

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
- RX210 グループ
- RX230、RX231 グループ
- RX64M グループ
- RX71M 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。

関連ドキュメント

- Firmware Integration Technology ユーザーズマニュアル(R01AN1833)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- e²studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)
- Renesas e² studio スマート・コンフィグレータ ユーザーガイド(R20AN0451)

目次

1. 概要	4
1.1 LPC FIT モジュールとは	4
1.2 LPC FIT モジュールの概要	4
1.3 API の概要	5
1.4 状態遷移図	6
2. API 情報	8
2.1 ハードウェアの要求	8
2.2 ソフトウェアの要求	8
2.3 サポートされているツールチェーン	8
2.4 使用する割り込みベクタ	8
2.5 ヘッダファイル	8
2.6 整数型	8
2.7 コンパイル時の設定	8
2.8 コードサイズ	9
2.9 引数	10
2.9.1 R_LPC_OperatingModeSet	10
2.9.2 R_LPC_LowPowerModeConfigure	11
2.9.3 R_LPC_ReturnClockSwitch	11
2.10 戻り値	12
2.11 コールバック関数	12
2.12 FIT モジュールの追加方法	13
2.13 for 文、while 文、do while 文について	14
3. API 関数	15
R_LPC_OperatingModeSet ()	15
R_LPC_LowPowerModeConfigure ()	17
R_LPC_LowPowerModeActivate ()	18
R_LPC_ReturnClockSwitch ()	23
R_LPC_GetVersion ()	25
4. 使用例	26
4.1 高電力状態への遷移例 (RX100 シリーズの場合)	26
4.2 低電力状態への遷移例 (RX100 シリーズの場合)	27
4.3 高電力状態への遷移例 (RX210 の場合)	28
4.4 低電力状態への遷移例 (RX210 の場合)	29
5. デモプロジェクト	30
5.1 lpc_demo_rskrx113	30
5.2 lpc_demo_rskrx231	30
5.3 lpc_demo_rskrx71M	31
5.4 ワークスペースにデモを追加する	31
5.5 デモのダウンロード方法	31
6. 付録	32

6.1 動作確認環境 32

6.2 トラブルシューティング 32

改訂記録 33

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	RX210	RX64M、RX71M
1. 高速動作モード 2. 中速動作モード 3. 4. 低速動作モード 5.	1. 高速動作モード 2. 中速動作モード 1A 3. 中速動作モード 1B 4. 低速動作モード 1 5. 低速動作モード 2	1. 高速動作モード 2. 3. 4. 低速動作モード 1 5. 低速動作モード 2

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	RX210	RX64M、RX71M
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 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 ディープ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 API の概要

表 1.3 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_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。

1.4 状態遷移図

図 1.1～図 1.3 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。

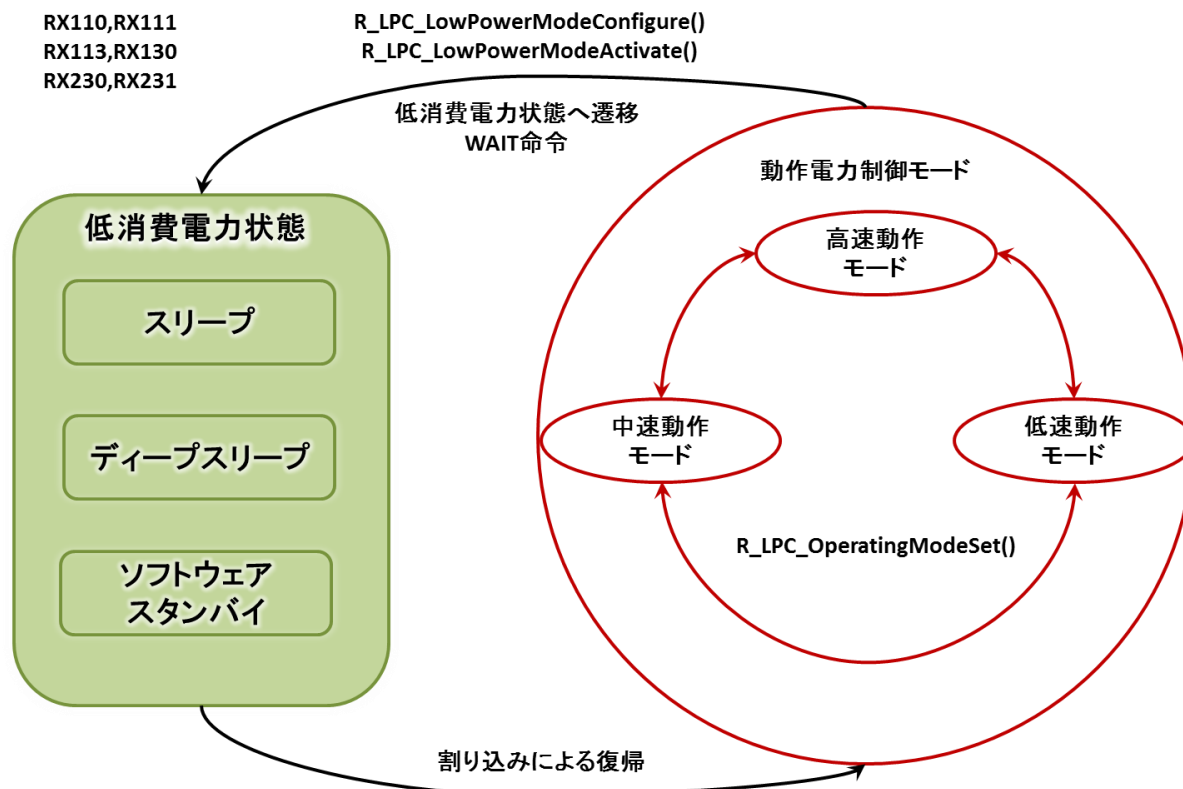


図 1.1 LPC FIT モジュールの状態遷移図 (RX110、RX111、RX113、RX130、RX230、RX231)

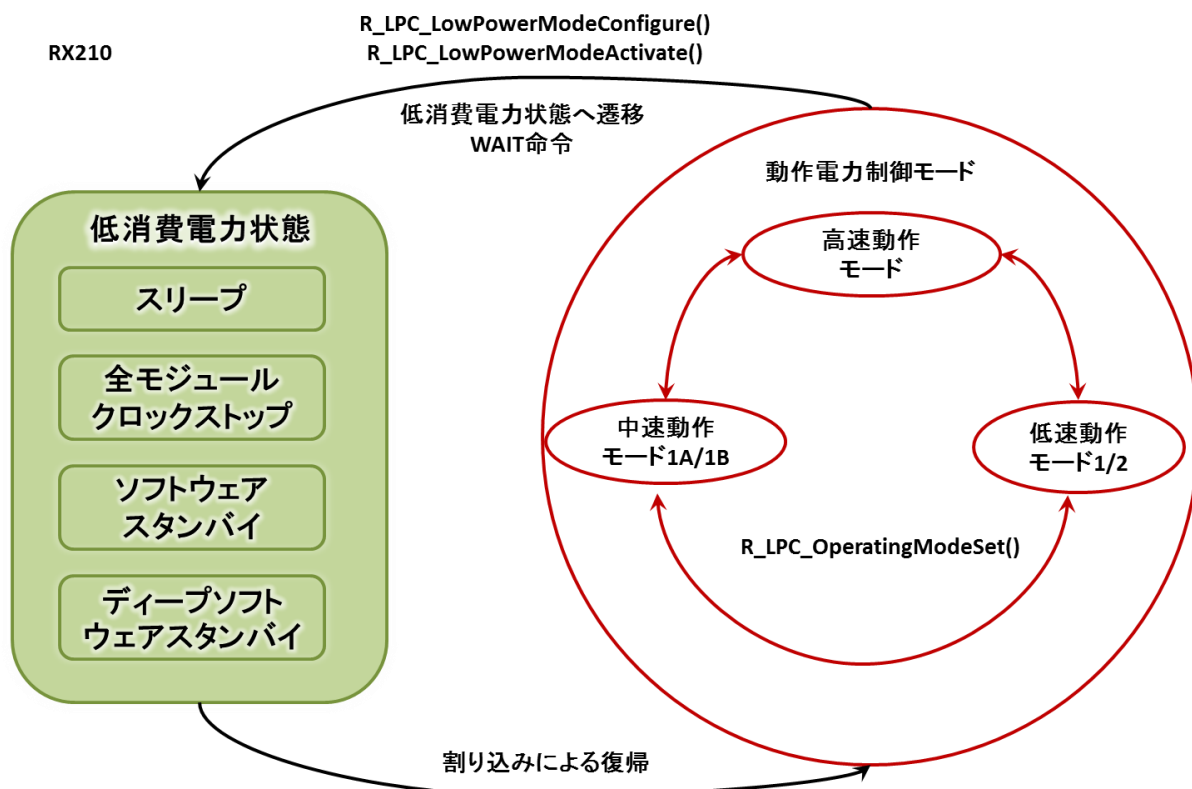


図 1.2 LPC FIT モジュールの状態遷移図 (RX210)

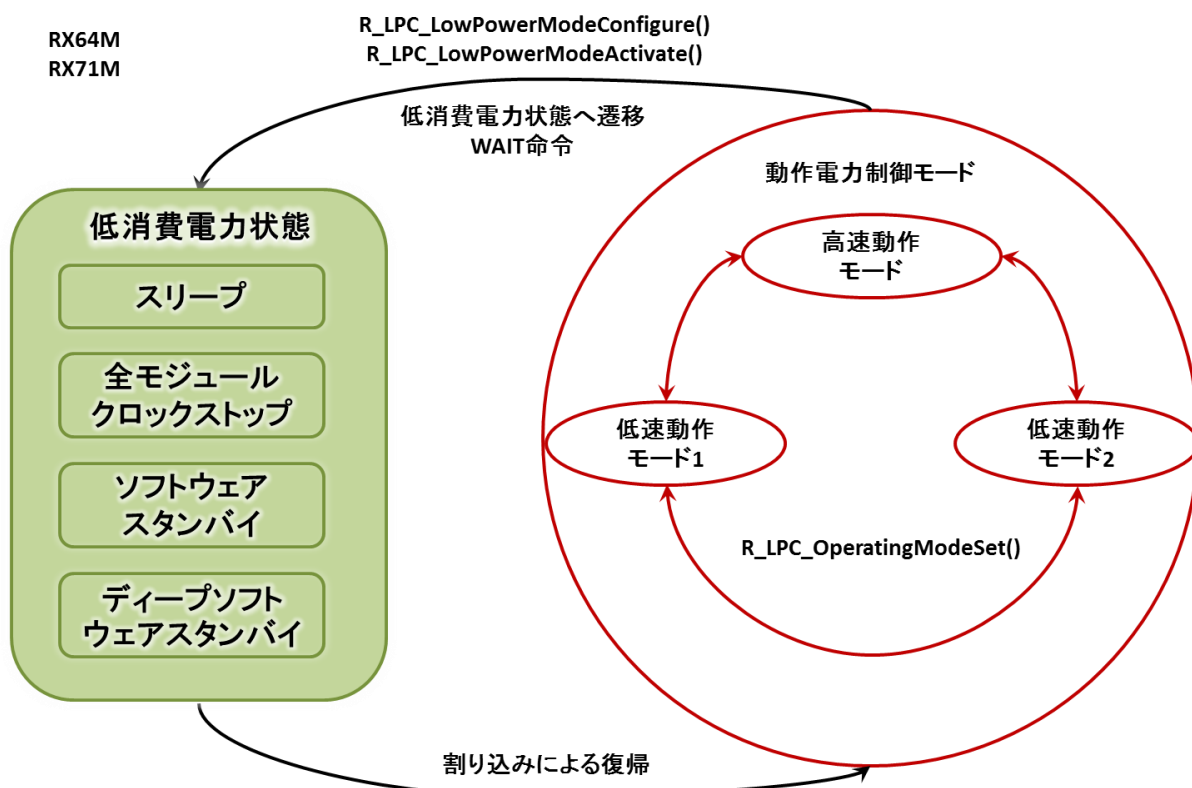


図 1.3 LPC FIT モジュールの状態遷移図 (RX64M、RX71M)

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)

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

本 FIT モジュールでは、割り込みベクタは使用していません。

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。
(RX210 のみ) LPC_CFG_SW_STANDBY_OPTIONS	6	本定義は RX210 のみが対象で、FHSSBYCR レジスタの設定に使用します。設定の詳細は、ユーザーズマニュアル ハードウェア編の 11.2.18「フラッシュ HOCO ソフトウェアスタンバイコントロールレジスタ」を参照してください。

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、RX700 シリーズの主なグループについての情報を掲載しています。

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 rev1.41

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.01.00

(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)

コンフィギュ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	
		パラメータチェックあり	パラメータチェックなし
RX110	ROM	1610 バイト	371 バイト
	RAM	40 バイト	0 バイト
	スタック	68 バイト	24 バイト
RX111	ROM	2287 バイト	371 バイト
	RAM	40 バイト	0 バイト
	スタック	80 バイト	24 バイト
RX113	ROM	2293 バイト	371 バイト
	RAM	40 バイト	0 バイト
	スタック	80 バイト	24 バイト
RX130	ROM	2285 バイト	371 バイト
	RAM	40 バイト	0 バイト
	スタック	80 バイト	24 バイト
RX230 RX231	ROM	1228 バイト	371 バイト
	RAM	40 バイト	0 バイト
	スタック	52 バイト	24 バイト
RX64M	ROM	1235 バイト	342 バイト
	RAM	40 バイト	0 バイト
	スタック	56 バイト	24 バイト
RX71M	ROM	1235 バイト	342 バイト
	RAM	40 バイト	0 バイト
	スタック	56 バイト	24 バイト

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].h (例 : r_lpc_rx64m_if.h) に記載されています。

2.9.1 R_LPC_OperatingModeSet

```
/* 動作電力制御モード (対象 : RX110, RX111, RX113, RX130, RX230, RX231) */
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 : RX210) */
typedef enum lpc_operating_mode
{
    LPC_OP_HIGH_SPEED = 0x00,
    LPC_OP_MIDDLE_SPEED_1A = 0x02,
    LPC_OP_MIDDLE_SPEED_1B = 0x03,
    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 : RX71M, RX64M) */
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;
```

2.9.2 R_LPC_LowPowerModeConfigure

```
/* 低消費電力状態（対象：RX110, RX111, RX113, RX130, RX230, RX231） */
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：RX210, RX71M, RX64M） */
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;
```

2.9.3 R_LPC_ReturnClockSwitch

```
/* 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, RX230, RX231） */
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：RX210, RX71M, RX64M） */
typedef enum lpc_clock_switch
{
    LPC_HOCO = 0x01,
    LPC_MAIN_OSC = 0x02,
}lpc_clock_switch_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、低消費電力状態に遷移する前に、ユーザが設定したコールバック関数を呼び出します。コールバック関数については「3 API 関数」の“`R_LPC_LowPowerModeActivate ()`”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、Smart Configurator を使用した(1)、(3)の追加方法を推奨しています。ただし、Smart Configurator 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(2)、(4)の方法を使用してください。

- (1) e² studio 上で Smart Configurator を使用して FIT モジュールを追加する場合
e² studio の Smart Configurator 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「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。
- (2) e² studio 上で FIT Configurator を使用して FIT モジュールを追加する場合
e² studio の FIT Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。
- (3) CS+上で Smart Configurator 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 Smart Configurator 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「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。
- (4) 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。

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。

「WAIT_LOOP」を記述している対象デバイス

- RX110、RX111、RX113 グループ
- RX130 グループ
- RX210 グループ
- RX230、RX231 グループ
- RX64M グループ
- RX71M グループ

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、RX230、RX231 で低速動作モードを選択した場合と RX210 で低速動作モード 2 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（RX71M、RX64M、RX210）、または表 11.4（RX110、RX111、RX113、RX130、RX230、RX231）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。

Reentrant

本関数は再入不可です。

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。

Reentrant

本関数は再入不可です。

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_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, RX230, RX231
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	RX210, RX64M, RX71M
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, RX230, RX231
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, RX230, RX231
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、ICLK、FCLK、BCLK、PCLKB、PCKD をすべて 2 分周以上に設定してください	LPC_ERR_ILLEGAL	RX210
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	RX210, RX64M, RX71M

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	RX210, RX64M, RX71M
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	RX210, RX64M, RX71M

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, RX230, RX231
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, RX230, RX231

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, RX230, RX231, RX210, RX64M, RX71M
DMAST.DMST ビットが“0”の状態です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, RX210, RX64M, RX71M
DTCST.DTCST ビットが“0”の状態です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, RX230, RX231, RX210, RX64M, RX71M
サブクロックをシステムクロックのクロックソースに使用する場合は、ソフトウェアスタンバイモードへ移行する際に、RTC (RCR3.RTCEN = 1)、または低速オンチップオシレータ (LOCOCR.LCSTP = 0) を動作させておいてください。	LPC_ERR_ILLEGAL	RX210
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, RX230, RX231, RX210, RX64M, RX71M

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	RX210, RX64M, RX71M
DMAST.DMST ビットが“0”の状態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	RX210, RX64M, RX71M
DTCST.DTCST ビットが“0”の状態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	RX210, RX64M, RX71M
サブクロックをシステムクロックのクロックソースに使用する場合は、ディープソフトウェアスタンバイモードへ移行する際に、RTC (RCR3.RTCEN = 1)、または低速オンチップオシレータ (LOCOCR.LCSTP = 0) を動作させておいてください。	LPC_ERR_ILLEGAL	RX210
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	RX210, RX64M, RX71M
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	RX210, RX64M, RX71M
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	RX210, RX64M, RX71M
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	RX210, RX64M, RX71M

Reentrant

本関数は再入不可です。

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_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、RX231:

- 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. RX64M、RX71M:

- 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。

Reentrant

本関数は再入不可です。

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

```
void 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。

Reentrant

この関数は再入可能（リエントラント）です。

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。

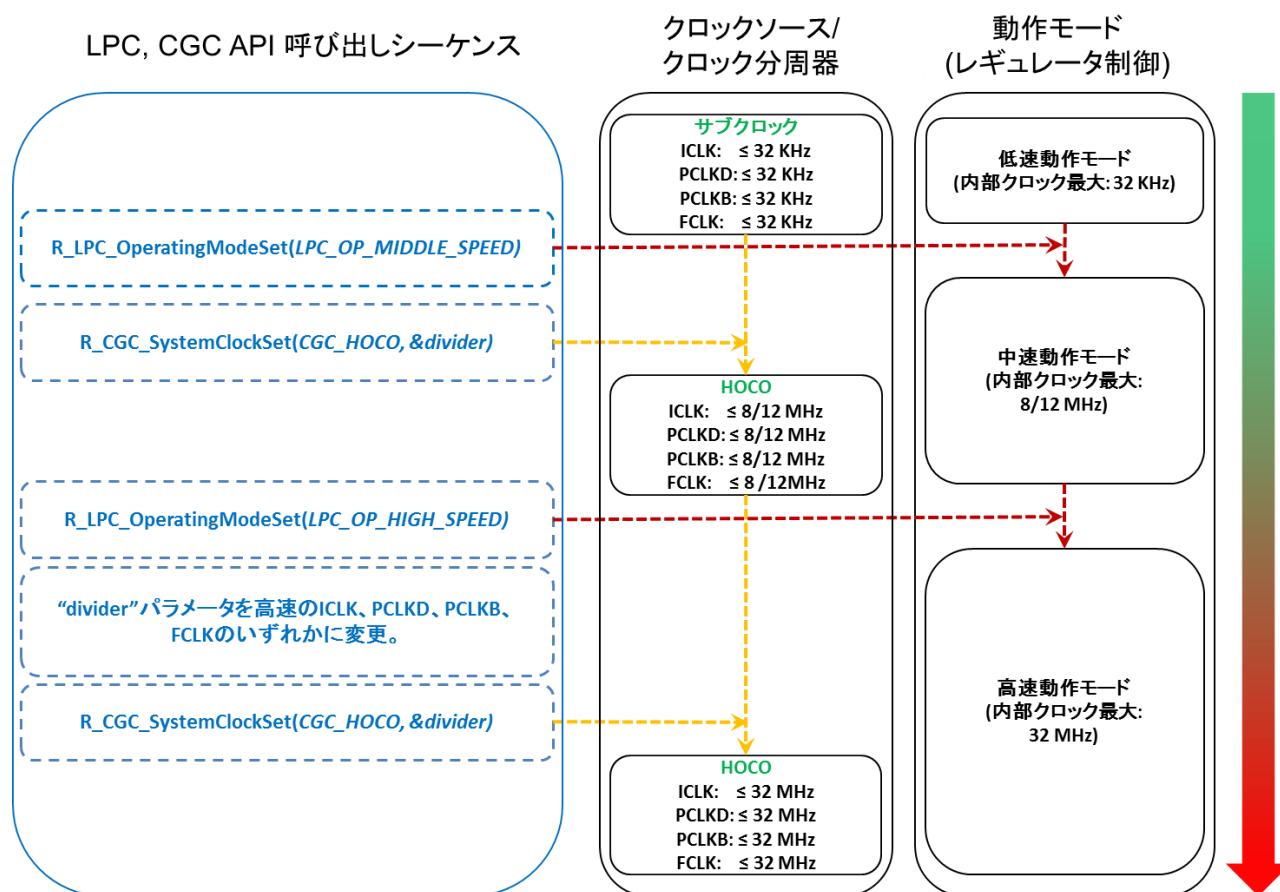
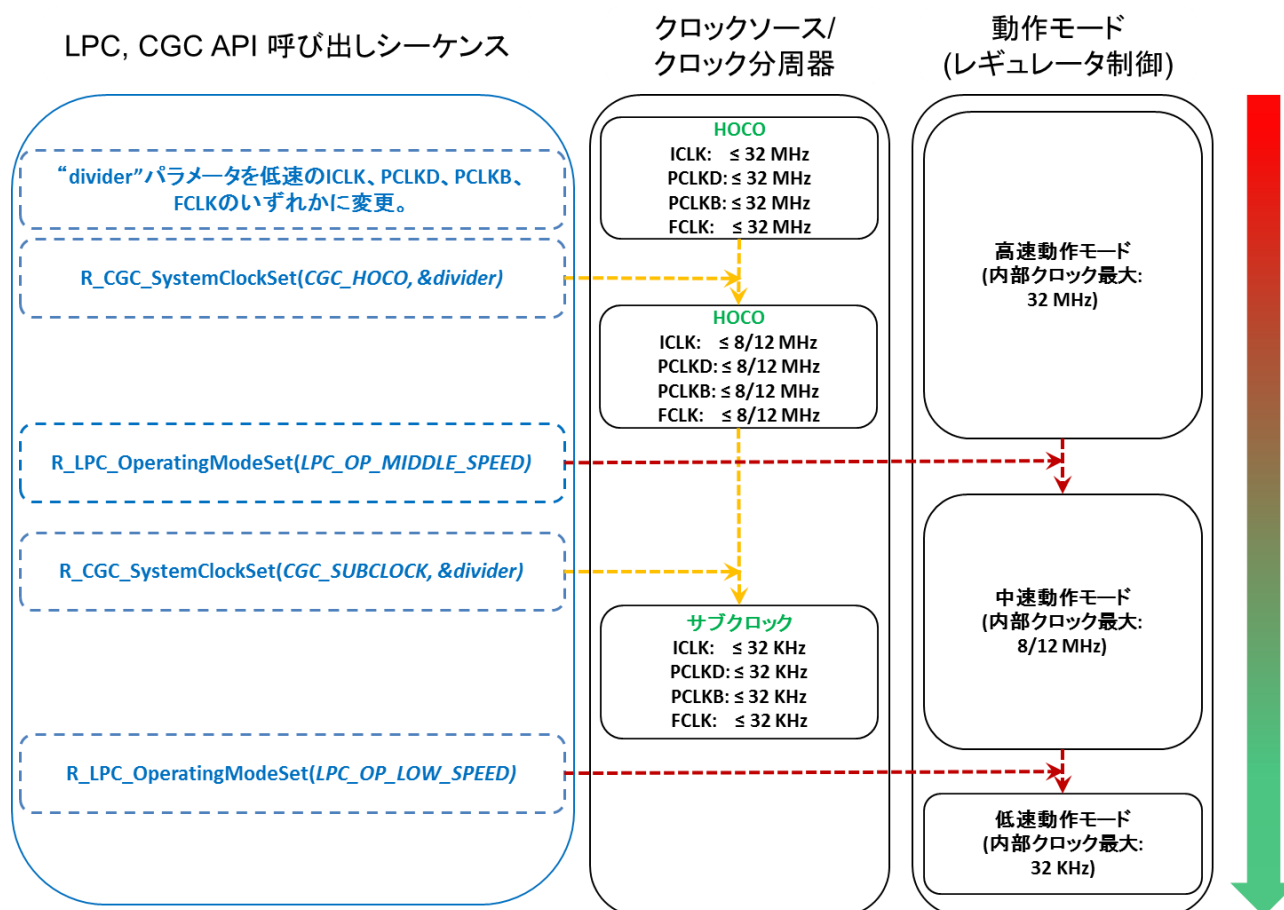


図 4.1 低電力状態から高電力状態への遷移例（RX110, RX111, RX113, RX130）

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。



4.3 高電力状態への遷移例（RX210 の場合）

RX210 MCU 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。

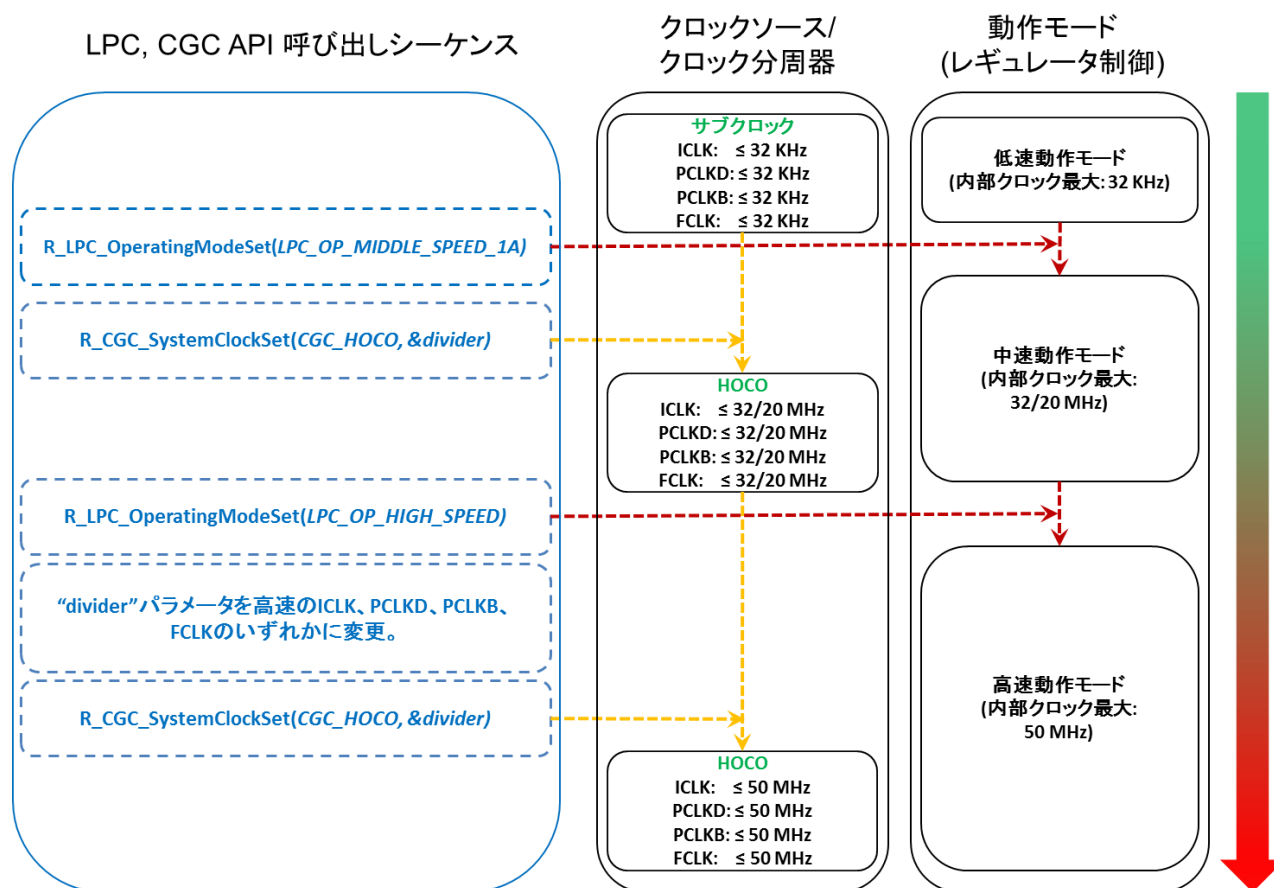


図 4.3 低電力状態から高電力状態への遷移例（RX210）

上記に加え、スリープモードからのウェイクアップ時にメインクロックソースを自動的に切り替える復帰クロックソース切り替え機能があります。

4.4 低電力状態への遷移例（RX210 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。

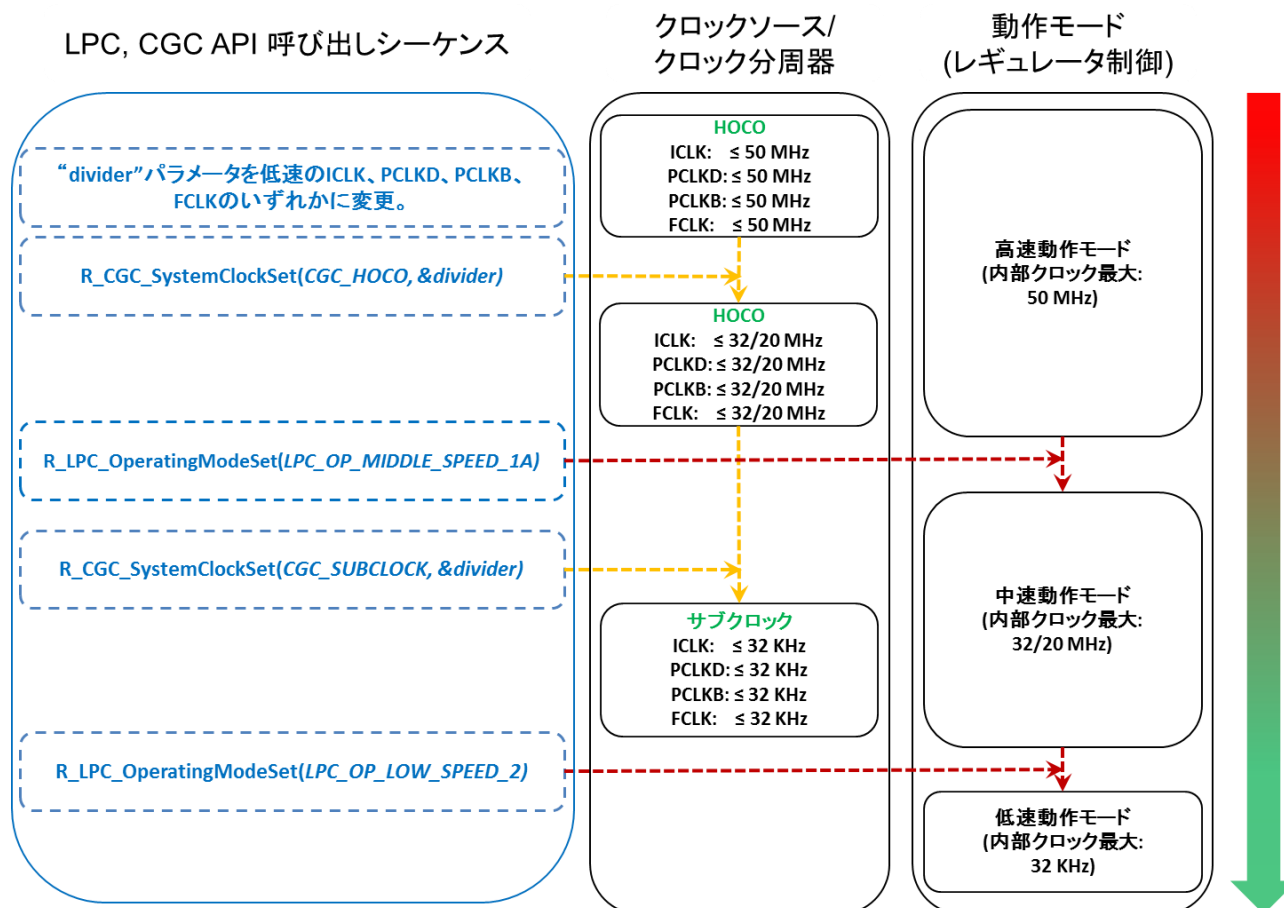


図 4.4 高電力状態から低電力状態への遷移例（RX210）

5. デモプロジェクト

デモプロジェクトはスタンドアロンプログラムです。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_rskrx113

RSKRX113 向けの RX113 消費電力低減機能（LPC）のデモプログラムです。デモでは、1 秒ごとに LED0 のトグルを繰り返します。SW2 が最初に押下された場合、スタンバイモードに遷移し、このモードでは CPU が動作しないため、LED0 はトグルを停止します。SW2 を再度押下すると、スタンバイモードから復帰し、LED は点滅を再開します。

1. サンプルコードをコンパイルしてダウンロードします。
2. デモプログラムを実行します。

注: R221（ボードの下側の 0Ω 抵抗）を取り外して、J11 に電流計を配置することで、MCU がノーマルモードとスタンバイモードを切り替える際の電流消費を観察できます。正確に電流を測定するには、いったんデバッガを使って MCU にプログラムを書き込み、その後デバッガを外して、テスト中は外部からボードに電源を供給します。測定が完了したら、J11 にジャンパを配置して、通常の動作を行います。

対応ボード

- RSKRX113

5.2 lpc_demo_rskrx231

RSKRX231 向けの RX231 消費電力低減機能（LPC）のデモプログラムです。デモでは以下に示す順番で電力のモードを切り替えます。

- (1) 高速動作モード
- (2) 中速動作モード
- (3) 低速動作モード
- (4) 高速動作モード（2 回目）
- (5) スリープモード（復帰クロックソース切り替え機能有効）
- (6) ソフトウェアスタンバイモード
- (7) ディープスリープモード

デモプログラムをコンパイルしたら、それをダウンロードして実行します。SW2 を押下して次の電力モードに遷移します。動作電力制御モードに遷移すると、そのモードのクロック速度に対応して LED0 が点滅します。低消費電力状態（スリープ、ソフトウェアスタンバイなど）では CPU が動作しないため、LED0 は点滅を停止します。各電力モード遷移時、メッセージが e² studio のコンソールウィンドウに送られ、MCU の状態を示します。デモプログラムが完了したとき、MCU はディープソフトウェアスタンバイモードの状態です。次の電力モードへ切り替えるには、プログラムをリセットしてください。

注: J8 に電流計を配置することで、MCU が動作電力制御モードと低消費電力状態を切り替える際の電流消費を観察できます。このとき、R55（J8 の横の 0Ω 抵抗）を取り外す必要があります。正確に電流を測定するには、いったんデバッガを使って MCU にプログラムを書き込み、その後デバッガを外して、テスト中は外部からボードに電源を供給します。測定が完了したら、R55 を再配置するか、J8 にジャンパを配置して、通常の動作を行います。

5.3 lpc_demo_rskrx71M

RSKRX71M 向けの RX71M 消費電力低減機能（LPC）のデモプログラムです。デモでは以下に示す順番で電力のモードを切り替えます。

- (1) 高速動作モード
- (2) 低速動作モード 1
- (3) 低速動作モード 2
- (4) スリープモード（復帰クロックソース切り替え機能有効）
- (5) 全モジュールクロックストップモード
- (6) ソフトウェアスタンバイモード
- (7) ディープソフトウェアスタンバイモード

デモプログラムをコンパイルしたら、それをダウンロードして実行します。SW2 を押下して次の電力モードに遷移します。動作電力制御モードに遷移すると、そのモードのクロック速度に対応して LED0 が点滅します。低消費電力状態（スリープ、ソフトウェアスタンバイなど）では CPU が動作しないため、LED0 は点滅を停止します。各電力モード遷移時、メッセージが e² studio のコンソールウィンドウに送られ、MCU の状態を示します。デモプログラムが完了したとき、MCU はディープソフトウェアスタンバイモードの状態です。次の電力モードへ切り替えるには、プログラムをリセットしてください。

注: J21 に電流計を配置することで、MCU が動作電力制御モードと低消費電力状態を切り替える際の電流消費を観察できます。このときは R217（J21 の横の 0Ω 抵抗）を取り外す必要があります。正確に電流を測定するには、いったんデバッグを使って MCU にプログラムを書き込み、その後デバッグを外して、テスト中は外部からボードに電源を供給します。測定が完了したら、J21 にジャンパを再配置して、通常の動作を行います。

5.4 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「完了」をクリックします。

5.5 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
モジュールのバージョン	Rev1.41

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)」
- e² 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 e² 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	2016.10.1	—	初版発行
1.41	2019.4.1	—	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

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イやマガジンケース、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、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブ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、リセットしてください。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。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）、交通管制（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)

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

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/

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。