RX ファミリ

LPC モジュール Firmware Integration Technology

要旨

RX ファミリには、消費電力を低減できる様々な機能が備わっています。本 FIT モジュールで提供される API を使って、RX ファミリの CPU に対して、各動作電力制御モードへの遷移、または低消費電力状態への遷移を簡単に設定できます。また、本モジュールはスリープモード復帰クロックソース切り替えにも対応しています。

本アプリケーションノートは、プロジェクトへの追加方法や使用例などを含め、Firmware Integration Technology (FIT)を使用した LPC（消費電力低減機能）モジュールについて説明します。以降、本モジュールを LPC FIT モジュールと称します。

対象デバイス

- RX110、RX111、RX113 グループ
- RX130 グループ
- RX140 グループ
- RX230、RX231 グループ
- RX23E-B グループ
- RX23W グループ
- RX26T グループ
- RX260、RX261 グループ
- RX64M グループ
- RX65N グループ
- RX660 グループ
- RX66N グループ
- RX671 グループ
- RX71M グループ
- RX72M グループ
- RX72N グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

対象コンパイラ

- Renesas Electronics C/C++ Compiler Package for RX Family
- GCC for Renesas RX
- IAR C/C++ Compiler for Renesas RX

各コンパイラの動作確認内容については 6.1 動作確認環境を参照してください。

目次

1. 概要

1.1 LPC FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方については、「2.12 FIT モジュールの追加方法」を参照してください。

1.2 LPC FIT モジュールの概要

下表に RX CPU に対応した動作電力制御モードを示します。

表 1.1 動作電力制御モード

動作電力制御モード			
RX110、RX111、RX113、RX130、RX230、RX231、RX23E-B、RX23W	RX140、RX260、RX261	RX64M、RX65N、RX66N、RX671、RX71M、RX72M、RX72N	RX660, RX26T
高速動作モード 中速動作モード 低速動作モード	高速動作モード 中速動作モード 中速動作モード 2 低速動作モード	高速動作モード 低速動作モード 1 低速動作モード 2	制御モードなし

各動作電力制御モードには、VCC の要求と内部クロックの最大周波数に対して、上限と下限が設けられています。

RX110、RX111、RX113 の高速動作モードの場合、すべての内部クロックをシステムクロックの最大周波数である 32MHz（VCC が 3.6V～2.7V の場合）に設定できます。一方、中速動作モードでは、最大速度は 12MHz（VCC が 3.6V～2.4V の場合）に制限されます。低速動作モードでは、サブクロックのみがシステムクロックとして使用でき、すべての内部クロックの最大速度が 32.768kHz に制限されます。

電圧、周波数の要求は RX ファミリのグループによって異なります。要求の詳細は各グループのハードウェアマニュアルでご確認ください。

動作電力制御モードに加えて、CPU が非動作時の状態として、以下に示す低消費電力状態があります。

表 1.2 低消費電力状態

低消費電力状態			
RX110、RX111、RX113、RX130、RX230、RX231、RX23E-B、RX23W	RX140、RX260、RX261	RX64M、RX65N、RX660、RX66N、RX671、RX71M、RX72M、RX72N	RX26T
スリープモード ディープスリープモード ソフトウェアスタンバイモード	スリープモード ディープスリープモード ソフトウェアスタンバイモード スヌーズモード	スリープモード 全モジュールクロックストップモード 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	スリープモード 全モジュールクロックストップモード ソフトウェアスタンバイモード

これらのモードでは、周辺機能やクロックソースが一部制限される場合があります。

1.3 LPC FIT モジュールの使用方法

1.3.1 LPC FIT モジュールを C++プロジェクト内で使用する方法

C++プロジェクトでは、LPC FIT モジュールのインタフェースヘッダファイルを extern "C"の宣言に追加してください。

```
extern "C"
{
    #include "r_smc_entry.h"
    #include "r_lpc_rx_if.h"
}
```

1.4 API の概要

表 1.3 API 関数一覧に本モジュールに含まれる API 関数を示します。

表 1.3 API 関数一覧

関数	関数説明
R_LPC_OperatingModeSet ()	MCU に対して動作電力制御モードを設定します。動作電力制御モードについては表 1.1 動作電力制御モードを参照してください。
R_LPC_LowPowerModeConfigure ()	MCU に対して低消費電力状態を設定します。低消費電力状態については表 1.2 低消費電力状態を参照してください。
R_LPC_LowPowerModeActivate ()	R_LPC_LowPowerModeConfigure ()で設定した低消費電力状態への遷移を実行します。
R_LPC_SnoozeModeConfigure ()	スヌーズモードへの遷移条件とソフトウェアスタンバイモードへの復帰条件及びスヌーズモードの解除条件を設定します。
R_LPC_ReturnClockSwitch ()	スリープモード復帰クロックソース切り替えを設定します。
R_LPC_GetVersion ()	本モジュールのバージョン番号を返します。

1.5　状態遷移図

図 1.1、図 1.2、図 1.3、図 1.4 および図 1.5 に本モジュールの状態遷移図を示します。

以下の図は動作電力制御モードと低消費電力状態、またモード間の切り替え時に呼び出される LPC FIT モジュール API との関係を示したものです。

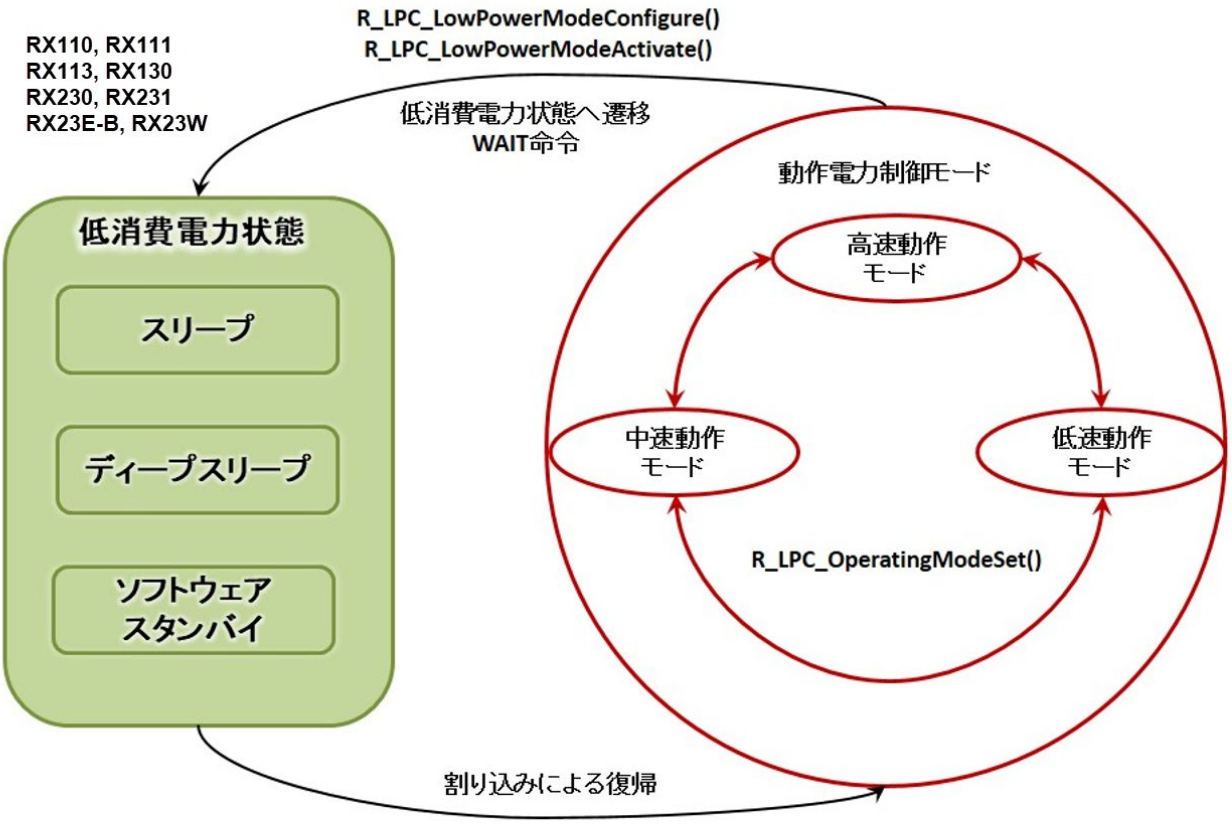

図 1.1　LPC FIT モジュールの状態遷移図 (RX110、RX111、RX113、RX130、RX230、RX231、RX23E-B、RX23W)

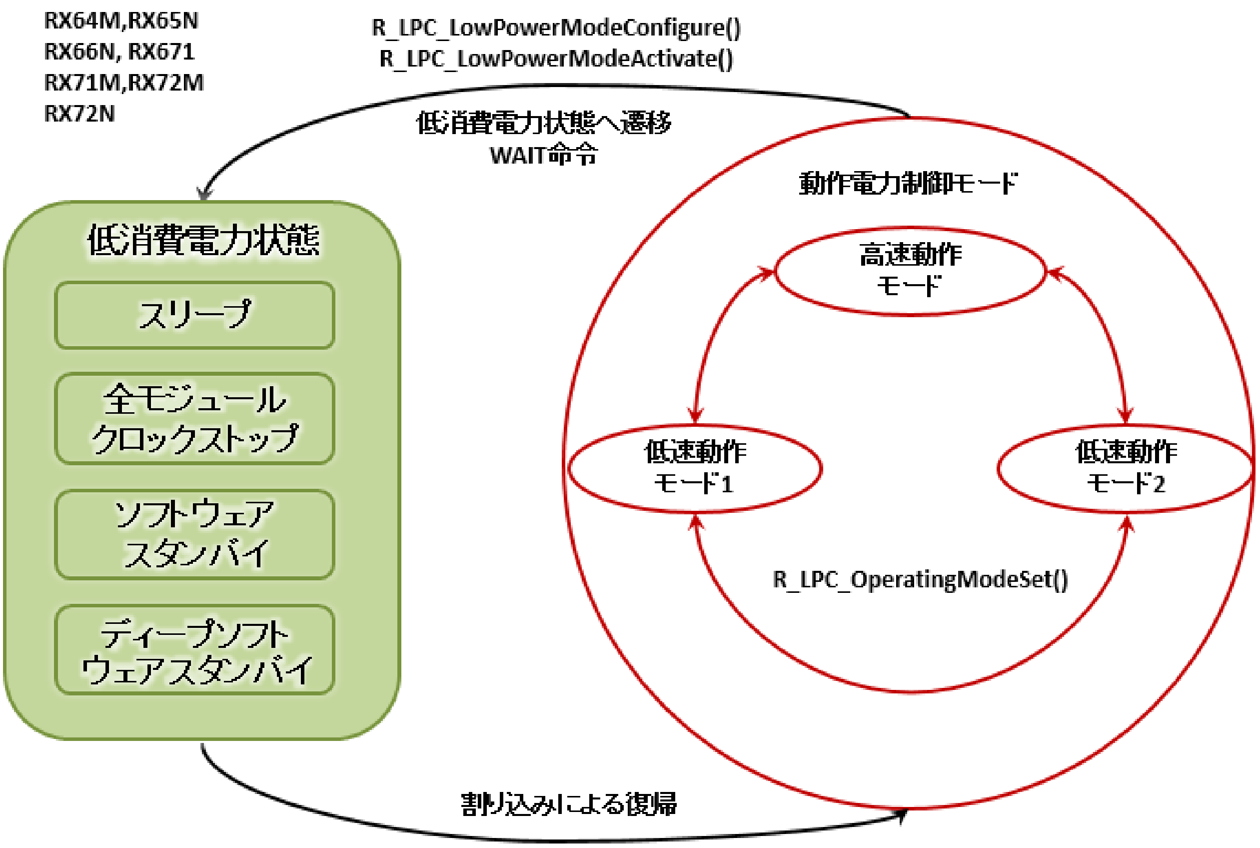

図 1.2　LPC FIT モジュールの状態遷移図（RX64M、RX65N、RX66N、RX671、RX71M、RX72M、RX72N）

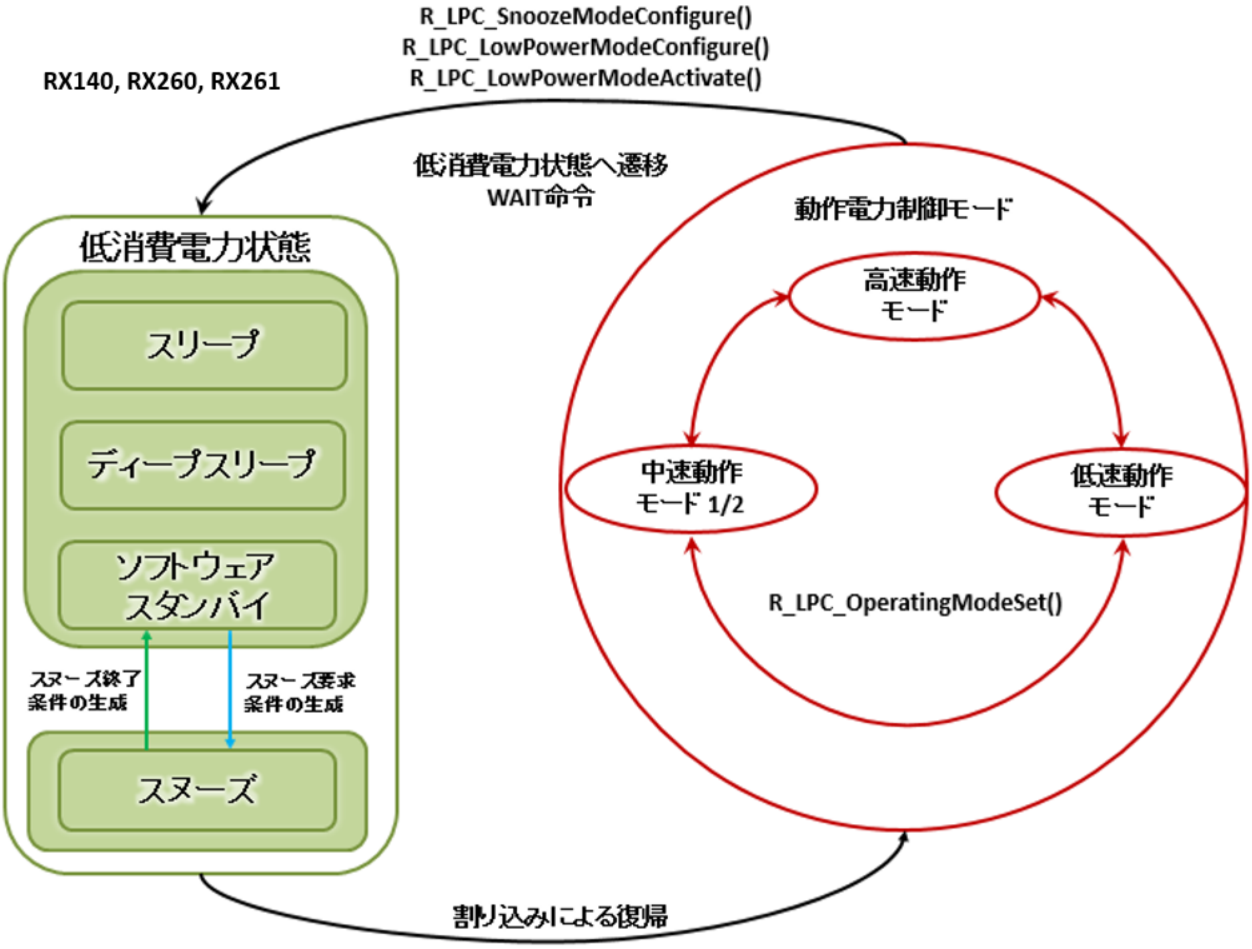

図 1.3　LPC FIT モジュールの状態遷移図（RX140、RX260、RX261）

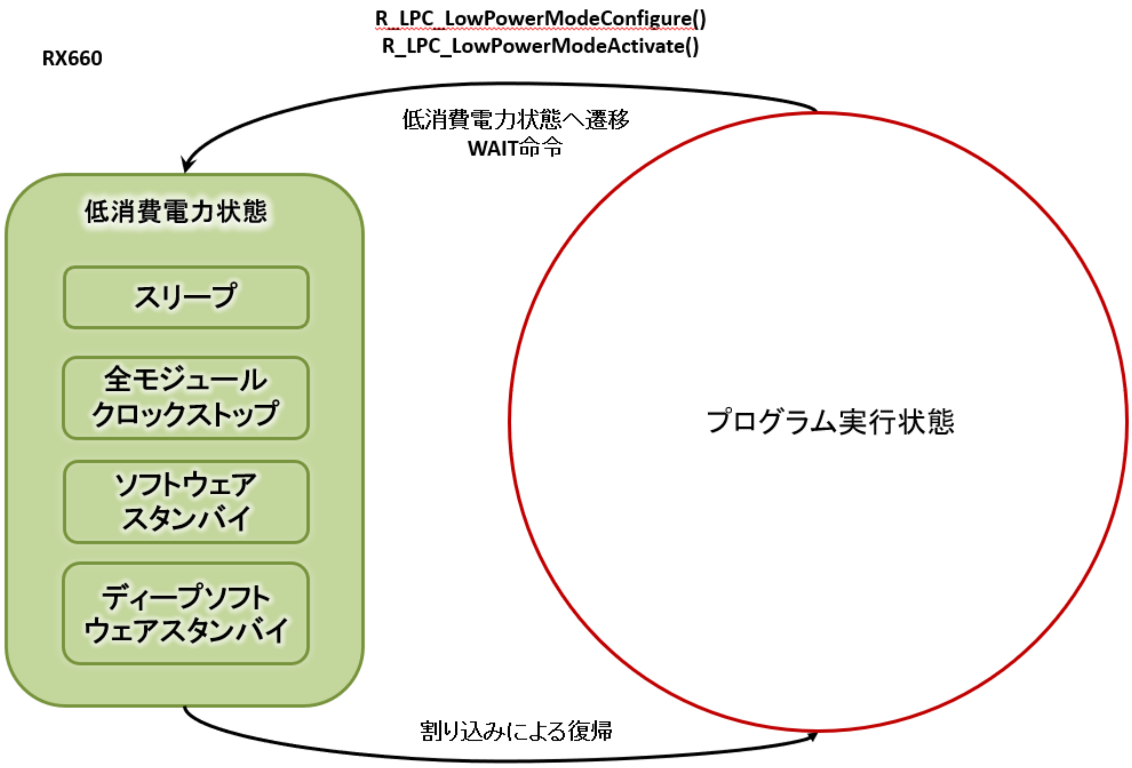

図 1.4　LPC FIT モジュールの状態遷移図 (RX660)

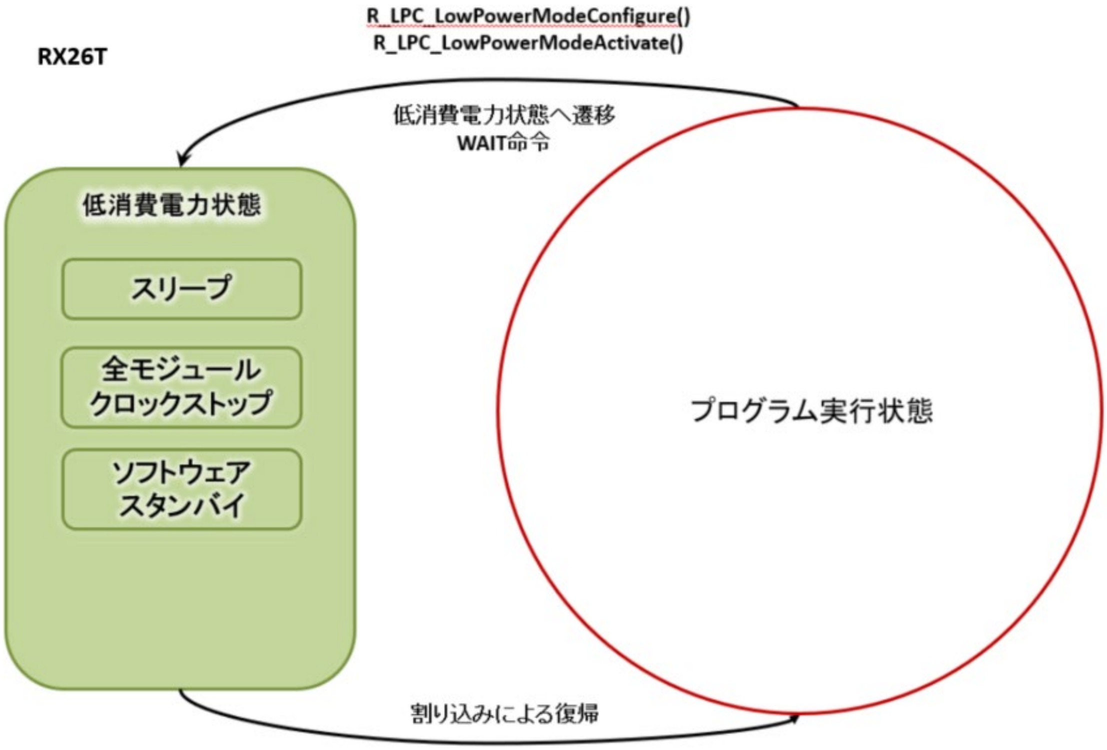

図 1.5　LPC FIT モジュールの状態遷移図 (RX26T)

2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- 1.概要の表 1.1 で示される動作電力制御モード
- 1.概要の表 1.2 で示される低消費電力状態

2.2 ソフトウェアの要求

このドライバは以下の FIT モジュールに依存しています。

- ボードサポートパッケージ（r_bsp）v5.20 以上

2.3 サポートされているツールチェーン

本 FIT モジュールは「6.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

2.4 使用する割り込みベクタ

R_LPC_SnoozeModeConfigure ()関数を実行すると、引数の設定に応じてスヌーズ解除割り込みが有効になります。

表 2.1 に本 FIT モジュールが使用する割り込みベクタを示します。

表 2.1 使用する割り込みベクタ一覧

デバイス	割り込みベクタ
RX140、 RX260、 RX261	スヌーズ解除割り込み（ベクタ番号: 81）

2.5 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r_lpc_rx_if.h に記載しています。

2.6 整数型

このドライバは ANSI C99 を使用しています。これらの型は stdint.h で定義されています。

2.7　コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、r_lpc_rx_config.h で行われます。

オプション名および設定値に関する説明を、下表に示します。

Configuration options in r_lpc _rx_config.h		
#define	**デフォルト値**	**説明**
LPC_CFG_PARAM_ CHECKING_ENABLE	1	本定義を 1 に設定するとパラメータチェック処理のコードを生成し、0 に設定すると生成しません。 本定義を BSP_CFG_PARAM_CHECKING_ENABLE に設定するとシステムのデフォルト設定が使用されます。

2.8 コードサイズ

本モジュールの ROM サイズ、RAM サイズ、最大使用スタックサイズを下表に示します。RX100 シリーズ、RX200 シリーズ、RX600 シリーズから代表して 1 デバイスずつ掲載しています。

ROM (コードおよび定数) と RAM (グローバルデータ) のサイズは、ビルド時の「2.7 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。

下表の値は下記条件で確認しています。

モジュールリビジョン: r_lpc_rx rev2.40

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.06.00

(統合開発環境のデフォルト設定に"-lang = c99"オプションを追加)

GCC for Renesas RX 8.3.0.202405

(統合開発環境のデフォルト設定に" -std=gnu99"オプションを追加)

IAR C/C++ Compiler for Renesas RX version 5.10.1

(統合開発環境のデフォルト設定)

コンフィグレーションオプション: デフォルト設定

ROM、RAM およびスタックのコードサイズ							
デバイス	分類	使用メモリ					
		Renesas Compiler		GCC		IAR Compiler	
		パラメータチェックあり	パラメータチェックなし	パラメータチェックあり	パラメータチェックなし	パラメータチェックあり	パラメータチェックなし
RX130	ROM (注 1)	7671 バイト	5763 バイト	9316 バイト	6308 バイト	5705 バイト	4698 バイト
	RAM (注 1)	3078 バイト	3038 バイト	2940 バイト	2940 バイト	1604 バイト	1564 バイト
	スタック (注 2)	4 バイト		-		52 バイト	
RX261	ROM (注 1)	7134 バイト	6374 バイト	8524 バイト	7348 バイト	6794 バイト	5618 バイト
	RAM (注 1)	6990 バイト	6950 バイト	6808 バイト	6768 バイト	4492 バイト	4452 バイト
	スタック (注 2)	4 バイト		-		52 バイト	
RX64M	ROM (注 1)	9175 バイト	8300 バイト	12340 バイト	10844 バイト	10171 バイト	8879 バイト
	RAM (注 1)	7520 バイト	7480 バイト	7420 バイト	7420 バイト	2204 バイト	2164 バイト
	スタック (注 2)	4 バイト		-		56 バイト	

注 1： BSP FIT モジュールの ROM/RAM サイズが含まれています。
注 2： BSP FIT モジュールのスタックサイズは含まれていません。

2.9 引数

API 関数の引数である列挙型を示します。この列挙型は、r_lpc_[device]_if.h (例：r_lpc_rx64m_if.h) に記載されています。

2.9.1 R_LPC_OperatingModeSet データ型

```
/* 動作電力制御モード（対象：RX110, RX111, RX113, RX130, RX230, RX231、RX23E-B、RX23W) */
typedef enum lpc_operating_mode
{
  LPC_OP_HIGH_SPEED = 0x00,
  LPC_OP_MIDDLE_SPEED = 0x02,
  LPC_OP_LOW_SPEED = 0x06,
  LPC_OP_INVALID_MODE
}lpc_operating_mode_t;

/* 動作電力制御モード（対象：RX64M, RX65N, RX66N, RX671, RX71M, RX72M, RX72N) */
typedef enum lpc_operating_mode
{
  LPC_OP_HIGH_SPEED = 0x00,
  LPC_OP_LOW_SPEED_1 = 0x06,
  LPC_OP_LOW_SPEED_2 = 0x07,
  LPC_OP_INVALID_MODE
}lpc_operating_mode_t;

/* 動作電力制御モード（対象：RX140, RX260, RX261) */
typedef enum lpc_operating_mode
{
  LPC_OP_HIGH_SPEED = 0x00,
  LPC_OP_MIDDLE_SPEED = 0x02,
  LPC_OP_MIDDLE_SPEED_2 = 0x04,
  LPC_OP_LOW_SPEED = 0x06,
  LPC_OP_INVALID_MODE
}lpc_operating_mode_t;
```

2.9.2 R_LPC_LowPowerModeConfigure データ型

```
  /* 低消費電力状態（対象：RX110, RX111, RX113, RX130, RX140, RX230, RX231、RX23E-B、RX23W, RX260, 
RX261）*/
  typedef enum lpc_low_power_mode
  {
    LPC_LP_SLEEP,
    LPC_LP_DEEP_SLEEP,
    LPC_LP_SW_STANDBY,
    LPC_LP_INVALID_MODE
  }lpc_low_power_mode_t;

  /* 低消費電力状態（対象：RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N）*/
  typedef enum lpc_low_power_mode
  {
    LPC_LP_SLEEP,
    LPC_LP_ALL_MODULE_STOP,
    LPC_LP_SW_STANDBY,
    LPC_LP_DEEP_SW_STANDBY,
    LPC_LP_INVALID_MODE
  }lpc_low_power_mode_t;

  /* 低消費電力状態（対象：RX26T）*/
  typedef enum lpc_low_power_mode
  {
    LPC_LP_SLEEP,
    LPC_LP_ALL_MODULE_STOP,
    LPC_LP_SW_STANDBY,
    LPC_LP_INVALID_MODE
  }lpc_low_power_mode_t;
```

2.9.3 R_LPC_ReturnClockSwitch データ型

```
  /* スリープモード復帰クロックソース切り替え（対象：RX110, RX111, RX113, RX130, RX140, RX230, RX231、
RX23E-B、RX23W, RX260, RX261）*/
  typedef enum lpc_clock_switch
  {
    LPC_LOCO = 0x00,
    LPC_HOCO = 0x01,
    LPC_MAIN_OSC = 0x02,
  }lpc_clock_switch_t;

  /* スリープモード復帰クロックソース切り替え（対象：RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, 
RX72N, RX26T）*/
  typedef enum lpc_clock_switch
  {
    LPC_HOCO = 0x01,
    LPC_MAIN_OSC = 0x02,
  }lpc_clock_switch_t;
```

2.9.4 R_LPC_SnoozeModeConfigure データ型

```
/* スヌーズモードへの遷移とソフトウェアスタンバイモードへの復帰及びスヌーズモード解除割り込みの条件（対象：RX140, RX260, RX261） */
typedef struct st_lpc_snooze_mode
{
  uint16_t snooze_operation;
  uint16_t snooze_release;
  lpc_snooze_interrupt_t snooze_interrupt;
} lpc_snooze_mode_t;

typedef struct st_lpc_snooze_interrupt
{
    uint8_t priority;
    lpc_snooze_callback_set_t pcallback;
} lpc_snooze_interrupt_t;
```

2.10 戻り値

API 関数の戻り値を示します。この列挙型は、API 関数のプロトタイプ宣言とともに r_lpc_rx_if.h に記載されています。

```
/* LPC API エラーコード */
typedef enum lpc_err
{
  LPC_SUCCESS,
  LPC_ERR_OSC_STOP_ENABLED,    // 発振停止検出機が許可の場合、ソフトウェアスタンバイモード
                               // には遷移できません。
  LPC_ERR_CLOCK_EXCEEDED,      // クロックの設定が動作電力制御モードの制限を違反しています。
  LPC_ERR_ILL_MAIN_CLK_FREQ,   // スリープからの復帰クロックを使用する場合の制限に違反しています。
  LPC_ERR_ILL_CLOCK_SOURCE,    // スリープモード復帰クロックソース切り替え有効時のクロック制限に
                               // 違反しています。
  LPC_ERR_P_E_MODE,            // フラッシュメモリの書き込み／消去中は、動作電力制御モードの
                               // 切り替えはできません。
  LPC_ERR_DEEP_SLEEP_STATUS,   // ディープスリープモードの制限に違反しています。
  LPC_ERR_ILL_PARAM,           // （未使用）。
  LPC_ERR_ILLEGAL              // 制限事項に違反しています。
}lpc_err_t;
```

2.11 コールバック関数

本モジュールでは、以下で説明するタイミングでユーザが設定したコールバック関数を呼び出します。コールバック関数はそれぞれの関数で設定します。

1.低消費電力状態に遷移する前に、コールバック関数を呼び出します。

詳細は「3 API 関数」の“R_LPC_LowPowerModeActivate ()”を参照してください。

2.スヌーズ解除割り込みが発生したタイミングで、コールバック関数を呼び出します。

詳細は「3 API 関数」の“R_LPC_SnoozeModeConfigure ()”を参照してください。

2.12 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、スマート・コンフィグレータを使用した(1)、(2)、(4)の追加方法を推奨しています。ただし、スマート・コンフィグレータは、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(3)の方法を使用してください。

(1) e^2 studio 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
e^2 studio のスマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: e^2 studio 編 (R20AN0451)」を参照してください。

(2) CS+上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
CS+上で、スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: CS+編 (R20AN0470)」を参照してください。

(3) CS+上で FIT モジュールを追加する場合
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。

(4) IAREW 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: IAREW 編 (R20AN0535)」を参照してください。

2.13 for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT_LOOP」で該当の処理を検索できます。

以下に記述例を示します。

```
while 文の例：
/* WAIT_LOOP */
While (0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

for 文の例：
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

do while 文の例：
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

3. API 関数

R_LPC_OperatingModeSet ()

動作電力制御モードを設定する関数です。動作電力制御モードについては表 1.1　動作電力制御モードを参照してください。

Format

```
lpc_err_t R_LPC_OperatingModeSet (
  lpc_operating_mode_t   e_mode    /* 動作電力制御モード */
)
```

Parameters

`lpc_operating_mode_t     e_mode`

動作電力制御モードを設定してください。（「2.9.1 R_LPC_OperatingModeSet」参照）

Return Values

LPC_SUCCESS	
LPC_ERR_CLOCK_EXCEEDED	*/*クロックの設定が動作電力制御モードの制限に違反しています。*/*
LPC_ERR_P_E_MODE	*/*フラッシュメモリの書き込み／消去中は、*/* */*動作電力制御モードの切り替えはできません。*/*

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

選択したモードと使用する MCU によって、内部クロックである ICLK、PCLKB、PCLKD、FCLK の最大速度が制限されます。例えば、RX110、RX111、RX113、RX130、RX140、RX230、RX231、RX23E-B、RX23W、RX260、RX261 で低速動作モードを選択した場合は、サブクロックのみをシステムクロックとして使用できます。クロックの制限についてはユーザーズマニュアル　ハードウェア編の表 11.3（RX64M、RX65N、RX66N、RX671、RX71M、RX72M、RX72N）、または表 11.4（RX110、RX111、RX113、RX130、RX140、RX230、RX231、RX23E-B、RX23W、RX260、RX261）を参照してください。本関数への引数が、選択された内部クロック周波数に対応していない場合はエラーが戻されます。クロックソースを低速の周波数から高速の周波数に切り替える場合は、動作電力制御モードが切り替える周波数に対応できるようにあらかじめ設定してください。周波数に対応できていない場合、CPU が正しく動作しません。

Example

```
lpc_err_t err;
err = R_LPC_OperatingModeSet (LPC_OP_MIDDLE_SPEED);
```

Special Notes:

動作電力制御モード、および内部クロックの周波数を切り替えるときは、内部クロックの周波数が動作電力制御モードによる制限に違反しないように設定する必要があります。動作電力制御モードおよび周波数を切り替える場合は以下の手順で行ってください。

1. 消費電流が小さいモードおよび低速クロック周波数から、消費電流が大きいモードおよび高速クロック周波数に切り替える場合：
   a. R_LPC_OperatingModeSet ()を使って、高速動作モードに遷移する。
   b. 内部クロック周波数を上げる。
2. 消費電流が大きいモードおよび高速クロック周波数から、消費電流が小さいモードおよび低速クロック周波数に切り替える場合：
   a. 内部クロック周波数を下げる。
   b. R_LPC_OperatingModeSet ()を使って、低速動作モードに遷移する。

R_LPC_LowPowerModeConfigure ()

WAIT 命令を実行したときの低消費電力状態を設定する関数です。低消費電力状態については表 1.2　低消費電力状態を参照してください。

Format

```
lpc_err_t R_LPC_LowPowerModeConfigure (
  lpc_low_power_mode_t   e_mode    /* 低消費電力状態 */
)
```

Parameters

`lpc_low_power_mode_t     e_mode`

低消費電力状態を設定してください。（「2.9.2 R_LPC_LowPowerModeConfigure」参照）

Return Values

LPC_SUCCESS

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

本関数は MCU に対して低消費電力状態（表 1.2 参照）を設定します。本関数では指定されたモードに従ってレジスタを設定します。低消費電力状態への遷移は実行しません。低消費電力状態への遷移は、R_LPC_LowPowerModeActivate ()関数を使って行います。

低消費電力状態に入ると CPU は停止しますが、周辺機能とクロックは低消費電力状態や設定により動作状態が異なります。詳細はユーザーズマニュアル　ハードウェア編を参照してください。

Special Notes:

なし

R_LPC_LowPowerModeActivate ()

R_LPC_LowPowerModeConfigure ()で設定した低消費電力状態への遷移を実行する関数です。

Format

```
lpc_err_t R_LPC_LowPowerModeActivate (
  void (*pcallback)(void* pdata)  /* コールバック関数 */
)
```

Parameters

`void (*pcallback) (void* pdata)`

低消費電力状態に遷移する前に呼び出す関数を設定して下さい。

Return Values

LPC_SUCCESS	
LPC_ERR_OSC_STOP_ENABLED	*/* 発振停止検出機能が許可の場合、ソフトウェアスタンバイ */*
	/ モードへは遷移できません。 */*
LPC_ERR_CLOCK_EXCEEDED	*/*クロックの設定が動作電力制御モードの制限に違反しています。*/*
LPC_ERR_ILL_CLOCK_SOURCE	*/* スリープモード復帰クロックソース切り替え有効時の */*
	/ クロック制限に違反しています。*/*
LPC_ERR_ILL_MAIN_CLK_FREQ	*/* スリープからの復帰クロックを使用する場合の制限に */*
	/ 違反しています。*/*
LPC_ERR_DEEP_SLEEP_STATUS	*/* ディープスリープモードの制限に違反しています。*/*
LPC_ERR_ILLEGAL	*/* 上記以外の制限事項に違反しています。 */*

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

本関数は wait()関数を呼び出して低消費電力状態への遷移を行います。

以下にユーザーズマニュアル　ハードウェア編に記載されている低消費電力状態への遷移手順を示します。

1. 割り込みを禁止します。
2. 低消費電力状態から復帰するための割り込み要因を設定します。
3. 最後の I/O レジスタへの書き込みが正常に完了したことを確認します。
4. 低消費電力状態に遷移するためのウェイト命令を実行します。ウェイト命令によって内部で割り込みが有効になります。

本関数は以下の手順を実行します。

1. 割り込みを禁止します。
2. 引数で設定したコールバック関数を呼び出します。コールバック関数では復帰割り込み要因を設定し、最後の I/O レジスタへの書き込みが正常に完了していることを確認してください。
   割り込み要因が既に設定されている場合は、引数に“FIT_NO_FUNC”を設定してください。
3. ウェイト命令を実行します。

本関数を実行した場合、以下の条件でエラーを返します。

表 3.1　スリープモード移行時の制限事項と戻り値

制限事項	違反時の戻り値	対象 CPU
スリープモード復帰クロック有効時、システムクロックはサブクロックを選択してください	LPC_ERR_ILL_CLOCK_SOURCE	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261
スリープモード復帰クロック有効時、システムクロックはサブクロック、または LOCO を選択してください	LPC_ERR_ILL_CLOCK_SOURCE	RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
スリープモードから復帰したときの動作電力制御モードが中速動作モードの場合、スリープモード復帰クロックに HOCO は選択しないでください	LPC_ERR_ILL_CLOCK_SOURCE	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261
スリープモードから復帰したときの動作電力制御モードが中速動作モード 2 の場合、スリープモード復帰クロックにメインロックは選択しないでください	LPC_ERR_ILL_CLOCK_SOURCE	RX260, RX261
PLL および PLL2 は VCC が 1.8V 以上の場合のみ使用可能です。	LPC_ERR_CLOCK_EXCEEDED	RX260, RX261
スリープモードから復帰したときの動作電力制御モードが中速動作モード、かつスリープモード復帰クロックにメインクロックを選択している場合は、内部クロックが中速動作モードの制限に違反しないようにしてください	LPC_ERR_ILL_MAIN_CLK_FREQ	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261
スリープモード復帰クロックに HOCO を選択している場合は、HOCO 電源を ON にしてください	LPC_ERR_ILL_CLOCK_SOURCE	RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N

表 3.2　全モジュールクロックストップモード移行時の制限事項と戻り値

制限事項	違反時の戻り値	対象 CPU
モジュールストップコントロールレジスタの設定が全モジュールクロックストップモードの条件を満たした状態にしてください	LPC_ERR_ILLEGAL	RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
フラッシュメモリ P/E モード中は全モジュールクロックストップに移行しないでください	LPC_ERR_ILLEGAL	RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N

表 3.3　ディープスリープモード移行時の制限事項と戻り値

制限事項	違反時の戻り値	対象 CPU
MSTPCRA.MSTPA28 ビットを“1”(DMAC/DTC モジュールストップ状態)にしてからディープスリープモードに移行してください	LPC_ERR_DEEP_SLEEP_STATUS	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261
フラッシュメモリ P/E モード中はディープスリープモードに移行しないでください	LPC_ERR_ILLEGAL	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261

表 3.4 ソフトウェアスタンバイモード移行時の制限事項と戻り値

制限事項	違反時の戻り値	対象 CPU
発振停止検出機能が有効な場合、ソフトウェアスタンバイモードには移行できません	LPC_ERR_OSC_STOP_ENABLED	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261, RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
DMAST.DMST ビットが“0”の状態でソフトウェアスタンバイモードに移行してください	LPC_ERR_ILLEGAL	RX230, RX231, RX23E-B, RX23W, RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
DTCST.DTCST ビットが“0”の状態でソフトウェアスタンバイモードに移行してください	LPC_ERR_ILLEGAL	RX110, RX111, RX113, RX130, RX140（注 1）, RX230, RX231, RX23E-B, RX23W, RX260（注 1）, RX261（注 1）, RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
フラッシュメモリ P/E モード中はソフトウェアスタンバイモードに移行しないでください	LPC_ERR_ILLEGAL	RX110, RX111, RX113, RX130, RX140, RX230, RX231, RX23E-B, RX23W, RX260, RX261, RX26T, RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N

注 1：RX140、RX260、RX261 においてスヌーズモードで DTC を使用する場合を除きます

表 3.5 ディープソフトウェアスタンバイモード移行時の制限事項と戻り値

制限事項	違反時の戻り値	対象 CPU
発振停止検出機能が有効な場合、ディープソフトウェアスタンバイモードには移行できません	LPC_ERR_OSC_STOP_ENABLED	RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
DMAST.DMST ビットが“0”の状態でディープソフトウェアスタンバイモードに移行してください	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
DTCST.DTCST ビットが“0”の状態でディープソフトウェアスタンバイモードに移行してください	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
フラッシュメモリ P/E モード中はディープソフトウェアスタンバイモードに移行しないでください	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671,RX71M, RX72M, RX72N
IWDT をオートスタートモードで使用している場合、OFS0.IWDTSLCSTP ビットが“0”(低消費電力モード遷移時 IWDT カウント継続)のときはディープソフトウェアスタンバイモードに移行できません	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671,RX71M, RX72M, RX72N
IWDT をレジスタスタートモードで使用している場合、IWDTCSTPR.SLCSTP ビットが“0” のときは、ディープソフトウェアスタンバイモードに移行できません	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671,RX71M, RX72M, RX72N
電圧検出回路において電圧監視 1 リセットの機能（LVD1CR0.LVD1RI=1）、または電圧監視 2 リセットの機能（LVD2CR0.LVD2RI=1）を選択している場合は、ディープソフトウェアスタンバイモードに移行できません	LPC_ERR_ILLEGAL	RX64M, RX65N, RX660, RX66N, RX671, RX71M, RX72M, RX72N
HOCO の FLL 機能有効(FLLCR1.FLLEN = 1) にした状態でソフトウェアスタンバイモードへ遷移しないでください	LPC_ERR_ILLEGAL	RX671, RX660

Example

```
  lpc_err_t err;

  err = R_LPC_LowPowerModeConfigure (LPC_LP_SLEEP);
err = R_LPC_LowPowerModeActivate (FIT_NO_FUNC);
```

Special Notes:

本 FIT モジュールでパラメータチェックを有効にしている場合、本関数は MCU が低消費電力状態に遷移する際に妨げとなる様々な条件をチェックします。開発フェーズではこの機能を有効にしておくことは重要ですが、リリース時はこの機能を無効にすることで、低消費電力状態への遷移の速度を上げることができます。

ディープスリープモード、またはソフトウェアスタンバイモードに遷移する前に、DTC で保留中の処理がないこと、DTC モジュールが停止していることを確認してください。

R_LPC_SnoozeModeConfigure ()

スヌーズモードへの遷移条件とソフトウェアスタンバイモードへの復帰条件及びスヌーズモードの解除条件を設定する関数です。

Format

```
lpc_err_t R_LPC_SnoozeModeConfigure (
lpc_snooze_mode_t * snooze_mode
)
```

Parameters

`lpc_snooze_mode_t * snooze_mode`

`snooze_mode->snooze_operation`
スヌーズモードへの遷移条件とソフトウェアスタンバイモードへの復帰条件を設定します。

表 3.6 パラメータ説明 (snooze_operation)をご参照ください。

`snooze_mode->snooze_release`
スヌーズモードの解除条件を設定します。

表 3.7 パラメータ説明(snooze_release) をご参照ください。

`snooze_mode->snooze_interrupt`
スヌーズ解除割り込みの割り込み優先レベルとコールバック関数を設定します。

表 3.8 パラメータ説明(snooze_interrupt)をご参照ください。

Return Values

LPC_SUCCESS

LPC_ERR_ILLEGAL

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

スヌーズモードへの遷移条件とソフトウェアスタンバイモードへの復帰条件及びスヌーズモードの解除割り込みの条件を設定します。

スヌーズ解除割り込みを使用してスヌーズモードを解除する場合、スヌーズ解除割り込みの条件の設定、および割り込み優先レベルとコールバック関数を設定します。スヌーズ解除割り込みを使用しない場合、スヌーズ解除割り込みの条件（snooze_release）に LPC_SNZ_RESET を設定します。LPC_SNZ_RESET を設定すると、スヌーズ解除割り込みは無効となり、割り込み優先レベルとコールバック関数の設定は無視されます。

スヌーズモードで動作可能な周辺機能については、ユーザーズマニュアル ハードウェア編を参照してください。パラメータの設定については表 3.6 ～表 3.8 を参照してください。本関数は、スヌーズモードが存在するデバイスでのみ使用することができます。

表 3.6 パラメータ説明 (snooze_operation)

マクロ定義	マクロ値	機能
LPC_SNZ_RESET	(0x0000)	設定情報を初期化する。
LPC_SNZ_SCI5_REQ_EXIT	(0x0002)	RXD5の立ち下がりを検出するとスヌーズモードに遷移し、受信したデータとSCI5.CDRレジスタの値が一致しなかった場合、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_SCI5_DTC_REQ_EXIT	(0x0003)	RXD5の立ち下がりを検出するとスヌーズモードに遷移し、受信したデータとSCI5.CDRレジスタの値が一致しなかった場合、または受信したデータをDTCで転送した後、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_LPT_REQ	(0x0008)	LPTコンペアマッチ1によりスヌーズモードに遷移し、ソフトウェアスタンバイモードには戻らず、スヌーズモードを保持。
LPC_SNZ_LPT_DTC_REQ_EXIT	(0x000c)	LPTコンペアマッチ1によりスヌーズモードに遷移し、LPTコンペアマッチ1起因のDTC転送が1回終了すると、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_S12AD_REQ	(0x0020)	LPTコンペアマッチ1によりスヌーズモードに遷移してA/D変換を実施し、ソフトウェアスタンバイモードには戻らず、スヌーズモードを保持。
LPC_SNZ_S12AD_DTC_REQ_EXIT	(0x0030)	LPTコンペアマッチ1によりスヌーズモードに遷移してA/D変換を実施し、A/Dコンバータの変換終了起因のDTC転送が1回終了すると、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_CTSU_REQ_EXIT	(0x0080)	LPTコンペアマッチ1によりスヌーズモードに遷移してCTSUの計測動作を開始し、CTSUからのスヌーズ終了要求でソフトウェアスタンバイモードに戻る。
LPC_SNZ_REMC_REQ_EXIT	(0x0200)	REMCによるクロックの開始要求に応じてスヌーズモードに遷移し、REMCによるクロックの終了要求が検出すると、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_S12AD_2SEL_REQ_EXIT	(0x0800)	LPTコンペアマッチ1によりスヌーズモードに遷移して、S12ADコンペア条件が満たされない場合、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_S12AD_3SEL_REQ_EXIT	(0x2000)	LPTコンペアマッチ1によりスヌーズモードに遷移して、S12ADコンペア条件が満たされると、ソフトウェアスタンバイモードに戻る。
LPC_SNZ_DTC_ENABLE	(0x8000)	スヌーズモードでの DTC 転送を許可。

表 3.7 パラメータ説明(snooze_release)

マクロ定義	マクロ値	機能
LPC_SNZ_RESET	(0x0000)	設定情報を初期化する。
LPC_SNZ_SCI5_ERROR_RELEASE	(0x0001)	スヌーズ解除割り込みにSCI5 受信エラー割り込み要因を選択する。
LPC_SNZ_SCI5_FULL_RELEASE	(0x0004)	スヌーズ解除割り込みにSCI5 受信データフル割り込み要因を選択する。
LPC_SNZ_SCI5_DTC_RELEASE	(0x0008)	スヌーズ解除割り込みにSCI5 受信データフルによるDTC 転送完了イベントを選択する。
LPC_SNZ_LPT_MATCH1_RELEASE	(0x0010)	スヌーズ解除割り込みにLPT コンペアマッチ1割り込み要因を選択する。
LPC_SNZ_LPT_DTC_RELEASE	(0x0020)	スヌーズ解除割り込みにLPT コンペアマッチ1によるDTC 転送完了イベントを選択する。
LPC_SNZ_S12AD_RELEASE	(0x0040)	スヌーズ解除割り込みにS12AD 変換終了割り込み要因を選択する。
LPC_SNZ_S12AD_DTC_RELEASE	(0x0080)	スヌーズ解除割り込みにS12AD 変換終了によるDTC 転送完了イベントを選択する。
LPC_SNZ_CTSU_RELEASE	(0x0100)	スヌーズ解除割り込みにCTSU 測定終了割り込み要因を選択する。
LPC_SNZ_S12AD2_RELEASE	(0x0400)	スヌーズ解除割込みにS12AD比較条件不成立イベントを割り込み要因に選択する。
LPC_SNZ_S12AD3_RELEASE	(0x1000)	スヌーズ解除割り込みにS12AD比較条件成立イベントを割り込み要因に選択する。

表 3.8 パラメータ説明(snooze_interrupt)

構造体メンバ	設定値	説明
priority	型：uint8_t 設定値：0～15	スヌーズ解除割り込みの割り込み優先レベルを設定します。スヌーズ解除割り込みを使用しない場合、設定値は無視されます。
pcallback	型：lpc_snooze_callback_set_t 設定値：関数ポインタ	スヌーズ解除割り込みのコールバック関数を設定します。スヌーズ解除割り込みでコールバック関数を使用しない場合、FIT_NO_FUNCを設定してください。 スヌーズ解除割り込みを使用しない場合、設定値は無視されます。

Example

```
lpc_err_t err;
lpc_snooze_mode_t snooze_mode;


/* スヌーズモードを使用しない場合、またはスヌーズモードの設定を初期化する場合 */
snooze_mode.snooze_operation = LPC_SNZ_RESET;
snooze_mode.snooze_release = LPC_SNZ_RESET;
snooze_mode.snooze_interrupt.priority = 0;
snooze_mode.snooze_interrupt.pcallback = FIT_NO_FUNC;


/* SCI5 の受信データ不一致でスヌーズモード終了(ソフトウェアスタンバイ遷移)、受信データフルでスヌーズモードを解除する場合 */
snooze_mode.snooze_operation = LPC_SNZ_SCI5_REQ_EXIT;
snooze_mode.snooze_release = LPC_SNZ_SCI5_FULL_RELEASE;
snooze_mode.snooze_interrupt.priority = 5;
snooze_mode.snooze_interrupt.pcallback = FIT_NO_FUNC; /* スヌーズ解除割り込みでコールバック関数を使用しない場合*/


/* SCI5 の受信データを DTC 転送したらスヌーズモード終了(ソフトウェアスタンバイ遷移)、DTC 転送終了割り込みでスヌーズモードを解除する場合 */
snooze_mode.snooze_operation = LPC_SNZ_SCI5_DTC_REQ_EXIT | LPC_SNZ_DTC_ENABLE;
snooze_mode.snooze_release = LPC_SNZ_SCI5_DTC_RELEASE;
snooze_mode.snooze_interrupt.priority = 5;
snooze_mode.snooze_interrupt.pcallback =
(lpc_snooze_callback_set_t)&lpc_callback; /* スヌーズ解除割り込みでコールバック関数を使用する場合*/


err = R_LPC_SnoozeModeConfigure (&snooze_mode);
err = R_LPC_LowPowerModeConfigure (LPC_LP_SW_STANDBY);
err = R_LPC_LowPowerModeActivate (FIT_NO_FUNC);
```

Special Notes

- スヌーズモードで各周辺機能を動作させるためには、ソフトウェアスタンバイに遷移する前に、使用する周辺機能ごとの条件を満たす必要があります。詳細はユーザーズマニュアル ハードウェア編 の消費電力低減機能を参照してください。

- スヌーズモードの解除割り込みの条件はスヌーズモード以外の動作モードでも有効です。スヌーズモード以外の動作モードでスヌーズ解除割り込みを発生させたくない場合、ソフトウェアスタンバイに遷移する手前で設定を行い、スヌーズモード解除後に本関数でスヌーズモードの設定を初期化してください。

R_LPC_ReturnClockSwitch ()

スリープモード復帰クロックソース切り替えを設定します。

Format

```
lpc_err_t R_LPC_ReturnClockSwitch (
  lpc_clock_switch_t     e_clock_source,     /* スリープモード復帰クロックソース */
  bool                   enable              /* スリープモード復帰クロックソース切替許可 */
)
```

Parameters

`lpc_clock_switch_t   e_clock_source`

スリープモードからの復帰時に使用するクロックソースを設定して下さい。

（「2.9.3 R_LPC_ReturnClockSwitch」参照）

`bool   enable`

スリープモードからの復帰時に、クロックソース切り替えを有効または無効にします。本パラメータが“1”の場合のみ、e_clock_source によって選択されたクロックソースが有効になります。

Return Values

LPC_SUCCESS

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

本関数は復帰クロックソース切り替え機能に関する設定を行います。この設定で、スリープモードからの復帰時、クロックソースの HOCO、LOCO、またはメインクロックへの切り替えを有効します。復帰クロックソース切り替え機能を有効にする場合、以下の項目を満たすようにしてください。

1. RX110、RX111、RX113 :
   - スリープモード遷移時、システムクロックにはサブクロックを選択してください。スリープモードからの復帰時、動作電力制御モードはスリープモード遷移前に選択されていた動作電力制御モードになります。
   - スリープモードからの復帰クロックソースにメインクロックが選択されている場合、スリープモードからの復帰時に中速動作モードに遷移します。スリープモードから復帰後の内部クロックが中速動作モードの制限に違反しないようにしてください。
2. RX130、RX140、RX230、RX231、RX23E-B、RX23W、RX260、RX261 :
   - スリープモード遷移時、システムクロックにはサブクロックを選択してください。スリープモードからの復帰時、動作電力制御モードはスリープモード遷移前に選択されていた動作電力制御モードになります。
   - 復帰モードが中速動作モードで、スリープモードからの復帰クロックソースにメインクロックが選択されている場合、スリープモードから復帰後の内部クロックが中速動作モードの制限に違反しないようにしてください。
3. RX26T、RX64M、RX65N、RX660、RX66N、RX671、RX71M、RX72M、RX72N:
   - スリープモード遷移時にクロックソースとして LOCO、またはサブクロックを選択してください。

Example

```
lpc_err_t err;

err = R_LPC_ReturnClockSwitch (LPC_MAIN_OSC, true);
```

Special Notes

なし

R_LPC_GetVersion ()

この関数は本モジュールのバージョン番号を返します。

Format

```
uint32_t R_LPC_GetVersion (void);
```

Parameters

なし

Return Values

バージョン番号

Properties

r_lpc_rx_if.h にプロトタイプ宣言されています。

Description

この関数は本モジュールのバージョン番号を返します。バージョン番号は符号化され、最上位の 2 バイトがメジャーバージョン番号を、最下位の 2 バイトがマイナーバージョン番号を示しています。

Example

```
uint32_t version;

version = R_LPC_GetVersion ();
```

Special Notes:

なし

4. 使用例

4.1 高電力状態への遷移例（RX100 シリーズの場合）

RX110、RX111、RX113、RX130 にはチップに供給する電力を制御する内部レギュレータが備えられています。高電力状態に正しく遷移するためには、前もって高めの電力を供給するようにレギュレータを設定する必要があります。

以下に低電力状態から高電力状態へ遷移するための処理と API 呼び出しの例を示します。この例では、処理開始時、システムクロックソースにサブクロックを、動作電力制御モードに低速動作モードが選択されているものとします。

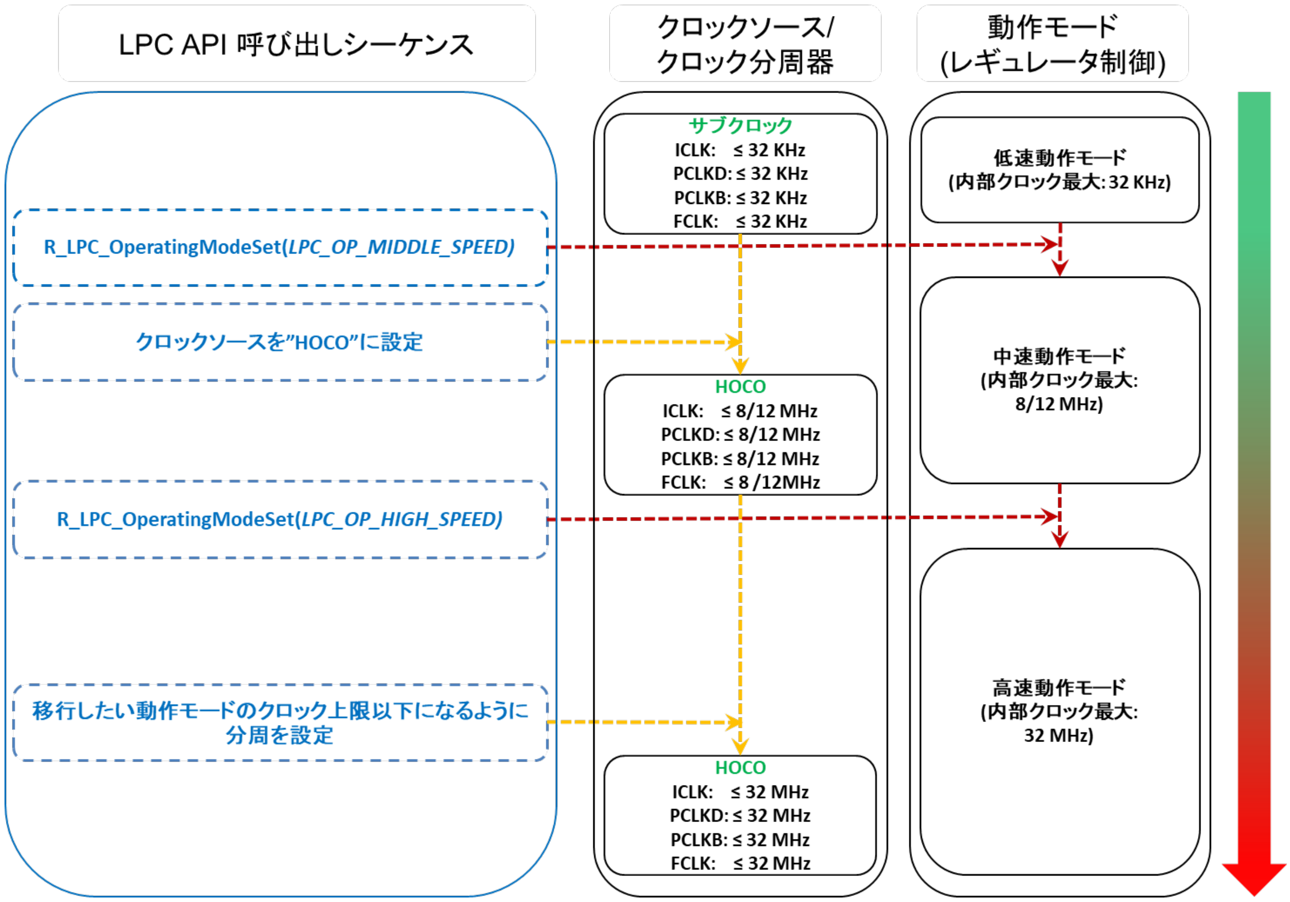

図 4.1 低電力状態から高電力状態への遷移例（RX110, RX111, RX113, RX130）

4.2 低電力状態への遷移例（RX100 シリーズの場合）

低電力状態へ遷移する時は、レギュレータがチップに供給する電力を落とす前に低電力状態に遷移しておく必要があります。

以下に高電力状態から低電力状態へ遷移するための処理と API 呼び出しの例を示します。この例では、処理開始時、システムクロックソースに HOCO を、動作電力制御モードに高速動作モードが選択されているものとします。

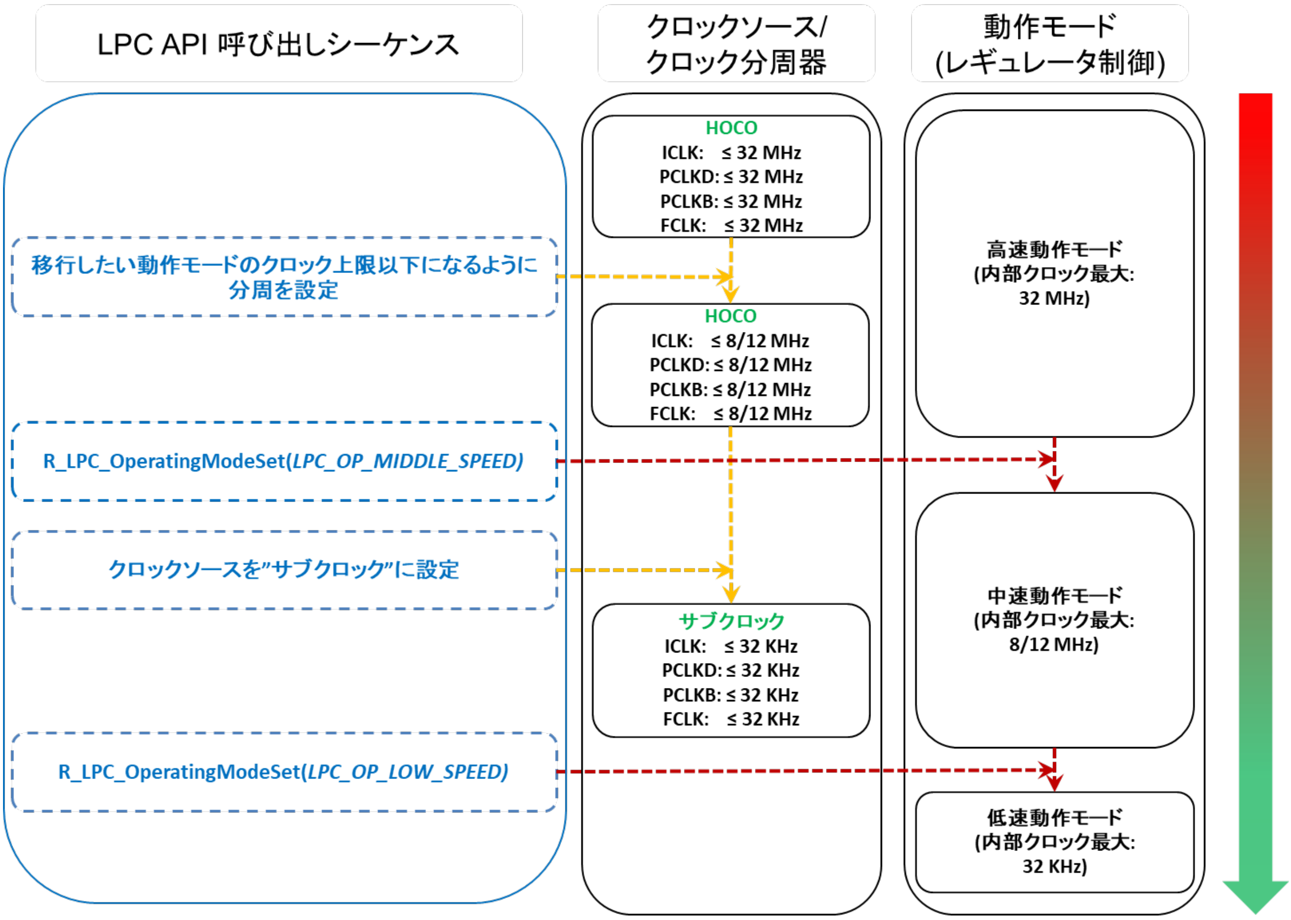

図 4.2 高電力状態から低電力状態への遷移例（RX110, RX111, RX113, RX130）

5. デモプロジェクト

デモプロジェクトには、FIT モジュールとそのモジュールが依存するモジュール（例：r_bsp）を使用するmain()関数が含まれます。本 FIT モジュールには以下のデモプロジェクトが含まれます。

5.1 lpc_demo_rskrx231, lpc_demo_rskrx64m, lpc_demo_rskrx65n, lpc_demo_rskrx231_gcc, lpc_demo_rskrx64m_gcc, lpc_demo_rskrx65n_gcc

これら LPC FIT モジュールのデモプログラムは Renesas RSKRX231、RSKRX64M、RSKRX65N デモボード用に設計されています。このプログラムは、R_LPC_LowPowerModeConfigure ()を使用して低消費電力モードを設定し、R_LPC_LowPowerModeActivate () を使用して低消費電力モードを有効にし、IRQ 割り込みでウェイクアップする方法を示しています。コードをコンパイルし、ボードにダウンロード、実行すると、初期化後に LED0 が点灯します。デフォルトでは、MCU は動作電力制御モード(高速モード)に遷移し、LED1 が点滅します。10 秒後、MCU は低消費電力モード(LPC_LP_SLEEP モード)に遷移し、LED1 が消灯します。ユーザが SW1 を押すと、MCU は低消費電力モードを終了し、LED1 が点滅します。

セットアップと実行

1. RSK ボードを PC に接続します (Renesas E1 Emulator を使用)。サンプルアプリケーションをビルドし、ボードにダウンロードして実行します。

2. ログをチェックして、プログラムが MCU を LPC_LP_SLEEP モードに設定していることを確認します。

3. ログと LED を確認して、プログラムが LPC_LP_SLEEP モードをアクティブにしていることを確認します。

4. SW1 を押して、IRQ 割り込みでウェイクアップします。

注: 手順 (3) では、Renesas Debug Virtual Console ウィンドウにメッセージが出力され、ユーザーに SW1 を押すように促されます。

サポートされるボード

RSKRX231、RXKRX64M、RSKRX65N

5.2 ワークスペースにデモを追加する

デモプロジェクトは、本アプリケーションノートで提供されるファイルの FITDemos サブディレクトリにあります。ワークスペースにデモプロジェクトを追加するには、「ファイル」 >> 「インポート」を選択し、「インポート」ダイアログから「一般」の「既存プロジェクトをワークスペースへ」を選択して「次へ」ボタンをクリックします。「インポート」ダイアログで「アーカイブ・ファイルの選択」ラジオボタンを選択し、「参照」ボタンをクリックして FITDemos サブディレクトリを開き、使用するデモの zip ファイルを選択して「終了」をクリックします。

5.3　デモのダウンロード方法

デモプロジェクトは、RX Driver Package には同梱されていません。デモプロジェクトを使用する場合は、個別に各 FIT モジュールをダウンロードする必要があります。「スマートブラウザ」の「アプリケーションノート」タブから、本アプリケーションノートを右クリックして「サンプル・コード（ダウンロード）」を選択することにより、ダウンロードできます。

6.　付録

6.1　動作確認環境

本 FIT モジュールの動作確認環境を以下に示します。

表 6.1　動作確認環境（Rev.1.41）

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.3.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev1.41

表 6.2　動作確認環境（Rev.1.42）

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.5.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev1.42
使用ボード	Renesas Solution Starter Kit for RX23W (型名：RTK5523Wxxxxxxxxxx)

表 6.3　動作確認環境（Rev.2.00）

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.6.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 4.8.4.201902 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.12.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.00
使用ボード	Renesas Solution Starter Kit for RX23W (型名：RTK5523Wxxxxxxxxxx)

表 6.4　動作確認環境 (Rev.2. 01)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.7.0 IAR Embedded Workbench for Renesas RX 4.14.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.02.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.201904 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.14.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.01
使用ボード	Renesas Starter Kit for RX130 (型名：RTK5005130xxxxxxxx) Renesas Starter Kit for RX231 (型名：R0K505231xxxxxx) Renesas Starter Kit+ for RX64M (型名：R0K50564Mxxxxxx) Renesas Starter Kit+ for RX65N (型名：RTK500565Nxxxxxxxx) Renesas Starter Kit+ for RX72M (型名：RTK5572Mxxxxxxxxxx) Renesas Starter Kit+ for RX72N (型名：RTK5572Nxxxxxxxxxx)

表 6.5　動作確認環境 (Rev.2. 02)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e2 studio 2021-01 IAR Embedded Workbench for Renesas RX 4.20.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.02.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202004 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.20.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.02
使用ボード	Renesas Starter Kit+ for RX671 (型名：RTK55671Exxxxxxxx)

表 6.6　動作確認環境（Rev.2. 03)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2021-07 IAR Embedded Workbench for Renesas RX 4.20.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.03.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202102 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.20.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.03
使用ボード	Target board for RX140 (型名：RTK5RX140xxxxxxxxx)

表 6.7　動作確認環境（Rev.2. 04)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2022-04 IAR Embedded Workbench for Renesas RX 4.20.3
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.04.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202104 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 4.20.3 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.04
使用ボード	Renesas Starter Kit for RX660 (型名：RTK556609HCxxxxxBJ)

表 6.8　動作確認環境（Rev.2. 10)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2022-07 IAR Embedded Workbench for Renesas RX 4.20.3
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.04.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202104 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 4.20.3 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.10
使用ボード	Renesas Starter Kit for RX231 (型名：R0K505231Sxxxxxxx) Renesas Starter Kit for RX64M (型名：R0K50564Mxxxxxx) Renesas Starter Kit+ for RX65N (型名：RTK5005651Cxxxxxxx)

表 6.9　動作確認環境（Rev.2. 20)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2022-10 IAR Embedded Workbench for Renesas RX 4.20.3
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.05.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202204 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 4.20.3 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.20
使用ボード	Renesas Flexible Motor Control Kit for RX26T(型名：RTK0EMXE70S00020BJ)

表 6.10　動作確認環境 (Rev.2. 30)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2023-04 IAR Embedded Workbench for Renesas RX 4.20.3
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.05.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202204 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 4.20.3 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.30
使用ボード	Renesas Solution Starter Kit for RX23E-B（型名：RTK0ES1001C00001BJ）

表 6.11　動作確認環境 (Rev.2. 40)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e² studio 2024-07 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.06.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202405 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 5.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.40
使用ボード	Evaluation Kit for RX261（型名：RTK5EK2610S00011BJ）

表 6.12　動作確認環境 (Rev.2. 50)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e2 studio 2024-07 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.06.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202405 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 5.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.50
使用ボード	Evaluation Kit for RX261（型名：RTK5EK2610S00011BJ）

表 6.13　動作確認環境 (Rev.2. 51)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e2 studio 2025-01 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.07.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.3.0.202411 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99 リンクオプション：「Optimize size (サイズ最適化) (-Os)」を使用する場合、統合開発環境のデフォルト設定に以下のオプションを追加 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 5.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.51
使用ボード	-

6.2 トラブルシューティング

(1) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「Could not open source file "platform.h"」エラーが発生します。

A : FIT モジュールがプロジェクトに正しく追加されていない可能性があります。プロジェクトへの追加方法をご確認ください。

- CS+を使用している場合
  アプリケーションノート RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」
- e^2 studio を使用している場合
  アプリケーションノート RX ファミリ e^2 studio に組み込む方法 Firmware Integration Technology (R01AN1723)」

また、本 FIT モジュールを使用する場合、ボードサポートパッケージ FIT モジュール(BSP モジュール)もプロジェクトに追加する必要があります。BSP モジュールの追加方法は、アプリケーションノート「ボードサポートパッケージモジュール(R01AN1685)」を参照してください。

(2) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「This MCU is not supported by the current r_lpc_rx module.」エラーが発生します。

A : 追加した FIT モジュールがユーザプロジェクトのターゲットデバイスに対応していない可能性があります。追加した FIT モジュールの対象デバイスを確認してください。

改訂記録

Rev.	発行日	改訂内容 ページ	ポイント
1.40	2016.10.01	—	初版発行
1.41	2019.04.01	—	機能関連 Smart Configurator での GUI によるコンフィグオプション設定機能に対応 ■内容 GUI によるコンフィグオプション設定機能に対応するため、設定ファイルを追加。
		1	・要旨修正 ・「関連ドキュメント」に Renesas e2 studio スマート・コンフィグレータ ユーザーガイドを追加
		4	「1.1 LPC FIT モジュールとは」タイトルを変更
		5	「1.3 API の概要」を追加
		6	「1.4 状態遷移図」タイトルを変更
		8	「ハードウェアリソースの要求」を削除 「制限事項」を削除 「2.3 サポートされているツールチェーン」を変更 「2.4 使用する割り込みベクタ」を追加 「2.5 ヘッダファイル」を変更 「2.6 整数型」を変更 「2.7 コンパイル時の設定」を変更
		9	「2.8 コードサイズ」を変更
		10	「2.9 引数」を変更
		12	「2.11 コールバック関数」を追加
		13	「2.12 FIT モジュールの追加方法」を変更
		14	「2.13 for 文、while 文、do while 文について」を追加
		31	「5.5 デモのダウンロード方法」を追加
		32	「6.1 動作確認環境」を追加
			「6.2 トラブルシューティング」を追加
1.42	2019.07.01	—	RX23W のサポートを追加
		23	R_LPC_ReturnClockSwitch ()の説明に RX230 を追加
2.00	2019.11.14	—	RX210 を対象デバイスから除外
		1	関連ドキュメントを削除
		8 27	以下のコンパイラをサポート - GCC for Renesas RX - IAR C/C++ Compiler for Renesas RX
		13	「WAIT_LOOP」を記述している対象デバイスの章を削除
		24	R_LPC_GetVersion ()の Format の誤記を修正
		27	「デモプロジェクト」の章を削除

Rev.	発行日	改訂内容	
		ページ	ポイント
2.01	2020.06.10	—	RX65N、RX66N、RX72M、RX72N のサポートを追加
		1	対象コンパイラを追加
		7	「2.2 ソフトウェアの要求」　依存する r_bsp モジュールのリビジョンを追加
		8	「2.8 コードサイズ」を変更
		12	「2.12 FIT モジュールの追加方法」を変更
		14-24	「3. API 関数」の、各 API の Reentrant を削除
		28	表 5.4　動作確認環境 (Rev.2.01)を追加
		プログラム	以下を修正しました。 ■対象デバイス 全デバイス ■内容 複数の周辺機能から同時にアクセスされる可能性があるレジスタがあり、そのレジスタへの書き込みのアトミック性が確保できるように処理を変更。
2.02	2021.04.02	—	RX671 のサポート追加
2.03	2021.07.31	—	RX140 のサポートを追加する
		1	対象デバイスに RX140 を追加
		3	「1.2　LPC FIT モジュールの概要」に、RX140 の動作電力制御モードを追加
		3	「1.2　LPC FIT モジュールの概要」に、RX140 の低消費電力状態を追加
		4	API 関数（R_LPC_ SnoozeModeConfigure）を追加
		7	「1.4　状態遷移図」RX140 の LPC FIT モジュールの状態遷移図を追加
		8	「2.2 ソフトウェアの要求」依存する r_bsp モジュールのリビジョンを変更
		8	「2.4 使用する割り込みベクタ」スヌーズの割り込み番号を追加
		9	「2.8　コードサイズ」を変更
		10	「2.9.1 R_LPC_OperatingModeSet」に、RX140 の動作電力制御モードの引数を追加
		11	「2.9.4　R_LPC_ SnoozeModeConfigure」に、RX140 のスヌーズモードの引数を追加
		12	「2.11 コールバック関数」コールバック関数の説明を追加しました
		23	関数 R_LPC_ SnoozeModeConfigure を追加
		33	表 5.6　動作確認環境 (Rev.2.03)を追加

Rev.	発行日	改訂内容	
		ページ	ポイント
2.04	2021.12.31	—	RX660 のサポートを追加する
		1	対象デバイスに RX660 を追加
		3	「1.2　LPC FIT モジュールの概要」に、RX660 の低消費電力状態を追加
		3	「1.2　LPC FIT モジュールの概要」に、RX660 の動作電力制御モードを追加
		4	「LPC FIT モジュールの使用方法」を追加
		5	「図 1.3」状態遷移図を追加
		8	「1.4　状態遷移図」RX660 の LPC FIT モジュールの状態遷移を追加
		10	「2.8　コードサイズ」を変更
		12	「2.9.2 R_LPC_LowPowerModeConfigure」に、RX660 の低消費電力状態の引数を追加
		20	「表 3.0.14　スリープモード移行時の制限事項と戻り値」に、RX660 のスリープモード移行時の制限事項と戻り値を追加
		20	「表 3.2　全モジュールクロックストップモード移行時の制限事項と戻り値」に、RX660 の全モジュールクロックストップモード移行時の制限事項と戻り値を追加
		21	「表 3.4　ソフトウェアスタンバイモード移行時の制限事項と戻り値」に、RX660 のソフトウェアスタンバイモード移行時の制限事項と戻り値を追加
		22	「表 3.5　ディープソフトウェアスタンバイモード移行時の制限事項と戻り値」に、RX660 と RX671 のディープソフトウェアスタンバイモード移行時の制限事項と戻り値を追加
		28-29	「3.5　R_LPC_ReturnClockSwitch ()」に、RX660 の説明を追加
		35	表 5.7　動作確認環境 (Rev.2.04)を追加
2.10	2022.06.28	33	「5. デモプロジェクト」を追加。
		33	「5. デモプロジェクト」に RSK RX231、RSK RX64M、RSKRX65N を追加。
		38	「6.1 動作確認環境」： Rev.2.10 に対応する表を追加。
		プログラム	デモプロジェクトの追加

Rev.	発行日	改訂内容	
		ページ	ポイント
2.20	2022.08.15	-	RX26T のサポートを追加する
		1	対象デバイスに RX26T を追加
		4	「1.2　LPC FIT モジュールの概要」に、RX26T の低消費電力状態を追加
		4	「1.1　LPC FIT モジュールの概要」に、RX26T の動作電力制御モードを追加
		6	「図 1.5」状態遷移図を追加
		10	「1.5　状態遷移図」RX26T の LPC FIT モジュールの状態遷移図を追加
		12	「2.8　コードサイズ」を変更
		15	「2.9.2 R_LPC_LowPowerModeConfigure」に、RX26T の低消費電力状態の引数を追加
		15	「2.9.3 R_LPC_ReturnClockSwitch」に、RX26T の低消費電力状態の引数を追加
		23	「表 3.0.15　スリープモード移行時の制限事項と戻り値」に、RX26T のスリープモード移行時の制限事項と戻り値を追加
		23	「表 3.2　全モジュールクロックストップモード移行時の制限事項と戻り値」に、RX26T の全モジュールクロックストップモード移行時の制限事項と戻り値を追加
		24	「表 3.4　ソフトウェアスタンバイモード移行時の制限事項と戻り値」に、RX26T のソフトウェアスタンバイモード移行時の制限事項と戻り値を追加
		30	「R_LPC_ReturnClockSwitch ()」に、RX26T の説明を追加
		40	「6.9 動作確認環境」： Rev.2.20 に対応する表を追加。
		プログラム	RX26T のサポートを追加。

Rev.	発行日	改訂内容	
		ページ	ポイント
2.30	2023.05.29	-	RX23E-B のサポートを追加する
		1	対象デバイスに RX23E-B を追加
		4	「1.2　LPC FIT モジュールの概要」に、RX23E-B の動作電力制御モードを追加
		4	「1.2　LPC FIT モジュールの概要」に、RX23E-B の低消費電力状態を追加
		6	図 1.1 に RX23E-B を追加。
		12	「2.8　コードサイズ」を変更
		13	「2.9.1 R_LPC_OperatingModeSet Data」に、RX23E-B の低消費電力状態の引数を追加
		14	「2.9.2 R_LPC_LowPowerModeConfigure」に、RX23E-B の低消費電力状態の引数を追加
		14	「2.9.3 R_LPC_ReturnClockSwitch」に、RX23E-B の低消費電力状態の引数を追加
		16	「2.12 FIT モジュールの追加方法」から FIT configurator の記述を削除
		18	「R_LPC_OperatingModeSet ()」に、RX23E-B の説明を追加
		22	「表 3.1　全モジュールクロックストップモード移行時の制限事項と戻り値」に、RX23E-B の全モジュールクロックストップモード移行時の制限事項と戻り値を追加
		22	「表 3.3　ソフトウェアスタンバイモード移行時の制限事項と戻り値」に、RX23E-B のソフトウェアスタンバイモード移行時の制限事項と戻り値を追加
		23	「表 3.4　ソフトウェアスタンバイモード移行時の制限事項と戻り値」に、RX23E-B のソフトウェアスタンバイモード移行時の制限事項と戻り値を追加
		29	「R_LPC_ReturnClockSwitch ()」に、RX23E-B の説明を追加
		40	「6.10 動作確認環境」： Rev.2.30 に対応する表を追加。
		プログラム	RX23E-B のサポートを追加。

Rev.	発行日	改訂内容	
		ページ	ポイント
2.40	2024.06.28	1、4、8、11、14-16、23、24	RX260、RX261 のサポートを追加。
		13	「2.8 コードサイズ」を変更。
		19	「R_LPC_OperatingModeSet ()」に、RX260、RX261 の説明を追加。
		21	R_LPC_LowPowerModeActivate ()関数に「LPC_ERR_CLOCK_EXCEEDED」の戻り値を追加。
		22	RX260,RX261 の中速動作モード 2 でメインクロックが選択できない制限を追加。
		22	PLL および PLL2 は VCC が 1.8V 以上の場合のみ使用できる制限を追加。
		27	「表 3.6 パラメータ説明 (snooze_operation)」: LPC_SNZ_REMC_REQ_EXIT、 LPC_SNZ_S12AD_2SEL_REQ_EXIT および LPC_SNZ_S12AD_3SEL_REQ_EXIT マクロを追加。
		28	「表 3.7 パラメータ説明 (snooze_release)」: LPC_SNZ_S12AD2_RELEASE および LPC_SNZ_S12AD3_RELEASE マクロを追加。
		30	「R_LPC_ReturnClockSwitch ()」に、RX260、RX261 の説明を追加。
		41	「6.11 動作確認環境」 : Rev.2.40 に対応する表を追加。
		プログラム	RX260、RX261 のサポートを追加。 すべてのデバイスの重複マクロ（LPC_CLOCK_ACTIVE と LPC_CLOCK_INACTIVE）を削除。 R_LPC_OperatingModeSet ()、R_LPC_ReturnClockSwitch ()、R_LPC_LowPowerModeActivate ()関数のコメントを修正。
2.50	2024.11.01	42	「6.12 動作確認環境」 : Rev.2.50 に対応する表を追加。
		プログラム	RX260、RX261 の中速動作モード、中速動作モード 2 の PLL、PLL2 とメインクロックの制限を lpc_freq_range_check()、lpc_lowpower_activate_check()関数に変更。
2.51	2025.03.15	42	「6.13 動作確認環境」 : Rev.2.51 に対応する表を追加。
		プログラム	FIT モジュールの免責事項と著作権を更新。

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、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。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も、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

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。

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/