モードを設定する複数のオプション
- さまざまな IO ポートまたはピン設定 (pin configuration) を使用して、MCU のさまざまな低消費電力モード (LPM) を設定する複数のオプション
- スレッド動作 (threaded) と非スレッド動作 (non-threaded) の両方をサポート

### 1.2 Synergy MCU Power Profile の概要 (Synergy MCU Power Profile Overview)

Synergy MCU グループは、電力制御モードと低消費電力モード (Low Power Mode、LPM) の 2 種類の制御レジスタ (control register) があり、さまざまな消費電力や性能の要求に応えることができます。この章は、これらのモードの基本的な概念と使用方法を説明します。制御レジスタの設定に関する詳細については、それぞれの MCU 製品固有の『Synergy MCU グループユーザズマニュアル』 (Synergy Microcontroller Group User's Manual)、また API の詳細については『Synergy SSP ユーザズマニュアル』 (SSP User's Manual) を参照してください。

#### 1.2.1 Synergy MCU の電力制御モード (Synergy MCU Power Control Modes)

ノーマル (Normal) 動作モードも、さまざまなクロックソース (clock source) と動作周波数による適切な動作電力制御モードを選択することで、消費電力を低減できます。この機能は、複数の電力と性能の設定ポイントを備えており、その選択により柔軟な動作とデバイスの最適化を実現します。システムクロックは、外部から供給されるクロック、またはメインクロック発振器やサブクロック発振器のようなオンチップオシレータからも選択できます。さらに、クロック生成回路 (Clock Generation Circuit、CGC) レジスタでさまざまな分周比を設定すると、システムクロック (ICLK)、周辺回路モジュールクロック (PCLKB、PCLKD)、外部バスクロック (BCLK)、その他のクロックを調整できます。Synergy MCU S3 シリーズに対応するクロックソースのブロック図を以下に示します。

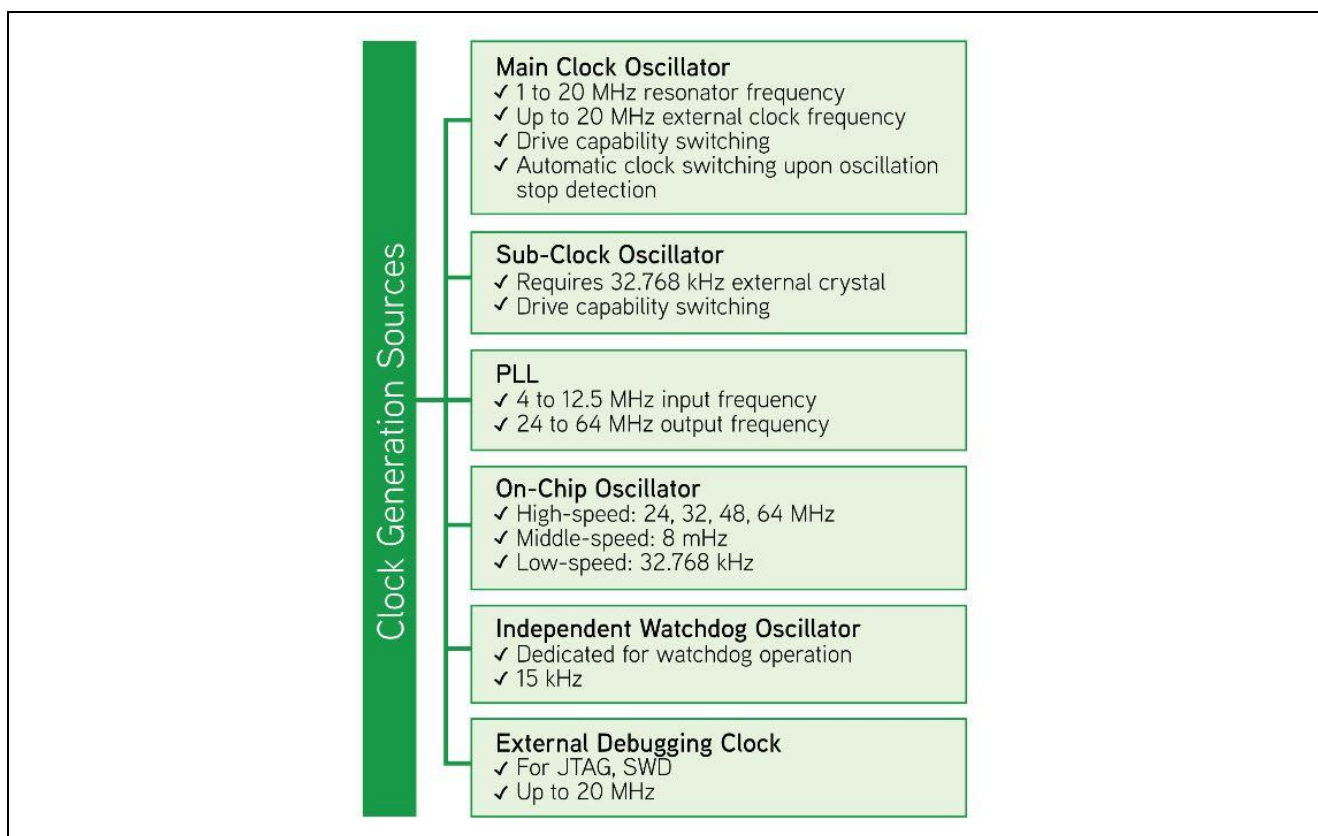


図 1 Synergy MCU S3 シリーズに対応するクロック生成ソース

Synergy MCU グループは、High-speed モード、Middle-speed モード、Low-voltage モード、Low-speed モード、Subosc-speed モードなど複数の電力制御モードを持ち、各モードは異なるクロック生成ソース (clock generation mode) と周波数範囲を使用します。それぞれの消費電力レベルは、記載したモードの順序が後になるほど低くなります。ただし、電力制御モードの選択は、設計のトレードオフであり、処理時間、静的および動的な消費電力、漏れ電流 (leakage current)、信頼性などの要因を考慮する必要があります。

現在の各 Synergy MCU グループの電力制御モードを、次の表に示します。

表 1 以下の Synergy MCU で利用できる電力制御モード

Synergy MCU グループ	電力制御モード				
	High-speed モード	Middle-speed モード	Low-voltage モード	Low-speed モード	Subosc-speed モード
S124	○	○	○	○	○
S128	○	○	○	○	○
S3A3	○	○	○	○	○
S3A6	○	○	○	○	○
S3A7	○	○	○	○	○
S5D5	○			○	○
S5D9	○			○	○
S7G2	○			○	○

注: それぞれの Synergy MCU で、電力制御モードの定義が異なっていることがあります。各電力モードの動作周波数と電圧は、それぞれの MCU 製品固有の Synergy MCU グループユーザーズマニュアルで規定されています。例えば、S3A7 MCU は『Synergy S3A7 ユーザーズマニュアル：マイクロコントローラ』に記載されている 5 つの既定の電力制御モードを使用し、それぞれの消費電力レベルは次の図に示すとおりです。

Mode	Oscillator							Power Consumption high ↓ low
	PLL*1	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator	IWDT- dedicated on-chip oscillator	
High-speed	Available	Available	Available	Available	Available	Available	Available	↓
Middle-speed	Available	Available	Available	Available	Available	Available	Available	
Low-voltage	N/A	Available	Available	Available	Available	Available	Available	
Low-speed	N/A	Available	Available	Available	Available	Available	Available	
Subosc-speed	N/A	N/A	N/A	Available	N/A	Available	Available	

図 2 S3A7 MCU の電力制御モードと消費電力の関係

各電力制御モードは、それぞれの Synergy MCU のユーザーズマニュアルで定義されています。

### 1.2.2 Synergy MCU の低消費電力モード (LPM) (Synergy MCU Low Power Modes)

消費電力をさらに低減できるように、各 Synergy MCU は低消費電力モード (LPM) をユーザに提供しており、単一の周辺回路の動作を有効にしつつ、CPU と他の周辺回路のクロックのゲーティング (gated) ゲーティングまたはパワーダウンします。LPM モードには 4 種類あります。スリープ (Sleep) モード、ソフトウェアスタンバイ (Software Standby) モード、ディープソフトウェアスタンバイ (Deep Software Standby) モード、スヌーズ (Snooze) モードです。例えばスリープモードは、コアへのクロック供給をゲーティングゲーティングしつつ、周辺回路を使用可能にすることができます。ソフトウェアスタンバイモードでは、コアと大半の周辺回路へのクロック供給はゲーティングゲーティングされますが、データは保持されます。Renesas Synergy™ S7 シリーズと S5 シリーズの MCU ではディープソフトウェアスタンバイモードも利用でき、この場合はコアと大半の周辺回路が電源オフになります。これらのモードが Synergy MCU の消費電力とスループットに及ぼす効果を、以下に示します。

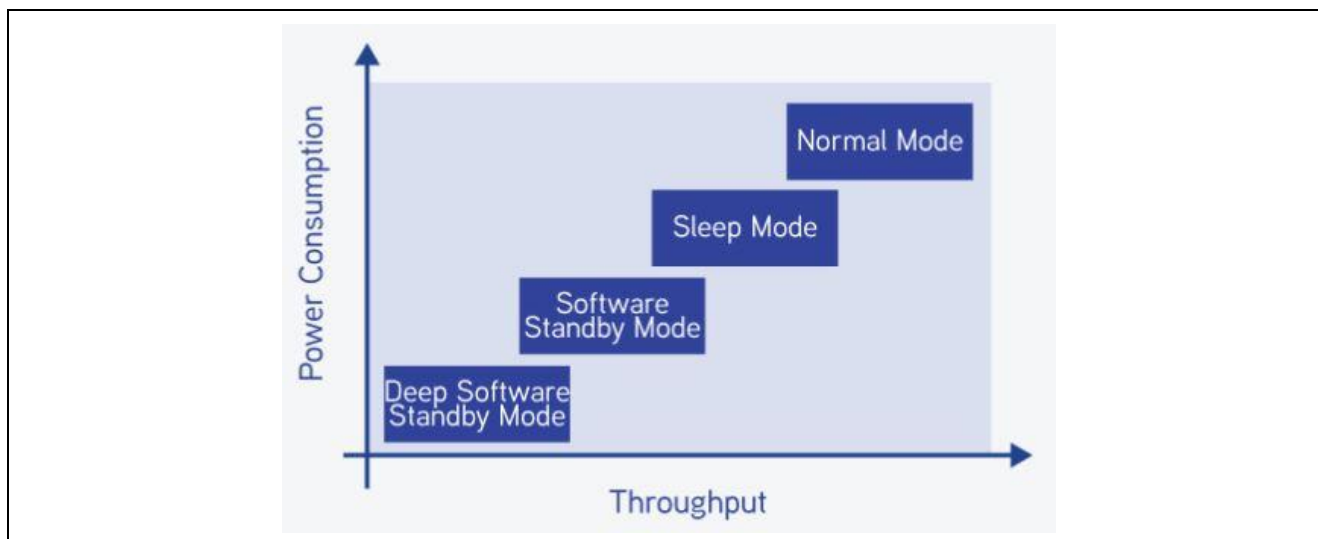


図 3 LPM モードの消費電力とスループット

表 2 Synergy MCU がサポートしている低消費電力モード

Synergy MCU グループ	低消費電力モード			
	スリープ モード	ソフトウェア スタンバイモード	ディープソフトウェア スタンバイモード	スヌーズモード
S124	○	○		○
S128	○	○		○
S3A3	○	○		○
S3A6	○	○		○
S3A7	○	○		○
S5D5	○	○	○	○
S5D9	○	○	○	○
S7G2	○	○	○	○

### 1.2.2.1 スリープモード (Sleep Mode)

スリープモードでは、CPU は動作を停止しますが、その内部レジスタの内容は保持されます。他の周辺回路の機能は停止しません。任意の割り込み、RES ピンによるリセット、パワーオンリセット、電圧モニタリセット、SRAM パリティエラーリセット、または IWDT か WDT のアンダーフローに起因するリセットが発生したときに、CPU をウェイクアップ (wake up) させることができます。

### 1.2.2.2 ソフトウェアスタンバイモード (SBY, Software Standby Mode)

SBY では、CPU、大半のオンチップ周辺回路の機能、および発振器は停止します。ただし、CPU の内部レジスタの内容と SRAM のデータ、オンチップ周辺回路機能の状態、I/O ポートの状態は保持されます。ソフトウェアスタンバイモードを使用すると、大半の発振器が停止するため消費電力を大幅に低減できます。ソフトウェアスタンバイモードをキャンセルできるのは、ウェイクアップ割り込み有効化レジスタ (Wake-UP Interrupt Enable Register、WUPEN) で規定されている割り込みのみです。



### 1.2.2.3 ディープソフトウェアスタンバイモード (DSBY, Deep-Software Standby Mode)

DSBY では、発振器、SRAM、フラッシュなどより多くのコンポーネントを停止させるため、SBY に比べていっそうの消費電力低減を実現できます。このモードでは、CPU の内部レジスタの内容と SRAM のデータ、オンチップ周辺回路の機能の状態、I/O ポートを保持することが可能です。

### 1.2.2.4 スヌーズ (Snooze, SNZ)

スヌーズ機能は、消費電流を大幅な低減して柔軟な動作を実現します。スヌーズ機能はソフトウェアスタンバイモードの拡張版であり、CPU をウェイクアップせずにいくつかの周辺回路モジュールを動作させることができます。いくつかの割り込みソース (interrupt source) を使用してソフトウェアスタンバイモードからスヌーズモードに移行すること、またソフトウェアスタンバイモードで使用可能な割り込みを使用してスヌーズモードからウェイクアップすることができます。

次の表に LPM モードの概要を示します。詳細な設定は、それぞれの MCU 製品の『Synergy MCU グループ ユーザーズマニュアル』(Synergy Microcontroller Group User's Manual) に掲載されています。

表 3 低消費電力モード (LPM) とそれぞれの基本的な設定

動作モード	コア	フラッシュ	SRAM	RTC、AGT、Vbatt、LVD	その他の周辺回路	IO ピン	スヌーズ
通常	動作	選択可能	選択可能	選択可能	選択可能	選択可能	N/A
スリープ	クロックをゲーティング	選択可能	選択可能	選択可能	選択可能	選択可能	N/A
ソフトウェアスタンバイ	クロックをゲーティング	データを保持	データを保持	選択可能	クロックをゲーティング	状態を保持	使用可能
ディープソフトウェアスタンバイ	電源オフ	電源オフ	部分的に電源オフ	選択可能	電源オフ	状態を保持	N/A

注: 「選択可能」とは、制御レジスタを使用して、動作または無効化を選択できることを意味します。一部のモジュールは、モジュールストップビットを使用して制御することもできます。

### 1.2.2.5 低消費電力モード (LPM) 相互間の遷移 (Transitions between the Low Power Modes)

通常モードと LPM モードの間での遷移は、次の図に示すように抽象化できます。

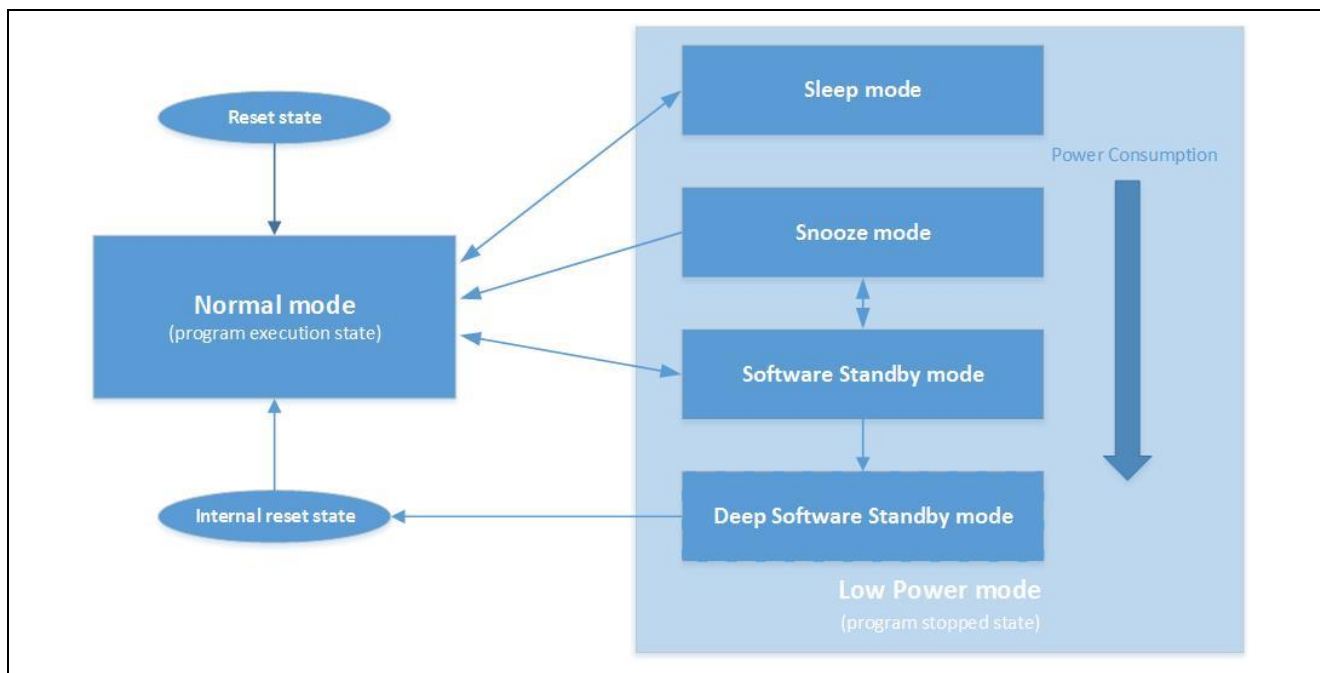


図 4 通常モードと 4 つの LPM モード間での遷移

注: この図は使用可能な LPM モードのみを示していますが、各モードの遷移条件を明示していません。Synergy MCU グループごとに LPM モードの数が異なり、遷移条件も異なります。また、実際の消費電力も LPM モードの設定に依存します。製品固有の Synergy MCU グループユーザーズマニュアルを参照してください。

## 2. Power Profile v2 フレームワークの動作概要 (Power Profile v2 Framework Operational Overview)

PPv2 フレームワークは LPM v2 ドライバ、CGC ドライバ、IO ポートドライバを使用するときに、Synergy MCU の低消費電力プロファイルをサポートするための汎用 API を提供しています。この API は、MCU の消費電力全般に対する高度な制御インタフェースであり、ThreadX RTOS の使用にかかわらず、アプリケーションでこの API を活用できます。API は内部で、LPM v2、IOPORT、SSP の CGC ドライバに依存しており、MCU の電力モードを制御するための使いやすいインタフェースを提供します。

### 2.1 Power Profile v2 フレームワーク(Power Profile v2 Framework)

PPv2 フレームワークは、MCU の消費電力を制御するために、**実行プロファイル (Run Profile)** および**低消費電力プロファイル (Low Power Profile)** という 2 つのメインプロファイルを提供しています。モジュールの構造を、次の図に示します。

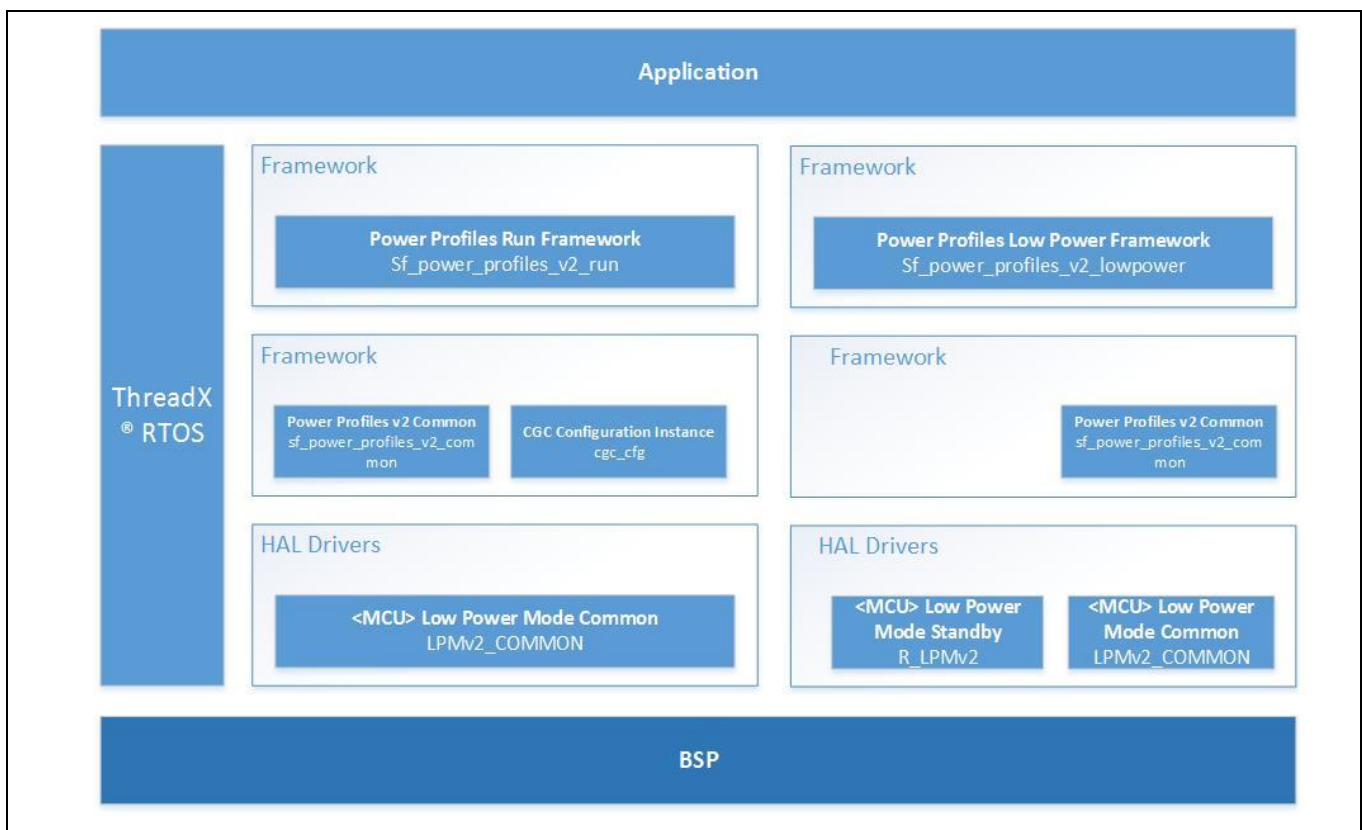


図 5 PPv2 フレームワークのスタック

### 2.1.1 PPv2 フレームワークの実行プロファイル (PPv2 Framework Run Profile)

実行プロファイル (**Run Profile**) は CGC クロック設定 (CGC Clocks configuration) と IO ポートピン設定を使用して、通常の動作モードで使用する MCU のシステムクロックと IO ポートピンを設定します。**RunApply ()** API を使用してこのプロファイルの機能を実装し、指定した順序で以下のタスクを実行します。

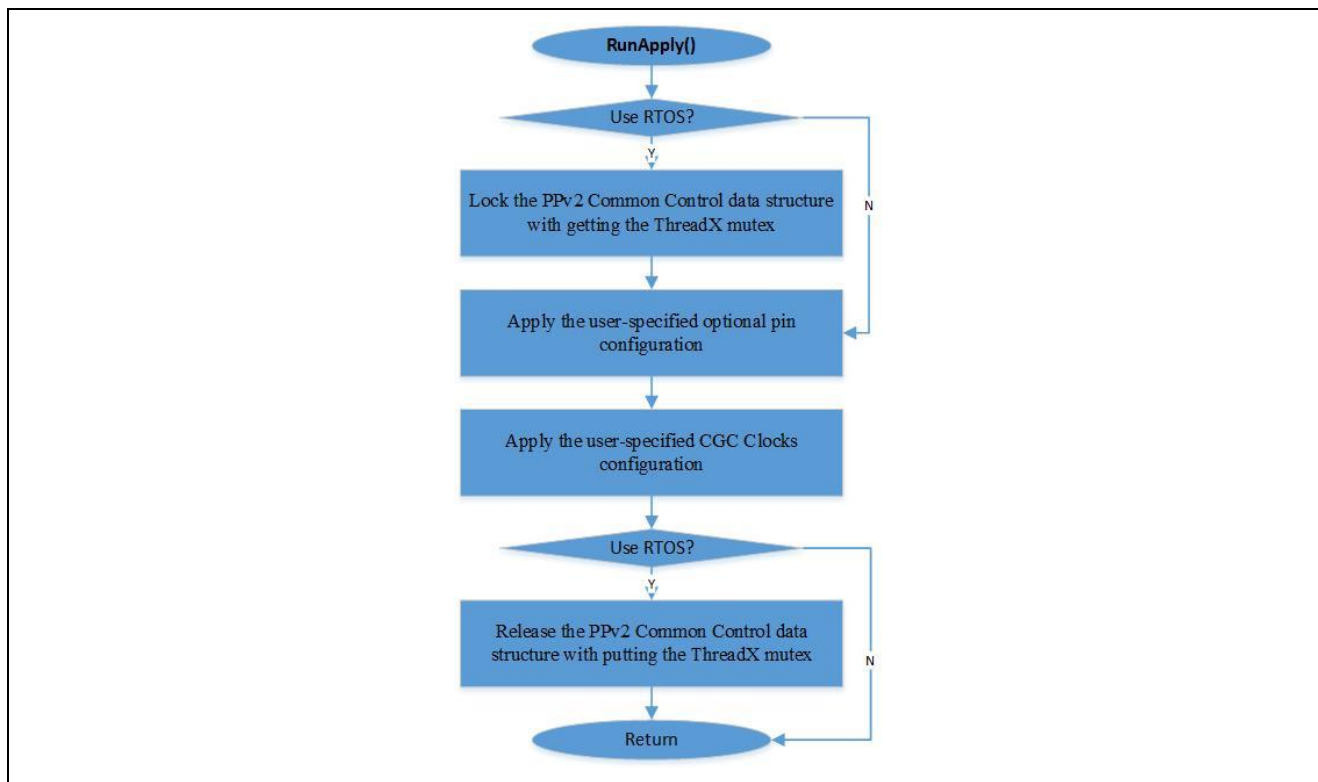


図 6 PPv2 の RunApply () プロセス

### 2.1.2 PPv2 フレームワークの低消費電力プロファイル (PPv2 Framework Low Power Profile)

低消費電力プロファイル (Low Power Profile) では、低消費電力モード (low power mode) への移行前と低消費電力モードからの復帰後の低消費電力モードと IO ポートの状態を設定するために、LPM v2 設定および pre-LPM と post-LPM の IO ポート設定を使用します。使用可能な低消費電力モードの詳細については、『SSP ユーザーズマニュアル』 (SSP User's Manual) と、それぞれの MCU 製品の『Synergy MCU グループユーザーズマニュアル』 (Synergy Microcontroller Group User's Manual) を参照してください。

内部関数 `LowPowerApply()` API は、以下のタスクを順に実行します。

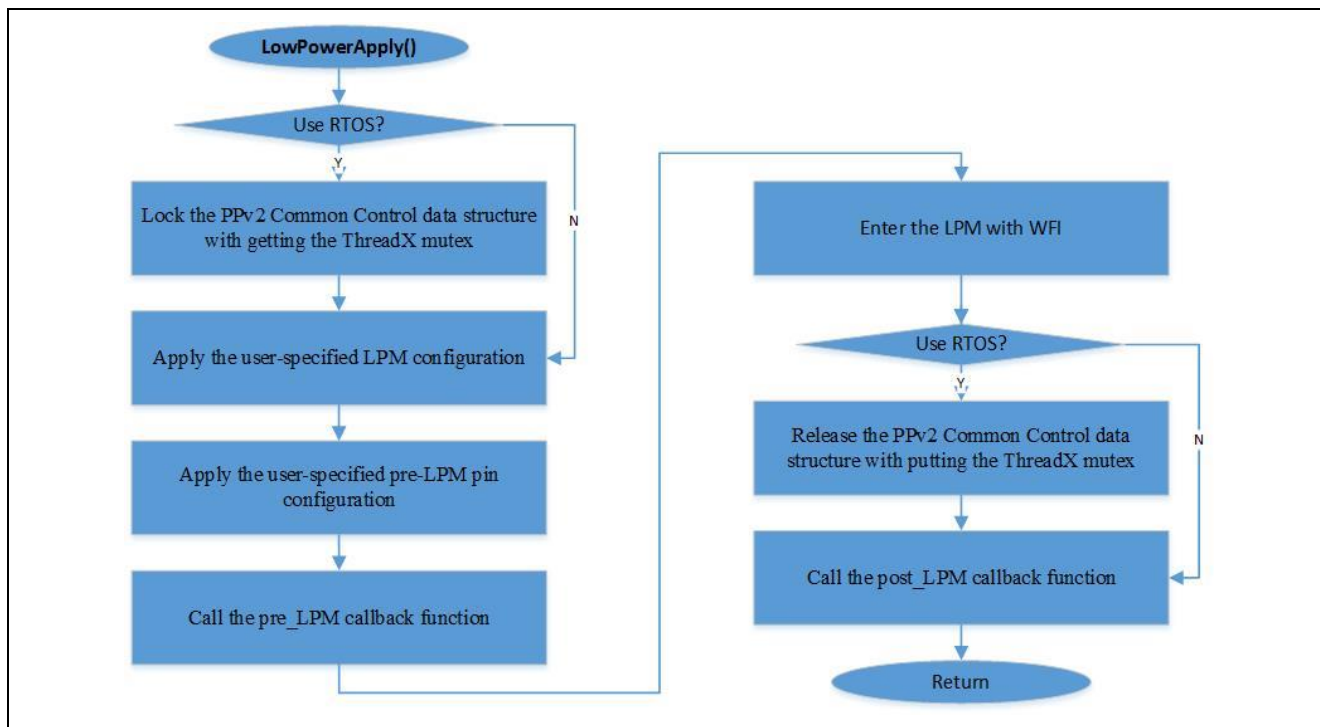


図 7 The PPv2 の LowPowerApply() プロセス

## 2.2 PPv2 低消費電力モードの動作に関する注意 (PPv2 Low Power Modes Operational Notes)

以下の低消費電力モードに関連する注意は、PPv2 フレームワークとドライバの現在のリリースを観察した結果に基づくものであり、今後、ユーザからのフィードバックや Power Profile パッケージの新しいリリースに伴って更新されることがあります。

- **Power Profile v1 と PPv2**

Power Profiles v1 と PPv2 フレームワークには互換性がありません。同じプロジェクトで、Power Profile v1 と PPv2 両方のフレームワークを使用しないでください。すべての新規プロジェクトで、アプリケーションが PPv2 フレームワークを使用することを推奨します。

- **デフォルトで、PPv2 アプリケーションに対して LPM v2 ドライバインスタンスが追加される**

MCU 固有の LPM v2 ドライバは、2 つの構造体 (structure) `lpmv2_mcu_cfg_t` と `lpmv2_api_t` において LPM 動作 (operation) の有効化、無効化と設定を行う API と設定を定義します。その後、自動的に生成された `common_data.c` ファイルの中でこのドライバがインスタンス化されます。したがって、PPv2 フレームワークインスタンス (PPv2 framework instance) は、LPM v2 ドライバインスタンス (LPM v2 Driver instance) をベースとしています。

- **PPv2 の runApply() 関数を使用するための付加的な CGC ドライバインスタンス**

デフォルトの CGC ドライバは、すべての Synergy アプリケーションプロジェクトに包含され、`common_data.c` ファイル内でインスタンス化されますが、プロジェクトのコードサイズは増加しません。PPv2 の `runApply()` 関数の CGC クロック設定に対応する、CGC ドライバのもう 1 つのインスタンスが必須です。

- **別のピン設定に対応する I/O ポートドライバインスタンス**

PPv2 電源プロファイル (PPv2 power profile) では異なるピン設定テーブルを定義することが可能ですが、自動的に生成されるファイル `common_data.c` の中では、I/O ポートドライバインスタンス `ioport_instance_t` のみが、インスタンス化されます。

- **ThreadX を使用する場合の動作**

前述の `RunApply()` および `LowPowerApply()` の動作説明にあるように、PPv2 フレームワーク API は、ThreadX が実装される場合には、マルチスレッドアプリケーションに対応するミューテックス (mutex) など、ThreadX 固有 (intrinsic) のオブジェクトを使用します。

- **マルチスレッドアプリケーションに関する特別な検討事項**

ソフトウェアスタンバイ、ディープソフトウェアスタンバイ、またはスヌーズの LPM モードでは、Systick に対応するソースクロック (source clock) を無効として設定することが可能です。したがって、PPv2 の LPM モードを使用してマルチスレッド RTOS プロジェクトを使用する場合は、特別な検討事項が必要になります。

- **LPM モードでのデバッガの使用方法**

デバッガを接続した状態では、MCU はソフトウェアスタンバイモードやディープソフトウェアスタンバイに移行すること、またはこれらのモードを維持することができない可能性があります。デバッガを接続することにより MCU がソフトウェアスタンバイモードやディープソフトウェアスタンバイからウェイクアップ (wake up) する可能性があります。

## 2.3 PPv2 モジュールの制限 (PPv2 Module Limitations)

- プロジェクトが ThreadX を使用していない場合、Power Profiles V2 フレームの `open` 関数は、`main` より前に自動的に呼び出されることはありません。初期化は、`g_common_init()` を明示的に呼び出すか、API 関数 `open()` を明示的に呼び出す方法で、実行する必要があります。これは PPv2 の制限ではなく、RTOS を使用しない状態をサポートするフレームワークモジュールによる影響です。
- PPv2 フレームワークは、MCU の周辺回路の開始または停止を行いません。LPM モードで周辺回路を停止するか選択するためのプロパティビュー (property view) が使用できないからです。したがって、周辺回路はユーザが手動 (manually) で停止する必要があります。
- PPv2 フレームワークの現在のバージョンがサポートしているのは、以下の Synergy MCU のみです。S124、S128、S3A7、S5D9、および S7G2
- PPv2 フレームワークの現在のバージョンは、LPM スタンバイモード (LPM Standby mode) のうち、スヌーズモード (Snooze mode) から通常モード (Normal mode) への遷移をサポートしていません。

### 3. Power Profile v2 モジュールの API の概要 (Power Profile v2 Module APIs Overview)

この章では、フレームワークレイヤにおける PPv2 の API 関数について、また、それらの関数を使用する場合の動作の概要を示します。

#### 3.1 PPv2 フレームワーク API の関数 (PPv2 Framework API Function)

フレームワークのプルダウンメニューから、PPv2 プロファイルのいずれかが追加済みで、ユーザが使用中のインスタンスに対応するプロファイルの設定データ構造体が `sf_power_profile_v2_api.h` 内で宣言されていることを前提にします。この場合、この中に、`p_ctrl`、`p_cfg`、`p_api` という 3 つの要素 (part) があります。

```
/* This structure encompasses everything that is needed to use an instance of this interface. (この構造体は、このインタフェースのインスタンスを使用するために必要とされるすべての要素を網羅しています。)*/  
typedef struct st_sf_power_profiles_v2_instance  
{  
    sf_power_profiles_v2_ctrl_t * p_ctrl;    ///< Pointer to the control structure for this instance (このインスタンスに対応する制御構造体へのポインタ)  
    sf_power_profiles_v2_cfg_t const * p_cfg;    ///< Pointer to the configuration structure for this instance (このインスタンスに対応する設定構造体へのポインタ)  
    sf_power_profiles_v2_api_t const * p_api;    ///< Pointer to the API structure for this instance (このインスタンスに対応する API 構造体へのポインタ)  
} sf_power_profiles_v2_instance_t;
```

制御構造体 (control structure) `sf_power_profiles_v2_ctrl_t` は、以下のように定義されています。

```
typedef struct st_sf_power_profiles_v2_ctrl
{
    uint32_t      open;          ///< Used by driver to check if pointer to control block is
    valid (制御ブロックへのポインタが有効かどうかを確認する目的でドライバが使用します)
    #if (1 == BSP_CFG_RTOS)
        TX_MUTEX      mutex;          ///< Mutex used to protect access to lower level driver
    hardware registers (下位レベルドライバからハードウェアレジスタへのアクセスを保護する目的で使用するミュー
    テックス)
    #endif /* (1 == BSP_CFG_RTOS) */
} sf_power_profiles_v2_ctrl_t;
```

設定構造体 (configuration structure) `sf_power_profiles_v2_cfg_t` は、以下のように定義されています。

```
typedef struct st_sf_power_profiles_v2_cfg
{
    /** Pointer to additional settings (not currently in use) (付加的な設定へのポインタ
    (現在は使用されていません)) */
    void          const * p_extend;
} sf_power_profiles_v2_cfg_t;
```

実行プロファイル (Run Profile) の設定は、次の構造体 (structure) を使用しています。

```
typedef struct st_sf_power_profiles_v2_run_cfg
{
    /** Pointer to IOPORT settings (IOPORT 設定へのポインタ) */
    ioport_cfg_t      const * p_ioport_pin_tbl;
    /** Pointer to a CGC configuration (CGC 設定へのポインタ) */
    cgc_clocks_cfg_t  const * p_clock_cfg;
    /** Pointer to additional settings (付加的な設定へのポインタ) */
    void          const * p_extend;
} sf_power_profiles_v2_run_cfg_t;
```

低消費電力プロファイル (Low Power Profiles) の設定は、次の構造体 (structure) を使用しています。

```
typedef struct st_sf_power_profiles_v2_low_power_cfg
{
    /** Pointer to IOPORT settings to apply after exiting the low power mode (低消費電力モードを
    終了した後に適用する IOPORT 設定へのポインタ) */
    ioport_cfg_t      const * p_ioport_pin_tbl_exit;
    /** Pointer to IOPORT settings to apply before entering low power mode (低消費電力モードに移行
    する前に適用する IOPORT 設定へのポインタ) */
    ioport_cfg_t      const * p_ioport_pin_tbl_enter;
    /** Pointer to an LPMv2 instance (LPMv2 インスタンスへのポインタ) */
    lpmv2_instance_t  const * p_lower_lvl_lpm;
    /** Callback function (コールバック関数) */
    void          (* p_callback)(sf_power_profiles_v2_callback_args_t
    * p_args);
    /** Placeholder for user data (ユーザデータに対応するプレースホルダ) */
    void          * p_context;
    /** Pointer to additional settings (付加的な設定へのポインタ) */
    void          const * p_extend;
} sf_power_profiles_v2_low_power_cfg_t;
```



PPv2 フレームワーク API である (sf\_power\_profiles\_v2\_api\_t) は、以下のように要約されます。

表 4 PPv2 フレームワーク API の要約

関数名	API 呼び出しの例と説明
.open	<pre>g_sf_power_profiles_v2_common.p_api-&gt; open(g_sf_power_profiles_v2_common.p_ctrl, g_sf_power_profiles_v2_common.p_cfg);</pre> <p>初期化された PPv2 フレームワーク</p> <p>[in,out] p_ctrl Pointer to a structure allocated by user. Elements initialized here.(See definition of sf_power_profiles_v2_ctrl_t) (ユーザが割り当てた構造体へのポインタ。ここで要素を初期化します(記載の定義を参照してください))</p> <p>[in] p_cfg Pointer to configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_cfg_t) (構造体へのポインタ。構造体の要素はユーザが設定する必要があります(記載の定義を参照してください))</p>
.runApply	<pre>g_sf_power_profiles_v2_common.p_api-&gt; runApply(g_sf_power_profiles_v2_common.p_ctrl, &amp;p_cfg);</pre> <p>実行プロファイル (Run profile) を適用します。</p> <p>[in] p_ctrl Pointer to control block set in the open() API above. [in] p_cfg Pointer to the run configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_run_cfg_t) (前述の open() API で設定した制御ブロックへのポインタ。実行構造体へのポインタ。構造体の要素はユーザが設定する必要があります(記載の定義を参照してください))</p>
.lowPowerApply	<pre>g_sf_power_profiles_v2_common.p_api-&gt; lowPowerApply(g_sf_power_profiles_v2_common.p_ctrl,&amp;p_cfg);</pre> <p>低消費電力プロファイル (Low Power Profile) を適用します。</p> <p>[in] p_ctrl Pointer to control block set in the open() API above. [in] p_cfg Pointer to the low power configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_low_power_cfg_t) (前述の open() API で設定した制御ブロックへのポインタ。低消費電力構造体へのポインタ。構造体の要素はユーザが設定する必要があります(記載の定義を参照してください))</p>
.versionGet	<pre>g_sf_power_profiles_v2_common.p_api-&gt;versionGet(&amp;version);</pre> <p>バージョンを取得し、その値を、バージョンへのポインタである p_version に書き込みます。</p> <p>[out] p_version Code and API version used. (使用しているコードと API のバージョン)</p>
.close	<pre>g_sf_power_profiles_v2_common.p_api-&gt; close(g_sf_power_profiles_v2_common.p_ctrl);</pre> <p>フレームワークを閉じます。</p> <p>[in] p_ctrl Pointer to control block set in the open() API above. (前述の open() API で設定した制御ブロックへのポインタ)</p>

注: 関数のデータ構造体、型定義 (typedef)、定義 (define)、API データ、API 構造体、関数の変数に関する詳細な動作説明と定義については、『SSP ユーザーズマニュアル』(SSP User's Manual) で、該当モジュールに関する API リファレンスを参照してください。

PPv2 フレームワーク API の戻り値 (return value) は、以下のように定義されています。

名前	説明
SSP_SUCCESS	関数が成功しました。
SSP_ERR_ASSERTION	アサーションエラーです。
SSP_ERR_IN_USE	フレームワークは既に初期化されています。
SSP_ERR_INVALID_HW_CONDITION	互換性のないシステムクロック設定です。
SSP_ERR_NOT_OPEN	デバイスが開いていません。
SSP_ERR_UNSUPPORTED	このモジュールはこの関数をサポートしていません。
SSP_ERR_INTERNAL	内部エラーです。

注： 下位レベルドライバは Common Error Codes (一般的なエラーコード) を返すことがあります。戻り値に関連するすべての定義の詳細については、『SSP ユーザーズマニュアル』で、該当モジュールに関する API リファレンスを参照してください。

#### 4. アプリケーションに PPv2 フレームワークを包含する (Including the PPv2 Framework in an Application)

PPv2 フレームワークのモジュールをアプリケーションに包含するために 2 つの方法があります。

- ThreadX を使用する
- ThreadX を使用しない

##### 4.1 ThreadX を使用して PPv2 フレームワークを包含する (Including the PPv2 Framework using ThreadX)

アプリケーション内で PPv2 フレームワークを使用する場合の代表的なステップは、以下のとおりです。

- ステップ 1: PPv2 の実行プロファイル (PPv2 Run Profile) モジュールを 1 つのスレッドに追加します。
- ステップ 2: 電力制御モード (power control mode) を設定し、ピン設定を変更します (実行プロファイル)。
- ステップ 3: PPv2 の低消費電力プロファイル (PPv2 Low Power Profile) モジュールを 1 つのスレッドに追加します。
- ステップ 4: LPM モードを設定して、遷移条件 (transision condition)、および LPM 移行前 (pre-LPM) モードと LPM 終了後 (post-LPM) モードの各ピン設定を変更します。

この章は、ステップ 2 と 3 に注目し、PPv2 フレームワークのプロファイルモジュールをスレッドに追加します。ステップ 2 と 4 のプロファイル設定については、5 章で説明します。

##### 4.1.1 PPv2 の実行プロファイルモジュールを 1 つのスレッドに追加 (Adding PPv2 Run Profile Module into a Thread)

以下のことを前提とします。まず、『e<sup>2</sup> studio ISDE ユーザーガイド』(e<sup>2</sup> studio ISDE User Guide)に従って、選択した MCU デバイスに対応した Synergy C プロジェクトが既に作成済みであること。およびその後、[Thread] タブで [New Thread] をクリックした後、プロパティビューで [Name] と [Symbol] の各エントリをクリックし、新しいスレッドに対応するわかりやすい名前とシンボルを入力する方法で、新しいスレッドも作成済みであること。

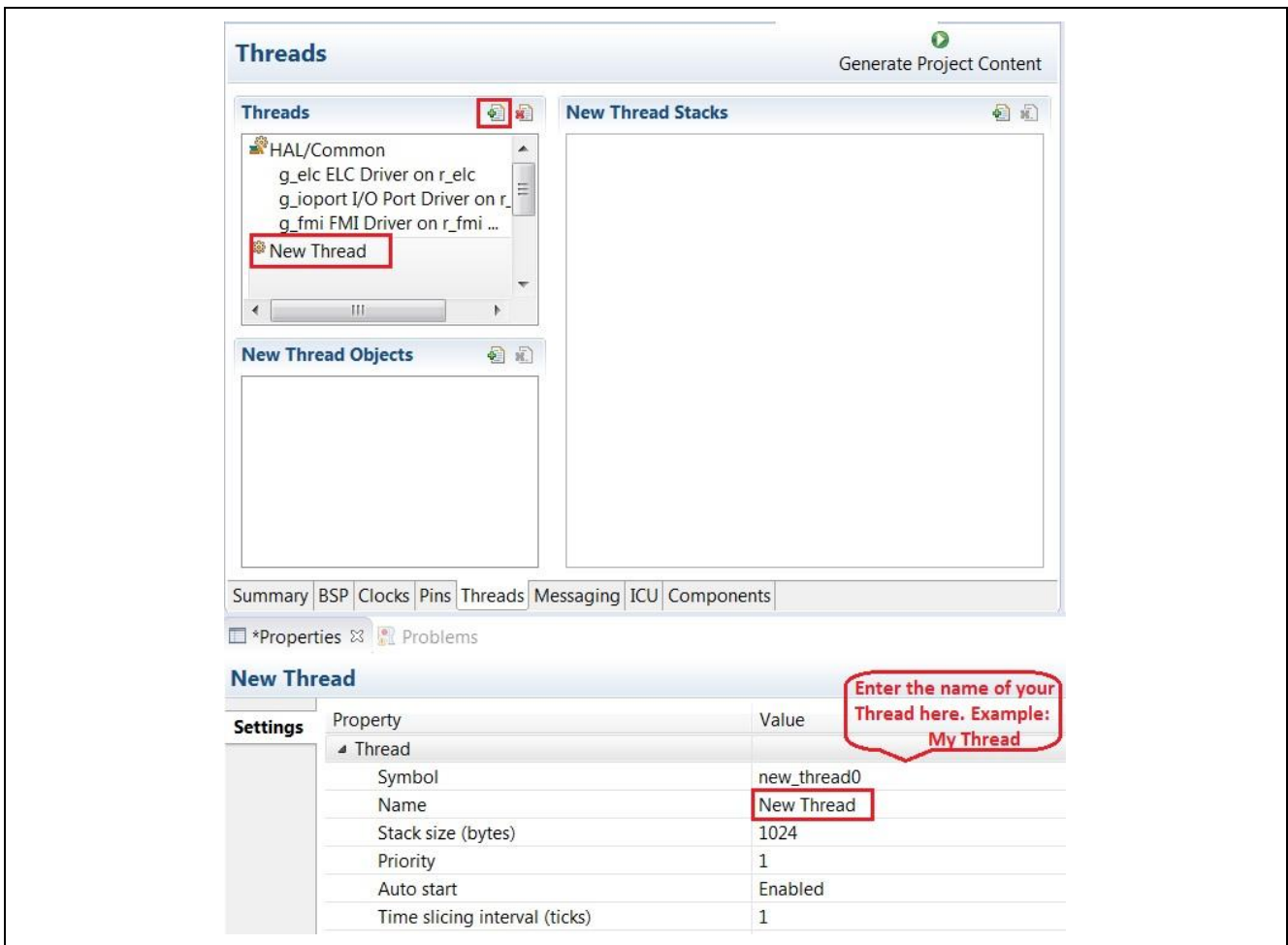


図 8 [Thread] タブで新しいスレッドを追加

選択したスレッドで、[New] をクリックし、プルダウンメニューから実行プロファイルモジュール (Run profile module) を選択します。

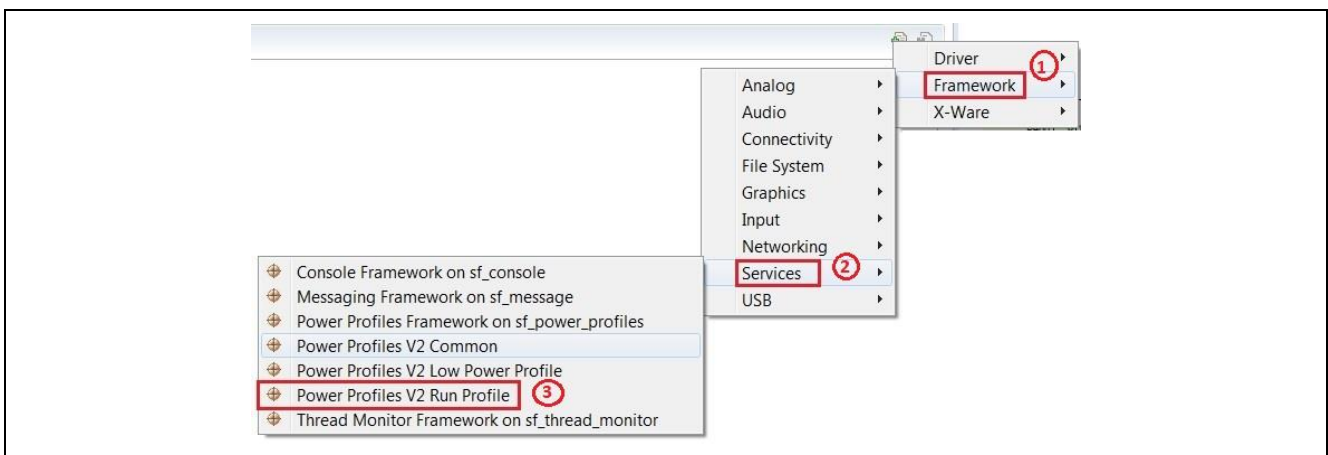


図 9 PPv2 の実行プロファイルモジュールを追加

次に、この実行プロファイルモジュールの名前を、意味のなす名前に変更します。例えば、`g_sf_power_profiles_v2_run_high_speed_mode` です。

注: プロジェクト内で、PPv2 モジュールの各インスタンスに対して一意の名前を割り当てる必要があります。

Property	Value
Module g_sf_power_profiles_v2_run_high_speed_mode	Power Profiles V2 Run Profile
Name	g_sf_power_profiles_v2_run_high_speed_mode
Pin configuration table	NULL

図 10 実行プロファイルモジュールの命名

図 11 のように、作成した電力制御モード (power control mode) が SSP configurator の中で表示されます。その動作周波数を定義するには、CGC 設定インスタンス (CGC Configuration Instance) モジュールのプロパティビューでクロックをオンまたはオフに切り替え、クロック分周器を変更した後、システムクロックを選択する方法が使用できます。その詳細結果は 5.1.2 章で説明されます。

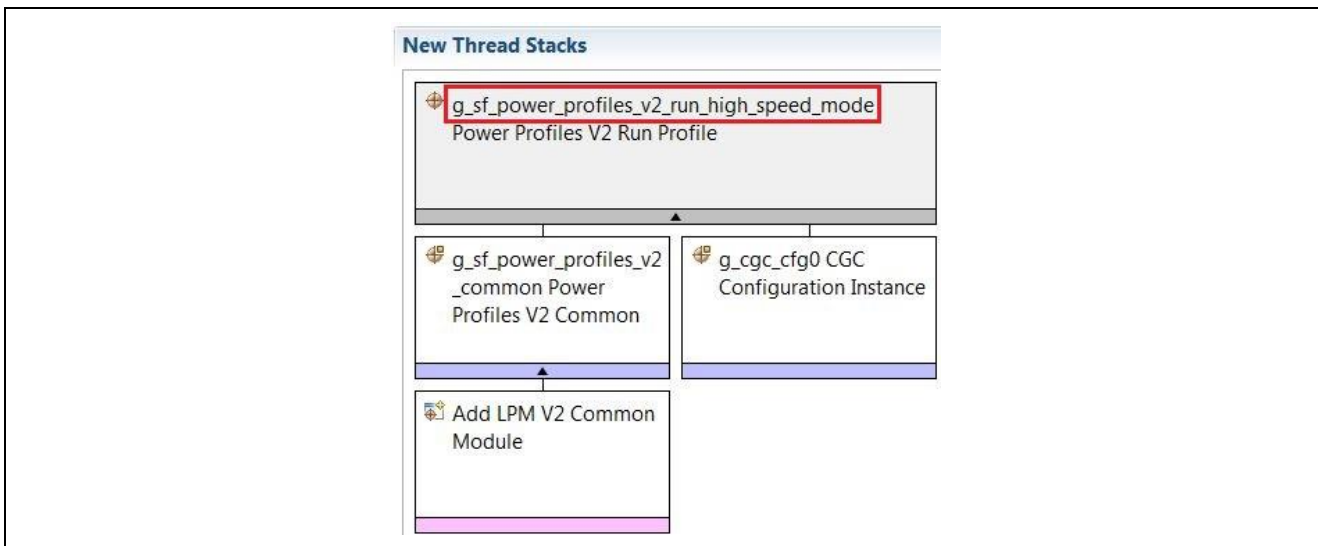


図 11 PPv2 の実行プロファイル内で定義した電力制御モード

図 12 に、[New] プルダウンメニューから LPM 共通 (LPM common) モジュールのいずれかを選択する状況を示します。このモジュールは、選択した Synergy MCU グループ (下図では S3A7) ごとに対応する PPv2 フレームワークの設定を保持しています。

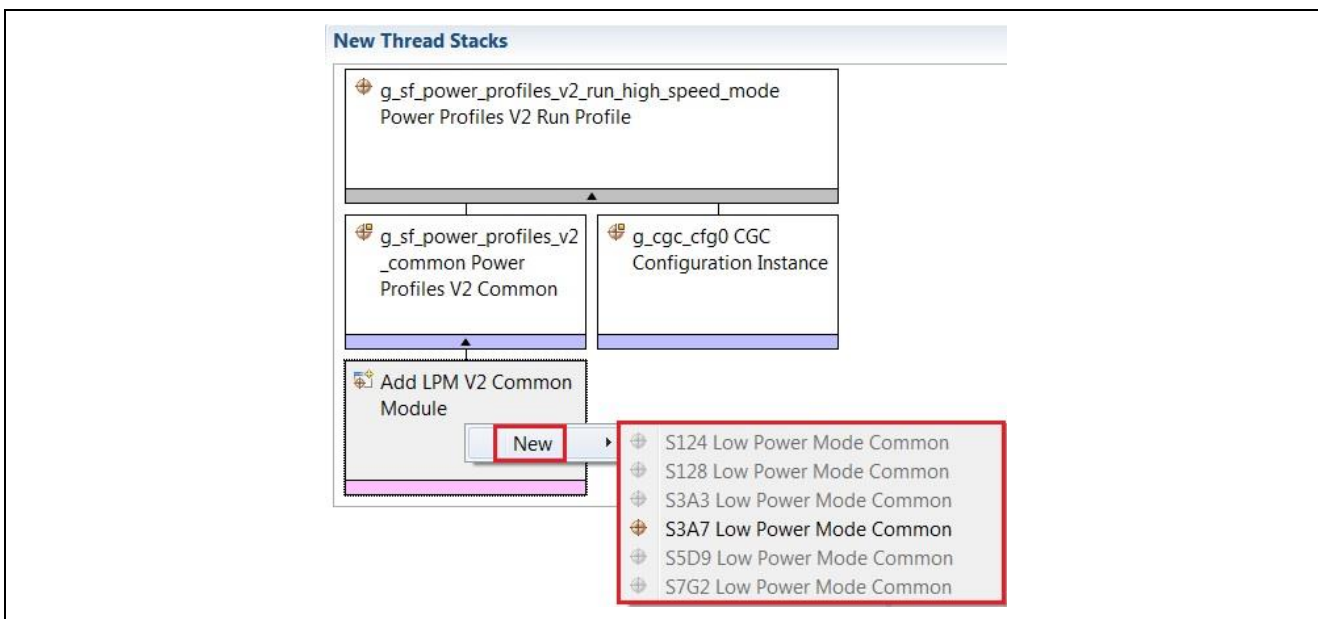


図 12 LPM v2 共通モジュールを追加

#### 4.1.2 PPv2 の低消費電力プロファイルモジュールをスレッドに追加 (Adding the PPv2 Low Power Profile Module into a Thread)

選択したスレッドで、[New] をクリックし、プルダウンメニューから低消費電力プロファイル (Low Power Profile) モジュールを選択します。

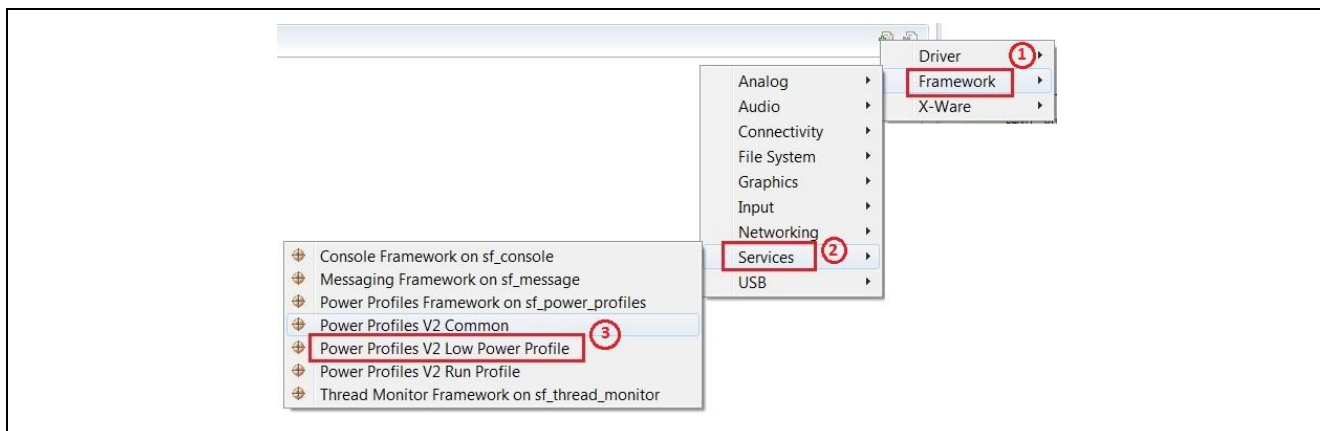


図 13 PPv2 の低消費電力プロファイルモジュールを追加

次に、SSP configurator によって LPM 低消費電力プロファイル (LPM Low Power Profile) モジュールが生成されます。選択した MCU で使用可能な LPM モードが、有効になります。

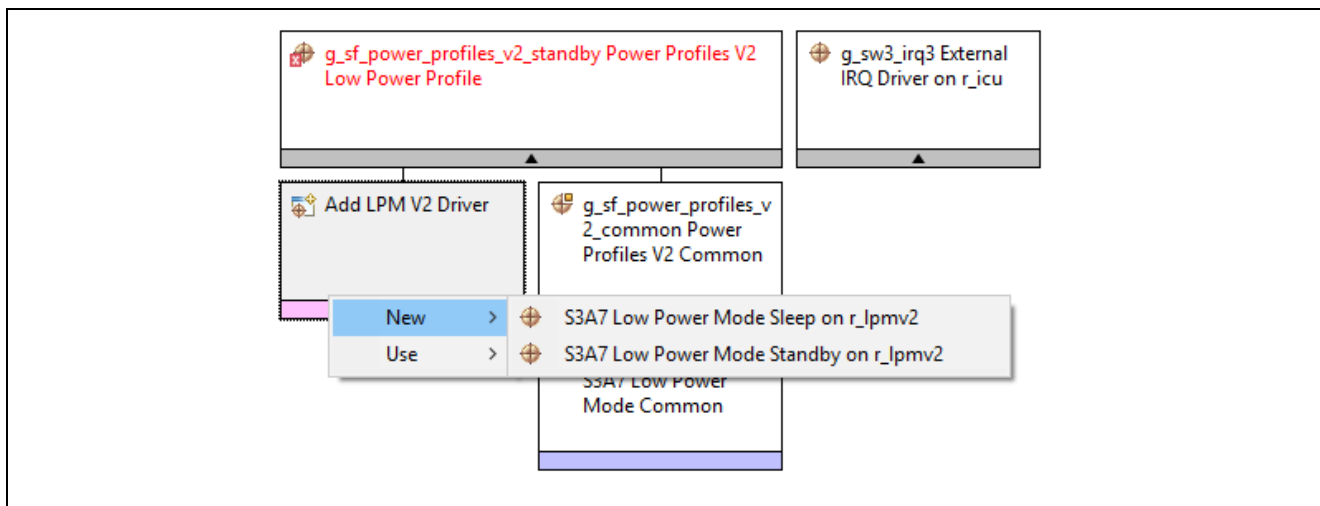


図 14 PPv2 の低消費電力プロファイルモジュール

開発中アプリケーションで、ユーザはスリープモード、スタンバイモード、またはディープスタンバイモードを表す PPv2 フレームワークモジュールを追加することができます。

#### 4.2 ThreadX を使用せずに PPv2 フレームワークを包含する (Including PPv2 Framework without using ThreadX)

ThreadX ベースのアプリケーションに PPv2 を包含するステップと同様に、PPv2 フレームワークのプロファイルモジュールも、スレッド化されていないプロジェクト(non-thread project)に追加することもできます。

- ステップ 1: PPv2 の実行プロファイルモジュールを(Run Profile module) HAL/共通スタック (HAL/Common Stacks) に追加します。
- ステップ 2: 電力制御モード(power control module)を設定し、ピン設定を変更します (実行プロファイル)。
- ステップ 3: PPv2 の低消費電力プロファイルモジュール(PPv2 Low Power Profile module)を HAL/共通スタック (HAL/Common Stacks) に追加します。

**ステップ 4:** LPM モードを設定して、遷移条件、および LPM 移行前 (pre-LPM) モードと LPM 終了後 (post-LPM) モードの各ピン設定を変更します。

ステップ 2 と 4 のプロファイル設定については、5 章で説明します。

#### 4.2.1 PPv2 の実行プロファイルモジュールを HAL/共通スタックに追加 (Adding PPv2 Run Profile Module into the HAL/Common Stacks)

ここでは、e2 studio ISDE ユーザーガイドに従って、MCU デバイス向けの Synergy C プロジェクト (C project) が既に生成され、[add] (追加) ボタンにより PPv2 フレームワーク実行プロファイルモジュール (PPv2 Framework Run Profile module) が追加されていることを前提に説明しています。

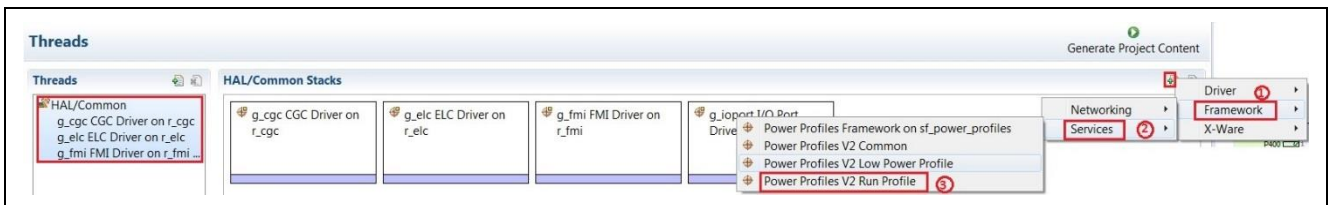


図 15 PPv2 の実行プロファイルモジュールを HAL/共通スタックに追加

プロパティビュー (property view) で、この実行プロファイルモジュールに対して固有の名前を割り当てます。例えば、`g_sf_power_profiles_v2_run_high_speed_mode` とします。次に、LPM 共通 (LPM common) モジュールの [New] をクリックして、Synergy MCU グループに対応する PPv2 フレームワーク設定 (PPv2 framework configuration) のいずれかを選択します。図に示す図 9～図 13 のような設定です。

#### 4.2.2 PPv2 の低消費電力プロファイルモジュールを HAL/共通スタックに追加 (Adding PPv2 Low Power Module into the HAL/Common Stacks)

4.1.2 章で PPv2 低消費電力プロファイルモジュールを 1 つのスレッドに追加した操作と同様に、[Add] ボタンをクリックした後に、プルダウンメニューから低消費電力プロファイル (Low Power Profile) モジュールを選択することで、複数の PPv2 LPM プロファイルモジュールを追加することができます。

### 5. PPv2 フレームワークモジュールの設定 (Configuring PPv2 Framework Modules)

ここまでの章で、PPv2 フレームワークモジュールをアプリケーションに包含する様々な方法を説明しました。次の手順は、PPv2 フレームワークプロファイルモジュールを設定する方法を示し、詳細な設定パラメータとそれらの推奨値を提供することです。これによりユーザは、プロジェクトアプリケーション内でこれらのモジュールを利用することができるようになります。

PPv2 フレームワークのプロファイルモジュールを設定する方法を説明する前に、新しいスレッドのプロパティについて説明します。このスレッドは、図 8 (4.2.2 章) に示すように、PPv2 フレームワークのプロファイルと組み合わせて使用することができます。

表 5 新しいスレッドに合わせた設定

ISDE のプロパティ	値	説明
Symbol	New_thread0 (デフォルト)	ユーザは別の名前を指定することもできます
Name	New Thread (デフォルト)	ユーザは別の名前を指定することもできます
Stack size (バイト数)	1024 (デフォルト)	アプリケーションに依存します
Priority	1 (デフォルト)	ユーザは個別のアプリケーションに基づいて、この priority (優先順位) を調整することができます
Auto start	Enabled (デフォルト)	ユーザはアプリケーションの実装に基づいて、この設定項目を調整することができます
Time slicing interval (ティック数)	1 (デフォルト)	ユーザは個別のアプリケーションに基づいて、この interval (間隔) を調整することができます

### 5.1 PPv2 フレームワークの実行プロファイルの設定 (Configuration of PPv2 Framework Run Profile)

ここでは、4章の操作により PPv2 フレームワークの実行プロファイルがプロジェクトに既に組み込まれ、そして後述のモジュール設定を使用することを前提に説明しています。

#### 5.1.1 PPv2 フレームワークの実行プロファイルの I/O 設定 (I/O Configuration of PPv2 Framework Run Profile)

図 16 に、以下の High-speed モードのような実行プロファイルと、その設定項目を示します

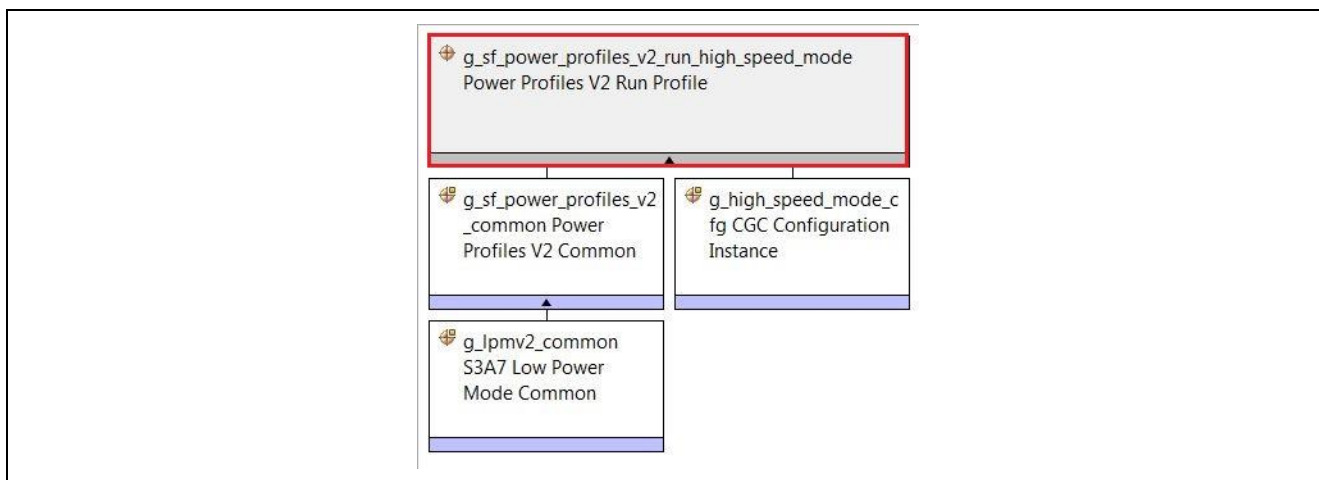


図 16 PPv2 の実行プロファイルモジュール

表 6 実行プロファイルに関する設定項目

ISDE のプロパティ	値	説明
Name	g_sf_power_profiles_v2_run_high_speed_mode	モジュール名
Pin configuration table	NULL (デフォルト)	ピン設定テーブル

1 つの電力モードに対応するピン設定テーブルを作成し、その後、実行プロファイルのプロパティ (Run profile property) にリンクさせることができます。それ以外の場合では、電力モードは、g\_bsp\_pin\_cfg によって与えられるデフォルトのピン割り当てを使用します。

カスタムのピン設定テーブルを作成する基本的なステップを以下に示します。

- S3A7-PK.pincfg のような特定のボードピン設定をコピーするには、設定をマウスの右ボタンでクリック (右クリック) し、メニューから [Copy] 操作を選択した後、同じプロジェクトに対して [Paste] を実行して貼り付けます。
- 新しいピン設定ファイルの名前を変更します。例えば、S3A7-DK2\_RUN.pincfg です。



図 17 [Copy] and [Paste] を使用した新しいピン設定ファイルの作成

- 図 18 のように、SSP Configurator の [Pins] タブで、プルダウンメニューから新しいピン設定を選択します。その後、生成しようとするピン設定の名前を指定します。

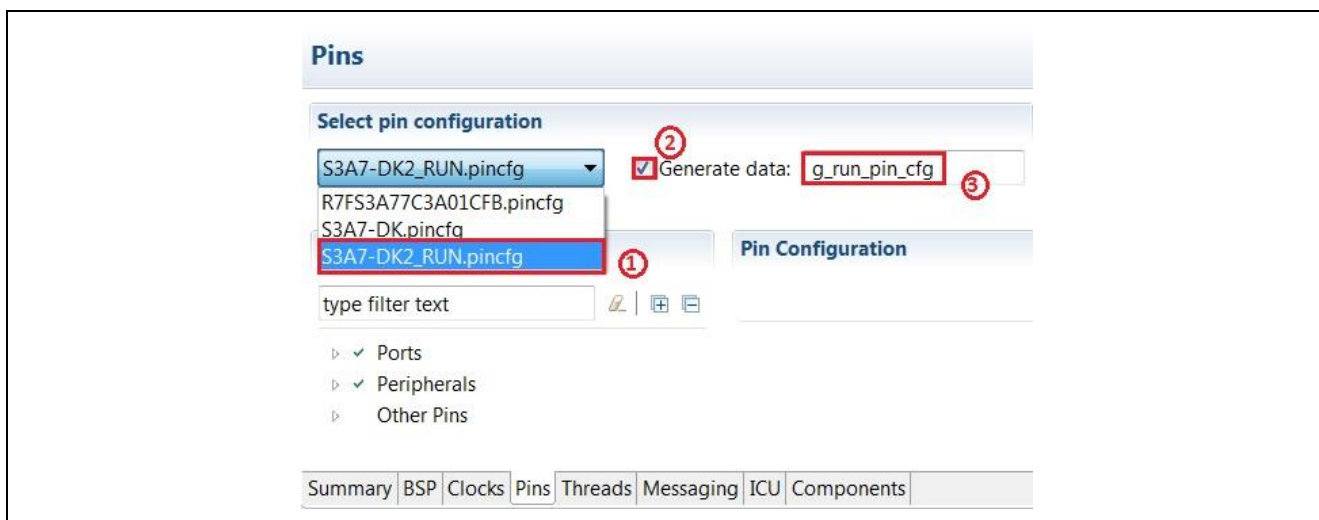


図 18 割り当てようとする新しいピン設定ファイルを選択

- 『Synergy SSP ユーザーズマニュアル』(SSP User's Manual) で説明されているように、各ピンの I/O 機能を設定した後、SSP Configurator の [Generate Project Content] ボタンをクリックすると、新しいピン設定が生成されます。

### 5.1.2 PPv2 フレームワークの実行プロファイルの CGC 設定 (CGC Configuration of PPv2 Framework Run Profile)

この実行プロファイルに対応する電力制御モードを定義するには、CGC 設定インスタンス (CGC Configuration Instance) のプロパティビューを使用します。

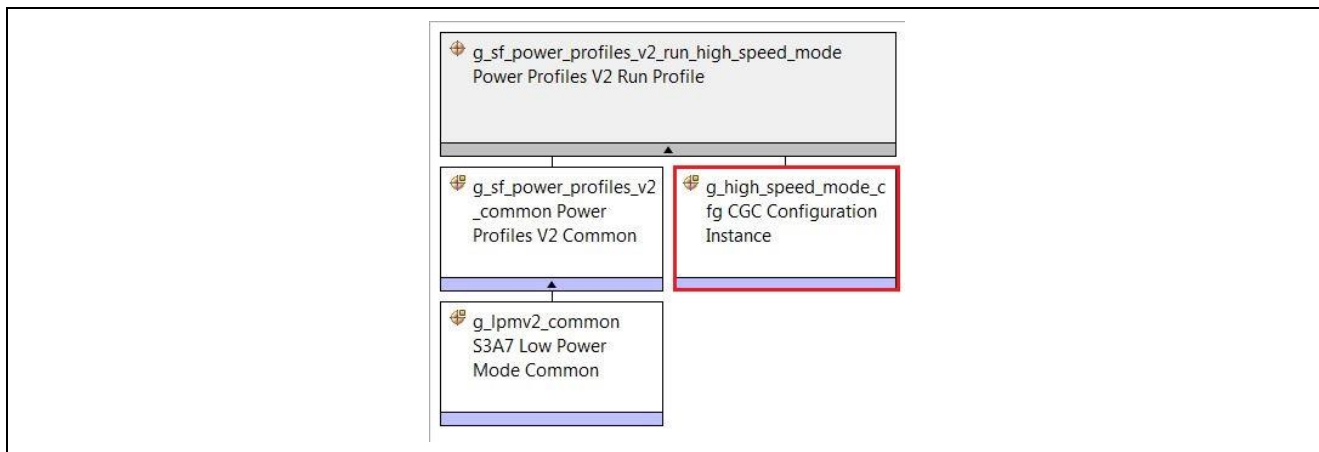


図 19 実行プロファイル (Run Profile) モジュール内の CGC 設定インスタンス



表 7 に、CGC 設定のプロパティビューで使用可能な各種パラメータを示します。

**表 7 実行プロファイル (Run Profile) モジュールに関する CGC 設定項目**

ISDE のプロパティ	値	説明
Name	g_cgc_cfg	モジュール名
System Clock	HOCO (デフォルト)、MOCO、LOCO、Main Oscillator、Sub-Clock、PLL	システムクロックのソースを設定します
LOCO State Change	None (デフォルト)、Stop、Start	LOCO の状態変更の選択肢
MOCO State Change	None (デフォルト)、Stop、Start	MOCO の状態変更の選択肢
HOCO State Change	None (デフォルト)、Stop、Start	HOCO の状態変更の選択肢
Sub-Clock State Change	None (デフォルト)、Stop、Start	サブクロックの状態変更の選択肢
Main Clock State Change	None (デフォルト)、Stop、Start	メインクロックの状態変更の選択肢
PLL State Change	None (デフォルト)、Stop、Start	PLL ソースクロックの選択肢
PLL Source Clock	HOCO (デフォルト)、MOCO、LOCO、Main Oscillator、Sub-Clock、PLL	PLL のソースを設定します
PLL Divisor	1 (デフォルト)、2、3、4	PLL 出力周波数の除数 (分周比)
PLL Multiplier	10.0(デフォルト)、10.5、11.0、11.5、12.0、12.5、13.0、13.5、14.0、14.5、15.0、15.5、16.0、16.5、17.0、17.5、18.0、18.5、19.0、19.5、20.0、20.5、21.0、21.5、22.0、22.5、23.0、23.5、24.0、24.5、25.0、25.5、26.0、26.5、27.0、27.5、28.0、28.5、29.0、29.5、30.0、31.0	PLL 出力周波数の乗数 (通倍比)
PCLKA Divisor	1 (デフォルト)、2、4、8、16、64	周辺回路のクロック A の除数 (分周比)
PCLKB Divisor	1 (デフォルト)、2、4、8、16、64	周辺回路のクロック B の除数 (分周比)
PCLKC Divisor	1 (デフォルト)、2、4、8、16、64	周辺回路のクロック C の除数 (分周比)
PCLKD Divisor	1 (デフォルト)、2、4、8、16、64	周辺回路のクロック D の除数 (分周比)
BCLK Divisor	1 (デフォルト)、2、4、8、16、64	外部バスのクロックの除数 (分周比)
FCLK Divisor	1 (デフォルト)、2、4、8、16、64	フラッシュのクロックの除数 (分周比)
ICLK Divisor	1 (デフォルト)、2、4、8、16、64	システムクロックの除数 (分周比)

注： これらの CGC パラメータに対する割り当ては、各電力制御モードで使用可能な発振器の仕様と、各動作クロックで想定する周波数の組み合わせを満たす必要があります。これらの関係は、それぞれの MCU 製品の『Synergy MCU ユーザーズマニュアル』(Synergy MCU User's Manual)の「CGC Block Diagram」(CGC ブロック図)で示されています。

これらのクロックに対応する最大動作周波数範囲は、それぞれの MCU 製品の『MCU ユーザーズマニュアル』に掲載されている範囲から外れないようにする必要があります。『S3A7 MCU グループユーザーズマニュアル』の図 20 はこれらの仕様を表す表を掲載しており、同様の内容を以下に示します。

Item	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	CPU, DTC, DMAC, Flash, SRAM	Up to 48 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (QSPI, SPI, SCI, SCE5, SDHI, CRC, IrDA, GPT bus-clock)	Up to 48 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (DAC12, IIC, SSI, DOC, CAC, CAN, AGT, POEG, CTSU)	Up to 32 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock C (PCLKC)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (ADC14 conversion clock)	Up to 64 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (GPT count clock)	Up to 64 MHz Division ratio: 1/2/4/8/16/32/64
FlashIF clock (FCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	FlashIF	1 MHz to 32 MHz (P/E) Up to 32 MHz (Read) Division ratio: 1/2/4/8/16/32/64
External bus clock (BCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	External bus	Up to 24 MHz Division ratio: 1/2/4/8/16/32/64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK	EBCLK pin	Up to 12 MHz Division ratio: 1 or 2
USB clock (UCLK)	PLL	USBFS	48 MHz

図 20 Synergy S3A7 MCU グループの内部クロックの最大動作周波数範囲

現在のプロパティビューは、ユーザが指定した割り当ての自動的な検証を実行しないため、代わりに SSP Configurator の CGC パネルを使用して、割り当てを最初にチェックしてください。図 21 に、CGC 設定に関する High-speed モードの例を示します。

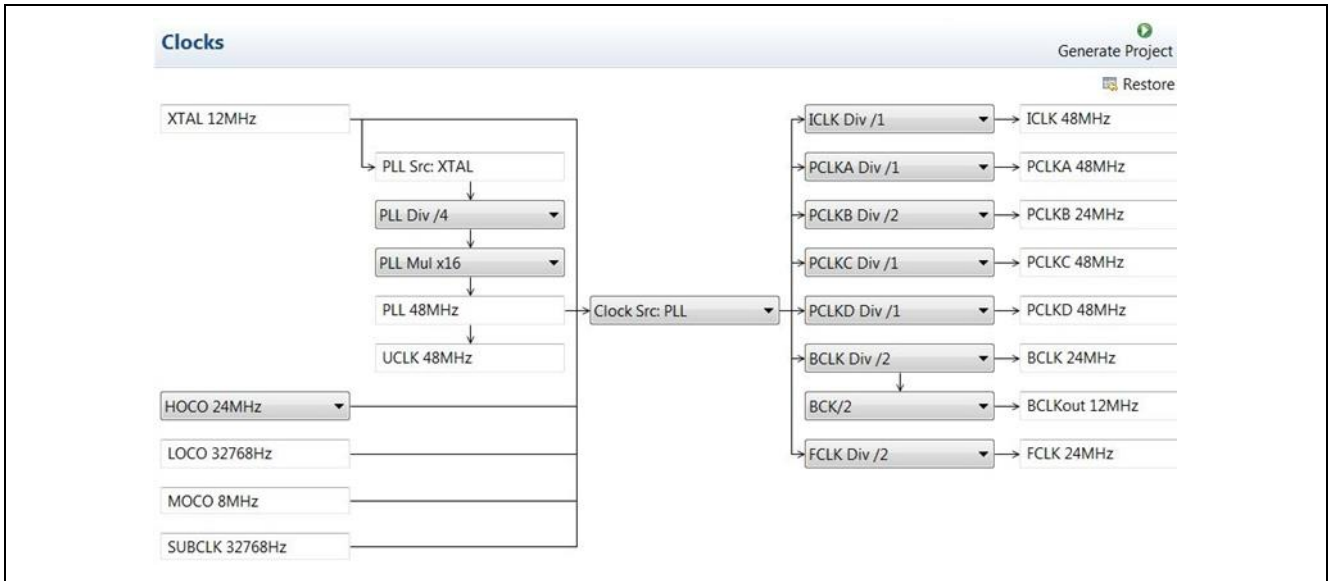


図 21 電力制御モードに対応するクロック設定の確認

同様に、他の電力制御モードに対して CGC 設定を定義することもできます。

## 5.2 PPv2 フレームワークの低消費電力プロファイルの設定 (Configuration of PPv2 Framework Low Power Profile)

Synergy MCU グループによっては、低消費電力プロファイル (Low Power Profile) モジュールのプルダウンメニューで、次の 3 つの LPM モードを選べる場合があります。スリープ (Sleep)、ソフトウェアスタンバイ (Software Standby)、ディープソフトウェアスタンバイ (Deep Software Standby) です。

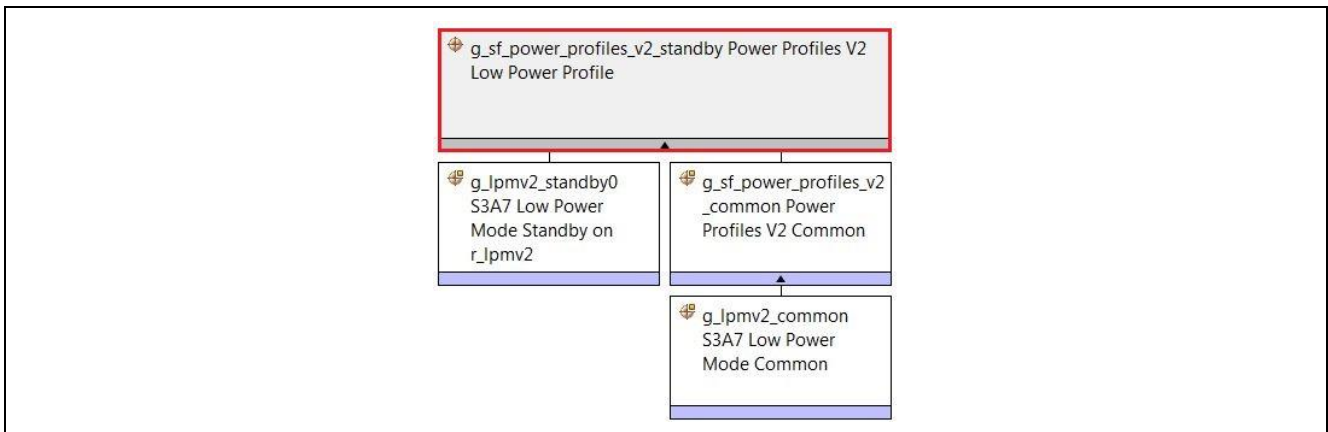


図 22 PPv2 の低消費電力プロファイルのスタンバイモジュール

低消費電力プロファイル (Low Power Profile) モジュールが既に 1 つのスレッドに追加されたらと想定すると、プロファイルに対して次の設定が割り当てられます。

表 8 PPv2 の低消費電力プロファイルに関する設定項目

ISDE のプロパティ	値	説明
Name	g_sf_power_profiles_v2_low_power_0	モジュール名
Callback (Low Power Exit Event N/A when using Deep Software Standby) (コールバック (ディープソフトウェアスタンバイを使用する場合、低消費電力の終了イベントは使用不可))	NULL (デフォルト)	LPM に移行する前のイベントと LPM 終了後のイベントを処理するためのコールバック関数
Low power entry pin configuration table (低消費電力モードに移行するときのピン設定テーブル)	NULL (デフォルト)	LPM に移行する前のピン設定テーブル
Low power exit pin configuration table (低消費電力モードを終了するときのピン設定テーブル)	NULL (デフォルト)	LPM を終了した後のピン設定テーブル

コールバック関数 (callback function) を使用して、以下のイベントを処理することができます。

- SF\_POWER\_PROFILES\_V2\_EVENT\_PRE\_LOW\_POWER
- SF\_POWER\_PROFILES\_V2\_EVENT\_POST\_LOW\_POWER

以下の 2 つのピン設定テーブルを使用して、IO ポートの機能を変更することができます。

- Low power entry pin configuration table (低消費電力モードに移行するときのピン設定テーブル)
- Low power exit pin configuration table (低消費電力モードを終了するときのピン設定テーブル)

PPv2 API 関数 LowPowerApply () は、これらのテーブルを内部で使用します。

### 5.2.1 LPM スリープモードの設定 (Configuration of the LPM Sleep Mode)

LPM スリープ (Sleep) モードの設定は容易です。いずれの割り込みが発生した場合でも、MCU はスリープモードからウェイクアップするためです。そのため、モジュール名を変更するだけで済みます。

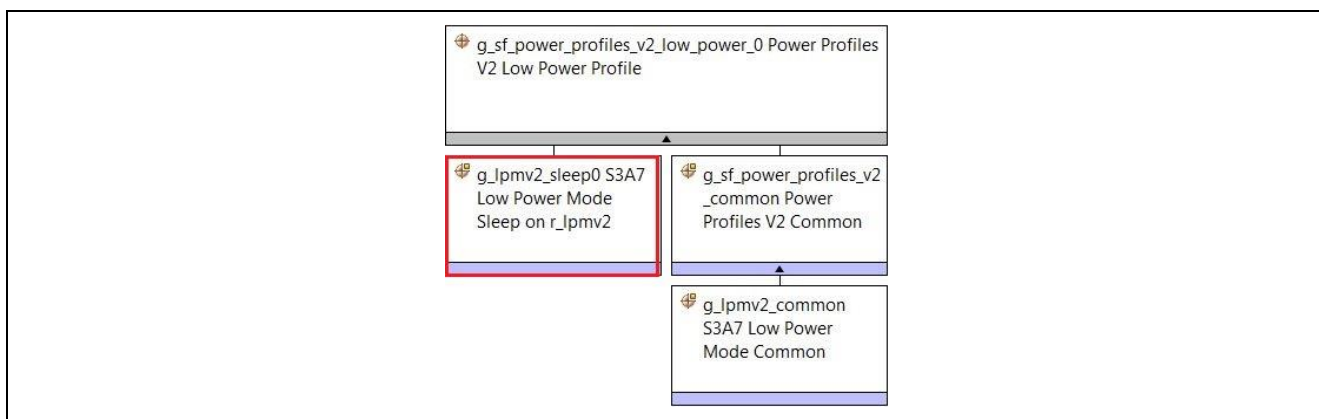


図 23 PPv2 の低消費電力プロファイルのスリープドライバ r\_lpmv2

表 9 PPv2 プロファイルのスリープモジュールに関する設定項目

ISDE のプロパティ	値	説明
Parameter Checking (パラメータのチェック)	BSP (デフォルト)、Enabled、Disabled	パラメータチェックを有効または無効にします
Name	g_lpmv2_sleep0	モジュール名

### 5.2.2 LPM スタンバイモードの設定 (Configuration of the LPM Standby Mode)

スタンバイモードの設定は、モード終了のトリガ (exit trigger) や、スタンバイモードとスヌーズモードの間でのいくつかの遷移条件 (transition condition) を設定します。これらの遷移条件は、PPv2 の現在のリリースでは、スタンバイモードの特殊な状況として扱われます。スヌーズ設定の詳細については、次の章を参照してください。

図 24 に、S3A7 MCU のスタンバイモジュールの例を示します。

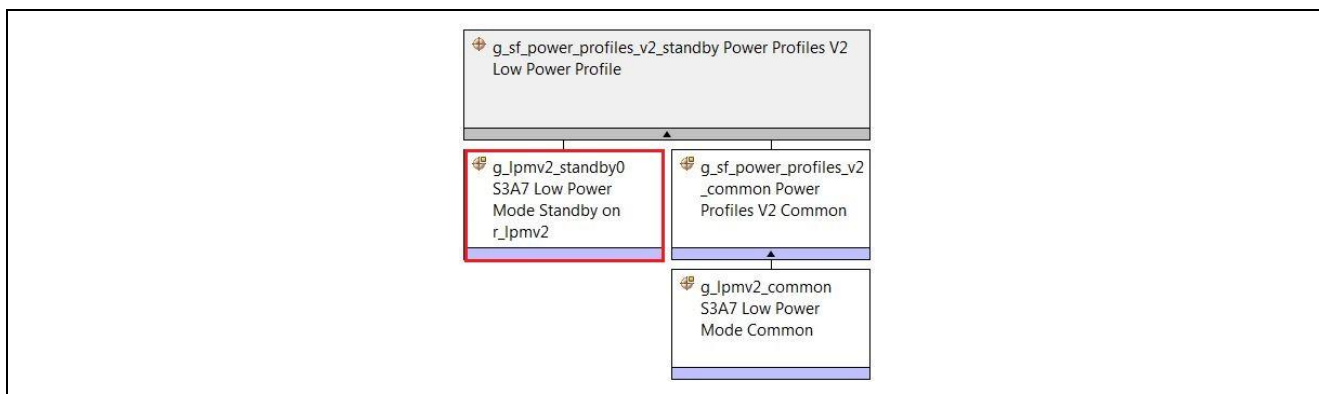


図 24 PPv2 の低消費電力プロファイルのスタンバイドライバ r\_lpmv2

表 10 PPv2 プロファイルのスタンバイモジュール r\_lpmv2 に関する設定項目

ISDE のプロパティ	値	説明
Parameter Checking (パラメータのチェック)	BSP (デフォルト)、Enabled、Disabled	パラメータチェックを有効または無効にします
Name	g_lpmv2_standby0	モジュール名
Choose the low power mode (低消費電力モードの選択)	Standby (デフォルト)、Standby with Snooze enabled	Low power mode selection (低消費電力モードの選択肢)
Output port state in standby and Deep Software Standby, applies to address output, data output, and other bus control pins (スタンバイとディープソフトウェアスタンバイモードにおける出力ポートの状態であり、アドレス出力、データ出力、他のバス制御ピンに適用されます)	No Change (デフォルト)、High Impedance state	出力ポートの状態の選択肢
スタンバイモード終了のソース (トリガ)		以下のフィールドを選択してください
IRQ[0:15]	Enabled、Disabled (デフォルト)	外部割り込み IRQ0 ~ IRQ15 のいずれかを選択します
IWDT	Enabled、Disabled (デフォルト)	IWDT の選択肢
Key Interrupt (キー入力による割り込み)	Enabled、Disabled (デフォルト)	キー入力による割り込みの選択肢

ISDE のプロパティ	値	説明
LVD1 Interrupt (LVD1 による割り込み)	Enabled、Disabled (デフォルト) (デフォルト: Disabled)	LVD1 の選択肢
LVD2 Interrupt (LVD2 による割り込み)	Enabled、Disabled (デフォルト)	LVD2 の選択肢
Analog Comparator High-speed 0 Interrupt (アナログコンパレータ高速 0 による割り込み)	Enabled、Disabled (デフォルト)	アナログコンパレータの選択肢
RTC Period (RTC 期間)	Enabled、Disabled (デフォルト)	RTC 期間の選択肢
RTC Alarm (RTC アラーム)	Enabled、Disabled (デフォルト)	RTC アラームの選択肢
USBFS	Enabled、Disabled (デフォルト)	USBFS の選択肢
AGT1 underflow (AGT1 アンダーフロー)	Enabled、Disabled (デフォルト)	AGT1 アンダーフローの選択肢
AGT1 Compare Match A (AGT1 比較の一致 A)	Enabled、Disabled (デフォルト)	AGT1 CMA の選択肢
AGT1 Compare Match B (AGT1 比較の一致 B)	Enabled、Disabled (デフォルト)	AGT1 CMB の選択肢
I2C 0	Enabled、Disabled (デフォルト)	I2C 0 の選択肢
<b>スヌーズ (Snooze) モードの設定</b>		
Snooze Entry Source (スヌーズへの移行のソース)	RXD0 falling edge (デフォルト)、IRQ0:15、KINT (キーによる割り込み)、ACMPLP (低速アナログコンパレータ)、RTC Alarm、RTC Period、AGT1 Underflow、AGT1 Compare Match A、AGT1 Compare Match B	スヌーズ (Snooze) モードへの移行のソース
<b>Snooze Exit Source (スヌーズ終了のソース)</b>		以下のフィールドを選択してください
AGT1 Underflow (AGT1 アンダーフロー)	Enabled、Disabled (デフォルト)	AGT1 アンダーフローの選択肢
DTC Transfer Completion (DTC 転送の完了)	Enabled、Disabled (デフォルト)	DTC 転送の完了に関する選択肢
DTC Transfer Completion Negated signal (DTC 転送の完了に対するネゲート信号)	Enabled、Disabled (デフォルト)	DTC 転送の完了に対するネゲート信号の選択肢
ADC0 Compare Match (ADC0 比較の一致)	Enabled、Disabled (デフォルト)	ADC0 比較の一致に関する選択肢
ADC0 Compare Mismatch (ADC0 比較の不一致)	Enabled、Disabled (デフォルト)	ADC0 比較の不一致に関する選択肢
SCI0 Address Match (SCI0 アドレスの一致)	Enabled、Disabled (デフォルト)	SCI0 アドレスの一致に関する選択肢
DTC state in Snooze Mode (スヌーズモードにおける DTC の状態)	Enabled、Disabled (デフォルト)	スヌーズモードにおける DTC の状態に関する選択肢

### 5.2.3 LPM スヌーズモードの設定 (Configuration of the LPM Snooze Mode)

図 25 に示すように、PPv2 ソフトウェアスタンバイモードのプロパティビューにおいて、[Choose the low power mode] (低消費電力モードの選択) フィールドで [Standby with Snooze Enabled] (スヌーズを有効とするスタンバイ) を選択することで、スヌーズモードを有効化できます。

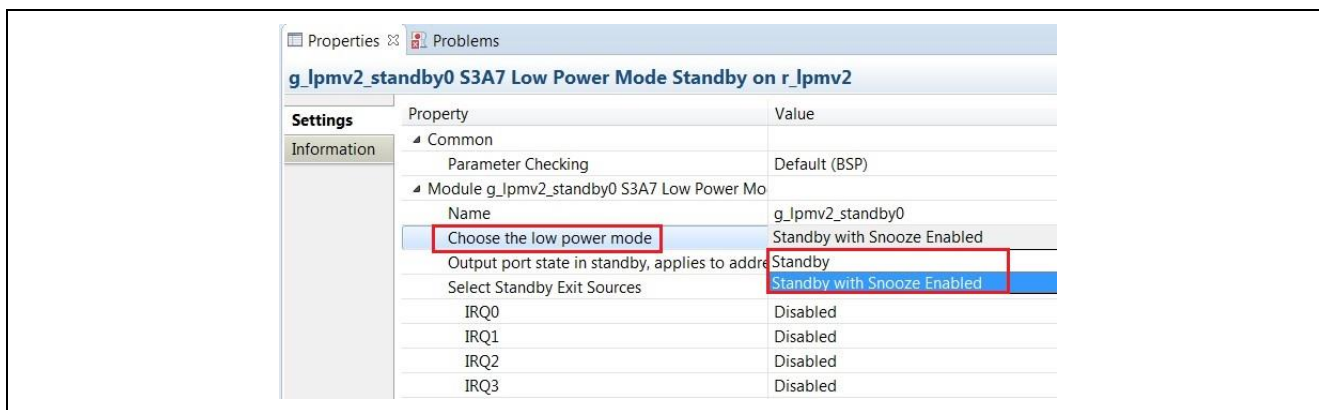


図 25 ソフトウェアスタンバイモードの一部としてスヌーズモードを有効化

### 5.2.4 LPM ディープソフトウェアスタンバイモードの設定 (Configuration of the LPM Deep Software Standby Mode)

ディープソフトウェアスタンバイ (Deep Software Standby) モードの設定とは、ディープソフトウェアスタンバイモードの終了トリガ (exit trigger) と内部の電力供給オプション (intenal power supply option) を設定することです。S5D9 を例とした、以下に示す設定例のように、PPv2 フレームワークを使用してください。

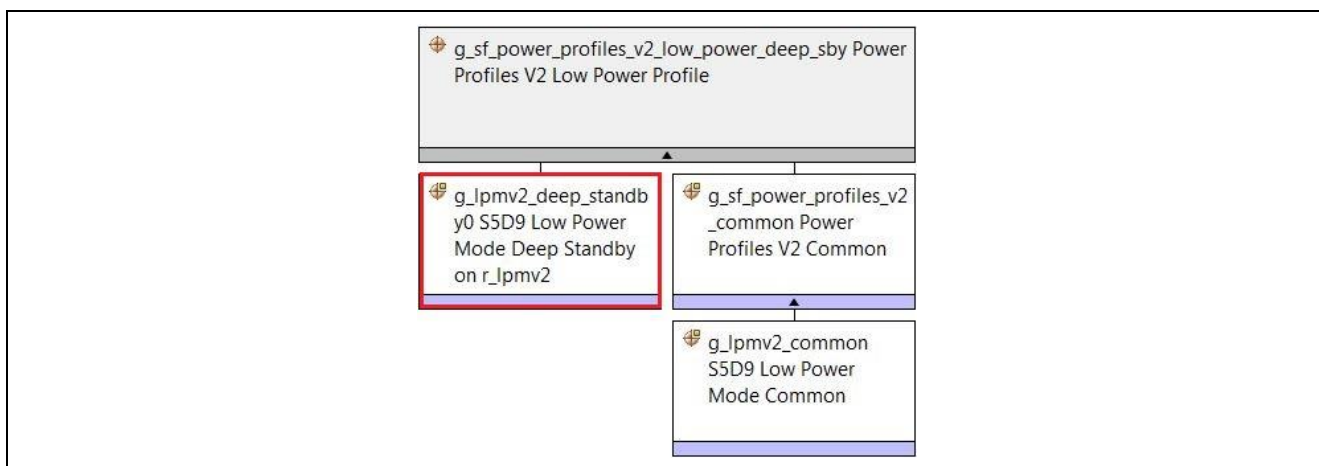


図 26 PPv2 の低消費電力プロファイルのディープソフトウェアスタンバイドライバ r\_lpmv2

表 11 PPv2 プロファイルのディープソフトウェアスタンバイモジュールに関する設定項目

ISDE のプロパティ	値	説明
Parameter Checking (パラメータのチェック)	BSP (デフォルト)、Enabled、Disabled	パラメータチェックを有効または無効にします
Name	g_lpmv2_deep_standby0	モジュール名
Output port state in standby and Deep Software Standby, applies to address output, data output, and other bus control output pins (スタンバイとディープソフトウェアスタンバイモードにおける)	High impedance state、No change (デフォルト)	スタンバイモードとディープソフトウェアスタンバイモードにおける出力ポートの状態に関する設定

ISDE のプロパティ	値	説明
出力ポートの状態、アドレス出力、データ出力、他のバス制御出力ピンに適用)		
Maintain or reset the IO port states on exit from Deep Software Standby mode (ディープソフトウェアスタンバイモードを終了するときに IO ポートの状態を保持またはリセット)	Maintain the IO port states (デフォルト)、Reset the IO port states	モード終了時の出力ポートの状態に関する設定
Internal power supply control in Deep Software Standby mode (ディープソフトウェアスタンバイモードでの内部電力供給の制御)	Maintain the internal power supply (内部電力供給を保持) (デフォルト)、Cut the power supply to standby RAM, Low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit (スタンバイ RAM、低速オンチップオシレータ、AGTn、USBFS/HS 再開検出ユニットへの電力供給を停止)、Cut the power supply to LVDn, standby RAM, Low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit (LVDn、スタンバイ RAM、低速オンチップオシレータ、AGTn、USBFS/HS 再開検出ユニットへの電力供給を停止)	ディープソフトウェアスタンバイモードでの内部電力供給の制御に関する設定
ディープソフトウェアスタンバイをキャンセルするソース/エッジ:		以下のフィールドを選択してください
IRQ0	Enabled、Disabled (デフォルト)	IRQ0 の選択肢
IRQ0 Edge (IRQ0 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ0 のエッジに関する選択肢
IRQ1	Enabled、Disabled (デフォルト)	IRQ1 の選択肢
IRQ1 Edge (IRQ1 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ1 のエッジに関する選択肢
IRQ2	Enabled、Disabled (デフォルト)	IRQ2 の選択肢
IRQ2 Edge (IRQ2 のエッジ)	Disabled、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ) (デフォルト)	IRQ2 のエッジに関する選択肢
IRQ3	Enabled、Disabled (デフォルト)	IRQ3 の選択肢
IRQ3 Edge (IRQ3 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ3 のエッジに関する選択肢
IRQ4	Enabled、Disabled (デフォルト)	IRQ4 の選択肢
IRQ4 Edge (IRQ4 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ4 のエッジに関する選択肢
IRQ5	Enabled、Disabled (デフォルト)	IRQ5 の選択肢
IRQ5 Edge (IRQ5 のエッジ)	Disabled、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ) (デフォルト)	IRQ5 のエッジに関する選択肢
IRQ6	Enabled、Disabled (デフォルト)	IRQ6 の選択肢
IRQ6 Edge (IRQ6 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ6 のエッジに関する選択肢
IRQ7	Enabled、Disabled (デフォルト)	IRQ7 の選択肢
IRQ7 Edge (IRQ7 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ7 のエッジに関する選択肢
IRQ8	Enabled、Disabled (デフォルト)	IRQ8 の選択肢
IRQ8 Edge (IRQ8 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ8 のエッジに関する選択肢
IRQ9	Enabled、Disabled (デフォルト)	IRQ9 の選択肢
IRQ9 Edge (IRQ9 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ9 のエッジに関する選択肢



ISDE のプロパティ	値	説明
IRQ10	Enabled、Disabled (デフォルト)	IRQ10 の選択肢
IRQ10 Edge (IRQ10 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ10 のエッジに関する選択肢
IRQ11	Enabled、Disabled (デフォルト)	IRQ11 の選択肢
IRQ11 Edge (IRQ11 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ11 のエッジに関する選択肢
IRQ12	Enabled、Disabled (デフォルト: Disabled)	IRQ12 の選択肢
IRQ12 Edge (IRQ12 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ12 のエッジに関する選択肢
IRQ13	Enabled、Disabled (デフォルト)	IRQ13 の選択肢
IRQ13 Edge (IRQ13 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ13 のエッジに関する選択肢
IRQ14	Enabled、Disabled (デフォルト)	IRQ14 の選択肢
IRQ14 Edge (IRQ14 のエッジ)	Disabled、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ) (デフォルト: Disabled)	IRQ14 のエッジに関する選択肢
IRQ15	Enabled、Disabled (デフォルト: Disabled)	IRQ15 の選択肢
IRQ15 Edge (IRQ15 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	IRQ15 のエッジに関する選択肢
LVD1	Enabled、Disabled (デフォルト)	LVD1 の選択肢
LVD1 Edge (LVD1 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	LVD1 のエッジに関する選択肢
LVD2	Enabled、Disabled (デフォルト)	LVD2 の選択肢
LVD2 Edge (LVD2 のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	LVD2 のエッジに関する選択肢
RTC Interval (RTC の間隔)	Enabled、Disabled (デフォルト)	RTC の間隔に関する選択肢
RTC Alarm (RTC アラーム)	Enabled、Disabled (デフォルト)	RTC アラームの選択肢
NMI	Enabled、Disabled (デフォルト: Disabled)	NMI の選択肢
NMI Edge (NMI のエッジ)	Disabled (デフォルト)、Rising Edge (立ち上がりエッジ)、Falling Edge (立ち下がりエッジ)	NMI のエッジに関する選択肢
USBFS	Enabled、Disabled (デフォルト)	USBFS の選択肢
UBSHS	Enabled、Disabled (デフォルト)	UBSHS の選択肢
AGT1	Enabled、Disabled (デフォルト)	AGT1 の選択肢

注: PPv2 フレームワークのプロパティダイアログ (property dialog) では、ディープソフトウェアスタンバイ (Deep Software Standby) モードでの内部電力供給に関して、既定の 2 つの設定を選択できます。それぞれの MCU 製品の内部コンポーネントへの内部電力供給の停止に関しては、『Synergy MCU グループユーザーズマニュアル』 (Synergy Microcontroller Group User's Manual) により多くの選択肢が掲載されています。例えば、『Synergy S5D9 MCU グループユーザーズマニュアル』の表 11.2 などです。

#### Maintain the internal power supply

Cut the power supply to standby RAM, low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit

Cut the power supply to LVDn, standby RAM, low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit

図 27 PPv2 フレームワークのディープソフトウェアスタンバイモードでの内部電力供給に関する既定の複数のオプション

### 5.3 PPv2 フレームワークの共通モジュールの設定 (Configuration of the LPM Snooze Mode)

PPv2 フレームワークには、実行プロファイルモジュール (Run profile module ) と低消費電力プロファイルモジュール (Low Power profile module) で共有される 2 つの Common (共通) モジュールがあります。

- 自動生成される Power Profile V2 Common モジュール

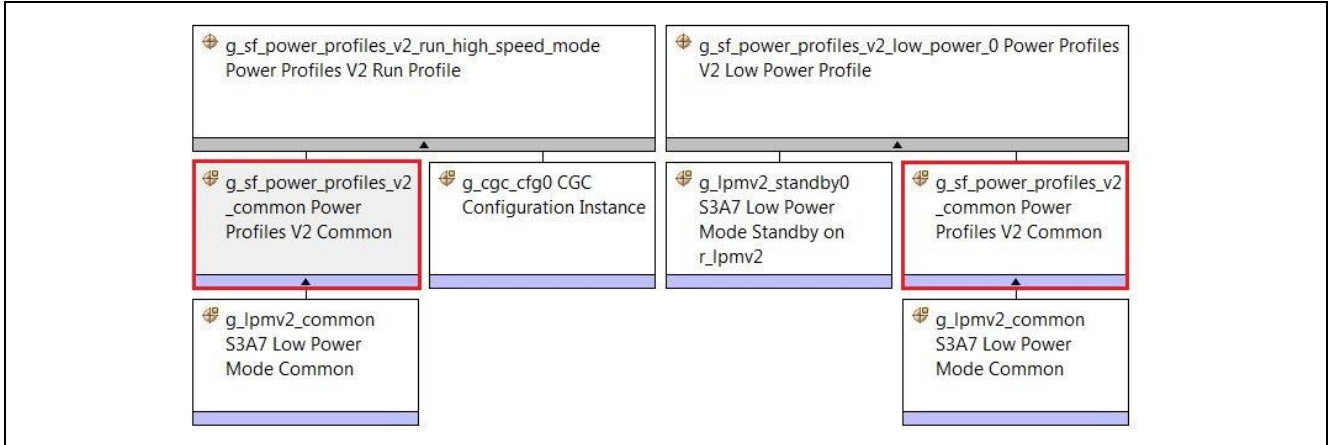


図 28 PPv2 の実行/低消費電力プロファイルモジュールの PPv2 Common モジュール

表 12 に、それぞれの設定項目の一覧を示します。

表 12 電力プロファイルの V2 Common に関する設定項目

ISDE のプロパティ	値	説明
Parameter Checking (パラメータのチェック)	BSP (デフォルト)、Enabled、Disabled	パラメータチェックを有効または無効にします
Name	g_sf_power_profiles_v2_common	モジュール名

- ユーザが追加した MCU 固有の Low Power Mode Common モジュール

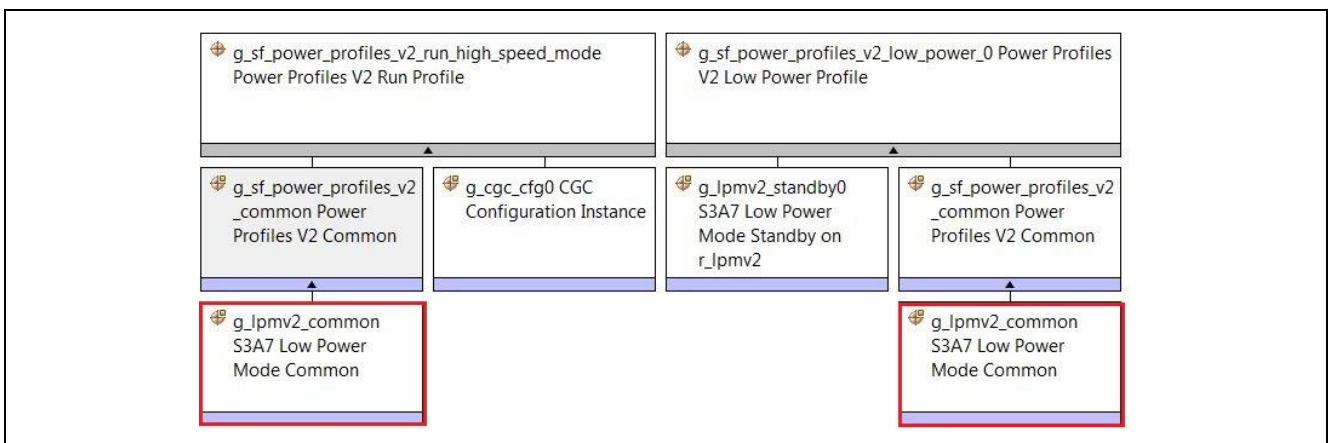


図 29 PPv2 の実行/低消費電力プロファイルモジュールの LPM Common Profile モジュール

表 13 に、それぞれの設定項目の一覧を示します。

**表 13 Low Power Mode Common に関する設定項目**

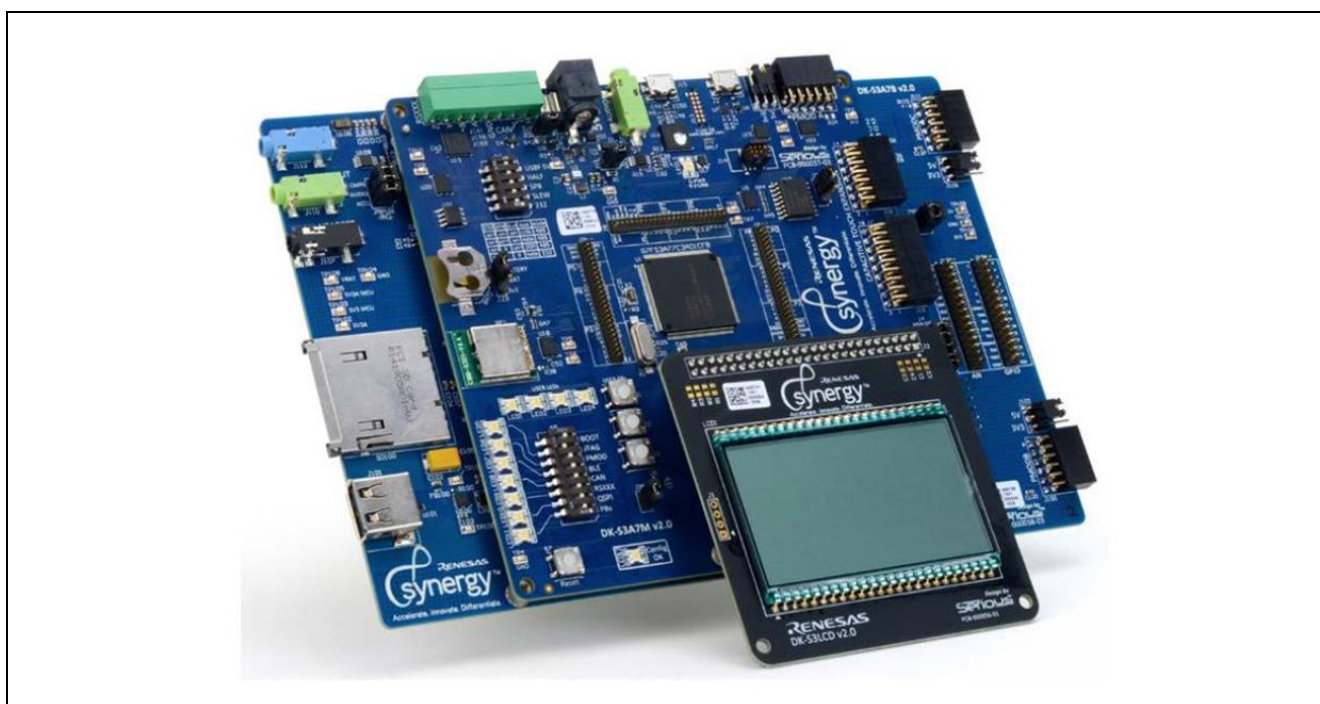
ISDE のプロパティ	値	説明
Parameter Checking (パラメータのチェック)	BSP (デフォルト)、Enabled、Disabled	パラメータチェックを有効または無効にします
Name	g_lpmv2_common	モジュール名

注: PPv2 フレームワークの実行プロファイルを追加した時点で、Synergy MCU S3A7 の LPM モジュールは 図 12 に既に追加済みです。

## 6. PPv2 フレームワークのサンプルアプリケーション (PPv2 Framework Application Example)

このアプリケーションプロジェクトの目標は、PPv2 フレームワークモジュールを使用して、RTOS 対応アプリケーションの作成方法を示すことです。このアプリケーションプロジェクトは、電力制御モードと LPM モードのさまざまな組み合わせや、複数のモード間遷移を列挙する総合的なテストプラットフォームを提供します。ただしここに掲載するのは、2 個の電力制御モードと 2 個の LPM モードを使用した簡略化バージョンです。

このプロジェクトで使用する開発ボードとして、DK-S3A7 を選択しました。このボードには、3 個の押しボタン、1 個のポテンショメータ (potentiometer)、MCU の状態変化を視覚的に表現するための複数の LED など、ユーザが制御できる要素があるからです。



**図 30 DK-S3A7 v2.0 上での PPv2 プロジェクトのデモンストレーション**

Synergy S3A7 MCU グループには、5 つの動作電力制御モードがあります。

- High-speed モード
- Middle-speed モード
- Low-speed モード
- Low-voltage モード
- Subosc-speed モード

Synergy S3A7 MCU グループには、3つのLPMモードがあります。

- スリープ (Sleep) モード
- ソフトウェアスタンバイ (Software Standby) モード
- スヌーズ (Snooze) モード

図 31 に、モードの遷移とそれらのトリガ条件を示します。詳細については、『Synergy S3A7 MCU グループ ユーザーズマニュアル』(Synergy S3A7 Microcontroller Group Users’s Manual)を参照してください。

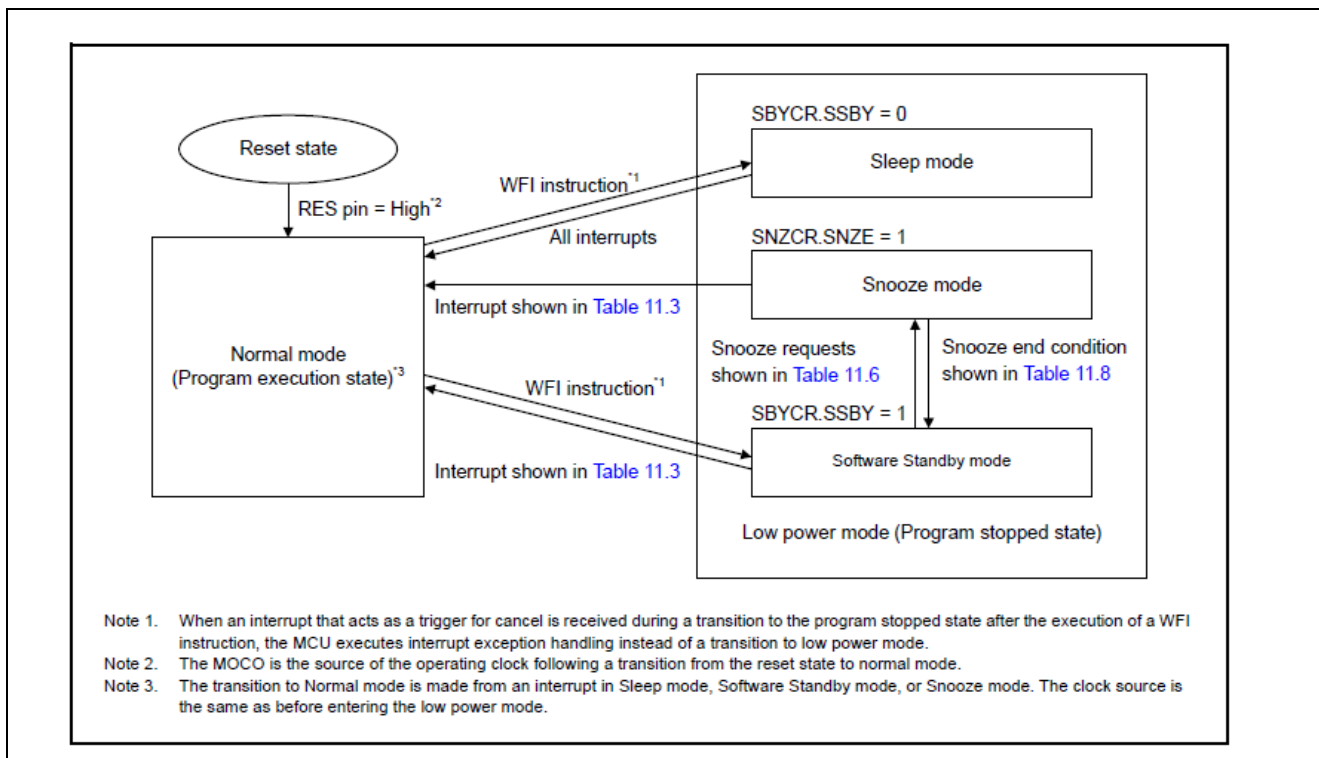


図 31 Synergy S3A7 MCU のLPMモードの遷移

### 6.1 プロジェクトの概要 (Project Overview)

このプロジェクトの動作は、以下のように要約できます。

- DK-S3A7 v2.0 ボードの S1 ボタンを押して電力制御モードを選択した後、S1 ボタンを押したままにして、選択した電力制御モードに移行します。現時点で実装されているのは、High-Speed モードと Middle-Speed モードのみです。これらのモード間の遷移を赤の実線で表記しています。

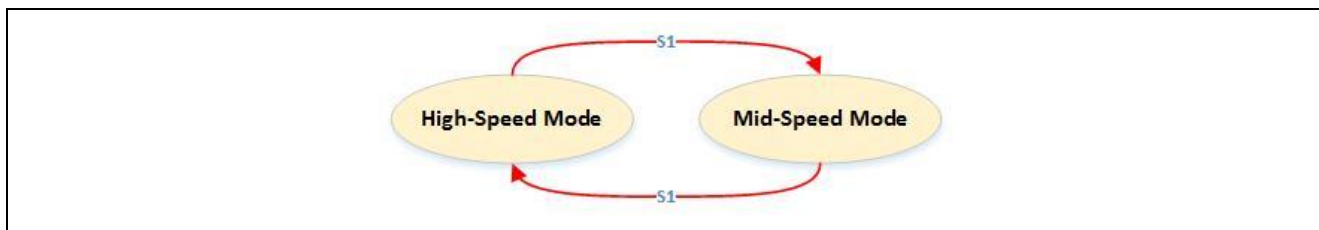


図 32 S1 ボタンを押して電力制御モードを選択

- DK-S3A7 v2.0 ボードの LED1\_G と LED2\_G を使用して、以下のように電力制御モードを表示します。

電力制御モード	LED1_G	LED2_G
High-speed モード (デフォルト)	点灯	点灯
Middle-speed モード	消灯	点灯

- S2 ボタンを押して LPM モードかノーマルモード (Normal mode) のいずれかを選択します。デフォルトはノーマル (Normal) モードです。その後、S2 ボタンを押したままにして、選択した LPM モードに移行します。現時点で実装されているのは、スタンバイ (Standby) モードのみです。このモードとノーマル (Normal) モードとの間の遷移を赤の実線で表記しています。

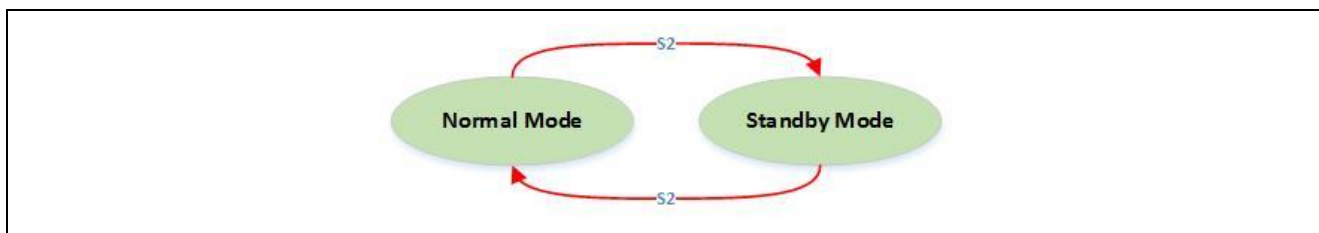


図 33 S2 ボタンを押して LPM モードを選択

- DK-S3A7 v2.0 ボードの LED1\_R と LED2\_R を使用して、以下のように電力制御モードを表現します。

LPM モード	LED1_R	LED2_R
通常 (Normal) モード (デフォルト)	点灯	点灯
ソフトウェアスタンバイ (Software Standby) モード	消灯	点灯

- ソフトウェアスタンバイ (Software Standby) モードに移行した時点で、CPU、大半のオンチップ周辺回路の機能、およびオシレータは停止します。
  - 注:LED1\_R と LED1\_G は、物理的に隣接した LED のペアのため、赤 (R) と緑 (G) の両方が点灯するとオレンジ色のように見えます。LED2\_R と LED2\_G のペアも、同様な見え方をします。

図 34 に、これらのモード間の遷移に関する状態遷移図を示します。

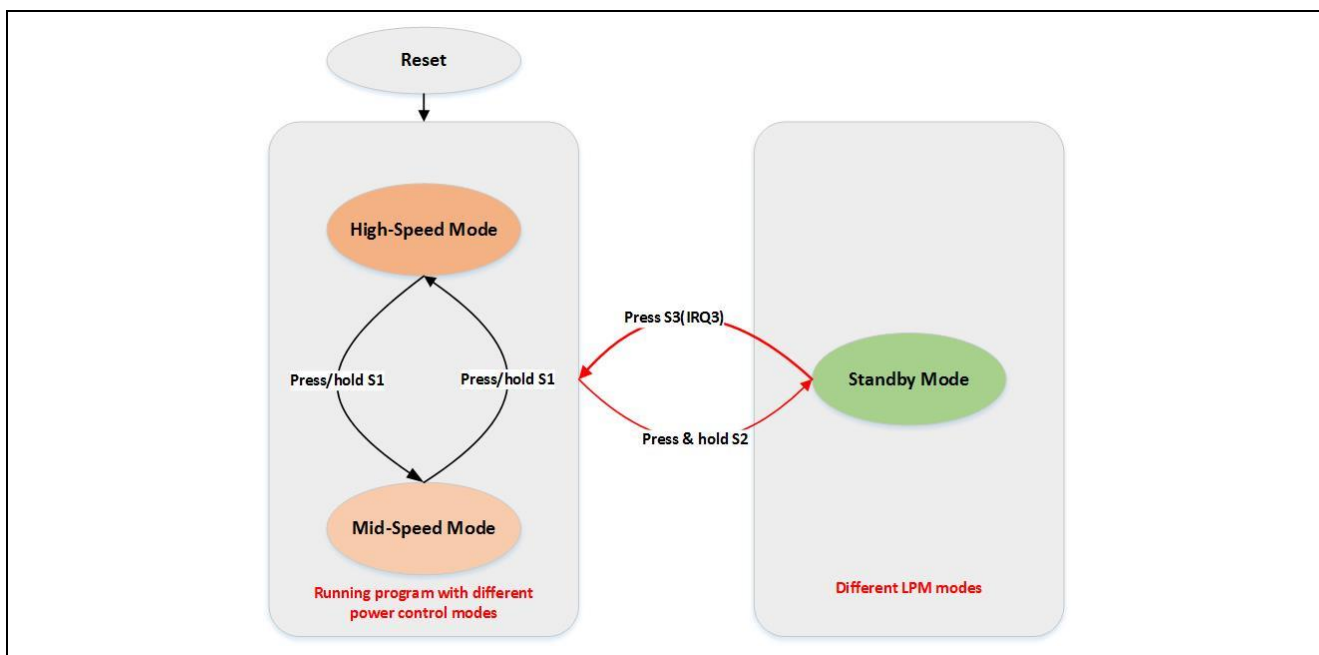


図 34 DK-S3A7 を使用した PPv2 プロジェクトにおける状態遷移図

## 6.2 プロジェクトのソフトウェアアーキテクチャ (Project Software Architecture)

このソフトウェアのアーキテクチャは、以下のように 2 つのパーティションに区分し、4 個のスレッドを使用して実装することが可能です。

- ユーザインタフェースからのイベントを取り扱う **frontend**:
  - **SW1\_thread:S1** ボタンの状態を確認するタイマを作成し、その後、いずれかの電力制御モード (power control mode) を選択するか、S1 タイマからのイベントに基づいて電力制御モードを変更するメッセージを送信するかを決定します。
  - **SW2\_thread: S2** ボタンの状態を確認するタイマを作成し、その後、いずれかの LPM モードを選択するか、S2 タイマからのイベントに基づいて LPM モードを変更するメッセージを送信するかを決定します。
- PPv2 フレームワーク API を使用して、ユーザが選択した電力制御モードと LPM モードに移行するための **backend**:
  - **Power\_control\_mode\_thread**: このスレッドはアプリケーション起動時に一時停止し、その後、thread SW1 によって再開され、MCU を、SW1 スレッドから受け取ったメッセージによって指定された電力制御モードに設定します。
  - **LPM\_mode\_thread**: このスレッドはアプリケーション起動時に一時停止し、その後、SW2 スレッドによって再開され、SW2 スレッドから受け取ったメッセージによって指定された LPM モードに移行します。

図 35 に、これらのパーティションとスレッドを示します。

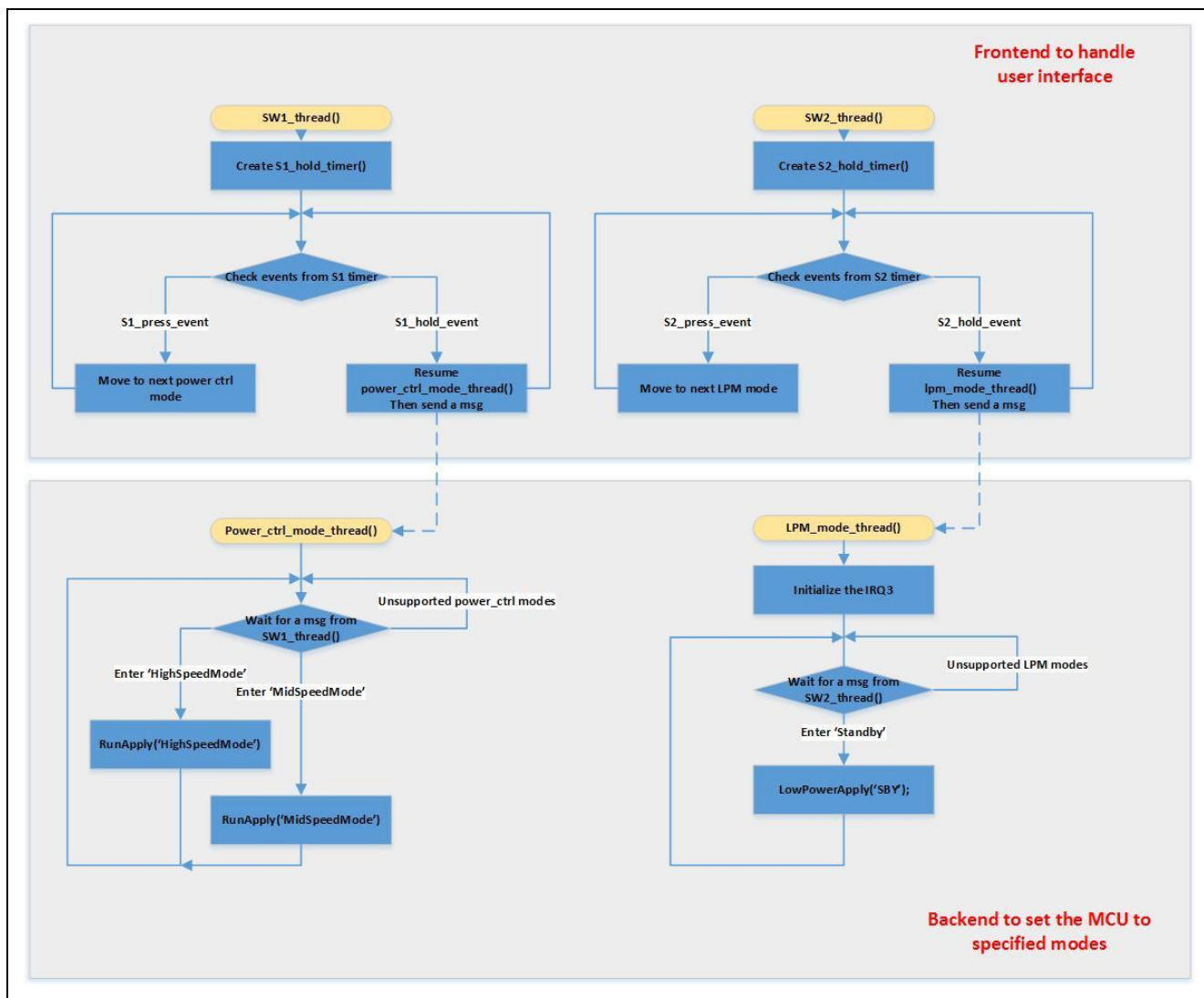


図 35 PPv2 プロジェクトのソフトウェアアーキテクチャ

### 6.3 プロジェクトの設定

4章に掲載した説明に従い、図 36 に ISDE e<sup>2</sup>studio と SSP Configurator を使用して作成した ThreadX イベントとメッセージキュー (message queue) に関連付けられている 4 個のスレッドを示します。

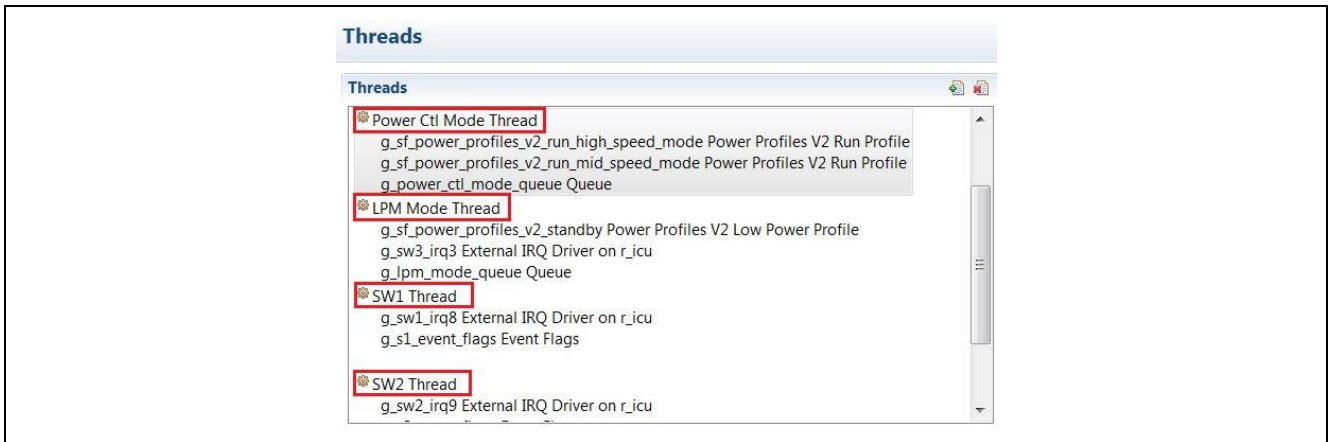


図 36 PPv2 アプリケーションプロジェクトで作成した 4 個のスレッド

#### 6.3.1 Power\_Ctl\_Mode\_Thread (電力制御モードスレッド) の設定 (Configuration of the power\_Ctl\_Mode\_Thread)

現時点で、2 種類の電力制御モードである high\_speed\_mode と mid\_speed\_mode が power\_ctl\_thread に対して追加されています。他の電力モードを追加する方法で、このスレッドを拡大することもできます。

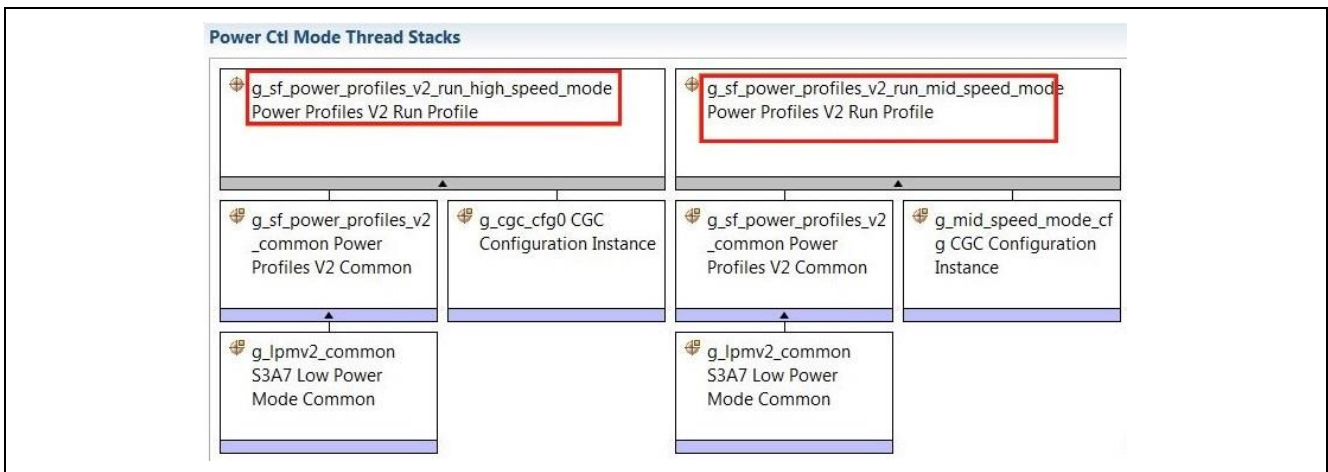


図 37 プロジェクト内の Power\_Ctl\_Mode\_Thread で使用する High-speed モードと middle speed モード

電力制御モードスレッド (Power Ctl Mode thread) は、アプリケーションの起動時に一時停止し、ユーザが電力制御モードを選択する目的で S1 を押しているときに SW1\_thread (SW1 スレッド) によって再開されます。図 38 に、そのプロパティビューを設定する方法を示します。

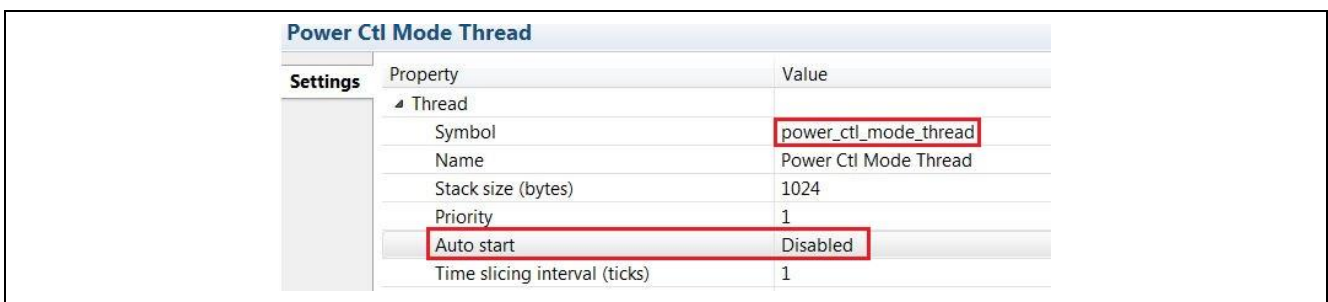


図 38 電力制御モードスレッドがアプリケーションの起動時に一時停止する

どちらの電力制御モードも、自らの IO ポート設定として、デフォルトのピン設定テーブルである `g_bsp_pin_cfg` を使用しています。

- `g_sf_power_profiles_v2_run_high_speed_mode` という CGC 設定は、システムクロックとして 24 MHz の HOCO を使用します。他のクロックは、以下のように生成します。

Module g_high_speed_mode_cfg CGC Configuration Instance	
Name	g_high_speed_mode_cfg
System Clock	HOCO
LOCO State Change	None
MOCO State Change	None
HOCO State Change	None
Sub-Clock State Change	None
Main Clock State Change	None
PLL State Change	None
PLL Source Clock	Main Oscillator
PLL Divisor	1
PLL Multiplier	10.0
PCLKA Divisor	1
PCLKB Divisor	2
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	2
FCLK Divisor	1
ICLK Divisor	1

図 39 プロジェクトで 24 MHz の高速 (High-speed) モードを使用する場合の CGC 設定

- `g_sf_power_profiles_v2_run_mid_speed_mode` という CGC 設定は、システムクロックとして 8 MHz の MOCO を使用します。他のクロックは、以下のように生成します。

Module g_mid_speed_mode_cfg CGC Configuration Instance	
Name	g_mid_speed_mode_cfg
System Clock	MOCO
LOCO State Change	None
MOCO State Change	None
HOCO State Change	None
Sub-Clock State Change	None
Main Clock State Change	None
PLL State Change	None
PLL Source Clock	Main Oscillator
PLL Divisor	1
PLL Multiplier	10.0
PCLKA Divisor	1
PCLKB Divisor	2
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	2
FCLK Divisor	1
ICLK Divisor	1

図 40 プロジェクトで 8 MHz の Middle-speed モードを使用する場合の CGC 設定



### 6.3.2 LPM\_Mode\_Thread (LPM モードスレッド) の設定

現在、1 個の PPv2 フレームワークモジュール、Standby モジュール、および外部 IRQ HAL ドライバがこのスレッドに対して追加されています。ただし、図 41 に示すように、他の PPv2 低消費電力 (Low Power) モジュールを追加する方法で、このスレッドを簡単に拡張することもできます。

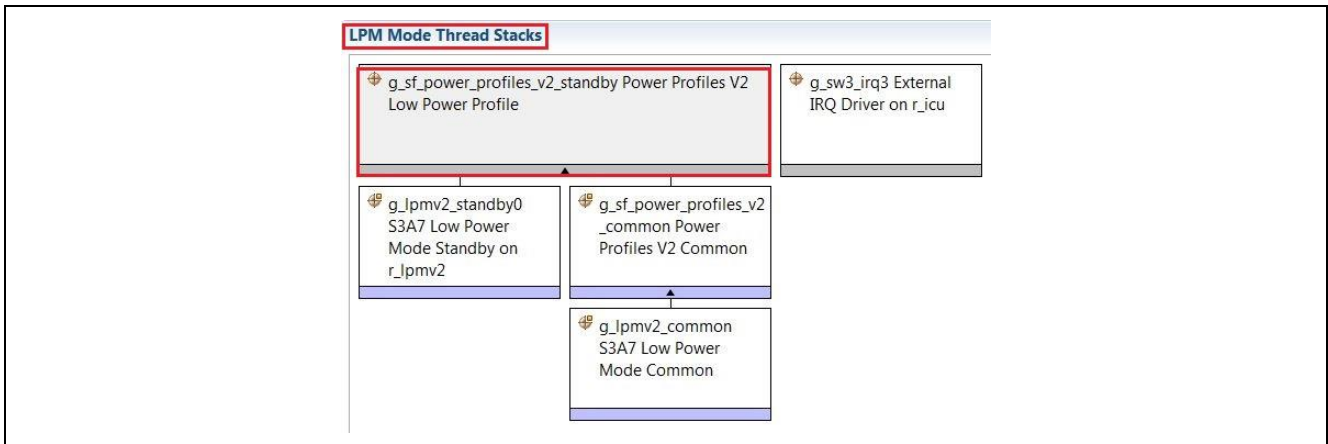


図 41 PPv2 アプリケーションプロジェクト内の LPM\_thread\_mode

- 電力制御モードスレッドと同様、LPM モードスレッドは一時停止し、その後、SW2\_thread (SW2 スレッド) がこのスレッドを再開するまで一時停止しています。
- ソフトウェアスタンバイ (Software Standby) モードへの移行前と終了後それぞれの IO 機能は、2 個のピン設定テーブルである g\_run\_pin\_cfg と g\_sby\_pin\_cfg を使用して定義します。

Module g_sf_power_profiles_v2_standby Power Profiles V2 Low Power Profile	
Name	g_sf_power_profiles_v2_standby
Callback (Low Power Exit Event N/A when using Deep Software Standby)	cb_pre_post_sby_fun
Low power entry pin configuration table	g_sby_pin_cfg
Low power exit pin configuration table	g_run_pin_cfg

図 42 PPv2 アプリケーションプロジェクトで使用する 2 個のピン設定テーブル

図 43 に、これら 2 個のピン設定の違いを示します。

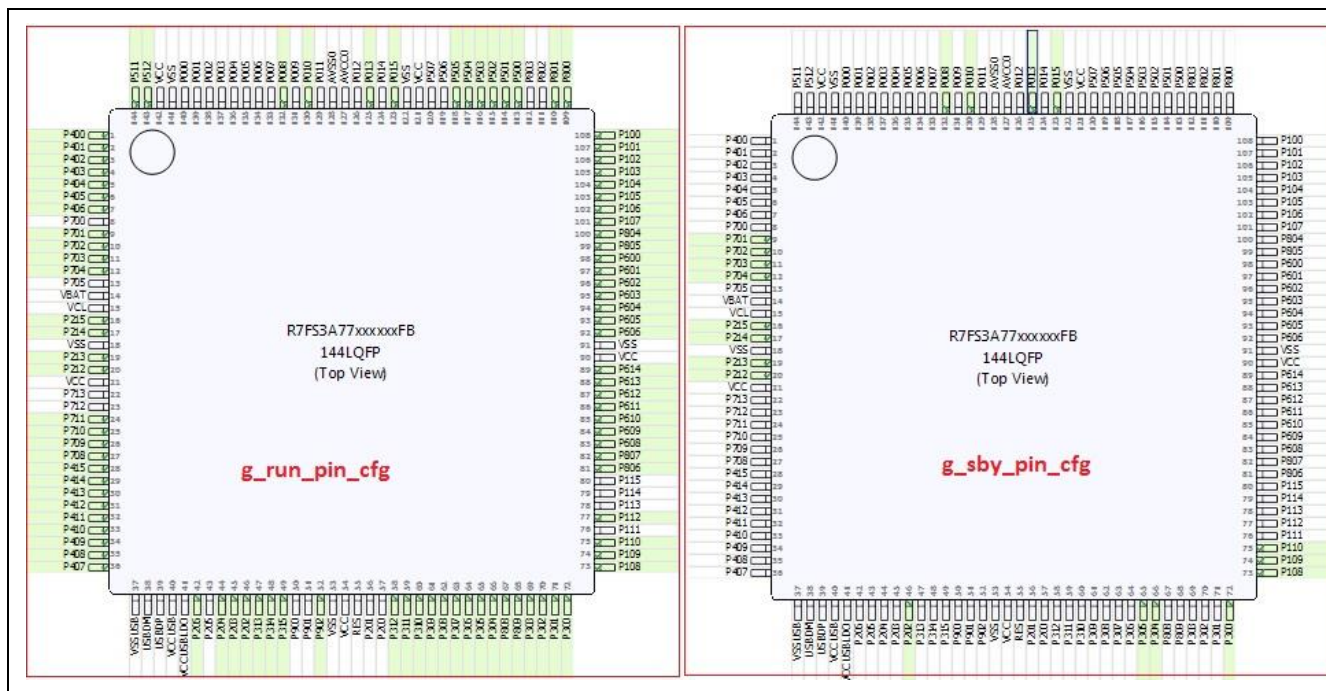


図 43 プロジェクトで使用する 2 個のピン設定の違い

前の図からわかるように、割り当て済みのピンは薄い緑色でマークしてあり、g\_run\_pin\_cfg を使用すると、g\_sby\_pin\_cfg を使用する場合より多くの IO 操作が可能です。

- IRQ3 は、ソフトウェアスタンバイ (Software Standby) モードからノーマル (Normal) モードへの遷移条件として有効にしています。

Select Standby Exit Sources	Select fields below:
IRQ0	Disabled
IRQ1	Disabled
IRQ2	Disabled
IRQ3	Enabled
IRQ4	Disabled

図 44 スタンバイ (Standby) モードとノーマル (Normal) モードの間での遷移の設定

## 7. PPv2 フレームワークモジュールのサンプルアプリケーションの実行 (Running the PPv2 Framework Module Application Example)

この章では、DK-S3A7 ボードにプロジェクトを書き込み、実行する方法を説明します。

### 7.1 ボードの電源投入 (Powering up the Board)

- ボードに接続するには、以下の手順に従ってください。
  - スイッチ S5 の“JTAG”と“PBs”をオンにします
  - 付属している USB ケーブルの Micro USB コネクタを DK-S3A7 v2.0 ボードの J15 コネクタ (DEBUG\_USB) に接続します。

注: このキットは、SEGGER J-Link® On-board (OB) をオンボード搭載しています。J-Link は、DK-S3A7 ボードのフルデバッグ機能とプログラミング機能を実現します。

- USB ケーブルのもう一方のコネクタを、開発ワークステーションの USB ポートに接続します。

- ボードへの書き込みは、以下の手順に従ってください。  
プロジェクトを e<sup>2</sup> studio にインポートしてビルド/実行する方法については、『Renesas Synergy Project Import Guide』(Renesas Synergy プロジェクトインポートガイド) を参照してください。このマニュアルは、アプリケーションプロジェクトに付属しています。

## 7.2 デモの確認 (Verifying the Demo)

このプロジェクトをテストするためのユーザインタフェースを次の図に示します。

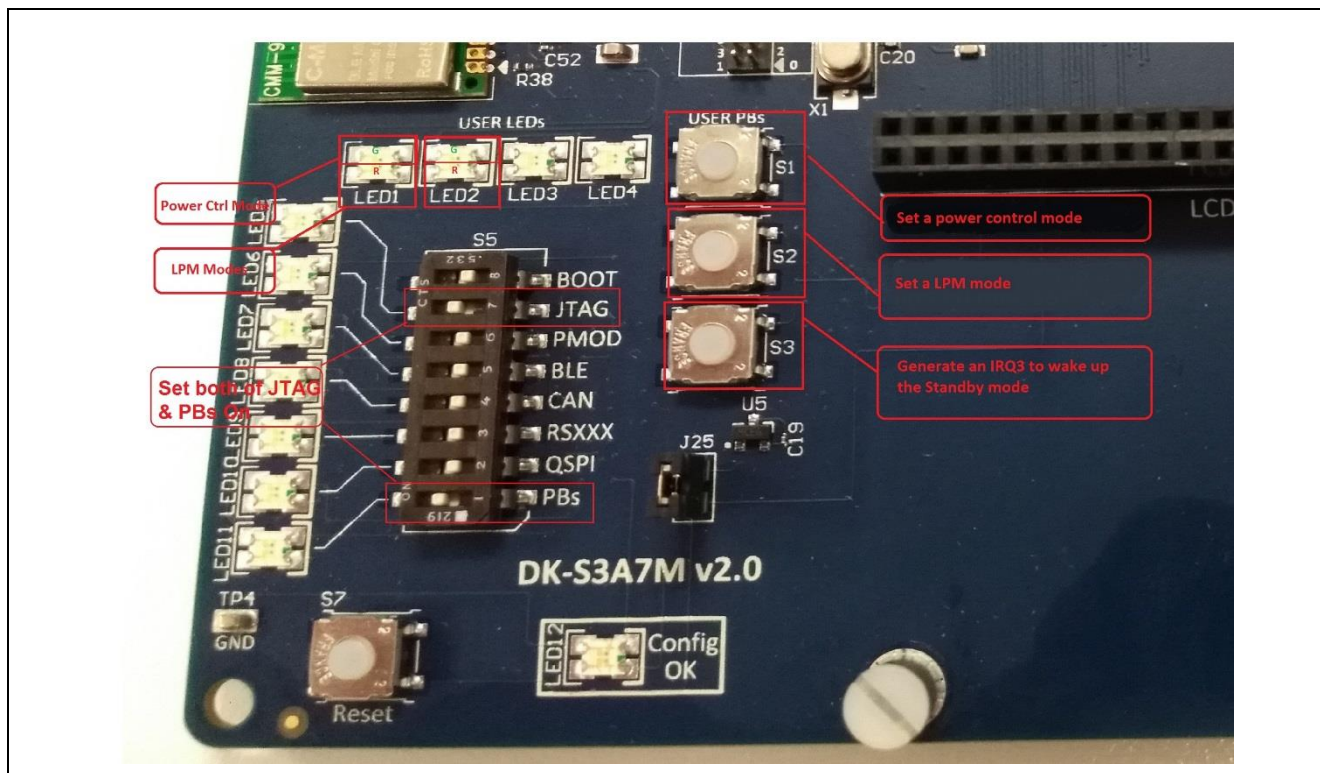


図 45 DK-S3A7 のデモのセットアップ

MCU がスタンバイ (Standby)/ディープスタンバイ (Deep Standby) モードに移行することをデバッガが妨げる可能性があります。そのため、S7 ボタンを押すか、J15 コネクタから USB 接続を取り外して再接続する方法によるフルのパワーオンリセットを実行して、ボードをリセットする必要があります。

S1、S2、S3 の各ボタンを使用して、選択した電力制御モードに移行します。モードは、LED によっても表示されます。

- さまざまな電力制御モードを見るには、S1 ボタンを押します。High-speed モード (デフォルト) と、Middle-speed モードが使用可能です。
- S1 ボタンを押したままにして、選択した電力制御モードに設定します。LED1\_G が 4 回点滅します。
- LPM モードのいずれかを選択するには、S2 ボタンを押します。デフォルトであるノーマル (Normal) モードと、ソフトウェアスタンバイ (Software Standby) モードが使用可能です。
- S3 ボタンを押すと、IRQ3 が生成され、MCU がソフトウェアスタンバイ (Software Standby) モードからウェイクアップします。
- 上記の操作を繰り返し、複数の電力制御モードと LPM モードの間での遷移を実行します。

## 8. 対象アプリケーションに合わせた PPv2 フレームワークモジュールのカスタマイズ (Customizing the PPv2 Framework Module for a Target Application)

- ユーザは簡単に、ポテンシオメータを、光センサやモーションセンサのような他のアナログ信号源に置き換えることができます。また、さまざまなウェイクアップ条件に対応するように ADC のウィンドウ (期間) 関数を再定義し、より複雑なアプリケーションを開発することもできます。
- ユーザは、別の CGC 設定と、スヌーズ (Snooze)、ソフトウェアスタンバイ (Software Standby)、ディープソフトウェアスタンバイ (Deep Software Standby) の各モードに対応する別の終了ソースを使用して、他の電力制御モードと LPM モードを追加することもできます。
- ユーザは、J22 でさまざまなモードの消費電力を測定することもできます。

注： それぞれの MCU 製品固有の『Synergy MCU グループユーザズマニュアル』 (Synergy Microcontroller Group User's Manual) の仕様に沿ったスタンバイ電流を実現するには、適切なボード改造を実施し、テスト環境をセットアップする必要があります。

## 9. 参考資料

IAR Systems. (n.d.). IAR Embedded Workbench, Getting Started with IAR Embedded Workbench for Renesas Synergy.

Keil. (2016). CMSIS-Pack (v 1.4.1). Retrieved from Usage and Description:  
<http://www.keil.com/pack/doc/CMSIS/Pack/html/pages.html>

Renesas Synergy™ Software Package, User's Manual.

USBX CDC ACM (Host).

USBX Device for CDCACM .

The Renesas e<sup>2</sup> studio ISDE 5.4.0.023.

Release Notes v1.2.0.

Renesas Synergy™ Project Import Guide

S3A7 User's Manual: Microcontrollers. Renesas Electronics.

USB 3.0 Promoter Group. (2014). USB 3.1 Specification.

## ホームページとサポート窓口

サポート: <https://synergygallery.renesas.com/support>

テクニカルサポート:

- アメリカ: <https://www.renesas.com/en-us/support/contact.html>
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

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。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018年7月13日	—	第 1.00 版 発行 英文版（資料番号 r20an0498eu0100-synergy-dk-s3a7-power-profiles-v2-framework、Rev1.00、発効日 2018 年 3 月 7 日）を 翻訳

## 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まれます。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、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、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)



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

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）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>