

RZ/A2M グループ

低消費電力モードの使用例

要旨

本アプリケーションノートは、RZ/A2M の低消費電力モード（スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモード）の使用例について説明します。

動作確認デバイス

RZ/A2M

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。

目次

1. 仕様	4
2. 動作確認条件	7
3. 関連アプリケーションノート	8
4. ハードウェア説明	9
4.1 低消費電力モードの概要	9
4.1.1 スリープモード	11
4.1.2 ソフトウェアスタンバイモード	12
4.1.3 ディープスタンバイモード	13
4.2 低消費電力モード遷移時の注意事項	16
4.3 使用端子一覧	17
5. ソフトウェア説明	18
5.1 動作概要	18
5.1.1 スリープモード	18
5.1.2 ソフトウェアスタンバイモード	20
5.1.3 ディープスタンバイモード	23
5.2 サンプルコード実行時の周辺機能の設定およびメモリ配置	29
5.2.1 周辺機能の設定	29
5.2.2 メモリマップ	31
5.2.3 サンプルコードのセクション配置	32
5.3 使用割り込み一覧	35
5.4 データ型一覧	35
5.5 定数一覧	36
5.6 構造体/共用体一覧	39
5.7 関数一覧	44
5.8 関数仕様	45
5.8.1 サンプル関数仕様	45
5.8.2 API 関数仕様	47
5.9 フローチャート	52
5.9.1 スリープモードへの遷移サンプル関数	52
5.9.2 スリープモードへの遷移関数	54
5.9.3 ソフトウェアスタンバイモードへの遷移サンプル関数	55
5.9.4 ソフトウェアスタンバイモードへの遷移関数	57
5.9.5 ディープスタンバイモードへの遷移サンプル関数	59
5.9.6 ディープスタンバイモードへの遷移関数	60
5.9.7 ディープスタンバイモード解除後に保持用内蔵 RAM から起動する処理	62
5.9.8 NMI 端子割り込みサンプル関数	63
5.9.9 IRQ0 端子割り込みサンプル関数	63
5.9.10 低消費電力モードへの遷移関数	64
5.10 サンプルコードの起動	65
5.10.1 スリープモード遷移コマンドの動作	66
5.10.2 ソフトウェアスタンバイモード遷移コマンドの動作	67

5.10.3 ディープスタンバイモード遷移コマンドの動作.....	68
5.10.4 ヘルプコマンドの動作	70
6. サンプルコード	71
7. 参考ドキュメント	71
改訂記録	72

1. 仕様

RZ/A2M では、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、ディープスタンバイモード、モジュールスタンバイ）をサポートしています。本サンプルコードでは、スリープモード、ソフトウェアスタンバイモード、およびディープスタンバイモードを使用し、低消費電力モードへの遷移と解除に必要な処理を行います。

本サンプルコードは、起動メッセージ出力後、ホスト PC 上のターミナルからコマンドを入力することにより、プログラム実行状態から低消費電力モードへ遷移しプログラム処理が停止します。サンプルコードでは、PJ_1 (IRQ0) 端子の立ち下がリエッジ、または NMI 端子の立ち下がリエッジを低消費電力モードの解除要因に設定しており、PJ_1 (IRQ0) 端子または NMI 端子から解除要求を受け付け後、低消費電力モードを解除します。

本アプリケーションノートでは、SPI マルチ I/O バスコントローラを SPIBSC、クロックパルス発振器を CPG、割り込みコントローラを INTC、OS タイマを OSTM、FIFO 内蔵シリアルコミュニケーションインタフェースを SCIFA、汎用入出力ポートを GPIO、低消費電力モードを STB、メモリ管理ユニットを MMU、ウォッチドッグタイマを WDT とします。

表 1.1 に使用する周辺機能と用途を、図 1.1 にサンプルコード実行時の動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
SPI マルチ I/O バスコントローラ (SPIBSC)	外部アドレス空間リードモードに設定し、CPU が SPI マルチ I/O バス空間に接続されたシリアルフラッシュメモリから直接リードするための信号を生成
クロックパルス発振器 (CPG)	RZ/A2M の動作周波数の生成
割り込みコントローラ (INTC)	OSTM チャンネル 0、OSTM チャンネル 2 および SCIFA チャンネル 4 の割り込み制御に使用 スリープモード、ソフトウェアスタンバイモード解除時の NMI 端子および IRQ0 端子割り込み制御に使用
OS タイマ (OSTM)	OSTM チャンネル 0 およびチャンネル 2 を使用 <ul style="list-style-type: none"> OSTM チャンネル 0 LED 点灯および消灯の周期を生成 OSTM チャンネル 2 OS Abstraction Layer による時間管理に使用
FIFO 内蔵シリアルコミュニケーションインタフェース (SCIFA)	SCIFA チャンネル 4 を用いて、ホスト PC との通信用として使用
汎用入出力ポート (GPIO)	SCIFA チャンネル 4 の兼用端子切り替えに使用、LED の点灯および消灯のための端子制御に使用 PJ_1 (IRQ0) 端子を低消費電力モードの解除用端子として使用
低消費電力モード (STB)	RZ/A2M の周辺 IO のモジュールスタンバイを解除するために使用、保持用内蔵 RAM をライト許可するために使用 スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモードの遷移および解除に使用 ディープスタンバイモード遷移時の発振安定時間の設定に使用
メモリ管理ユニット (MMU)、L1 キャッシュ、L2 キャッシュ	RZ/A2M の外部アドレス空間において、L1 キャッシュの有効領域の指定やメモリタイプの指定などの変換テーブルを生成。L1 キャッシュおよび L2 キャッシュを有効に設定
ウォッチドッグタイマ (WDT)	ソフトウェアスタンバイモード遷移時の発振安定時間の設定に使用

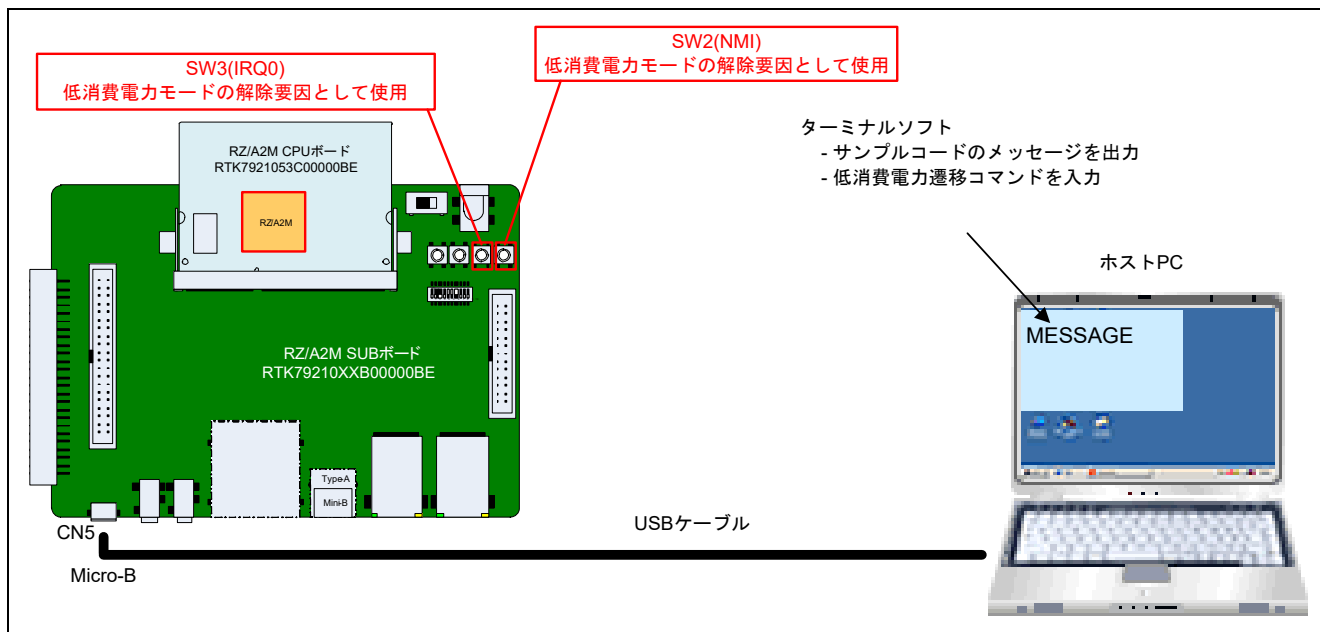


図 1.1 動作環境

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件 (1/2)

項目	内容
使用 MCU	RZ/A2M
動作周波数 (注)	CPU クロック (I ϕ) : 528MHz 画像処理クロック (G ϕ) : 264MHz 内部バスクロック (B ϕ) : 132MHz 周辺クロック 1 (P1 ϕ) : 66MHz 周辺クロック 0 (P0 ϕ) : 33MHz QSPI0_SPCLK : 66MHz CKIO : 132MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (1.8/3.3V 切替 I/O (PVcc_SPI)) : 3.3V 電源電圧 (内部) : 1.2V
統合開発環境	e2 studio V7.5.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update コンパイラオプション (ディレクトリパスの追加は除く) Release コンフィグレーション : -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug コンフィグレーション : -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

【注】 クロックモード 1 (EXTAL 端子からの 24MHz のクロック入力) で使用時の動作周波数です。

表 2.2 動作確認条件 (2/2)

項目	内容
動作モード	ブートモード 3 (シリアルフラッシュブート 3.3V 品)
ターミナルソフトの通信設定	<ul style="list-style-type: none"> 通信速度 : 115200bps データ長 : 8 ビット パリティ : なし ストップビット長 : 1 ビット フロー制御 : なし
使用ボード	RZ/A2M CPU ボード RTK7921053C00000BE RZ/A2M SUB ボード RTK79210XXB00000BE
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none"> シリアルフラッシュメモリ (SPI マルチ I/O バス空間に接続) メーカー名 : Macronix 社、型名 : MX25L51245GXD RL78/G1C (USB 通信とシリアル通信を変換し、ホスト PC との通信に使用) LED1 SUB ボードの SW2 および SW3 SW2 : NMI 端子への信号入力 SW3 : PJ_1 (IRQ0) 端子への信号入力

【注】 ソフトウェアスタンバイモードおよびディープスタンバイモードでは、デバッグインタフェースのクロックも停止するため、ご使用のエミュレータによってはエミュレータとボードを接続した状態で本サンプルコードを動作させることができません。このようなエミュレータをご使用の場合には、ボード上のシリアルフラッシュメモリにプログラムをダウンロードした後、エミュレータとボードの接続を切断した上で、ボード単体でサンプルコードを動作させてください。

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/A2M グループ 初期設定例 (R01AN4321)
- RZ/A2M グループ シリアルフラッシュメモリからのブート例 (R01AN4333)

4. ハードウェア説明

4.1 低消費電力モードの概要

RZ/A2M は、以下の低消費電力モードをサポートしています。低消費電力モードでは、CPU 動作、クロック、内蔵メモリ、一部内蔵周辺モジュール等の機能を停止したり、LSI の内部電源をオフにしたりすることにより、消費電力を低減させることができます。

- スリープモード
- ソフトウェアスタンバイモード
- ディープスタンバイモード
- モジュールスタンバイモード

本アプリケーションノートでは、主にスリープモード、ソフトウェアスタンバイモード、ディープスタンバイモードについて説明します。RZ/A2M の処理状態には、リセット状態、プログラム実行状態、低消費電力モードの3つの状態があります。図 4.1 に処理状態の状態遷移図を示します。

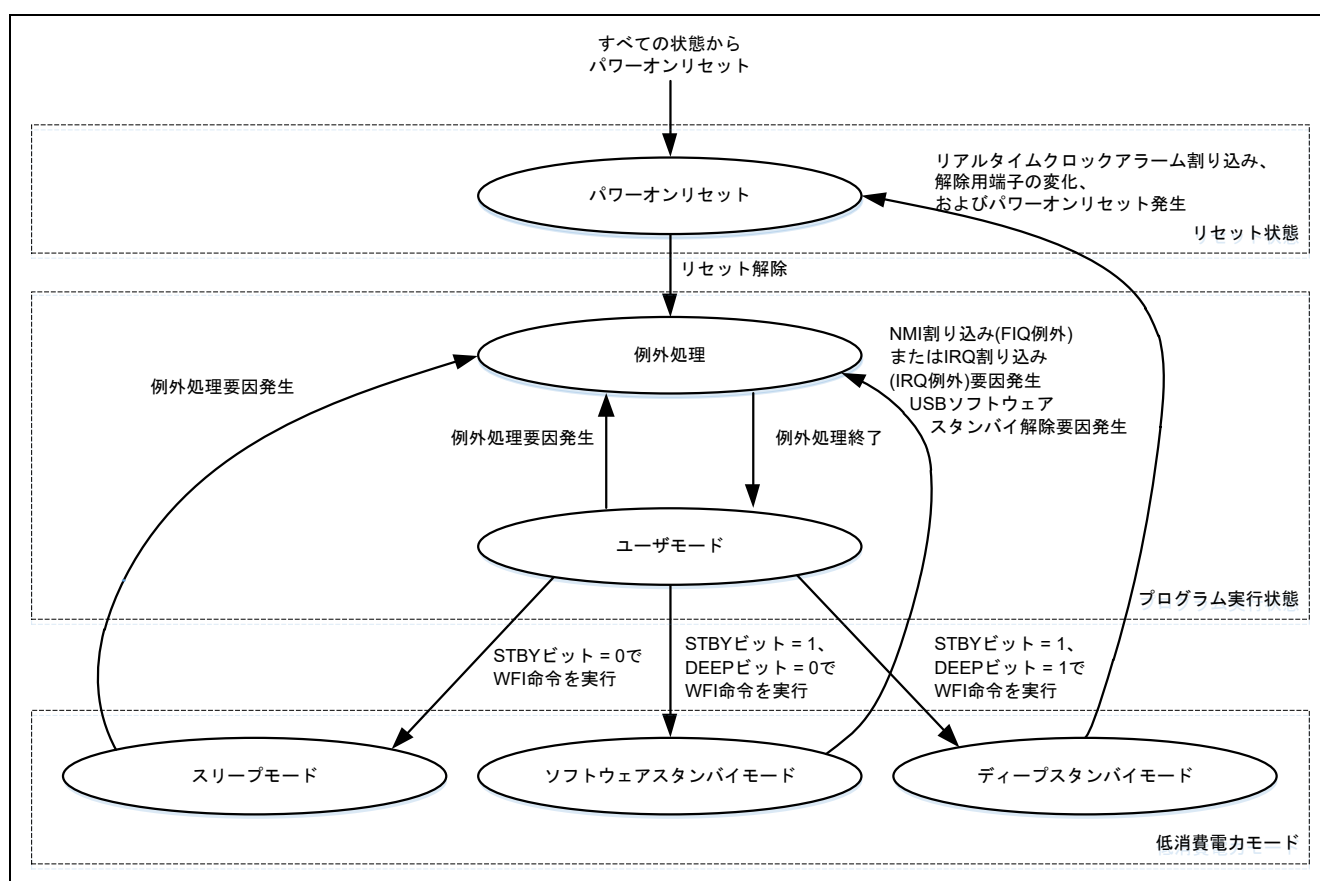


図 4.1 処理状態の状態遷移図

表 4.1 に低消費電力モードの概要を示します。

表 4.1 低消費電力モードの概要

低消費電力モード	遷移条件	状態					解除方法
		CPU、CPU レジスタ、1 次キャッシュ、TLB	2 次キャッシュ	大容量内蔵 RAM (保持用内蔵 RAM 含む)	内蔵周辺モジュール	LSI 内部電源	
スリープモード	STBCR1=0x00 (or 0x40) の状態で WFI 命令を実行	停止 内容は保持	動作	動作	動作	印加	<ul style="list-style-type: none"> ● 割り込み ● パワーオンリセット
ソフトウェアスタンバイモード	STBCR1=0x80 の状態で WFI 命令を実行	停止 内容は保持	停止 内容は保持	停止 内容は保持	停止	印加	<ul style="list-style-type: none"> ● NMI、IRQ 割り込み ● パワーオンリセット ● USB ソフトウェアスタンバイ解除要因信号の変化
ディープスタンバイモード	STBCR1=0xC0 の状態で WFI 命令を実行	停止 内容は非保持	停止 内容は非保持	停止 保持用内蔵 RAM の内容は保持 その他の大容量内蔵 RAM は非保持	停止	遮断	<ul style="list-style-type: none"> ● パワーオンリセット ● リアルタイムクロックアラーム割り込み 0,1 ● 解除用端子の変化 ● USB2.0 ホスト/ファンクションモジュールチャネル割り込み 0,1
モジュールスタンバイ	STBCR10～STBCR2 の MSTP ビットを 1 とする。	動作	動作	動作	指定モジュール停止	印加	<ul style="list-style-type: none"> ● MSTP ビットを 0 にクリア

4.1.1 スリープモード

スリープモードに遷移すると、CPU 動作を停止することにより消費電力を低減させることができます。スリープモード遷移中、CPU のレジスタ、大容量内蔵 RAM、キャッシュの内容は保持するため、スリープモード解除後に、スリープモード遷移直後の命令から処理を継続することができます。なお、プログラム実行状態およびスリープモード遷移中は、FRQCR レジスタの CKOEN[1:0]ビットの設定値に応じて、CKIO の出力状態は異なります。

表 4.2 通常時およびスリープモード時の CKIO の出力状態

FRQCR レジスタ CKOEN[1:0]ビットの設定値	プログラム実行状態の出力	スリープモード時の出力
b'00	出力	出力
b'01	出力	出力
b'10	出力	出力
b'11	出力オフ (Hi-z)	出力オフ (Hi-z)

(1) スリープモードへの遷移

RZ/A2M は、STBCR1 レジスタの STBY ビットが 0 の状態で WFI 命令を実行すると、プログラム実行状態からスリープモードに遷移します。

(2) スリープモードの解除

スリープモードは、割り込み (NMI 端子、IRQ 端子、内蔵周辺機能)、およびパワーオンリセットにより解除されます。スリープモード解除後の処理は、解除方法により異なります。表 4.3 にスリープモードの解除方法と解除後の処理を示します。

表 4.3 スリープモードの解除方法と解除後の処理

解除方法	解除後の処理
割り込み (IRQ 端子、 周辺モジュール)	IRQ 例外発生後に IRQ 例外処理、解除要因に応じた割り込み処理が実行された後、スリープモード遷移時の WFI 命令の次の命令から処理を継続。
NMI 端子	FIQ 例外発生後に FIQ 例外処理、NMI 端子割り込み処理が実行された後、スリープモード遷移時の WFI 命令の次の命令から処理を継続。
パワーオンリセット	リセット解除後、リセットベクタ (H'FFFF_0000 番地) からプログラムを実行。

4.1.2 ソフトウェアスタンバイモード

ソフトウェアスタンバイモードに遷移すると、CPU 動作、クロック、周辺モジュールを停止することにより、スリープモードよりもさらに消費電力を低減させることができます。ソフトウェアスタンバイモード中、CPU および周辺モジュールのレジスタ、大容量内蔵 RAM、キャッシュの内容は保持するため、ソフトウェアスタンバイモード解除後に、遷移前のソフトウェア処理を再開することができます。なお、プログラム実行状態およびソフトウェアスタンバイモード遷移中は、FRQCR レジスタの CKOEN[1:0]ビットの設定値に応じて、CKIO の出力状態は異なります。

表 4.4 ソフトウェアスタンバイモード時の CKIO の出力状態

FRQCR レジスタ CKOEN[1:0]ビットの設定値	プログラム実行状態の出力	ソフトウェアスタンバイ モード時の出力
b'00	出力	出力オフ (Hi-z)
b'01	出力	ローレベル出力
b'10	出力	出力 (不安定なクロック)
b'11	出力オフ (Hi-z)	出力オフ (Hi-z)

(1) ソフトウェアスタンバイモードへの遷移

RZ/A2M は、STBCR1 レジスタの STBY ビットが 1、DEEP ビットが 0 の状態で WFI 命令を実行すると、プログラム実行状態からソフトウェアスタンバイモードに遷移します。

(2) ソフトウェアスタンバイモードの解除

ソフトウェアスタンバイモードは、割り込み (NMI 端子、IRQ 端子)、USB ソフトウェアスタンバイ解除要因、およびリセット (パワーオンリセット) により解除されます。解除要因の検出後、ソフトウェアスタンバイ復帰発振安定時間 (注) の経過後に、スタンバイモードは解除されます。ソフトウェアスタンバイモード解除後の処理は、解除方法により異なります。表 4.5 にソフトウェアスタンバイモードの解除方法と解除後の処理を示します。ソフトウェアスタンバイ解除後、CKIO 端子からクロックが出力されます (CKOEN[1:0]ビットが b'11 以外の設定の場合)。

【注】 ソフトウェアスタンバイ復帰発振安定時間

WDT の WTCSCR レジスタの CKS[3:0]ビット (クロックソースの選択) と WTCNT レジスタ (カウンタの設定値) により設定します。WTCNT レジスタのカウンタのオーバフローが発生するまでの時間が、お客様のシステムのクロック発振安定時間以上となるように設定する必要があります。

表 4.5 ソフトウェアスタンバイモードの解除方法と解除後の処理

解除方法	解除後の処理
割り込み (IRQ 端子、USB)	IRQ 例外発生後に IRQ 例外処理、解除要因に応じた割り込み処理 (IRQ 端子割り込み処理または USB 割り込み処理) が実行された後、ソフトウェアスタンバイモード遷移時の WFI 命令の次の命令から処理を再開。
NMI 端子	FIQ 例外発生後に FIQ 例外処理、NMI 端子割り込み処理が実行された後、ソフトウェアスタンバイモード遷移時の WFI 命令の次の命令から処理を再開。
パワーオンリセット	リセット解除後、リセットベクタ (H'FFFF_0000 番地) からプログラムを実行。

4.1.3 ディープスタンバイモード

ディープスタンバイモードに遷移すると、CPU 動作、クロック、周辺モジュールを停止し、LSI の内部電源を遮断することにより、スリープモードやソフトウェアスタンバイと比べて、大幅に消費電力を低減させることができます。ただし、ディープスタンバイモード中は、CPU および周辺モジュールのレジスタ、大容量内蔵 RAM、キャッシュの内容は保持されず、ディープスタンバイモード解除後にリセット状態から処理を開始するため、周辺モジュールやプログラムが使用するメモリの内容を再初期化する必要があります。なお、プログラム実行状態およびディープスタンバイモード遷移中は、FRQCR レジスタの CKOEN[1:0] ビットの設定値に応じて、CKIO の出力状態は異なります。

表 4.6 ディープスタンバイモード時の CKIO の出力状態

FRQCR レジスタ CKOEN[1:0]ビットの設定値	プログラム実行状態の出力	ディープスタンバイモード時 の出力
b'00	出力	出力オフ (Hi-z)
b'01	出力	ローレベル出力
b'10	出力	ローレベル出力または ハイレベル出力
b'11	出力オフ (Hi-z)	出力オフ (Hi-z)

(1) ディープスタンバイモードへの遷移

STBCR1 レジスタの STBY ビットが 1、DEEP ビットが 1 の状態で WFI 命令を実行すると、プログラム実行状態からディープスタンバイモードに遷移します。

(2) ディープスタンバイモードの解除

ディープスタンバイモードは、割り込み (NMI 端子、リアルタイムクロック 0,1 のアラーム割り込み、USB2.0 ホスト/ファンクションモジュールチャネル 0,1 の解除要因)、解除用端子の変化、およびリセット (パワーオンリセット) により解除されます。解除要因の検出後、ディープスタンバイ復帰発振安定時間 (注) の経過後に、スタンバイモードは解除されます。ディープスタンバイモード解除後の処理は、いずれの解除方法でも、ディープスタンバイモード解除後にリセットベクタ (H'FFFF_0000 番地) からプログラムを実行します。

なお、外部端子によりディープスタンバイモードを解除する場合は、汎用入出力ポート (GPIO) にて解除要因に使用する端子の機能割り当てを行う必要がありません。

【注】 ディープスタンバイ復帰発振安定時間

DSCNT レジスタにより設定します。DSCNT レジスタのカウンタのオーバフローが発生するまでの時間が、お客様のシステムのクロック発振安定時間以上となるように設定する必要があります。

ディープスタンバイモードの解除後の動作は、EBUSKEEPE ビットと RAMBOOT ビットの設定内容により異なります。表 4.7 にディープスタンバイモード解除後の起動方法と端子状態を示します。

表 4.7 ディープスタンバイモード解除後の起動方法と端子状態

EBUSKEEPE ビット	RAMBOOT ビット	起動方法	ディープスタンバイモード解除後の端子状態
0	0	外部メモリから起動 (電源投入時のブートモードに応じたローダプログラムの配置アドレスから起動)	外部メモリ制御端子は保持しない。 ディープスタンバイ解除後外部メモリ制御端子は端子保持解除。 その他の端子は IOKEEP ビットクリア時に端子保持解除。
0	1	保持用内蔵 RAM の H'8000_0000 番地から起動	外部メモリ制御端子は保持しない。 ディープスタンバイ解除後外部メモリ制御端子は端子保持解除。 その他の端子は IOKEEP ビットクリア時に端子保持解除。
1	0	-	設定禁止
1	1	保持用内蔵 RAM の H'8000_0000 番地から起動	外部メモリ制御端子は保持する。 外部メモリ制御端子もその他の端子も、 IOKEEP ビットクリア時に端子保持解除。

(3) ディープスタンバイモード解除要因の判別

ディープスタンバイモードがパワーオンリセット以外の要因で解除された場合には、どの解除要因で解除されたかを、DSFR レジスタ、USBDSFR レジスタの値により判別することができます。

図 4.2 にディープスタンバイモード解除フローを示します。

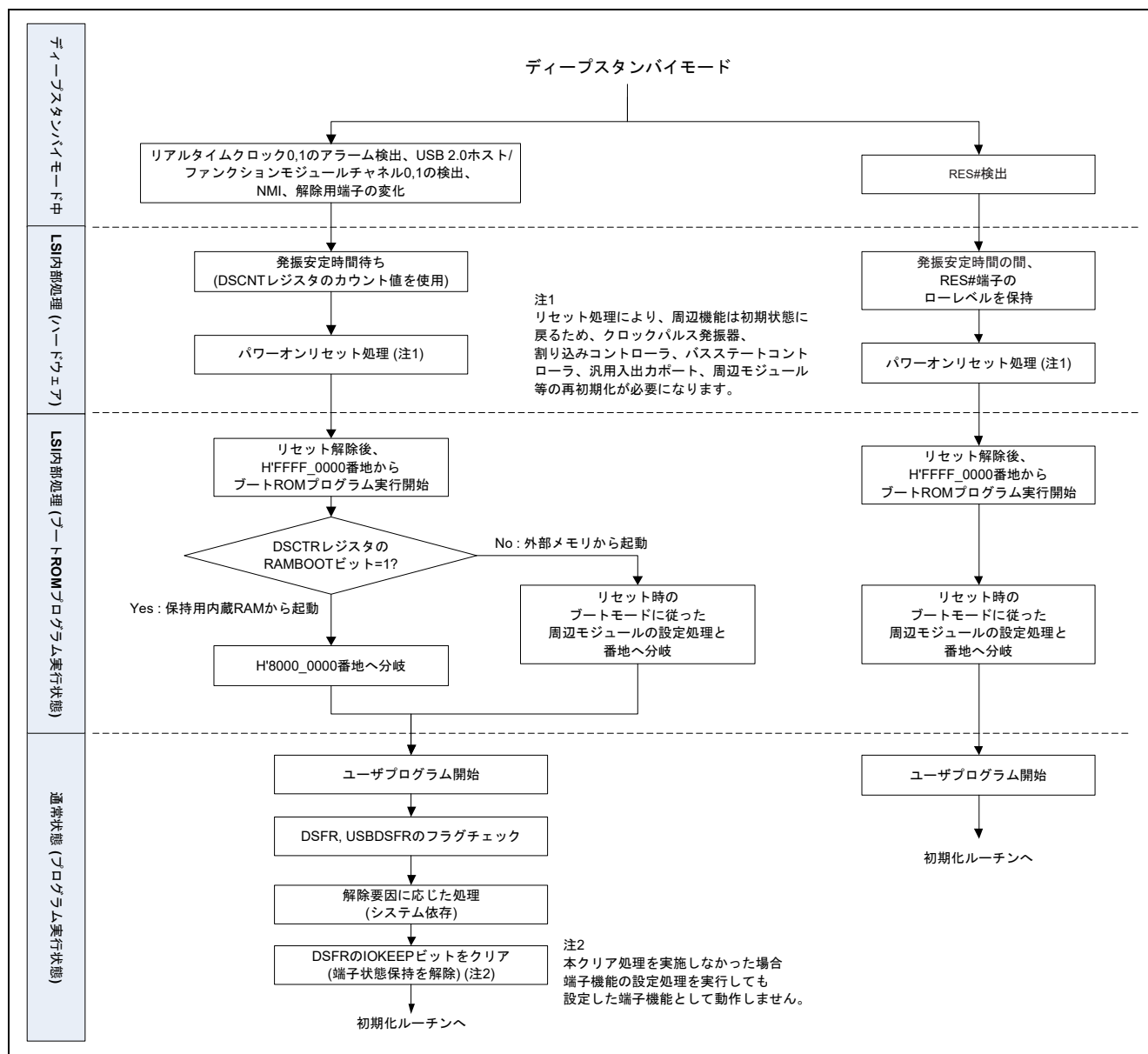


図 4.2 ディープスタンバイモード解除フロー

4.2 低消費電力モード遷移時の注意事項

ソフトウェアスタンバイモードおよびディープスタンバイモードでは、WFI 命令実行後、バスマスタからの発行済みリクエストの完了を待ってから、ハードウェアがバスマスタを自動的に停止させ、スタンバイモードへ遷移します。この際、発行済みリクエストが完了できない場合はスタンバイモードへ遷移できませんので、モジュールスタンバイ状態のモジュールに対するレジスタアクセスなどは実施しないでください。また、バスマスタから意図しないリクエストの発生を抑止するために、スタンバイモードへの遷移手順を実行する前に、予めソフトウェアにより各バスマスタを停止させてください。

4.3 使用端子一覧

表 4.8 に使用端子と機能を示します。

表 4.8 使用端子と機能

端子名	入出力	内容
MD_BOOT2	入力	ブートモードの選択（ブートモード 3 に設定） MD_BOOT2 : "L"、MD_BOOT1 : "H"、MD_BOOT0 : "H"
MD_BOOT1	入力	
MD_BOOT0	入力	
QSPI0_SPCLK	出力	シリアルフラッシュメモリのクロック
QSPI0_SSL	出力	シリアルフラッシュメモリのスレーブセレクト
QSPI0_IO0	入出力	シリアルフラッシュメモリのデータ 0
QSPI0_IO1	入出力	シリアルフラッシュメモリのデータ 1
QSPI0_IO2	入出力	シリアルフラッシュメモリのデータ 2
QSPI0_IO3	入出力	シリアルフラッシュメモリのデータ 3
RPC_RESET#	出力	シリアルフラッシュメモリのリセット
P6_0	出力	LED の点灯および消灯
RxD4 (P9_1)	入力	シリアル受信データ信号
TxD4 (P9_0)	出力	シリアル送信データ信号
NMI	入力	NMI 信号
IRQ0 (PJ_1)	入力	IRQ0 (PJ_1) 信号

【注】 #は負論理（またはアクティブロー）を示す記号です。

5. ソフトウェア説明

5.1 動作概要

サンプルコードでは、スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモードのそれぞれの低消費電力モードへの遷移処理および解除処理を行います。ここでは、それぞれの低消費電力モードの処理内容について説明します。

なお、サンプルコードでは、スリープモードおよびソフトウェアスタンバイモードのスタンバイ解除要因として IRQ0 端子割り込みを使用しますが、PJ_1 端子を IRQ0 端子機能に設定する処理は、main 関数の GPIO ドライバの Open 関数処理で行っています（NMI 端子は専用端子であるため、端子機能の設定は必要ありません）。ご使用の解除要因に応じて、解除に必要な処理を実装してください。

5.1.1 スリープモード

サンプルコードでは、起動メッセージの出力後、ターミナルよりコマンド"LPM_SM"を入力すると、スリープモードに遷移するための関数 Sample_LPM_Sleep_Mode 関数が実行され、スリープモードに遷移します。

また、スリープモード遷移中に SUB ボードの SW2（NMI）または SW3（IRQ0）を押すことにより、スリープモードが解除され、スリープモード遷移前の処理を継続します。

スリープモード遷移前と解除後にターミナルにメッセージを出力します。スリープモードの出力メッセージは「5.10.1 スリープモード遷移コマンドの動作」を参照してください。

(1) スリープモード遷移前の処理

スリープモードに遷移する前に、解除要因に使用する割り込みを設定します。

サンプルコードでは、IRQ0 端子割り込みおよび NMI 端子割り込みを使用するための設定を行います。割り込み要求が受け付けられるように設定を行い、IRQ0 端子割り込みおよび NMI 端子割り込みを受け付け時に実行される割り込みハンドラ関数を登録します。Sample_LPM_Sleep_Mode 関数にて、R_INTC_RegistIntFunc 関数を使用して、表 5.1 に示す割り込みハンドラ関数を登録しています。

表 5.1 スリープモード解除に使用する関数

割り込み	割り込み ID	登録する関数名
NMI	-	Sample_NMI_Interrupt 関数
IRQ0	36	Sample_IRQ0_Interrupt 関数

なお、サンプルコードでは、IRQ0 端子割り込みおよび NMI 端子割り込み以外でスリープモードが解除されないように、割り込みマスクレベルを制御しています。IRQ0 端子割り込みの割り込み優先レベルを"1"に設定し、スリープモード遷移前に R_INTC_SetMaskLevel 関数を使用して割り込みマスクレベルを"31"から"2"に変更することにより、割り込み優先レベルが 2 以下の割り込み（サンプルコードでは OSTM および SCIFA 割り込み）が発生しないようにしています。

解除要因に使用する IRQ0 端子および NMI 端子の割り込み要求の検出方法を（エッジセンスまたはレベルセンス）を設定します。

表 5.2 スリープモードモード解除要因の設定内容

解除要因	ボード上の資源	設定内容
NMI	SUB ボードの SW2	割り込み要求を NMI 入力の立ち下がリエッジで検出
IRQ0	SUB ボードの SW3	割り込み要求を IRQ0 入力の立ち下がリエッジで検出

(2) スリープモード遷移時の処理

スリープモードへの遷移は R_LPM_SleepTransition 関数をコールすることにより実現しています。R_LPM_SleepTransition 関数では、STBCR1 レジスタの STBY ビットに 0、DEEP ビットに 0 を設定した後、WFI 命令を実行します。

(3) スリープモード解除後の処理

スリープモード解除後の動作は、解除要因によって異なります。表 5.3 にスリープモード解除後の処理を示します。

スリープモードの解除後、割り込みマスクレベルをスリープモード遷移前の"31"に設定し、優先レベルが 31 より高い IRQ 例外を受け付けるようにしています。

表 5.3 スリープモード解除後の処理

解除要因	実行される例外	スリープモード解除後の動作
NMI	FIQ 例外	FIQ 例外処理からコールされる Sample_NMI_Interrupt 関数が実行された後、通常ルーチンに復帰して、WFI 命令の次の命令から処理を継続。
IRQ0	IRQ 例外	IRQ 例外処理からコールされる Sample_IRQ0_Interrupt 関数を実行した後、通常ルーチンに復帰して、WFI 命令の次の命令から処理を継続。
リセット	リセット例外	リセットベクタ（H'FFFF_0000 番地）からプログラム処理を実行。電源投入時と同じ処理を実行。

5.1.2 ソフトウェアスタンバイモード

サンプルコードでは、起動メッセージの出力後、ターミナルよりコマンド"LPM_SS"を入力すると、ソフトウェアスタンバイモードに移行するための関数 Sample_LPM_Software_Standby 関数が実行され、ソフトウェアスタンバイモードに移行します。

また、ソフトウェアスタンバイモード移行中に、SUB ボードの SW2 (NMI) または SW3 (IRQ0) を押すことにより、ソフトウェアスタンバイモードが解除され、ソフトウェアスタンバイモード移行前の処理を再開します。

FRQCR レジスタの CKOEN[1:0]ビットを b'00 に設定し、ソフトウェアスタンバイモード移行中は CKIO 端子の状態が出力オフ (Hi-z) になるようにしています。

ソフトウェアスタンバイモード移行前と解除後にターミナルにメッセージを出力します。ソフトウェアスタンバイモードの出力メッセージは「5.10.2 ソフトウェアスタンバイモード移行コマンドの動作」を参照してください。

(1) ソフトウェアスタンバイモード移行前の処理

ソフトウェアスタンバイモードに移行する前に、解除要因に使用する割り込みを設定します。

サンプルコードでは、IRQ0 端子割り込みおよび NMI 端子割り込みを使用するための設定を行います。割り込み要求が受け付けられるように設定を行い、IRQ0 端子割り込みおよび NMI 端子割り込みを受け付け時に実行される割り込みハンドラ関数を登録します。Sample_LPM_Software_Standby 関数にて、R_INTC_RegistIntFunc 関数を使用して、表 5.4 に示す割り込みハンドラ関数を登録しています。

表 5.4 ソフトウェアスタンバイモード解除に使用する関数

割り込み	割り込み ID	登録する関数名
NMI	-	Sample_NMI_Interrupt 関数
IRQ0	36	Sample_IRQ0_Interrupt 関数

なお、サンプルコードでは、IRQ0 端子割り込みおよび NMI 端子割り込み以外の割り込み要求と WFI 命令の実行が同時に発生することがないように、IRQ0 端子割り込みの割り込み優先レベルを"1"に設定し、割り込みマスクレベルを"31"から"2"に設定することにより、割り込み優先レベルが 2 以下の割り込み（サンプルコードでは OSTM および SCIFA 割り込み）が発生しないようにしています。

解除要因に使用する IRQ0 端子および NMI 端子の割り込み要求の検出方法を（エッジセンスまたはレベルセンス）を設定します。

表 5.5 ソフトウェアスタンバイモード解除要因の設定内容

解除要因	ボード上の資源	設定内容
NMI	SUB ボードの SW2	割り込み要求を NMI 入力の立ち下がリエッジで検出
IRQ0	SUB ボードの SW3	割り込み要求を IRQ0 入力の立ち下がリエッジで検出

コンフィグレーションテーブル (LPM_SC_TABLE[]) の内容を設定することにより、IRQ7~IRQ0 端子および USB モジュールの解除要因を指定することができます。なお、USB モジュールを動作させるためのソフトウェアは別途準備する必要があります。

(2) ソフトウェアスタンバイモード遷移時の処理

ソフトウェアスタンバイモードへの遷移は R_LPM_SStandbyTransition 関数をコールすることにより実現しています。R_LPM_SStandbyTransition 関数では、引数に指定した低消費電力モードのコンフィグレーションテーブルのデータを使用してソフトウェアスタンバイモードの解除要因、スタンバイ復帰発振安定時間などのレジスタ設定処理を実行した後、STBCR1 レジスタの STBY ビットに 1、DEEP ビットに 0 を設定し、WFI 命令を実行します。

本サンプルコードでは、コンフィグレーションテーブル (LPM_SC_TABLE[]) の設定内容により、ソフトウェアスタンバイモード遷移中の端子状態、解除要因、ソフトウェアスタンバイモード復帰発振安定時間を設定することができます。コンフィグレーションテーブル LPM_SC_TABLE[] の詳細は表 5.19～表 5.23 を参照してください。

表 5.6 コンフィグレーションテーブルで設定できるソフトウェアスタンバイモード項目一覧

設定項目	設定可能な値	サンプルコードでの設定内容
端子状態	ソフトウェアスタンバイモード遷移中の端子状態 <ul style="list-style-type: none"> • LPM_PIN_STATUS_HIZ 端子状態を保持しない (Hi-z) • LPM_PIN_STATUS_RETAINED 端子状態を保持する 	ソフトウェアスタンバイモード遷移中は端子状態を保持しない (Hi-z)
解除要因	ソフトウェアスタンバイモードの解除要因に IRQ 端子 (IRQ7～IRQ0) の使用有無を選択 <ul style="list-style-type: none"> • LPM_SSTANDBY_DISABLE_IRQ IRQ 端子を解除要因に使用しない • LPM_SSTANDBY_ENABLE_IRQ IRQ 端子を解除要因に使用する 	IRQ 端子 (IRQ7～IRQ0) をソフトウェアスタンバイモードの解除要因に使用する
	ソフトウェアスタンバイの解除要因に USB0、USB1 の割り込み要求の使用有無を選択 <ul style="list-style-type: none"> • LPM_SS_CANCEL_FACTOR_DISABLED USB を解除要因として使用しない • LPM_SS_CANCEL_FACTOR_ENABLED USB を解除要因に使用する 	USB0 および USB1 の割り込みをソフトウェアスタンバイモードの解除要因に使用しない
ソフトウェアスタンバイモード復帰発振安定時間	ソフトウェアスタンバイモード復帰発振安定時間を ms 単位で設定 (小数点 2 桁まで設定可能)	7.0ms

(3) ソフトウェアスタンバイモード解除後の処理

ソフトウェアスタンバイモード解除後の動作は、解除要因によって異なります。表 5.7 にソフトウェアスタンバイモード解除後の処理を示します。

ソフトウェアスタンバイモードの解除後、割り込みマスケレベルをソフトウェアスタンバイモード遷移前の"31"に設定し、優先レベルが 31 より高い IRQ 例外を受け付けるようにしています。

表 5.7 ソフトウェアスタンバイモード解除後の処理

解除要因	実行される例外	ソフトウェアスタンバイモード解除後の動作
NMI	FIQ 例外	FIQ 例外処理からコールされる Sample_NMI_Interrupt 関数が実行された後、通常ルーチンに復帰して、WFI 命令の次の命令から処理を再開。
IRQ0	IRQ 例外	IRQ 例外処理からコールされる Sample_IRQ0_Interrupt 関数を実行した後、通常ルーチンに復帰して、WFI 命令の次の命令から処理を再開。
リセット	リセット例外	リセットベクタ (H'FFFF_0000 番地) からプログラム処理を実行。電源投入時と同じ処理を実行。

【注】 ソフトウェアスタンバイモードへの遷移処理で、ソフトウェアスタンバイ復帰発振安定時間を設定するために、WDT のカウンタをストップしカウンタの値を変更しています。ソフトウェアスタンバイモードの解除後に WDT を使用する場合は、WDT を再初期化する必要があります。

5.1.3 ディープスタンバイモード

サンプルコードでは、起動メッセージの出力後、ターミナルよりコマンド"LPM_DS"+"1"または"LPM_DS"+"2"を入力すると、ディープスタンバイモードに遷移するための関数 Sample_LPM_Deep_Standby 関数が実行され、ディープスタンバイモードに遷移します。

また、ディープスタンバイモード遷移中に、SUB ボードの SW2 (NMI) または SW3 (IRQ0) を押すと、ディープスタンバイモードが解除され、リセットベクタ (H'FFFF_0000 番地) からディープスタンバイモードの解除後の処理を行います。

FRQCR レジスタの CKOEN[1:0]ビットを b'00 に設定し、ディープスタンバイモード遷移中は CKIO 端子の状態が出力オフ (Hi-z) になるようにしています。

ディープスタンバイモード遷移前と解除後にターミナルにメッセージを出力します。ディープスタンバイモードの出力メッセージは「5.10.3 ディープスタンバイモード遷移コマンドの動作」を参照してください。

(1) ディープスタンバイモード遷移前の処理

ディープスタンバイモードに遷移する前に、解除要因を設定します。

サンプルコードでは、PJ_1 端子および NMI 端子の変化を解除要因とするように設定しています。

表 5.8 ディープスタンバイモード解除要因の設定内容

解除要因	ボード上の資源	設定内容
NMI	SUB ボードの SW2	NMI 端子の立ち下がリエッジで検出
PJ_1	SUB ボードの SW3	PJ_1 端子の立ち下がリエッジで検出

コンフィグレーションテーブル (LPM_SC_TABLE[]) の内容を設定することにより、解除用端子、RTC のアラーム割り込み、USB モジュールの割り込みを解除要因として指定することができます。なお、RTC および USB モジュールを動作させるためのソフトウェアは別途準備する必要があります。

(2) ディープスタンバイモード遷移時の処理

ディープスタンバイモードへの遷移は `R_LPM_DStandbyTransition` 関数をコールすることにより実現しています。`R_LPM_DStandbyTransition` 関数では、引数に指定した低消費電力モードのコンフィグレーションテーブル (`LPM_SC_TABLE[]`) のデータを使用してディープスタンバイモードの解除要因、スタンバイ復帰発振安定時間などのレジスタ設定処理を実行した後、`STBCR1` レジスタの `STBY` ビットに 1、`DEEP` ビットに 1 を設定し、`WFI` 命令を実行します。

本サンプルコードでは、コンフィグレーションテーブル (`LPM_SC_TABLE[]`) の設定内容により、ディープスタンバイモード遷移中の端子状態、解除要因、ディープスタンバイモード復帰発振安定時間、保持用 RAM のデータ保持、ディープスタンバイ解除後のプログラム起動方法を設定することができます。コンフィグレーションテーブル `LPM_SC_TABLE[]` の詳細は表 5.19～表 5.23 を参照してください。

表 5.9 コンフィグレーションテーブルで設定できるディープスタンバイモード項目一覧

設定項目	設定可能な値	サンプルコードでの設定内容
端子状態	ディープスタンバイモード遷移中の端子状態 <ul style="list-style-type: none"> LPM_PIN_STATUS_HIZ 端子状態を保持しない (Hi-z) LPM_PIN_STATUS_RETAINED 端子状態を保持する 	ディープスタンバイモード遷移中は端子状態を保持しない (Hi-z)
解除要因	ディープスタンバイモードの解除要因に PK_4、PK_2、PJ_5、PJ_1、PH_0、PG_6、NMI、PG_2、PH_1、PE_1、P6_2、P3_3、P3_1 の選択と、検出方法の設定 <ul style="list-style-type: none"> LPM_PIN_CANCEL_FACTOR_DISABLED 解除要因に使用しない LPM_PIN_CANCEL_FACTOR_FALLING_EDGE 立ち下がりエッジで解除 LPM_PIN_CANCEL_FACTOR_RISING_EDGE 立ち上がりエッジで解除 	PJ_1 : 立ち下がりエッジ NMI : 立ち下がりエッジ その他の端子は、解除要因に使用しない設定
	ディープスタンバイモードの解除要因に USB0、USB1 の各 DP 端子、DM 端子の使用有無の選択 <ul style="list-style-type: none"> LPM_DS_CANCEL_FACTOR_DISABLED 解除要因として使用しない LPM_DS_CANCEL_FACTOR_ENABLED 信号変化を解除要因に使用する 	USB0、USB1 は解除要因として使用しない設定
	ディープスタンバイモードの解除要因に RTC0、RTC1 のアラーム割り込みの使用有無の選択 <ul style="list-style-type: none"> LPM_DS_CANCEL_FACTOR_DISABLED アラーム割り込みを解除要因に使用しない LPM_DS_CANCEL_FACTOR_ENABLED アラーム割り込みを解除要因に使用する 	RTC1、RTC0 は解除要因として使用しない設定
スタンバイ遷移中の保持用 RAM の内容の保持・非保持	保持用内蔵 RAM ページ 0~3 について、ディープスタンバイ中に保持 RAM の領域を保持するかしないかを選択 <ul style="list-style-type: none"> LPM_RETENTION_RAM_NOT_RETAINED 保持しない LPM_RETENTION_RAM_RETAINED 保持する 	保持用 RAM ページ 0 : 保持する 保持用 RAM ページ 1 : 保持しない 保持用 RAM ページ 2 : 保持しない 保持用 RAM ページ 3 : 保持しない
ディープスタンバイモード復帰発振安定時間	ディープスタンバイモード復帰発振安定時間を ms 単位で設定 (小数点 2 桁まで設定可能)	7.0ms
ディープスタンバイモード解除後のプログラム起動方法	ディープスタンバイモード解除後のプログラム起動方法の選択 <ul style="list-style-type: none"> LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT 外部メモリからプログラム起動 LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 保持用 RAM からプログラム起動(外部メモリ端子状態を保持) LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1 保持用 RAM からプログラム起動(外部メモリ端子状態を保持しない) 	外部メモリからプログラム起動または保持用 RAM メモリからプログラム起動

(3) ディープスタンバイモード解除後の処理

図 5.1 にディープスタンバイモード解除時のサンプルコード動作を示します。ディープスタンバイモードの解除後、解除要因によらず、リセットベクタ (H'FFFF_0000 番地) に配置されたブートプログラムを実行します。ディープスタンバイ遷移前に設定したディープスタンバイコントロールレジスタ (DSCTR) の RAMBOOT ビットおよび EBUSKEEPE ビットの設定内容によって、解除後の動作が異なります。

- 外部メモリ起動モード

RAMBOOT ビットに"0"を設定してディープスタンバイモードに遷移した場合、スタンバイ解除後にブートプログラムを実行し、ブートモードに対応した外部メモリから起動します。サンプルコードでは、ブートモード3を使用しており、ブートプログラム実行後に H'2000_0000 番地に分岐してローダプログラムを実行します。ローダプログラム実行後、H'2001_0000 番地に配置しているアプリケーションプログラムに分岐します。RAMBOOT が"0"の場合は、IOKEEP ビットをクリアするまで外部メモリ制御端子以外の端子状態が保持されているため、サンプルコードではローダプログラムで IOKEEP ビットをクリアする処理を行っています。

- 保持用内蔵 RAM 起動モード

RAMBOOT ビットに"1"を設定してディープスタンバイモードに遷移した場合、スタンバイ解除後にブートプログラムを実行し、保持用内蔵 RAM の先頭アドレス (H'8000_0000 番地) に分岐します。サンプルコードでは、保持用内蔵 RAM の先頭アドレスに配置した LPM_DSStandby_ReturnRamBoot 関数が実行されます。RAMBOOT ビットが"1"の場合は、ディープスタンバイモード遷移前に、保持用内蔵 RAM の領域 (H'8000_0000 番地から 128KB の領域) にディープスタンバイモード解除後に動作するプログラムを配置しておく必要があります。サンプルコードでは、INITSCT 関数のセクション初期化処理により、ディープスタンバイモード解除後に実行するプログラムを保持用内蔵 RAM へ転送しています。

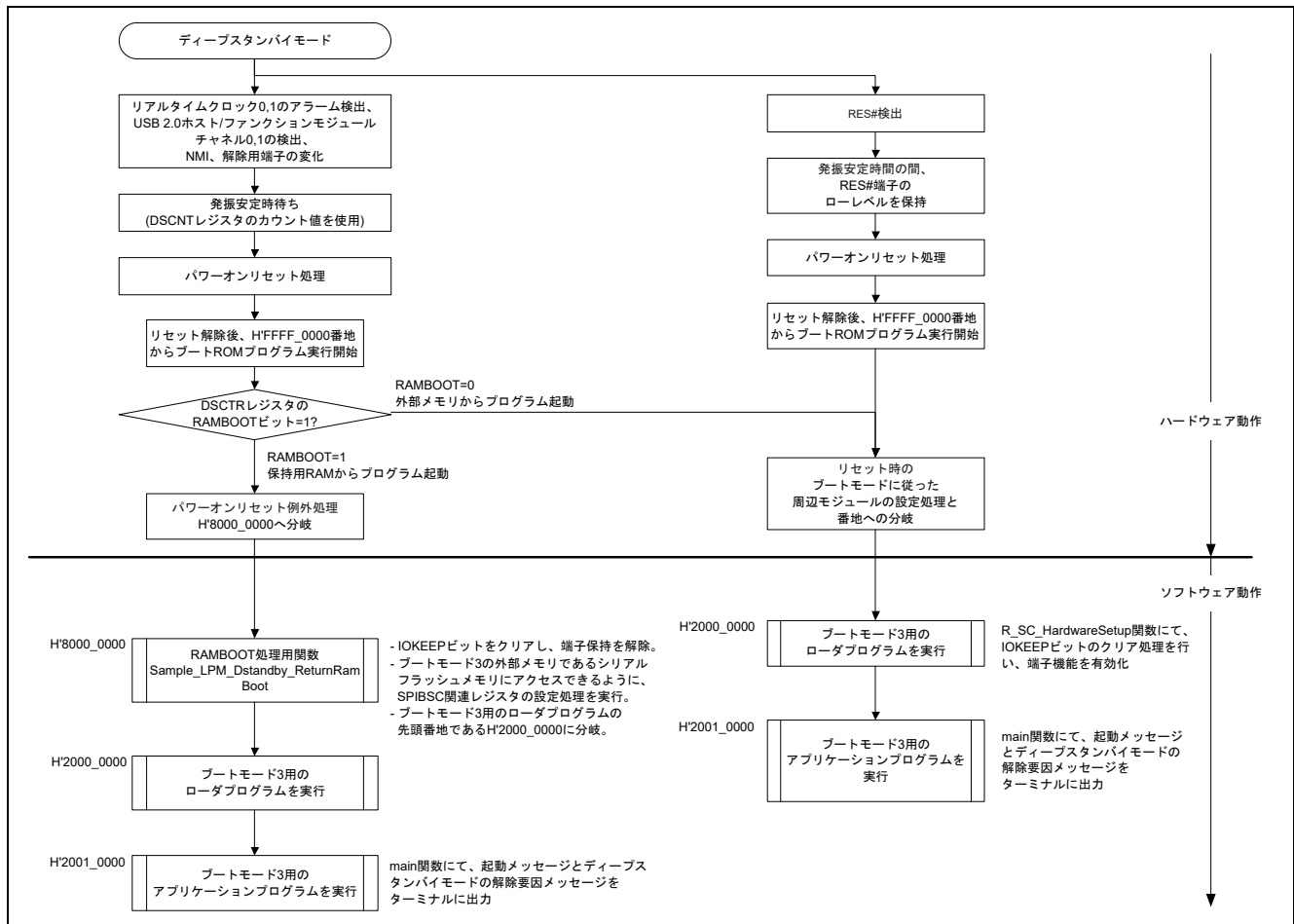


図 5.2 にディープスタンバイ解除後に保持用内蔵 RAM から起動する場合のプログラム配置イメージを示します。RAMBOOT ビットを"1"に設定してディープスタンバイモードを解除する場合、ディープスタンバイモード解除後に内蔵 ROM に格納されたブートプログラム実行後のプログラムのエントリポイントは、保持用内蔵 RAM の先頭アドレスである H'8000_0000 番地です。このため、ディープスタンバイモードに移移する前に、スタンバイ解除後に実行するプログラムを H'8000_0000 番地に格納する必要があります。

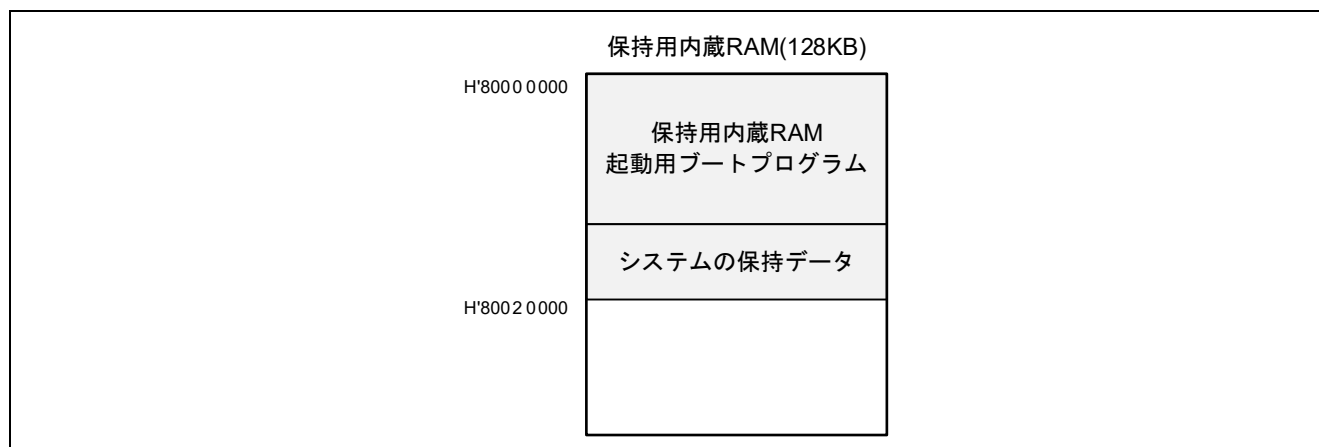


図 5.2 ディープスタンバイ解除後に保持用内蔵 RAM から起動する場合のプログラム配置イメージ

なお、RAMBOOT ビットを"1"に設定してディープスタンバイモードを解除する場合、内蔵 ROM のブートプログラムは、ブートモードに応じた各周辺モジュールのレジスタの設定処理を実行しないため、外部メモリにアクセスするための処理を、H'8000_0000 番地に格納する処理で実施する必要があります。ブートモード 3 で動作するシステムでは、SPI マルチ I/O バスコントローラに接続されたシリアルフラッシュメモリにアクセスできるようにするために、表 5.10 に示す周辺モジュールのレジスタ設定処理を実施する必要があります。

表 5.10 保持用 RAM からの復帰時にシリアルフラッシュメモリにアクセスするための周辺モジュール設定

周辺モジュール	レジスタ	レジスタ設定値
低消費電力モード	STBCR8	H'F7
クロックパルス発振器	SCLKSEL	H'0000
SPI マルチ I/O バスコントローラ	CMNCR	H'01AA A200
	SSLDR	H'0000 0000
	DRCR	H'0003 0100
	DRCMR	H'0003 0000
	DRENr	H'0000 4700
	DRDMCR	H'0000 0000
	DRDRENr	H'0000 0000
	PHYCNT	H'0000 0260
	PHYOFFSET1	H'3151 1144
汎用入出力ポート	PHYINT	H'0707 0002
	PPOC	H'0000 0101
	PSPIBSC	H'0555 5555

5.2 サンプルコード実行時の周辺機能の設定およびメモリ配置

5.2.1 周辺機能の設定

表 5.11 および表 5.12 にサンプルコード実行時の周辺機能の設定内容を示します。

表 5.11 周辺機能の設定内容 (1/2)

モジュール	設定内容
CPG	<p>CPU クロック : PLL 回路の周波数に対して $\times 1/2$ 倍に設定 内部バスクロック : PLL 回路の周波数に対して $\times 1/8$ 倍に設定 周辺クロック 1 (P1ϕ) : PLL 回路の周波数に対して $\times 1/16$ 倍に設定</p> <p>クロックモード 1 (分周器 1 : $\times 1/2$ 倍、PLL 回路 : $\times 88$ 倍) で、入力クロックが 24MHz の場合に以下の周波数となるように設定</p> <ul style="list-style-type: none"> • CPU クロック (Iϕ) : 528MHz • 画像処理クロック (Gϕ) : 264MHz • 内部バスクロック (Bϕ) : 132MHz • 周辺クロック 1 (P1ϕ) : 66MHz • 周辺クロック 0 (P0ϕ) : 33MHz • QSPI0_SPCLK : 66MHz (Bϕ 選択時) • CKIO クロック : 132MHz (Bϕ 選択時)
SPIBSC	<p>外部アドレス空間リードモードに設定し、CPU が SPI マルチ I/O バス空間に接続されたシリアルフラッシュメモリから、直接リードするための信号を生成するための設定</p>
OSTM	<p>チャンネル 0 およびチャンネル 2 をインターバルタイマモードに設定</p> <ul style="list-style-type: none"> • チャンネル 0 P1ϕ=66MHz の時に 500ms ごとに割り込み要求を発生するように、タイマカウントを設定 • チャンネル 2 P1ϕ=66MHz の時に 1ms ごとに割り込み要求を発生するように、タイマカウントを設定
SCIFA	<p>チャンネル 4 を調歩同期式モードに設定</p> <ul style="list-style-type: none"> • データ長 : 8 ビット • ストップビット長 : 1 ビット • パリティ : なし • データ転送方向 : LSB ファースト転送 <p>P1ϕ=66MHz の時に、クロックソースを分周なし、ボーレートジェネレータは倍速モード、ビットレートの 8 倍の基本クロックで動作するように設定。ビットレートが 115200bps となるように、ビットレート値に 71 を設定 (誤差 : -0.53%)</p>

表 5.12 周辺機能の設定内容 (2/2)

モジュール	設定内容
STB	<ul style="list-style-type: none"> 保持用内蔵 RAM へのライト許可 STBCR3, STBCR4, STBCR8 で OSTM0, OSTM2, SCIFA4, SPIBSC にクロックを供給 スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモードへの遷移処理および解除処理に使用 ディープスタンバイモードの解除要因として、PJ_1 端子（立ち上がりエッジ）および NMI 端子（立ち下がりエッジ）を設定 ディープスタンバイモードの復帰発振安定時間を 7ms に設定
GPIO	PORT6、PORT9、PORTJ の兼用端子機能を設定 <ul style="list-style-type: none"> P6_0 : LED の点灯および消灯 P9_1 : RxD4、P9_0 : TxD4 PJ_1 : IRQ0 端子機能に設定し、スリープモードおよびソフトウェアスタンバイモード解除用として使用。また、PJ_1 端子はディープスタンバイモード解除要因としても使用
INTC	INTC の初期設定および下記の割り込みについてハンドラの登録と実行 <ul style="list-style-type: none"> OSTM チャンネル 0 割り込み（割り込み ID : 88） OSTM チャンネル 2 割り込み（割り込み ID : 90） SCIFA チャンネル 4 割り込み（割り込み ID : 322,323） IRQ0 端子割り込み（割り込み ID : 36）
WDT	ソフトウェアスタンバイモードの復帰発振安定時間を 7ms に設定

5.2.2 メモリマップ

図 5.3 に RZ/A2M グループのアドレス空間と RZ/A2M CPU ボードのメモリマップを示します。

サンプルコードでは、ROM 領域を使用するコードおよびデータを SPI マルチ I/O バス空間に接続したシリアルフラッシュメモリに配置し、RAM 領域を使用するコードおよびデータを大容量内蔵 RAM に配置するようにしています。また、ディープスタンバイモード解除後に保持用内蔵 RAM から実行する場合のプログラム処理は、保持用内蔵 RAM（128KB）の H'8000_0000 番地以降に配置するようにしています。

	RZ/A2Mグループの アドレス空間	RZ/A2M CPUボード メモリマップ
H'FFFF FFFF	内蔵IO領域 および 予約領域 (2044MB)	内蔵IO領域 および 予約領域 (2044MB)
H'8040 0000		
H'8002 0000	大容量内蔵RAM (4MB)	大容量内蔵RAM (4MB)
H'8000 0000		
		} 保持用内蔵RAM(128KB)
	予約領域 (256MB)	予約領域 (256MB)
H'7000 0000		
H'6100 0000	OctaRAM™空間 (256MB)	-
H'6000 0000		
H'5400 0000	OctaFlash™空間 (256MB)	-
H'5000 0000		
H'4080 0000	HyperRAM™空間 (256MB)	-
H'4000 0000		HyperRAM™ (8MB)
H'3400 0000	HyperFlash™空間 (256MB)	-
H'3000 0000		HyperFlash™ (64MB)
H'2400 0000	SPIマルチI/Oバス 空間 (256MB)	-
H'2000 0000		シリアルフラッシュ メモリ (64MB)
H'1800 0000	内蔵IO領域および 予約領域 (128MB)	内蔵IO領域および 予約領域 (128MB)
H'1400 0000	CS5空間 (64MB)	-
H'1000 0000	CS4空間 (64MB)	-
H'0C00 0000	CS3空間 (64MB)	-
H'0800 0000	CS2空間 (64MB)	-
H'0400 0000	CS1空間 (64MB)	-
H'0000 0000	CS0空間 (64MB)	-

図 5.3 メモリマップ

5.2.3 サンプルコードのセクション配置

サンプルコードでは、割り込み処理の高速化のため、例外処理ベクタテーブルと IRQ 割り込みハンドラを大容量内蔵 RAM 上に配置して、これらの処理を大容量内蔵 RAM 上で実行するようにしています。例外処理ベクタテーブルおよび IRQ 割り込みハンドラのプログラムデータをシリアルフラッシュメモリ領域から大容量内蔵 RAM 領域への転送処理、初期値なしデータセクションのゼロクリア処理、および初期値ありデータセクションの初期化処理は、INITSCT 関数を使用して行っています。INITSCT 関数では、section.c で定義されているセクション初期化用のテーブルデータを参照して、各セクションの初期化処理を行います。プログラムデータの配置は、リンカスクリプト (linker_script.ld) に記述しています。

なお、ディープスタンバイモード解除後に保持用内蔵 RAM から実行する場合のプログラム処理を、保持用内蔵 RAM に転送する処理も、INITSCT 関数を使用して行っています。

表 5.13 にサンプルコードで使用するセクション名とオブジェクト名一覧を示します。図 5.4 にサンプルコードの初期状態のセクション配置（ロードビュー）と、INITSCT 関数実行後のセクション配置（実行ビュー）を示します。

表 5.13 サンプルコードで使用するセクション名とオブジェクト名一覧

出力セクション名	入力セクション名 入力オブジェクト名	内容	ロード 領域	実行 領域
LOAD_MODULE 1	VECTOR_TABLE	例外処理ベクタテーブル	FLASH	FLASH
LOAD_MODULE 2	*r_cpg/*.o (.text .rodata .data)	CPG の設定処理	FLASH	LRAM
	*rza_io_regrw.o (.text .rodata .data)	IO レジスタアクセス処理		
	hwsetup.o (.text .rodata .data)	HardwareSetup の設定処理		
LOAD_MODULE 3	*r_cpg/*.o (.bss)	CPG の設定処理の初期値なしデー タ領域	-	LRAM
	*rza_io_regrw.o (.bss)	IO レジスタアクセス処理用の初期 値なしデータ領域		
LOAD_MODULE 4	RESET_HANDLER	リセット処理	FLASH	FLASH
	INIT_SECTION */sections.o	セクション初期化処理		
LOAD_MODULE 5	*lpm_deepstandby_ramboot.o (.text .rodata .data)	ディープスタンバイモード解除後に 保持用内蔵 RAM から実行する処理	FLASH	LRAM (注 3)
.data	VECTOR_MIRROR_TABLE	例外処理ベクタテーブル	FLASH	LRAM
	r_intc_.o (.text .rodata .data)	INTC ドライバ処理用コード領域		
	IRQ_FIQ_HANDLER	IRQ/FIQ ハンドラ処理		
.bss	なし	なし	-	LRAM
.uncached_RAM	*r_cache_*.o (.bss)	L1 および L2 キャッシュ設定処理用 初期値なしデータ領域 (注 2)	-	LRAM
	UNCACHED_BSS	初期値なしデータ領域 (非キャッ シュ設定)		
.uncached_RAM2	*r_cache_*.o (.text .rodata .data)	L1 および L2 キャッシュ設定処理 (注 2)	FLASH	LRAM
	UNCACHED_DATA	初期値ありデータ領域 (非キャッ シュ設定)		
.mmu_page_table	なし	MMU 変換テーブル領域	-	LRAM
.stack	なし	システムモードのスタック領域	-	LRAM
		IRQ モードのスタック領域		
		FIQ モードのスタック領域		
		SVC モードのスタック領域		
		アボート(ABT)モードの スタック領域		
.text2	*(.text .text.*)	デフォルトのコード領域	FLASH	FLASH
	(.rodata .rodata.)	デフォルトの定数データ領域		
.data2	*(.data .data.*)	デフォルトの初期値ありデータ領域	FLASH	LRAM
.bss2	*(.bss .bss.*) *(COMMON)	デフォルトの初期値なしデータ領域	-	LRAM
.heap	なし	ヒープ領域	-	LRAM

- 【注】 1. 表中のロード領域および実行領域において、FLASH はシリアルフラッシュメモリの領域を、LRAM は大容量内蔵 RAM の領域を表します。
2. このセクションは、キャッシュ無効領域に配置する必要があります。
3. 大容量内蔵 RAM の保持用内蔵 RAM (H'8000_0000 番地～H'8001_FFFF 番地) を示します。

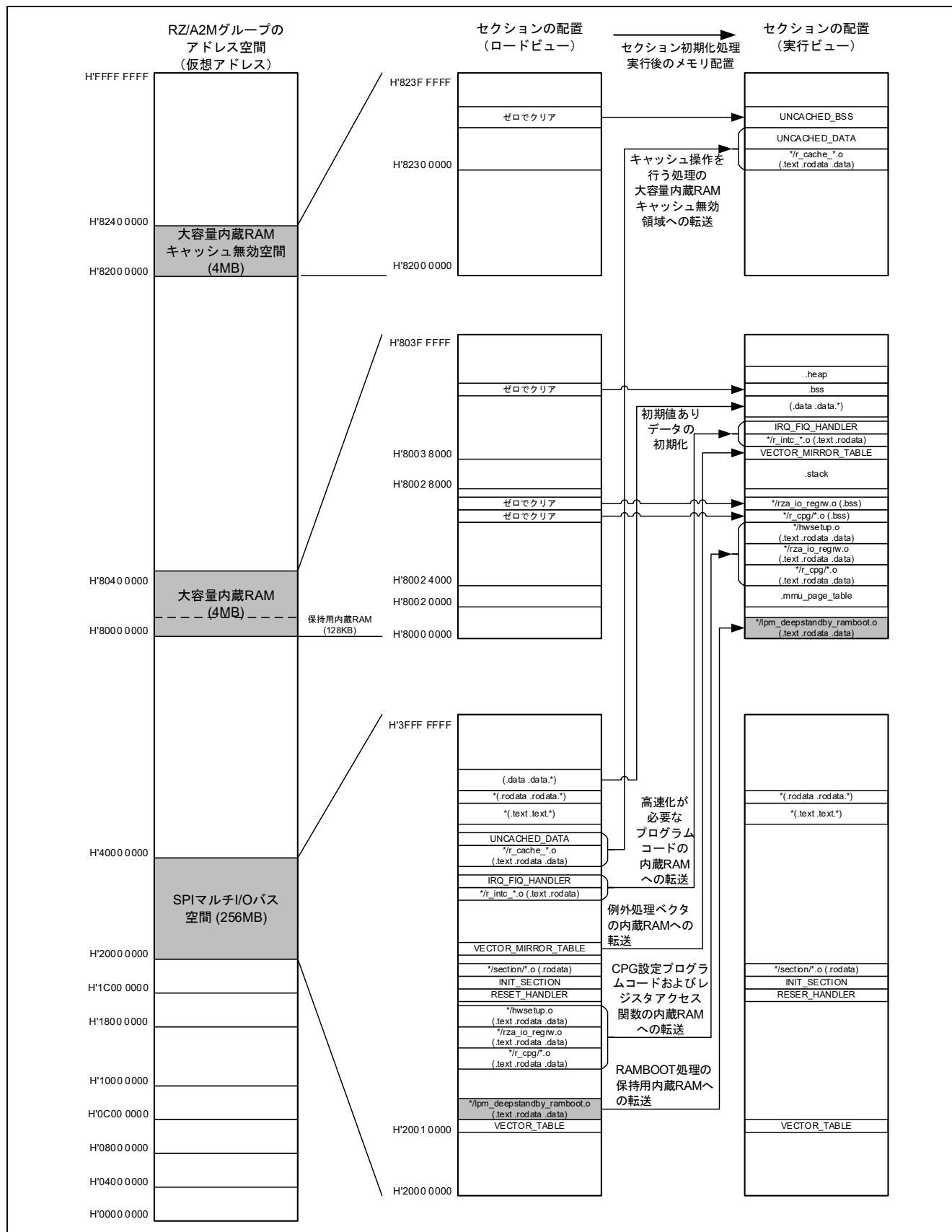


図 5.4 セクション配置

5.3 使用割り込み一覧

表 5.14 にサンプルコードで使用する割り込みを示します。

表 5.14 サンプルコードで使用する割り込み

割り込み要因（要因 ID）	優先度	処理概要
OSTM0（88）	3	500ms ごとに割り込みを発生
OSTM2（90）	30	1ms ごとに割り込みを発生 （OS Abstraction Layer による時間管理に使用）
RXI4（322）	30	SCIFA の TXI4 割り込みを発生
TXI4（323）	30	SCIFA の RXI4 割り込みを発生
IRQ0（36）	1	IRQ0 端子の立ち下がりエッジの検出により割り込みを発生。スリープモードおよびソフトウェアスタンバイモードの解除要因に使用
NMI	—	NMI 端子の立ち下がりエッジの検出により割り込みを発生。スリープモードおよびソフトウェアスタンバイモードの解除要因に使用

【注】 スリープモード、ソフトウェアスタンバイモードに遷移中に、NMI および IRQ0 割り込み以外の割り込み要求を受け付けないように、スタンバイ遷移前に割り込みマスクレベルに"2"を設定しています。スリープモード、ソフトウェアスタンバイモードを解除した後、他の割り込みを許可するために、割り込みマスクレベルを"31"に戻します。

5.4 データ型一覧

表 5.15 にサンプルコードで使用するデータ型一覧を示します。

表 5.15 サンプルコードで使用するデータ型一覧

シンボル	内容
char_t	8 ビット文字
bool_t	論理型。値は true（1）、false（0）
int_t	高速な整数、符号あり、本サンプルコードでは 32 ビット整数。
int8_t	8 ビット整数、符号あり（標準ライブラリ stdint.h にて定義）
int16_t	16 ビット整数、符号あり（標準ライブラリ stdint.h にて定義）
int32_t	32 ビット整数、符号あり（標準ライブラリ stdint.h にて定義）
int64_t	64 ビット整数、符号あり（標準ライブラリ stdint.h にて定義）
uint8_t	8 ビット整数、符号なし（標準ライブラリ stdint.h にて定義）
uint16_t	16 ビット整数、符号なし（標準ライブラリ stdint.h にて定義）
uint32_t	32 ビット整数、符号なし（標準ライブラリ stdint.h にて定義）
uint64_t	64 ビット整数、符号なし（標準ライブラリ stdint.h にて定義）
float32_t	32 ビット浮動小数
float64_t	64 ビット浮動小数
float128_t	128 ビット浮動小数

5.5 定数一覧

表 5.16～表 5.18 に低消費電力モードのサンプルプログラムで使用する定数を示します。これらの定数は、コンフィグレーションテーブル LPM_SC_TABLE[] の各メンバ変数を設定する時に使用します。コンフィグレーションテーブル LPM_SC_TABLE[] の詳細は表 5.19～表 5.23 を参照してください。

表 5.16 低消費電力モードのサンプルプログラムで使用する定数 (1/3)

定数名	設定値	内容
LPM_MODE_SLEEP	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の mode メンバの設定に使用します。 使用する低消費電力モードのテーブルのフォーマットを指定します。 <ul style="list-style-type: none"> スリープモードに遷移する場合 LPM_MODE_SLEEP を設定 ソフトウェアスタンバイモードに遷移する場合 LPM_MODE_SOFTWARE_STANDBY を設定 ディープスタンバイモードに遷移する場合 LPM_MODE_DEEP_STANDBY を設定
LPM_MODE_SOFTWARE_STANDBY	(1)	
LPM_MODE_DEEP_STANDBY	(2)	
LPM_SSTANDBY_DISABLE_IRQ	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ss_cancel_by_irq メンバの設定に使用します。 ソフトウェアスタンバイモードの解除要因に IRQ 端子を使用するかどうかを指定します。 <ul style="list-style-type: none"> IRQ 端子を解除要因に使用しない場合 LPM_SSTANDBY_DISABLE_IRQ IRQ 端子を解除要因に使用する場合 LPM_SSTANDBY_ENABLE_IRQ
LPM_SSTANDBY_ENABLE_IRQ	(1)	
LPM_PIN_CANCEL_FACTOR_DISABLED	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ds_cancel_by_pk_4, ds_cancel_by_pk_2, ds_cancel_by_pj_5, ds_cancel_by_pj_1, ds_cancel_by_ph_0, ds_cancel_by_pg_6, ds_cancel_by_pg_2, ds_cancel_by_ph_1, ds_cancel_by_pe_1, ds_cancel_by_p6_2, ds_cancel_by_p3_3, ds_cancel_by_p3_1 メンバの設定に使用します。 ディープスタンバイモードの解除要因に端子変化を使用するかどうか、使用する場合の検出方法を指定します。 <ul style="list-style-type: none"> 端子変化を解除要因に使用しない場合 LPM_PIN_CANCEL_FACTOR_DISABLED 端子変化を解除要因に使用し立ち下がりエッジに設定する場合 LPM_PIN_CANCEL_FACTOR_FALLING_EDGE 端子変化を解除要因に使用し立ち上がりエッジに設定する場合 LPM_PIN_CANCEL_FACTOR_RISING_EDGE
LPM_PIN_CANCEL_FACTOR_FALLING_EDGE	(1)	
LPM_PIN_CANCEL_FACTOR_RISING_EDGE	(2)	

表 5.17 低消費電力モードのサンプルプログラムで使用する定数 (2/3)

定数名	設定値	内容
LPM_SS_CANCEL_FACTOR_DISABLED	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ss_cancel_by_usb0_cc2_rd, ss_cancel_by_usb0_cc1_rd, ss_cancel_by_usb0_ovrclr, ss_cancel_by_usb0_vbusin, ss_cancel_by_usb0_dm, ss_cancel_by_usb0_dp, ss_cancel_by_usb1_cc2_rd, ss_cancel_by_usb1_cc1_rd, ss_cancel_by_usb1_ovrclr, ss_cancel_by_usb1_vbusin, ss_cancel_by_usb1_dm, ss_cancel_by_usb1_dp メンバの設定に使用します。 ソフトウェアスタンバイモードの解除要因に USB モジュールの割り込みを使用するかどうかを指定します。 <ul style="list-style-type: none"> • USB モジュールを解除要因に使用しない場合 LPM_SS_CANCEL_FACTOR_DISABLED • USB モジュールを解除要因に使用する場合 LPM_SS_CANCEL_FACTOR_ENABLED
LPM_SS_CANCEL_FACTOR_ENABLED	(1)	
LPM_DS_CANCEL_FACTOR_DISABLED	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ds_cancel_by_rtc0, ds_cancel_by_rtc1, ds_cancel_by_usb0_dm, ds_cancel_by_usb0_dp, ds_cancel_by_usb1_dm, ds_cancel_by_usb1_dp メンバの設定に使用します。 ディープスタンバイモードの解除要因に RTC アラーム割り込み, USB モジュールの割り込みを使用するかどうかを指定します。 <ul style="list-style-type: none"> • RTC, USB モジュールを解除要因に使用しない場合 LPM_DS_CANCEL_FACTOR_DISABLED • UTC, USB モジュールを解除要因に使用する場合 LPM_DS_CANCEL_FACTOR_ENABLED
LPM_DS_CANCEL_FACTOR_ENABLED	(1)	
LPM_PIN_STATUS_RETAINED	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の pin_status メンバの設定に使用します。 ソフトウェアスタンバイモード、ディープスタンバイモード遷移中の端子状態を指定します。 <ul style="list-style-type: none"> • スタンバイモード遷移中の端子状態を保持する場合 LPM_PIN_STATUS_RETAINED • スタンバイモード遷移中の端子状態を Hi-z 状態にする場合 LPM_PIN_STATUS_HIZ
LPM_PIN_STATUS_HIZ	(1)	

表 5.18 低消費電力モードのサンプルプログラムで使用する定数 (3/3)

定数名	設定値	内容
LPM_RETENTION_RAM_NOT_RETAINED	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ds_retention_ram_page0～ds_retention_ram_page3 メンバの設定に使用します。 ディープスタンバイモード遷移中に、保持用内蔵 RAM の内容を保持するかどうかを指定します。 <ul style="list-style-type: none"> 保持用 RAM の内容を保持しない場合 LPM_RETENTION_RAM_NOT_RETAINED 保持用 RAM の内容を保持する場合 LPM_RETENTION_RAM_RETAINED
LPM_RETENTION_RAM_RETAINED	(1)	
LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT	(0)	コンフィグレーションテーブル LPM_SC_TABLE[] の ds_reboot メンバの設定に使用します。 ディープスタンバイ解除後の起動方法を指定します。 <ul style="list-style-type: none"> 外部メモリから起動する場合 LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT 保持用内蔵 RAM から起動する（外部メモリ端子状態は保持しない）場合 LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 保持用内蔵 RAM から起動する（外部メモリ端子状態は保持する）場合 LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1
LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0	(1)	
LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1	(2)	

5.6 構造体/共用体一覧

表 5.19～表 5.23 に低消費電力モードのコンフィグレーションテーブルを示します。コンフィグレーションテーブルは、スタンバイモードの解除要因、スタンバイ遷移中の状態、解除後の動作を指定するために使用します。コンフィグレーションテーブルの低消費電力モードの定義は、スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモードのそれぞれについて 1 つのみ定義可能です。

表 5.19 コンフィグレーション構造体 (st_r_drv_lpm_sc_config_t) (1/5)

分類	メンバ名	内容
全モード共通	e_r_drv_lpm_mode_t mode	使用する低消費電力モードのテーブルのフォーマットを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_MODE_SLEEP LPM_MODE_SOFTWARE_STANDBY LPM_MODE_DEEP_STANDBY
ソフトウェアスタンバイ、ディープスタンバイ共通	e_r_drv_lpm_pin_status_t pin_status	ソフトウェアスタンバイ、ディープスタンバイ遷移中の端子状態を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_STATUS_RETAINED LPM_PIN_STATUS_HIZ
ソフトウェアスタンバイ	e_r_drv_lpm_cancel_factor_irq_t ss_cancel_by_irq	解除要因に IRQ 端子を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SSTANDBY_DISABLE_IRQ LPM_SSTANDBY_ENABLE_IRQ
	e_r_drv_lpm_cancel_factor2_usb0_cc2_rd_t ss_cancel_by_usb0_cc2_rd	解除要因に USB チャンネル 0 の CC2_RD の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_factor2_usb0_cc1_rd_t ss_cancel_by_usb0_cc1_rd	解除要因に USB チャンネル 0 の CC1_RD の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_factor2_usb0_ovrclr_t ss_cancel_by_usb0_ovrclr	解除要因に USB チャンネル 0 の OVRCLR の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_factor2_usb0_vbusin_t ss_cancel_by_usb0_vbusin	解除要因に USB チャンネル 0 の VBUSIN の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_factor2_usb0_dm_t ss_cancel_by_usb0_dm	解除要因に USB チャンネル 0 の DM の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED

表 5.20 コンフィグレーション構造体 (st_r_drv_lpm_sc_config_t) (2/5)

分類	メンバ名	内容
ソフトウェア スタンバイ	e_r_drv_lpm_cancel_factor2_t ss_cancel_by_usb0_dp	解除要因に USB チャンネル 0 の DP の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_cc2_rd	解除要因に USB チャンネル 1 の CC2_RD の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_cc1_rd	解除要因に USB チャンネル 1 の CC1_RD の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_ovrclr	解除要因に USB チャンネル 1 の OVRCLR の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_vbusin	解除要因に USB チャンネル 1 の VBUSIN の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_dm	解除要因に USB チャンネル 1 の DM の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_dp	解除要因に USB チャンネル 1 の DP の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	float64_t ss_osc_stab_time	ソフトウェアスタンバイモード復帰発振安定時間を ms 単位で指定します (小数点 2 桁まで有効)。 <ul style="list-style-type: none"> 設定可能な値 : 1 以上の小数

表 5.21 コンフィグレーション構造体 (st_r_drv_lpm_sc_config_t) (3/5)

分類	メンバ名	内容
ディープ スタンバイ	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pk_4	解除要因に PK_4 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pk_2	解除要因に PK_2 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pj_5	解除要因に PJ_5 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pj_1	解除要因に PJ_1 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_ph_0	解除要因に PH_0 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pg_6	解除要因に PG_6 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_nmi	解除要因に NMI 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_rtc1	解除要因に RTC チャンネル 1 のアラーム割り込みを使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED

表 5.22 コンフィグレーション構造体 (st_r_drv_lpm_sc_config_t) (4/5)

分類	メンバ名	内容
ディープ スタンバイ	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_rtc0	解除要因に RTC チャンネル 0 のアラーム割り込みを使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pg_2	解除要因に PG_2 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_ph_1	解除要因に PH_1 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pe_1	解除要因に PE_1 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p6_2	解除要因に P6_2 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p3_3	解除要因に P3_3 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p3_1	解除要因に P3_1 端子の信号変化を使用するかどうか、使用する場合は検出方法を指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_usb1_dm	解除要因に USB チャンネル 1 の DM の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値 : LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED

表 5.23 コンフィグレーション構造体 (st_r_drv_lpm_sc_config_t) (5/5)

分類	メンバ名	内容
ディープスタンバイ	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb1_dp	解除要因に USB チャンネル 1 の DP の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb0_dm	解除要因に USB チャンネル 0 の DM の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb0_dp	解除要因に USB チャンネル 0 の DP の信号変化を使用するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page3	ディープスタンバイモード遷移中に保持用内蔵 RAM ページ 3 の内容を保持するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page2	ディープスタンバイモード遷移中に保持用内蔵 RAM ページ 2 の内容を保持するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page1	ディープスタンバイモード遷移中に保持用内蔵 RAM ページ 1 の内容を保持するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page0	ディープスタンバイモード遷移中に保持用内蔵 RAM ページ 0 の内容を保持するかどうかを指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	float64_t ds_osc_stab_time	ディープスタンバイモード復帰発振安定時間を ms 単位で指定します (小数点 2 桁まで有効)。 <ul style="list-style-type: none"> 設定可能な値: 1 以上の小数
	e_r_drv_lpm_reboot_type_t ds_reboot_type	ディープスタンバイモード解除後の起動方法を指定します。 <ul style="list-style-type: none"> 設定可能な値: LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1

5.7 関数一覧

サンプルコードは、周辺機能を使用するためのインタフェース関数（API 関数）、サンプルコードを動作させるために必要なサンプル関数から構成されています。

表 5.24 にサンプル関数一覧を、表 5.25 に API 関数一覧を示します。

表 5.24 サンプル関数一覧

関数名	概要
Sample_LPM_Sleep_Mode	スリープモードへの遷移サンプル関数
Sample_LPM_Software_Standby	ソフトウェアスタンバイモードへの遷移サンプル関数
Sample_LPM_Deep_Standby	ディープスタンバイモードへの遷移サンプル関数
Sample_NMI_Interrupt	NMI 端子割り込みサンプル関数
Sample_IRQ0_Interrupt	IRQ0 端子割り込みサンプル関数
Sample_LPM_DStandby_ReturnRamBoot	保持用内蔵 RAM からのディープスタンバイモード解除後の処理

表 5.25 API 関数一覧

関数名	概要
R_LPM_Transition	低消費電力モードへの遷移関数
R_LPM_SleepTransition	スリープモードへの遷移関数
R_LPM_SSStandbyTransition	ソフトウェアスタンバイモードへの遷移関数
R_LPM_DStandbyTransition	ディープスタンバイモードへの遷移関数
R_INTC_SetIrqMask	ソフトウェアスタンバイモードの IRQ 端子割り込み要求のマスク制御
R_INTC_SetNMIConfig	NMI 端子割り込み要求の検出方法の設定
R_IRQ_SetSense	IRQ 端子割り込み要求の検出方法の設定

5.8 関数仕様

本サンプルコードで使用する関数仕様を示します。

5.8.1 サンプル関数仕様

Sample_LPM_Sleep_Mode	
概 要	スリープモードへの遷移サンプル関数
宣 言	int_t Sample_LPM_Sleep_Mode(void)
説 明	スリープモードに遷移するためのサンプル関数です。 ターミナルより、コマンド文字列"LPM_SM"を入力すると、本関数が実行されます。スリープモードの解除要因として、IRQ0 端子割り込み、NMI 端子割り込みの設定を行い、R_LPM_Transition 関数をコールしてスリープモードへの遷移処理を行います（上記の関数から R_LPM_SleepTransition 関数をコールしています）。
引 数	なし
リターン値	LPM_SUCCESS : 正常終了 LPM_ERROR : エラー終了

Sample_LPM_Software_Standby	
概 要	ソフトウェアスタンバイモードへの遷移サンプル関数
宣 言	int_t Sample_LPM_Software_Standby(st_r_drv_lpm_sc_config_t *p_config_tbl)
説 明	ソフトウェアスタンバイモードに遷移するためのサンプル関数です。 ターミナルより、コマンド文字列"LPM_SS"を入力すると、本関数が実行されます。ソフトウェアスタンバイモードの解除要因として、IRQ0 端子割り込み、NMI 端子割り込みの設定を行い、R_LPM_Transition 関数をコールしてソフトウェアスタンバイモードへの遷移処理を行います（上記の関数から R_LPM_SSstandbyTransition 関数をコールしています）。
引 数	st_r_drv_lpm_sc_con コンフィグレーションテーブルのポインタ fig_t *p_config_tbl
リターン値	LPM_SUCCESS : 正常終了 LPM_ERROR : エラー終了

Sample_LPM_Deep_Standby	
概 要	ディープスタンバイモードへの遷移サンプル関数
宣 言	int_t Sample_LPM_Deep_Standby(st_r_drv_lpm_sc_config_t *p_config_tbl)
説 明	ディープスタンバイモードに遷移するためのサンプル関数です。 ターミナルより、コマンド文字列"LPM_DS"+"1"または"LPM_DS"+"2"を入力すると、本関数が実行されます。ディープスタンバイモードの解除要因として、PJ_1 端子の立ち下がリエッジ、NMI 端子の立ち下がリエッジの設定を行い、R_LPM_Transition 関数をコールしてディープスタンバイモードへの遷移処理を行います（上記の関数から R_LPM_DSstandbyTransition 関数をコールしています）。
引 数	st_r_drv_lpm_sc_con コンフィグレーションテーブルのポインタ fig_t *p_config_tbl
リターン値	LPM_SUCCESS : 正常終了 LPM_ERROR : エラー終了

Sample_NMI_Interrupt	
概 要	NMI 端子割り込みサンプル関数
宣 言	void Sample_NMI_Interrupt(uint32_t int_sense)
説 明	スリープモード、ソフトウェアスタンバイモードの解除時に実行される NMI 端子割り込みのサンプル関数です。 サンプルコードでは、割り込みコントロールレジスタ 0 (IRQ0) の NMIF ビットに "0" をライトし、NMI 割り込み要求をクリアする処理を行います。
引 数	uint32_t int_sense : 未使用
リターン値	なし

Sample_IRQ0_Interrupt	
概 要	IRQ0 端子割り込みサンプル関数
宣 言	void Sample_IRQ0_Interrupt(uint32_t int_sense)
説 明	スリープモード、ソフトウェアスタンバイモードの解除時に実行される IRQ0 端子割り込みのサンプル関数です。 サンプルコードでは、IRQ 割り込み要求レジスタ (IRQRR) の IRQ0F ビットに "0" をライトし、IRQ0 割り込み要求をクリアする処理を行います。
引 数	uint32_t int_sense : 未使用
リターン値	なし
注意事項	複数の IRQ 端子割り込み要求を異なる割り込み優先レベルに設定して使用する場合に、IRQRR のリード値から IRQnF ビットをクリアする値を生成し、IRQRR にライトすると、ライトするタイミングによっては、保留中の別の IRQ 端子割り込み要求がクリアされることがあります。 サンプルコードでは、IRQRR のリード値から IRQ0F ビットをクリアする値を生成せずに、IRQ0F ビットのみがクリアされるように、IRQ7F~IRQ1F ビットに "1" を、IRQ0F ビットに "0" をライトするようにしています。

Sample_LPM_DStandby_ReturnRamBoot	
概 要	保持用内蔵 RAM からのディープスタンバイモード解除後の処理
宣 言	void Sample_LPM_DStandby_ReturnRamBoot(void)
説 明	ディープスタンバイモード解除後に保持用内蔵 RAM で実行する処理のサンプル関数です。ディープスタンバイコントロールレジスタ (DSCTR) の RAMBOOT ビットに "1" を設定してディープスタンバイモードに遷移し解除された場合に、内蔵 ROM のブートプログラム実行後にコールされます。 サンプルコードでは、IOKEEP ビットクリアによる端子状態保持の解除、外部メモリを使用するための周辺モジュールのレジスタ設定を行い、ローダプログラムの先頭アドレス (H'2000_0000 番地) に分岐します。
引 数	なし
リターン値	なし

5.8.2 API 関数仕様

R_LPM_Transition	
概 要	低消費電力モードへの遷移関数
宣 言	int_t R_LPM_Transition(e_r_drv_lpm_mode_t mode, e_r_drv_lpm_reboot_type_t ds_reboot_type)
説 明	<p>低消費電力モードに遷移するための処理を行います。</p> <p>引数 mode で指定された低消費電力モードに遷移するための API 関数をコールします。引数 mode に LPM_MODE_SOFTWARE_STANDBY または LPM_MODE_DEEP_STANDBY を指定した場合に、引数 mode および引数 ds_reboot_type で指定されたコンフィグレーションテーブルを使用します。（引数 mode に LPM_MODE_SLEEP を指定した場合には、コンフィグレーションテーブルは使用しません。）</p>
引 数	<p>e_r_drv_lpm_mode_t mode 遷移する低消費電力のモード</p> <p style="margin-left: 100px;">LPM_MODE_SLEEP : スリープモード</p> <p style="margin-left: 100px;">LPM_MODE_SOFTWARE_STANDBY : ソフトウェアスタンバイモード</p> <p style="margin-left: 100px;">LPM_MODE_DEEP_STANDBY : ディープスタンバイモード</p> <p>e_r_drv_lpm_reboot_type_t ds_reboot_type ディープスタンバイモード解除後の起動方法</p> <p style="margin-left: 100px;">LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT : 外部メモリから起動</p> <p style="margin-left: 100px;">LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 : 保持用内蔵 RAM から起動 (外部メモリ端子状態は保持しない)</p> <p style="margin-left: 100px;">LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1 : 保持用内蔵 RAM から起動 (外部メモリ端子状態は保持)</p>
リターン値	<p>DRV_SUCCESS : 正常終了</p> <p>DRV_ERROR : エラー終了</p>
注意事項	<p>本関数をコールして低消費電力モードに遷移する前に、システムに応じて、ご使用の低消費電力モードの解除要因となる割り込み関数の設定処理をしてください。</p> <p>IRQ7～IRQ0 端子割り込みにより、ソフトウェアスタンバイ処理を解除する場合は、本関数をコールする前に、端子機能を IRQ7～IRQ0 端子機能に設定してください。</p>

R_LPM_SleepTransition	
概 要	スリープモードへの遷移関数
宣 言	int_t R_LPM_SleepTransition(void)
説 明	スリープモードに遷移するための処理を行います。
引 数	なし
リターン値	DRV_SUCCESS : 正常終了 DRV_ERROR : エラー終了
注意事項	本関数をコールしてスリープモードに遷移する前に、システムに応じて、スリープモードの解除要因となる割り込み関数の設定処理を行ってください。 コンフィグレーションテーブルに複数の低消費電力モードを定義した場合には、本関数を直接コールせずに、R_LPM_Transition 関数をコールしてください。

R_LPM_SStandbyTransition	
概 要	ソフトウェアスタンバイモードへの遷移関数
宣 言	int_t R_LPM_SStandbyTransition(st_r_drv_lpm_sc_config_t *p_config_tbl)
説 明	ソフトウェアスタンバイモードに遷移するための処理を行います。 引数で指定したコンフィグレーションテーブル（表 5.19～表 5.23 を参照）を使用して、ソフトウェアスタンバイモード遷移中の端子状態、ソフトウェアスタンバイモードの解除要因、スタンバイ復帰発振安定時間を設定し、ソフトウェアスタンバイモードに遷移します。
引 数	st_r_drv_lpm_sc_config_tbl コンフィグレーションテーブルのポインタ
リターン値	DRV_SUCCESS : 正常終了 DRV_ERROR : エラー終了
注意事項	本関数をコールしてソフトウェアスタンバイモードに遷移する前に、システムに応じて、ソフトウェアスタンバイモードの解除要因となる割り込み関数の設定処理をしてください。 IRQ7～IRQ0 端子割り込みにより、ソフトウェアスタンバイ処理を解除する場合は、本関数をコールする前に、端子機能を IRQ7～IRQ0 端子機能に設定してください。 コンフィグレーションテーブルに複数の低消費電力モードを定義した場合には、本関数を直接コールせずに、R_LPM_Transition 関数をコールしてください。

R_LPM_DStandbyTransition	
概 要	ディープスタンバイモードへの遷移関数
宣 言	int_t R_LPM_DStandbyTransition(st_r_drv_lpm_sc_config_t *p_config_tbl)
説 明	ディープスタンバイモードに遷移するための処理を行います。 引数で指定したコンフィグレーションテーブル（表 5.19～表 5.23 を参照）を使用して、ディープスタンバイモード遷移中の端子状態、ディープスタンバイモード解除要因の設定、保持用内蔵 RAM の内容の保持設定、スタンバイ復帰発振安定時間、解除時のプログラム起動方法を設定し、ディープスタンバイモードに遷移します。
引 数	st_r_drv_lpm_sc_con コンフィグレーションテーブルのポインタ fig_t *p_config_tbl
リターン値	DRV_SUCCESS : 正常終了 DRV_ERROR : エラー終了
注意事項	本関数をコールしてディープスタンバイモードに遷移する前に、システムに応じて、ディープスタンバイモードの解除要因となるモジュールの設定処理を行ってください。 コンフィグレーションテーブルに複数の低消費電力モードを定義した場合には、本関数を直接コールせずに、R_LPM_Transition 関数をコールしてください。

R_INTC_SetIrqMask	
概 要	ソフトウェアスタンバイモードの IRQ 端子割り込み要求のマスク制御
宣 言	e_r_drv_intc_err_t R_INTC_SetIrqMask(e_r_drv_irq_mask_t mask)
説 明	ソフトウェアスタンバイモードの解除要因に IRQ 端子（IRQ7～IRQ0）割り込みを使用するかどうかを設定する処理を行います。 引数 mask で指定された値を使用して、IRQ 割り込み要求レジスタ（IRQRR）の IRQMSK ビットに設定します。 サンプルコードでは、引数 mask に IRQ_MASK_OFF を指定して本関数をコールし、IRQ 端子割り込みがソフトウェアスタンバイモードの解除要因となるように設定しています。
引 数	e_r_drv_irq_mask_t : マスク値 mask : IRQ_MASK_ON（IRQ 解除要因をマスクする） IRQ_MASK_OFF（IRQ 解除要因をマスクしない）
リターン値	INTC_SUCCESS : 正常終了 INTC_ERR_INVALID : 異常終了
注意事項	汎用入出力ポートを IRQ7～IRQ0 端子機能に切り替える時、ソフトウェアスタンバイモードの解除要因がセットされる場合があります。IRQ7～IRQ0 端子をソフトウェアスタンバイモードの解除要因に使用する場合は、汎用入出力ポートを IRQ7～IRQ0 端子機能に切り替える前に、IRQMSK ビットは 0 に設定してください。本関数の引数 mask に IRQ_MASK_ON を指定すると、IRQMSK ビットに"0"を設定することができます。

R_INTC_SetNMICConfig	
概 要	NMI 端子割り込み要求の検出方法を設定
宣 言	e_r_drv_intc_err_t R_INTC_SetNMICConfig(const st_r_drv_nmi_cfg_t * p_nmi_config)
説 明	引数 p_nmi_config で指定した端子状態になると、NMI 端子割り込みの要求を検出するように設定します。
引 数	const : NMI 端子割り込み発生要因 st_r_drv_nmi_cfg_t * : NMI_SENSE_FALLINGEDGE（立ち下がリエッジ） p_nmi_config : NMI_SENSE_RISINGEDGE（立ち上がりエッジ）
リターン値	INTC_SUCCESS : 正常終了

R_IRQ_SetSense	
概 要	IRQ 端子割り込み要求の検出方法の設定
宣 言	e_r_drv_intc_err_t R_IRQ_SetSense(e_r_drv_irq_num_t irq_num, e_r_drv_irq_sense_t sense)
説 明	IRQ 端子割り込み要求の検出方法を設定します。 引数 irq_num で指定した IRQ7~IRQ0 の端子に対して、引数 sense で指定した端子 状態で IRQ 端子割り込み要求を検出するように設定します。
引 数	e_r_drv_irq_num_t : IRQ 番号 irq_num INTC_IRQ0 (IRQ0 端子) INTC_IRQ1 (IRQ1 端子) INTC_IRQ2 (IRQ2 端子) INTC_IRQ3 (IRQ3 端子) INTC_IRQ4 (IRQ4 端子) INTC_IRQ5 (IRQ5 端子) INTC_IRQ6 (IRQ6 端子) INTC_IRQ7 (IRQ7 端子) e_r_drv_irq_sense_t : IRQ 端子割り込み発生要因 sense IRQ_SENSE_LEVEL (ローレベル) IRQ_SENSE_FALLINGEDGE (立ち下がリエッジ) IRQ_SENSE_RISINGEDGE (立ち上がりエッジ) IRQ_SENSE_BOTHEDGE (両エッジ)
リターン値	INTC_SUCCESS : 正常終了 INTC_ERR_INVALID_NUM : 引数 irq_num が範囲外 INTC_ERR_INVALID_SENSE : 引数 sense が範囲外

5.9 フローチャート

5.9.1 スリープモードへの遷移サンプル関数

図 5.5 および図 5.6 にスリープモードへの遷移サンプル関数のフローチャートを示します。

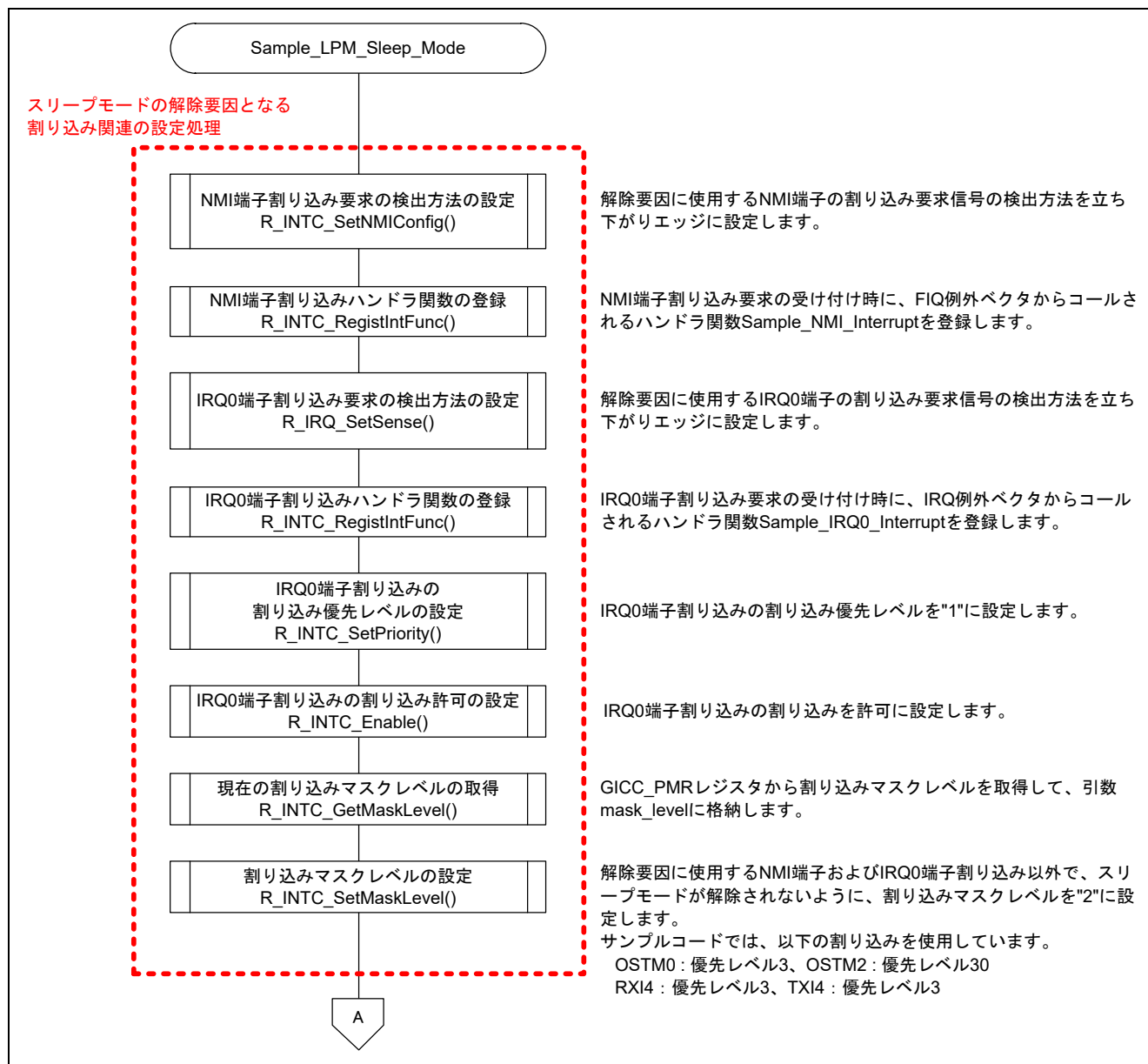


図 5.5 スリープモードへの遷移サンプル関数 (1/2)

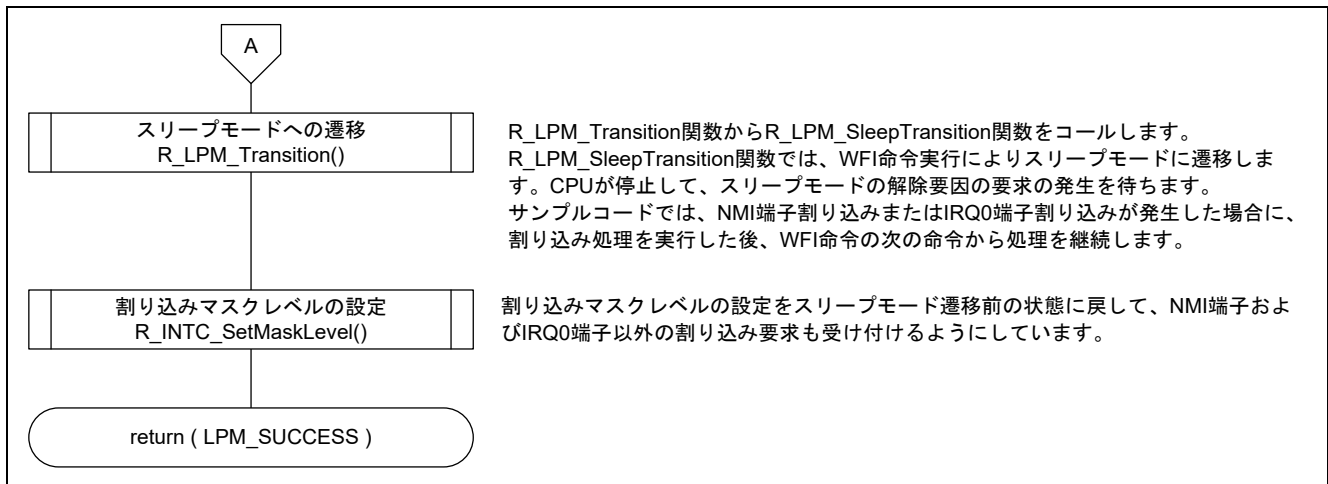


図 5.6 スリープモードへの遷移サンプル関数 (2/2)

5.9.2 スリープモードへの遷移関数

図 5.7 にスリープモードへの遷移のフローチャートを示します。

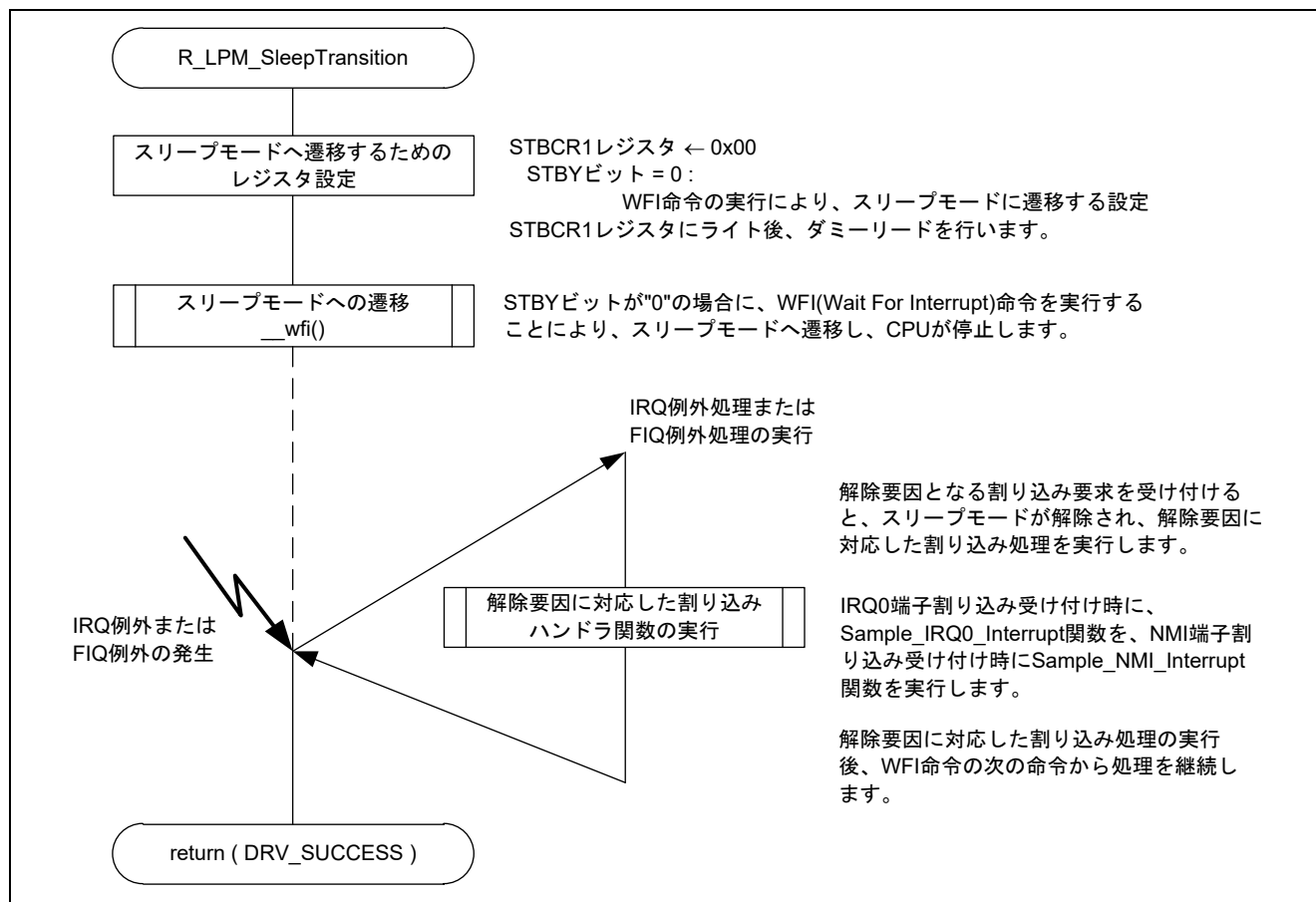


図 5.7 スリープモードへの遷移関数

5.9.3 ソフトウェアスタンバイモードへの遷移サンプル関数

図 5.8 および図 5.9 にソフトウェアスタンバイモードへの遷移のフローチャートを示します。

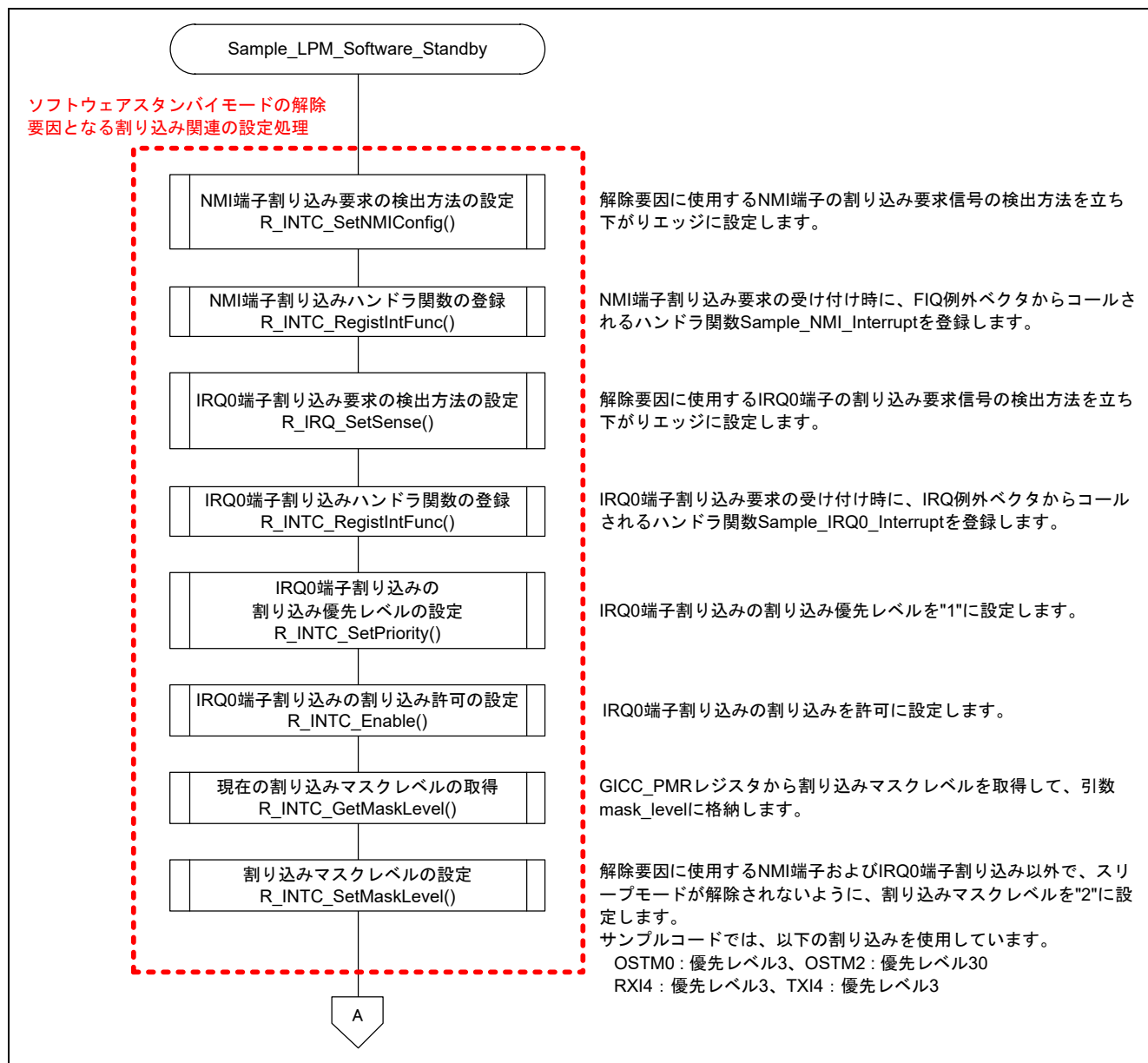


図 5.8 ソフトウェアスタンバイモードへの遷移サンプル関数 (1/2)

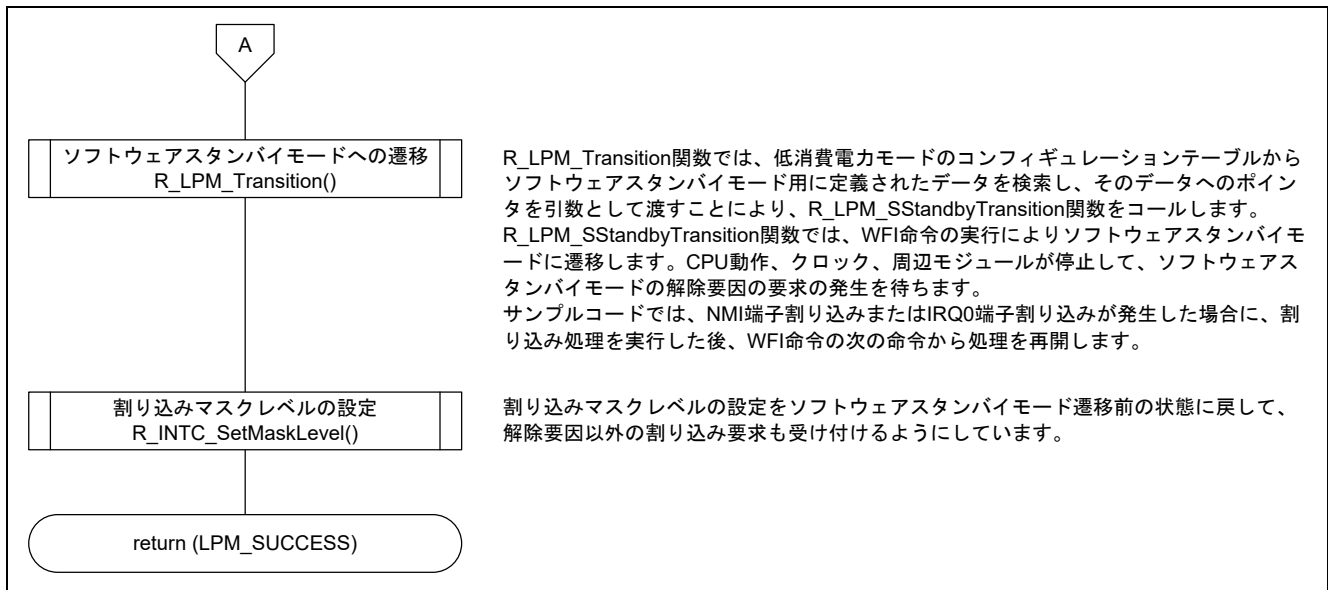


図 5.9 ソフトウェアスタンバイモードへの遷移サンプル関数 (2/2)

5.9.4 ソフトウェアスタンバイモードへの遷移関数

図 5.10 および図 5.11 にソフトウェアスタンバイモードへの遷移関数フローチャートを示します。

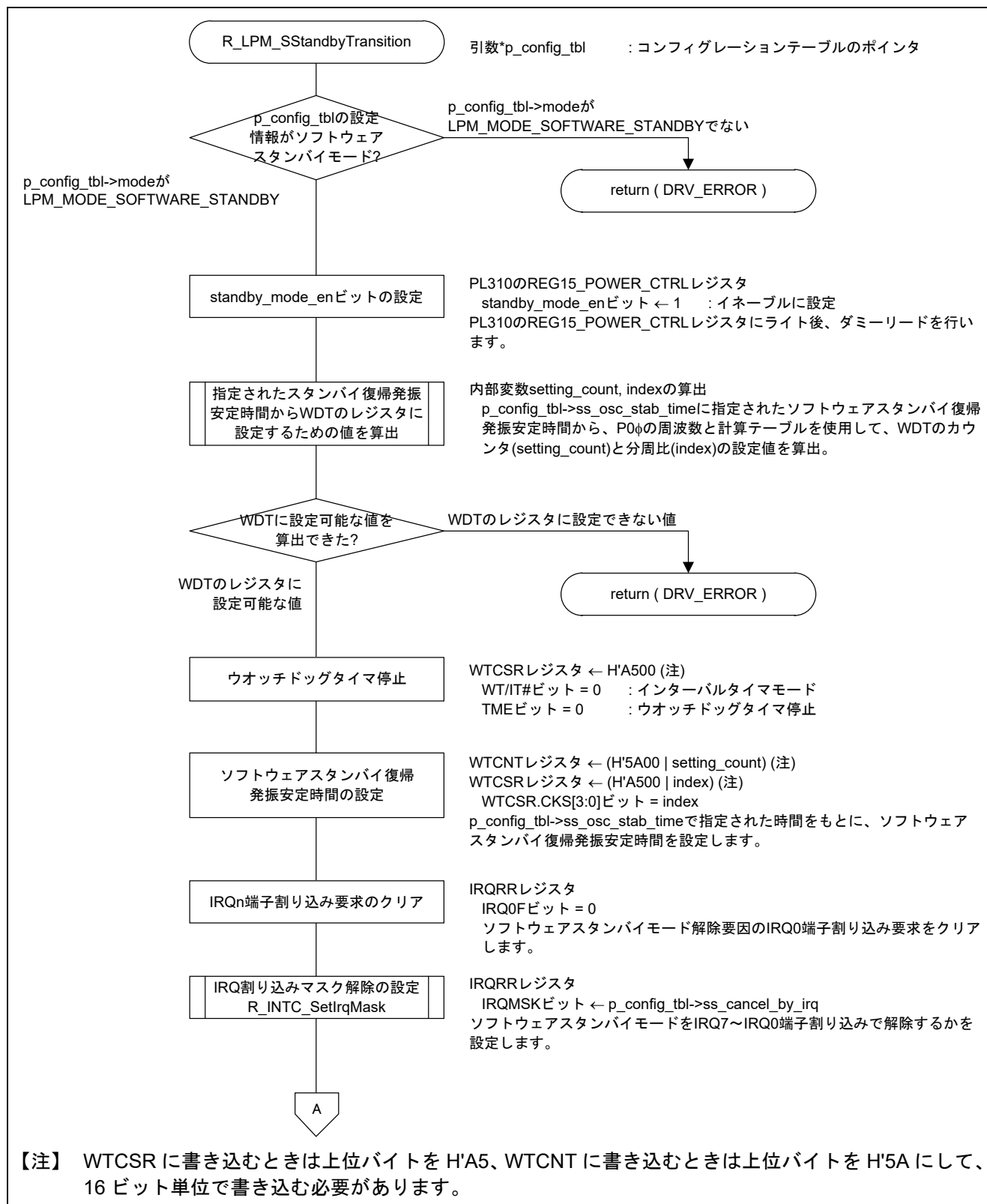


図 5.10 ソフトウェアスタンバイモードへの遷移関数 (1/2)

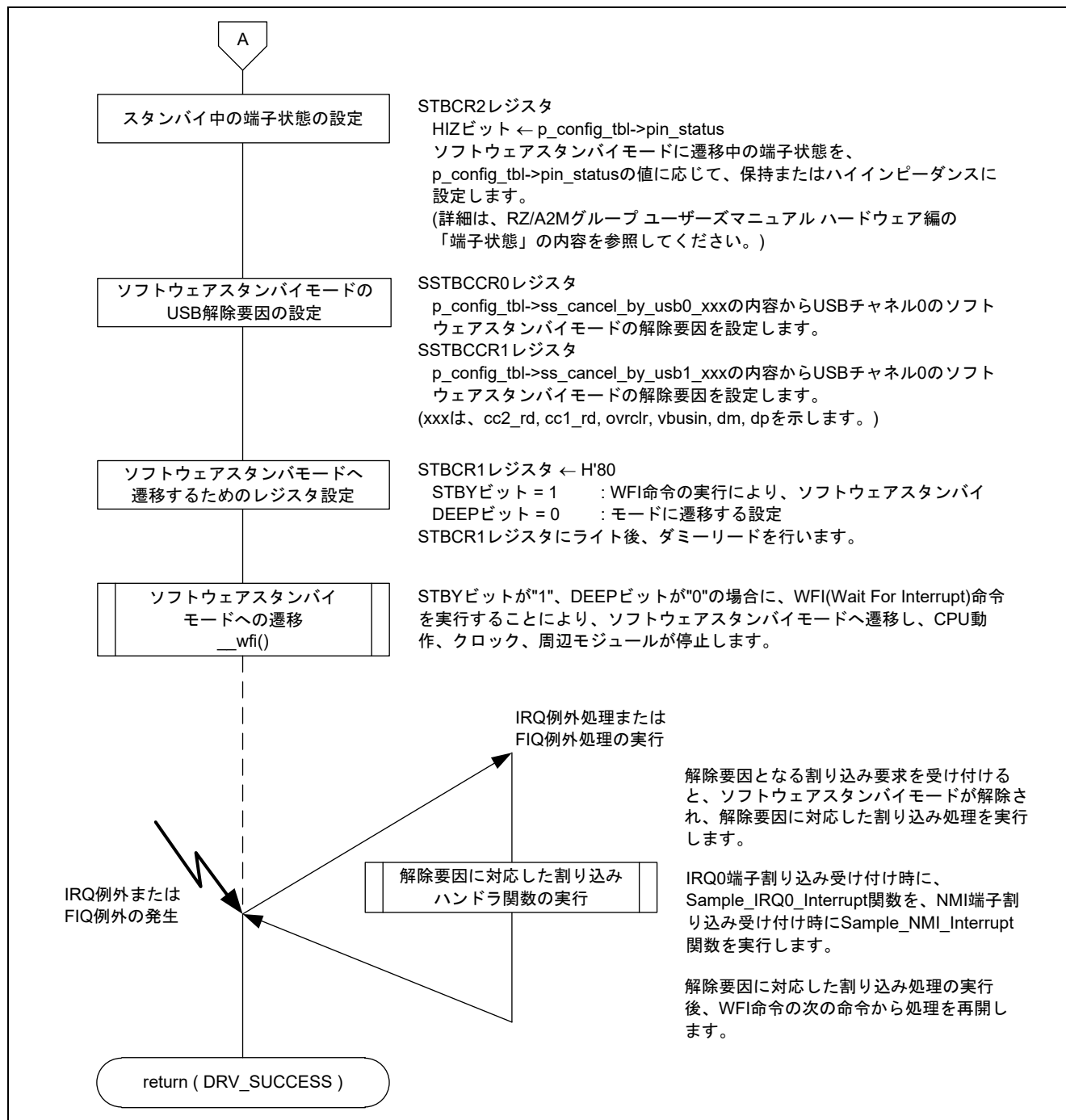


図 5.11 ソフトウェアスタンバイモードへの遷移関数 (2/2)

5.9.5 ディープスタンバイモードへの遷移サンプル関数

図 5.12 にディープスタンバイモードへの遷移サンプル関数のフローチャートを示します。

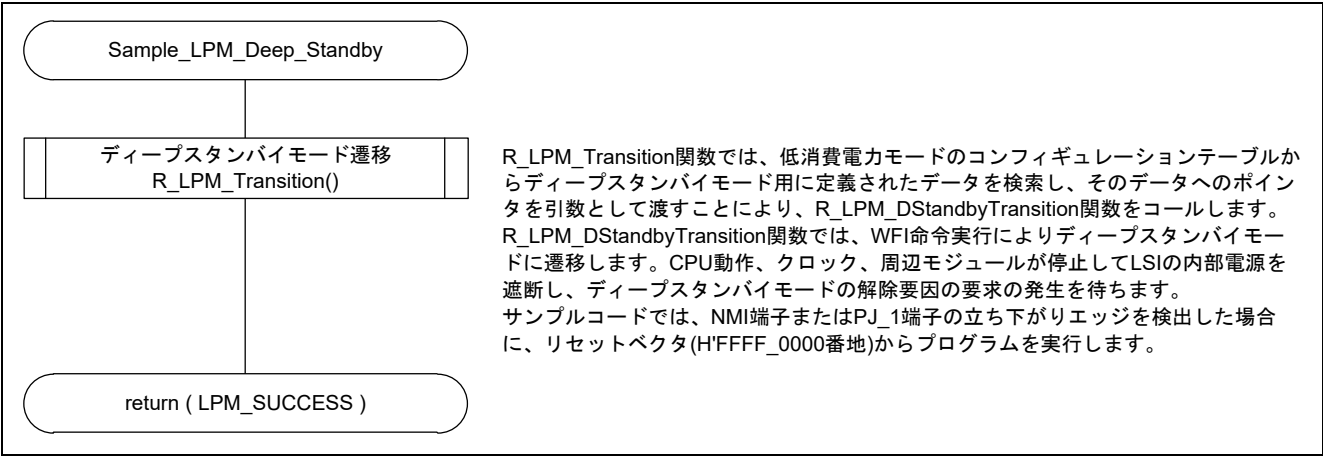


図 5.12 ディープスタンバイモードへの遷移サンプル関数

5.9.6 ディープスタンバイモードへの遷移関数

図 5.13 および図 5.14 にディープスタンバイモードへの遷移関数のフローチャートを示します。

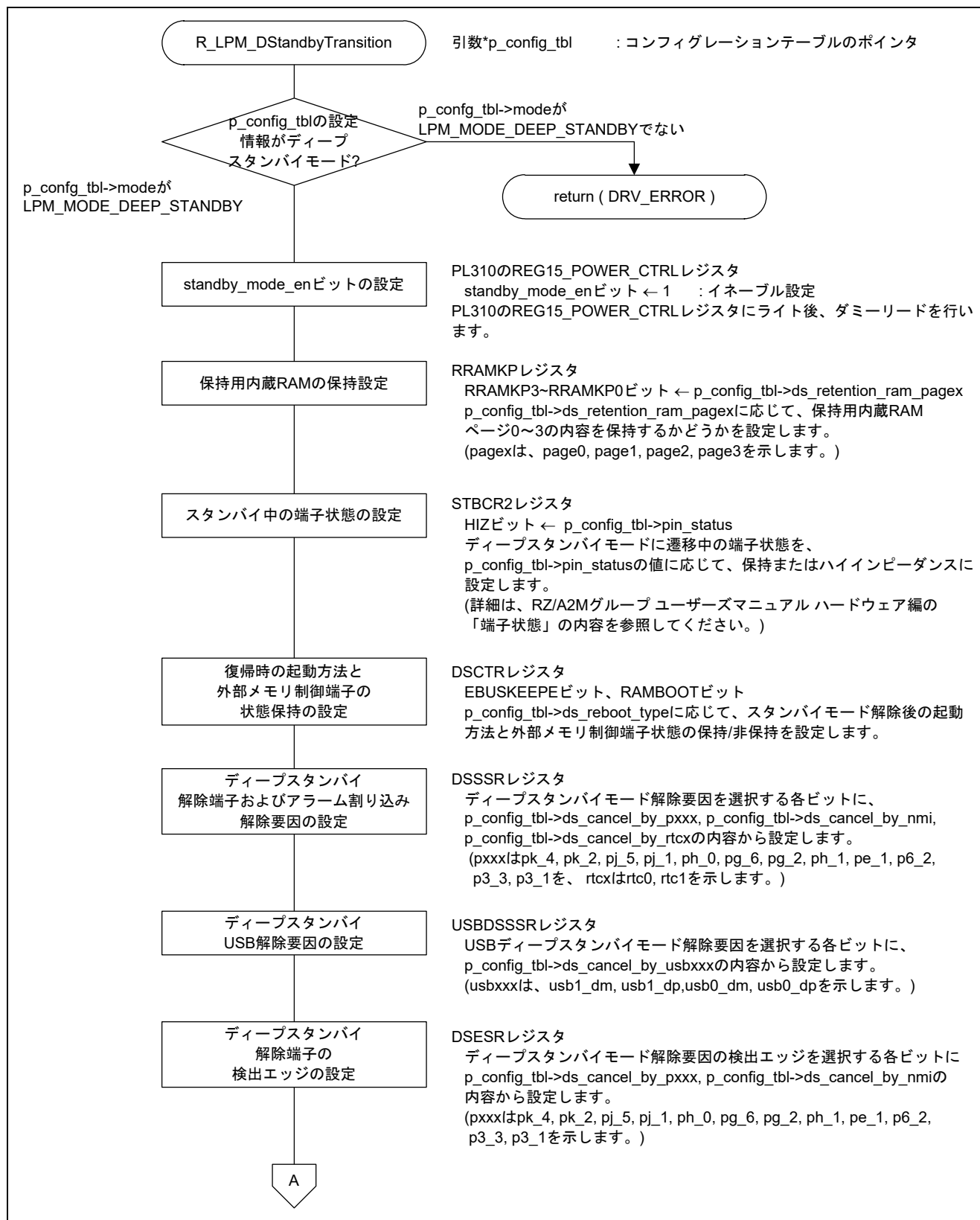


図 5.13 ディープスタンバイモードへの遷移関数 (1/2)

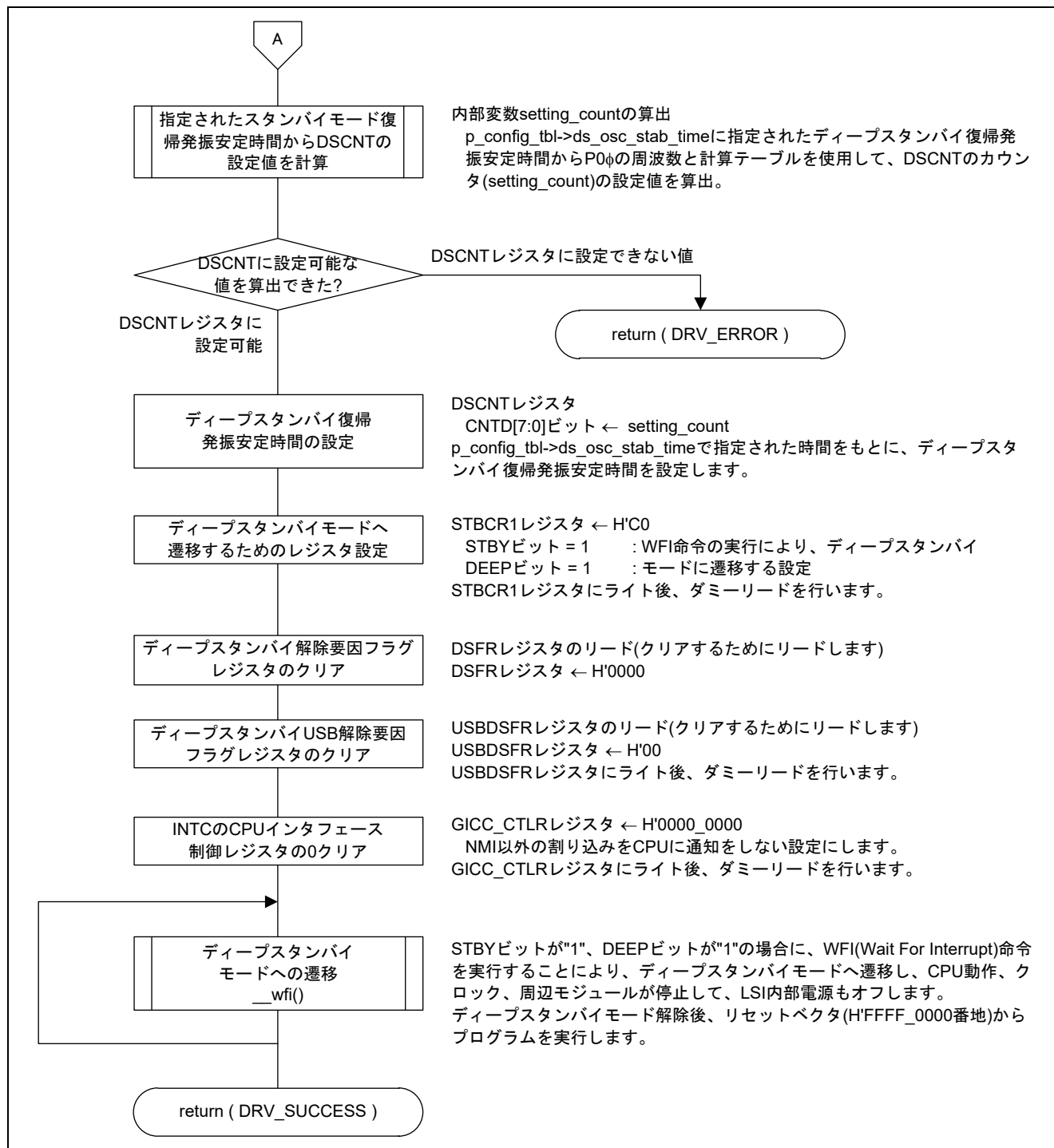


図 5.14 ディープスタンバイモードへの遷移関数 (2/2)

5.9.7 ディープスタンバイモード解除後に保持用内蔵 RAM から起動する処理

図 5.15 にディープスタンバイモード解除後に保持用内蔵 RAM から起動する処理のフローチャートを示します。

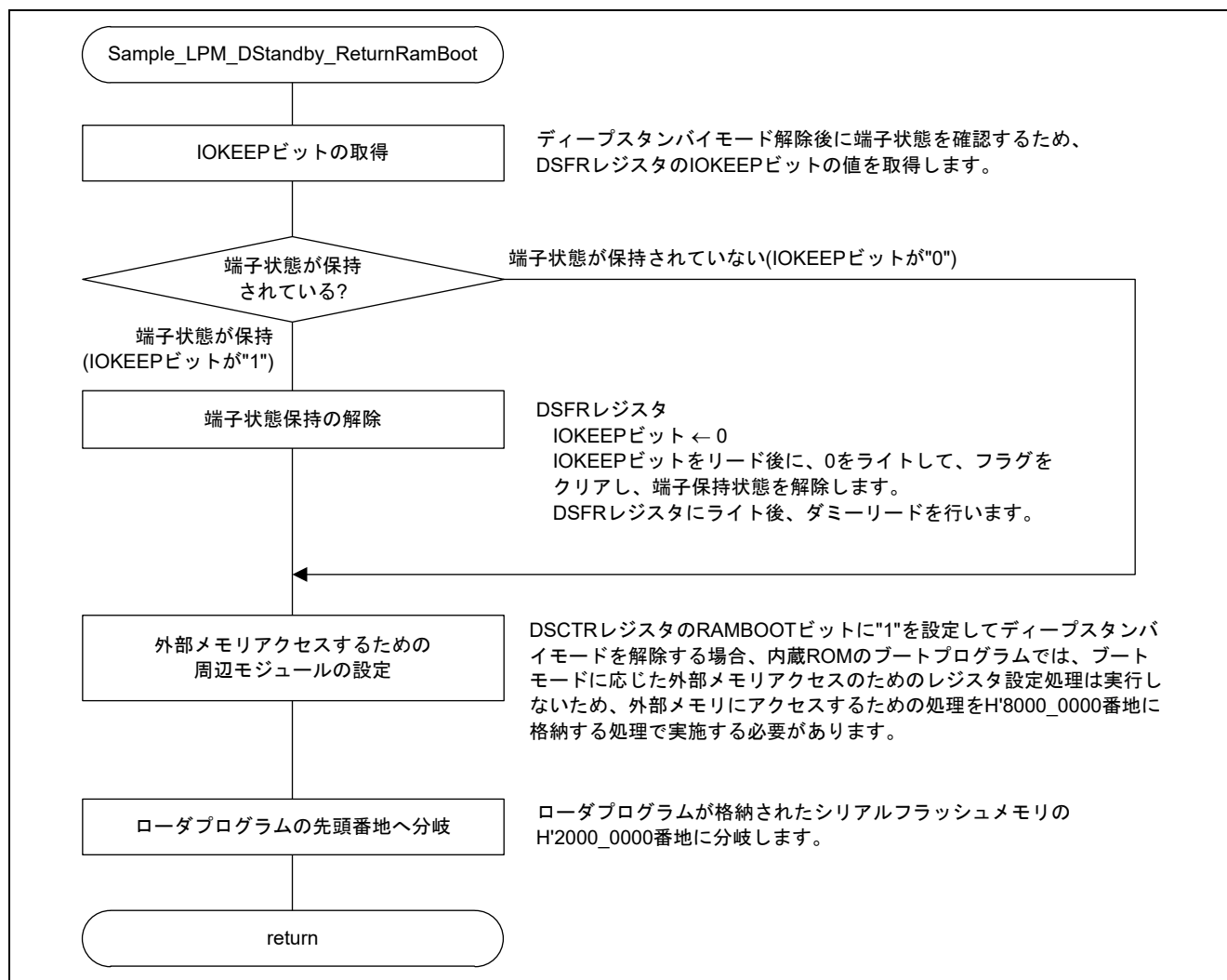


図 5.15 ディープスタンバイモード解除後に保持用内蔵 RAM から起動する処理

5.9.8 NMI 端子割り込みサンプル関数

図 5.16 に NMI 端子割り込みサンプル関数のフローチャートを示します。

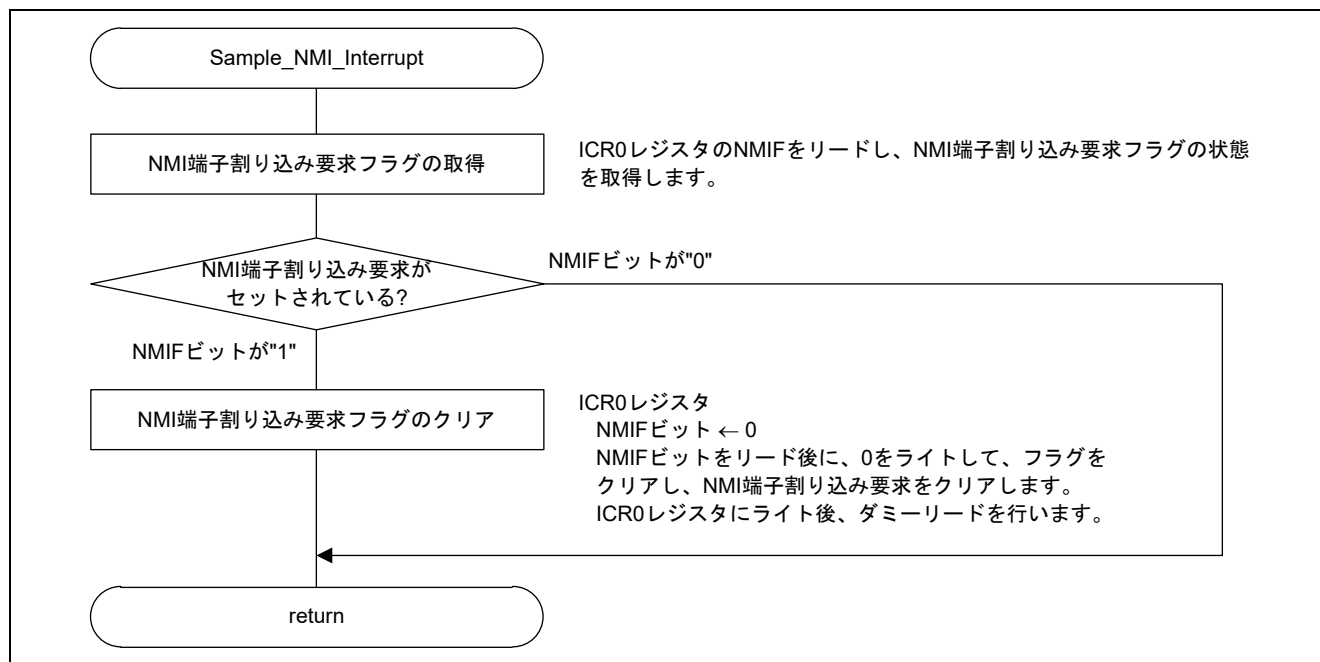


図 5.16 NMI 端子割り込みサンプル関数

5.9.9 IRQ0 端子割り込みサンプル関数

図 5.17 に IRQ0 端子割り込みサンプル関数のフローチャートを示します。

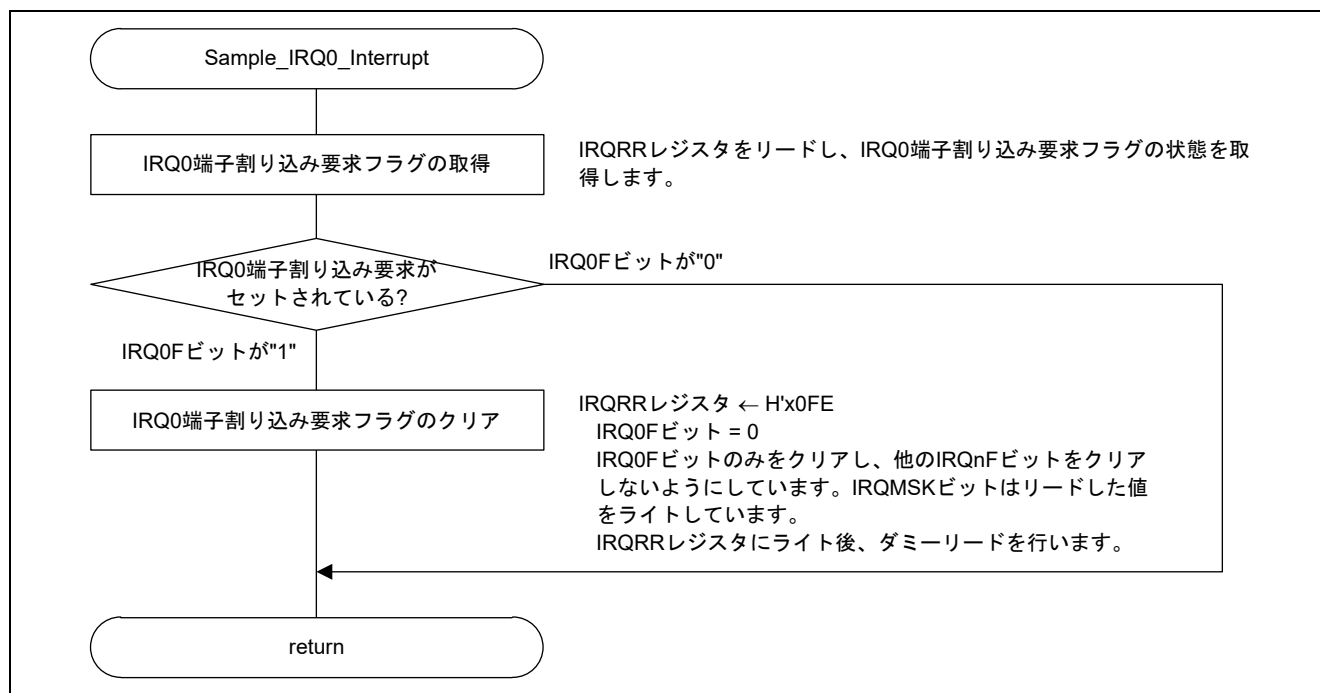


図 5.17 IRQ0 端子割り込みサンプル関数

5.9.10 低消費電力モードへの遷移関数

図 5.18 に低消費電力モードへの遷移関数のフローチャートを示します。

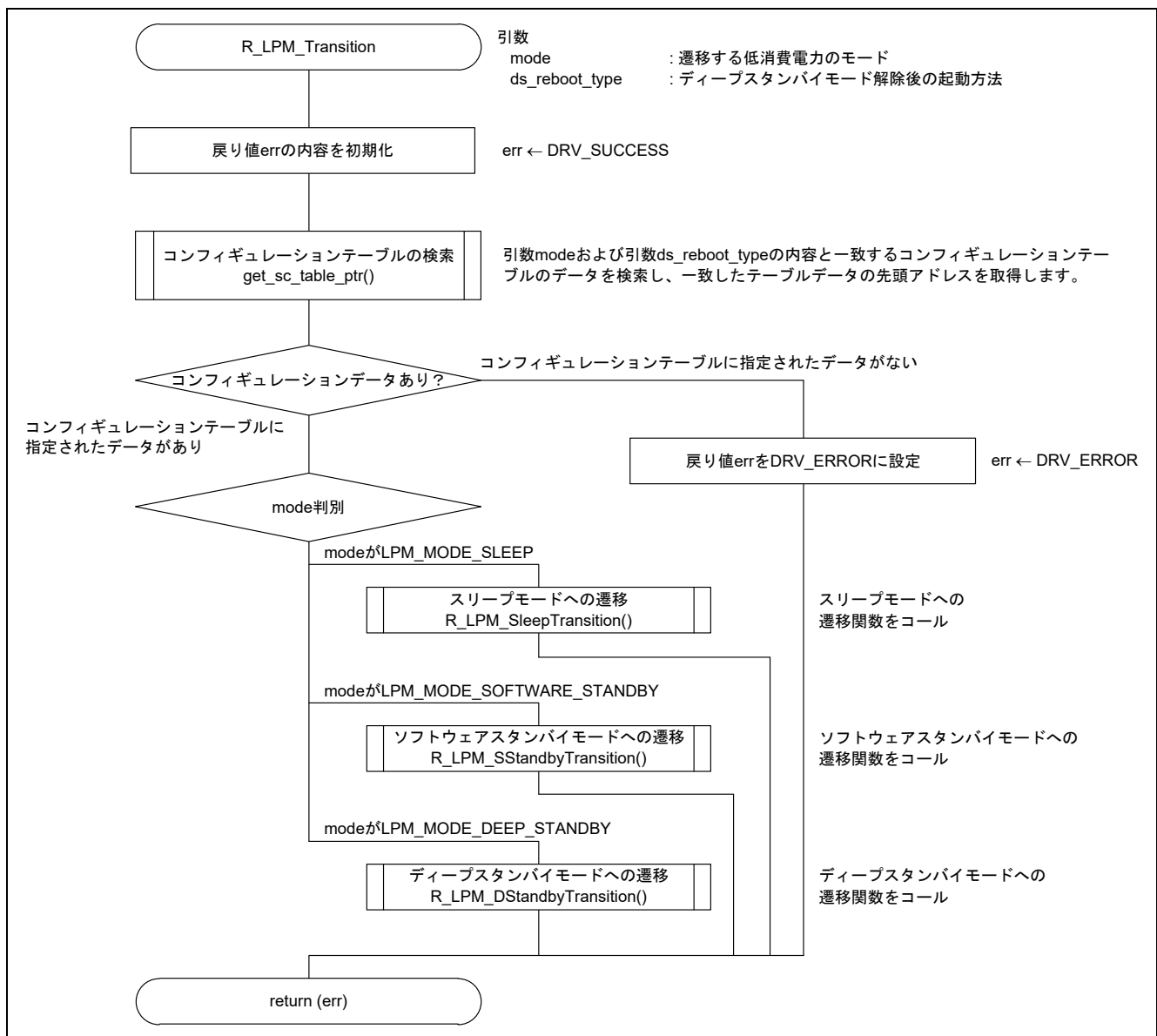


図 5.18 低消費電力モードへの遷移関数

5.10 サンプルコードの起動

サンプルコードでは、ターミナルからコマンドを入力することで、各低消費電力モードに遷移します。表 5.26 にサンプルコマンド動作一覧を示します。

表 5.26 サンプルコマンド動作一覧

コマンド		コマンド関数名	内容
第一 パラメータ	第二 パラメータ		
LPM_SM	—	sapmle_lpm_ cmd_sleep_mode	スリープモードに遷移します。 SW2 または SW3 の押し下げを検出後に、スリープモードを解除します。
LPM_SS	—	sapmle_lpm_ cmd_software_standby	ソフトウェアスタンバイモードに遷移します。 SW2 または SW3 の押し下げを検出後に、ソフトウェアスタンバイモードを解除します。
LPM_DS	1	sapmle_lpm_ cmd_deep_standby	ディープスタンバイモードに遷移します。 SW2 または SW3 の押し下げを検出後に、ディープスタンバイモードを解除し、内蔵 ROM のブートプログラムを実行後、ローダプログラムを実行します（外部メモリ起動モード）。
LPM_DS	2	sapmle_lpm_ cmd_deep_standby	ディープスタンバイモードに遷移します。 SW2 または SW3 の押し下げを検出後に、ディープスタンバイモードを解除し、内蔵 ROM のブートプログラムを実行後、保持用内蔵 RAM に格納されたプログラムを実行します（保持用内蔵 RAM 起動モード）。

ボードの電源投入後は、図 5.19 の(1)に示すメッセージを出力します。Ver.x.xx はサンプルコードのメイン処理のバージョン、Copyright (C) yyyy はアプリケーションノートの発行年号を示します。"SAMPLE>"プロンプトの出力後、コマンドを入力することができます。

表示メッセージ
<div style="text-align: right;">(1)</div> RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved. SAMPLE>

図 5.19 サンプルコード起動時のターミナル表示例

5.10.1 スリープモード遷移コマンドの動作

サンプルコードでは、サンプルコードの起動後、ターミナルソフトから"LPM_SM"+"Enter キー"を入力すると、スリープモード遷移コマンドが動作します。スリープモード遷移コマンドが動作すると、図 5.20 の(1)のメッセージを出力します。"Please Press Enter Key"が出力された後に"Enter キー"を入力すると、スリープモードに遷移します。スリープモードに遷移後、解除要因の入力待ちの状態となります。

また、スリープモードの解除要因であるスイッチ（SW2 または SW3）を押すと、スリープモードが解除され、図 5.20 の(2)に示すメッセージを出力します。

表示メッセージ	
<div>SAMPLE>LPM_SM Transition to Sleep Mode Press the enter key on the terminal to enter sleep mode Sleep mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by IRQ0, SW2 : Canceled by NMI) Please Press Enter Key</div>	(1)
<div>***** Return from Sleep Mode ***** SAMPLE></div>	(2)

図 5.20 スリープモード遷移コマンド実行時のターミナル表示例

5.10.2 ソフトウェアスタンバイモード遷移コマンドの動作

サンプルコードでは、サンプルコードの起動後、ターミナルソフトから"LPM_SS"+"Enter キー"を入力すると、ソフトウェアスタンバイモード遷移コマンドが動作します。ソフトウェアスタンバイモード遷移コマンドが動作すると、図 5.21 の(1)のメッセージを出力します。"Please Press Enter Key"が出力された後に"Enter キー"を入力すると、ソフトウェアスタンバイモードに遷移します。ソフトウェアスタンバイモードに遷移後、解除要因の入力待ちの状態となります。

また、ソフトウェアスタンバイモードの解除要因であるスイッチ（SW2 または SW3）を押すと、ソフトウェアスタンバイモードが解除され、図 5.21 の(2)に示すメッセージを出力します。

表示メッセージ	
<div>SAMPLE> LPM_SS Transition to Software Standby Mode Press the enter key on the terminal to enter software standby mode Software stanby mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by IRQ0, SW2 : Canceled by NMI) Please Press Enter Key</div>	(1)
<div>***** Return from Software Standby Mode ***** SAMPLE></div>	(2)

図 5.21 ソフトウェアスタンバイモード遷移コマンド実行時のターミナル表示例

5.10.3 ディープスタンバイモード遷移コマンドの動作

サンプルコードでは、サンプルコードの起動後、ターミナルソフトから"LPM_DS"+"1"+"Enter キー"または"LPM_DS"+"2"+"Enter キー"を入力すると、ディープスタンバイモード遷移コマンドが動作します。

(1) 外部メモリ起動モード

"LPM_DS"+"1"+"Enter キー"を入力すると、図 5.22 の(1)のメッセージを出力します。"Please Press Enter Key"が出力された後に"Enter キー"を入力すると、ディープスタンバイモードに遷移します。ディープスタンバイモードに遷移後、解除要因の入力待ちの状態となります。

また、ディープスタンバイモードの解除要因であるスイッチ（SW2 または SW3）を押すと、ディープスタンバイモードが解除され、外部メモリ起動モードで起動し、図 5.22 の(2)に示すメッセージを出力します。

表示メッセージ	
<div>SAMPLE>LPM_DS 1 Transition to Deep Standby Mode(EBUSKEEPE=0, RAMBOOT=0) Press the enter key on the terminal to enter deep standby mode Deep standby mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by PJ_1, SW2 : Canceled by NMI) Please Press Enter Key</div>	(1)
<div>***** Return from Deep Standby Mode ***** RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved. SAMPLE></div>	(2)

図 5.22 ディープスタンバイモード遷移コマンド実行時のターミナル表示例（外部メモリ起動モード）

(2) 保持用内蔵 RAM 起動モード

"LPM_DS"+"2"+"Enter キー"を入力すると、図 5.23 の(1)のメッセージを出力します。"Please Press Enter Key"が出力された後に"Enter キー"を入力すると、ディープスタンバイモードに遷移します。ディープスタンバイモードに遷移後、解除要因の入力待ちの状態となります。

また、ディープスタンバイモードの解除要因であるスイッチ（SW2 または SW3）を押すと、ディープスタンバイモードが解除され、保持用内蔵 RAM 起動モードで起動し、図 5.23 の(2)に示すメッセージを出力します。

表示メッセージ	
<div>SAMPLE>LPM_DS 2 Transition to Deep Standby Mode(EBUSKEEPE=0, RAMBOOT=1) Press the enter key on the terminal to enter deep standby mode Deep standby mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by PJ_1, SW2 : Canceled by NMI) Please Press Enter Key</div>	(1)
<div>***** Return from Deep Standby Mode ***** RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved. SAMPLE></div>	(2)

図 5.23 ディープスタンバイモード遷移コマンド実行時のターミナル表示例（保持用 RAM 起動モード）

5.10.4 ヘルプコマンドの動作

サンプルコードでは、サンプルコードの起動後、ターミナルソフトから"HELP"+"Enter キー"を入力すると、図 5.24 の(1)に示すようにサンプルコードでサポートされるコマンドの内容を表示します。

表示メッセージ	
(1)	
SAMPLE>HELP	
[LPM_SM command]	
Transition to Sleep Mode	
command format:	
LPM_SM	
argument:	
None	
[LPM_SS command]	
Transition to Software Standby Mode	
command format:	
LPM_SS	
argument:	
None	
[LPM_DS command]	
Transition to Deep Standby Mode	
command format:	
LPM_DS 'value'	
argument:	
'value' - 1 or 2	
(1:External Memory Startup Mode,	
2:On-chip Data Retention RAM Startup Mode)	
SAMPLE>	

図 5.24 ヘルプコマンド実行時のターミナル表示例

6. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A2M グループ ユーザーズマニュアル ハードウェア編

(最新版をルネサス エレクトロニクスホームページから入手してください。)

RTK7921053C00000BE (RZ/A2M CPU ボード) ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

RTK79210XXB00000BE (RZ/A2M SUB ボード) ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

(最新版を Arm ホームページから入手してください。)

Arm Cortex™-A9 Technical Reference Manual Revision: r4p1

(最新版を Arm ホームページから入手してください。)

Arm Generic Interrupt Controller Architecture Specification - Architecture version2.0

(最新版を Arm ホームページから入手してください。)

Arm CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3

(最新版を Arm ホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：統合開発

統合開発環境 e² studio のユーザーズマニュアルは、ルネサス エレクトロニクスホームページから入手してください。

(最新版をルネサス エレクトロニクスホームページから入手してください。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
Rev.1.00	Apr.15.19	—	初版発行
Rev.1.10	May.17.19	P7	表 2.1 動作確認条件 (1/2) コンパイルオプション"-mthumb-interwork"を削除
Rev.1.11	Aug.28.19	P44 P47 P64	API 関数として R_LPM_Transition 関数を新規に追加 <ul style="list-style-type: none"> 「表 5.25 API 関数一覧」に追加した関数の概要を記載 「5.8.2 API 関数仕様」に追加した関数の仕様を記載 「図 5.18 低消費電力モードへの遷移関数」に追加した関数のフローを追加
		P48 P48 P49	以下の API 関数コール時の注意事項を追加 <ul style="list-style-type: none"> R_LPM_SleepTransition 関数 R_LPM_SStandbyTransition 関数 R_LPM_DStandbyTransition 関数
		P45 P45 P45 P53 P56 P59	各低消費電力モードに遷移するためのサンプル関数からコールされる関数を R_LPM_Transition 関数に変更 <ul style="list-style-type: none"> Sample_LPM_Sleep_Mode 関数の関数説明を変更 Sample_LPM_Software_Standby 関数の関数説明を変更 Sample_LPM_Deep_Standby 関数の関数説明を変更 図 5.6 スリープモードへの遷移サンプル関数 (2/2) R_LPM_SleepTransition 関数から R_LPM_Transition 関数に変更 図 5.9 ソフトウェアスタンバイモードへの遷移サンプル関数 (2/2) R_LPM_SStandbyTransition 関数から R_LPM_Transition 関数に変更 図 5.12 ディープスタンバイモードへの遷移サンプル関数 R_LPM_DStandbyTransition 関数から R_LPM_Transition 関数に変更
		P66 ～ P70	サンプルコードのコマンド実行時の出力メッセージを変更 <ul style="list-style-type: none"> 5.10.1 スリープモード遷移コマンドの動作 5.10.2 ソフトウェアスタンバイモード遷移コマンドの動作 5.10.3 ディープスタンバイモード遷移コマンドの動作 5.10.4 ヘルプコマンドの動作
		P33 P34	R_SC_HardwareSetup 関数の処理を行う入力セクションを追加したことに伴い、以下の図および表を変更 <ul style="list-style-type: none"> 表 5.13 サンプルコードで使用するセクション名とオブジェクト名一覧 図 5.4 セクション配置
Rev.1.12	Nov.13.19		

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、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、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。