

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様が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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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とその総株主の議決権の過半数を直接または間接に保有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。

H8/300H Tiny シリーズ

低消費電力モード切り替え (H8/3664)

要旨

H8/3664 の低消費電力モード切り替え機能によりアクティブ/サブアクティブ/スリープ/サブスリープ/スタンバイモードの切り替えを行ないます。

動作確認デバイス

H8/300H Tiny シリーズ –H8/3664–

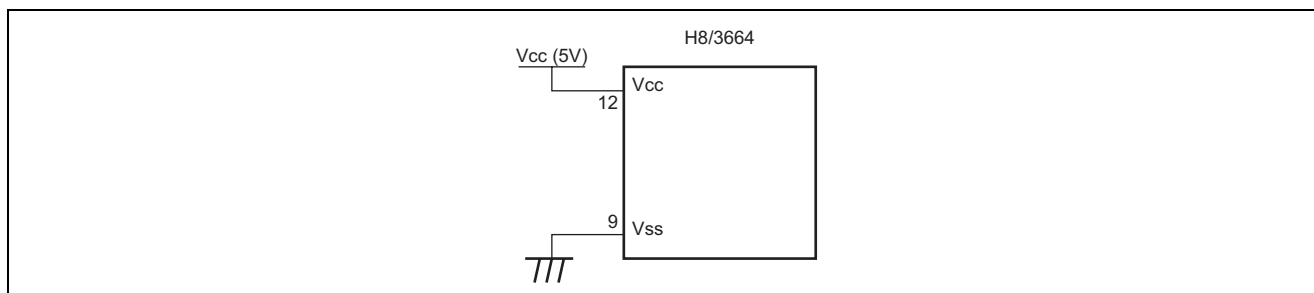
目次

1. 概要	2
2. 構成	2
3. サンプルプログラム	3
4. 参考文献	15

1. 概要

H8/3664 の低消費電力モード切り替え機能によりアクティブ/サブアクティブ/スリープ/サブスリープ/スタンバイモードの切り替えを行ないます。

2. 構成



仕様

H8/3664 の動作周波数：16MHz

3. サンプルプログラム

3.1 機能

1. 直接遷移によりアクティブ/サブアクティブモードに遷移します。
2. スリープ/サブスリープ/スタンバイモードに遷移します。割り込みにより指定の動作周波数でアクティブ/サブアクティブモードに復帰します。

3.2 組み込み方法

1. サンプルプログラム 11-A define 定義を組み込んでください。
2. サンプルプログラム 11-B プロトタイプ宣言を組み込んでください。
3. サンプルプログラム 11-C を共通ルーチンとして追加してください。

3.3 サンプルプログラムの変更

サンプルプログラムそのままでは、システムが動作しないことがあります。お客様のプログラムやシステム環境に合わせて修正を行なう必要があります。

1. IO レジスタの構造体定義は、ルネサス Web で無償入手できる定義ファイルをご利用になるとサンプルプログラムをそのまま使用することができます。独自に作成される場合は、サンプルプログラム中に使用している IO レジスタの構造体を適宜変更してください。

3.4 使用方法

1. 動作周波数を変更します。

```
com_frequency_change (int return_mode , int frequency )
```

引数	説明
return_mode	周波数変更後の動作モードを指定する。 ACTIVE_MODE(0) : アクティブモード SUB_ACTIVE_MODE(1) : サブアクティブモード
frequency	変更後の動作周波数を示す。 OSC_PER_1 (0x00) : ϕ (本サンプルプログラムでは 16MHz) OSC_PER_8 (0x10) : $\phi/8$ (本サンプルプログラムでは 2MHz)* OSC_PER_16 (0x14) : $\phi/16$ (本サンプルプログラムでは 1MHz)* OSC_PER_32 (0x18) : $\phi/32$ (本サンプルプログラムでは 0.5MHz)* OSC_PER_64 (0x1C) : $\phi/64$ (本サンプルプログラムでは 0.25MHz)* OSC_PER_W8 (0x00) : $\phi w/8$ (4.096KHz) OSC_PER_W4 (0x01) : $\phi w/4$ (8.192KHz) OSC_PER_W2 (0x02) : $\phi w/2$ (16.384KHz) 【注】 78.125kHz 未満 (動作周波数最小値) になる場合は, 設定不可。 (例) $\phi = 4\text{MHz}$ のとき, $\phi/64 = 62.5\text{KHz}$ のため OSC_PER_64 (0x1C) は指定不可

使用例

```
/*SCI3 受信割り込みを disable にする。*/
SCI3.SCR3.BYTE = 0x10;
/*TimerZ 割り込みを無効にする*/
TZ0.TIER.BIT.IMIEA = 0 ;
/* 周波数の変更を行なう*/
com_frequency_change(ACTIVE_MODE , OSC_PER_8) ;
// 直接遷移割り込みで復帰
// アクティブモード 動作周波数 $\phi/8$  に遷移
```

2. スリープ/サブスリープ/スタンバイモードに遷移します。割り込みが発生したとき、指定の動作周波数でアクティブ/サブアクティブモードに復帰します。

```
com_mode_change (int stop_mode , int return_mode , int frequency)
```

引数	説明
stop_mode	休止モードを指定する。 SLEEP_MODE(0) : スリープモード SUB_SLEEP_MODE(1) : サブスリープモード STANDBY_MODE(2) : スタンバイモード
return_mode	周波数変更後の動作モードを指定する。 ACTIVE_MODE(0) : アクティブモード SUB_ACTIVE_MODE(1) : サブアクティブモード
frequency	変更後の動作周波数を示す。 OSC_PER_1 (0x00) : ϕ (本サンプルプログラムでは 16MHz) OSC_PER_8 (0x10) : $\phi/8$ (本サンプルプログラムでは 2MHz)* OSC_PER_16 (0x14) : $\phi/16$ (本サンプルプログラムでは 1MHz)* OSC_PER_32 (0x18) : $\phi/32$ (本サンプルプログラムでは 0.5MHz)* OSC_PER_64 (0x1C) : $\phi/64$ (本サンプルプログラムでは 0.25MHz)* OSC_PER_W8 (0x00) : $\phi w/8$ (4.096KHz) OSC_PER_W4 (0x01) : $\phi w/4$ (8.192KHz) OSC_PER_W2 (0x02) : $\phi w/2$ (16.384KHz) 【注】 78.125kHz 未満 (動作周波数最小値) になる場合は、設定不可。 (例) $\phi = 4\text{MHz}$ のとき、 $\phi/64 = 62.5\text{KHz}$ のため OSC_PER_64 (0x1C) は指定不可

使用例

```
/*SCI3 受信割り込みを disable にする。*/
SCI3.SCR3.BYTE = 0x10;
/*TimerZ 割り込みを無効にする*/
TZ0.TIER.BIT.IMIEA = 0 ;
/* int0 enable にする*/
IENR1.BIT.IEN0 = 1 ;
com_mode_change(SLEEP_MODE,ACTIVE_MODE,OSC_PER_1) ;
// スリープモードになり, int0 割り込みで復帰
// アクティブモード  $\phi$  で復帰
```

【注】 サブアクティブモードに遷移した場合、下記のように動作に制限があります。

- ・タイマ A のインターバルタイマ使用不可。
- ・ウォッチドッグタイマ, I²C, タイマ W 使用不可。
- ・タイマ V, SCI3, A/D コンバータはリセットされるため各レジスタを再設定すること。

3.5 動作説明

図1に可能なモード間遷移を示します。プログラム実行状態からプログラム停止状態へは SLEEP 命令の実行によって遷移します。プログラム停止状態からプログラム実行状態へは割り込みによって復帰します。プログラム実行状態であるアクティブモードとサブアクティブモードの間ではプログラムの実行を停止することなく直接遷移することもできます。また、アクティブモードからアクティブモード、サブアクティブモードからサブアクティブモードへ直接遷移することにより、同一モードで動作周波数を変更することができます。RES 入力によりすべてのモードからリセット状態に遷移します。表1に SLEEP 命令実行時の各モードへの遷移条件と割り込みによる復帰先を、表2に各動作モードでの LSI の内部状態を示します

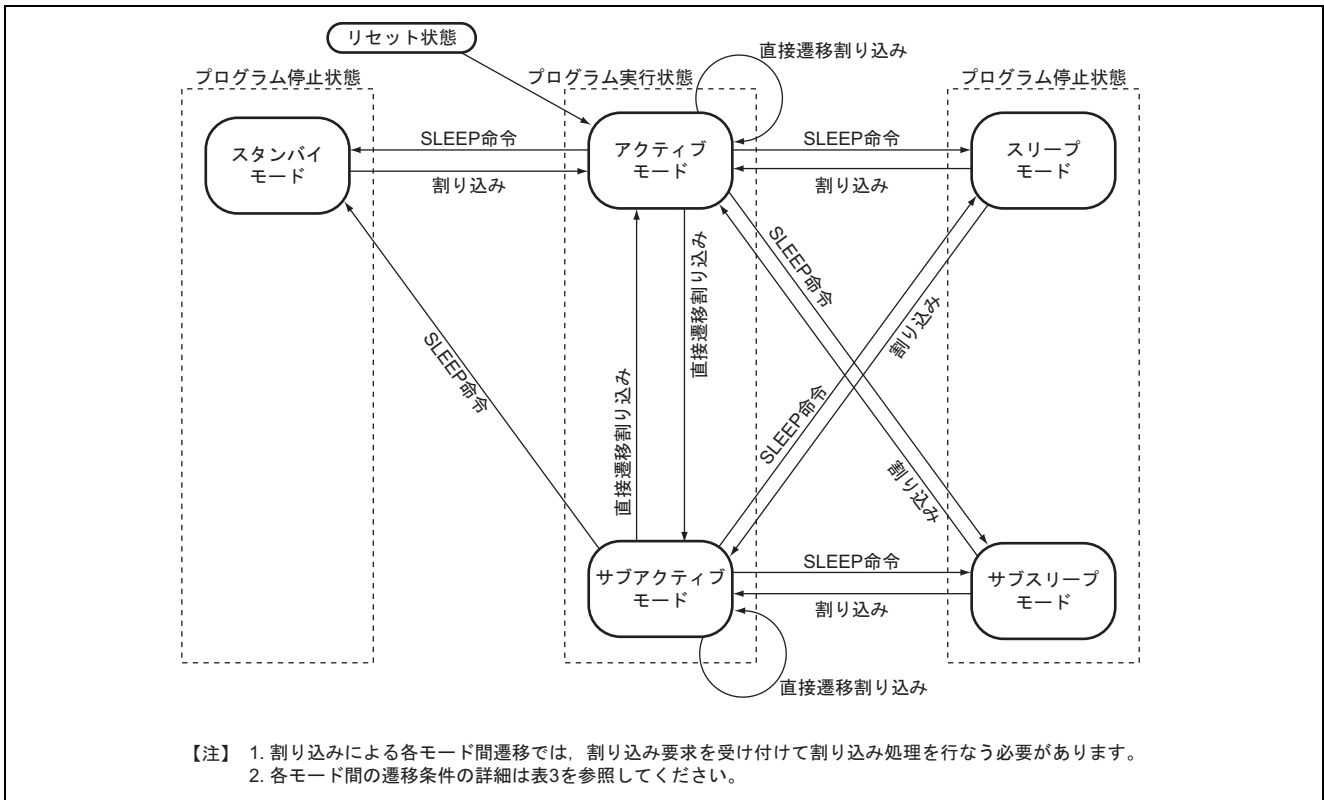


図1 モード遷移図

表1 SLEEP 命令実行後の状態と割り込みによる復帰先

SYSCR1 DTON	SYSCR2			SLEEP 命令実行後の状態	割り込みによる復帰先
	SSBY	SMSSEL	LSON		
0	0	0	0	スリープモード	アクティブモード
0	0	0	1	スリープモード	サブアクティブモード
0	0	1	0	サブスリープモード	アクティブモード
0	0	1	1	サブスリープモード	サブアクティブモード
0	1	x	x	スタンバイモード	アクティブモード
1	x	0*	0	アクティブモード (直接遷移)	X
1	x	x	1	サブアクティブモード (直接遷移)	

【注】 x : Don't care

* SMSSEL = 1 で状態遷移を行なった場合、タイマV, SCI3, A/D コンバータはリセットされ、各レジスタの値は初期値に戻ります。アクティブモード遷移後に、これらの機能を使用する場合は、各レジスタの再設定が必要です。

表2 各動作モードで LSI の状態

機能		アクティブ	スリープ	サブアクティブ	サブスリープ	スタンバイ
システムクロック発振器		動作	動作	停止	停止	停止
サブクロック発振器		動作	動作	動作	動作	動作
CPU	命令実行	動作	停止	動作	停止	停止
	レジスタ	動作	保持	動作	保持	保持
RAM		動作	保持	動作	保持	保持
I/O ポート		動作	保持	動作	保持	レジスタは保持, 出力はハイインピーダンス
外部割り込み	IRQ3 ~ IRQ0	動作	動作	動作	動作	動作
	WKP5 ~ WKP0	動作	動作	動作	動作	動作
周辺モジュール	タイマ A	動作	動作	時計用タイムベース機能選択時は動作, インターバルタイマ選択時は保持		
	タイマ V	動作	動作	リセット	リセット	リセット
	タイマ W	動作	動作	保持 (カウントクロックに内部クロックφを選択した場合, カウンタはサブクロックでカウントアップします*)		保持
	ウォッチドッグタイマ	動作	動作	保持 (ただし, カウントクロックに内部発振器を選択した場合は動作します*)		
	SCI3	動作	動作	リセット	リセット	リセット
	I ² C	動作	動作	保持*	保持	保持
	A/D コンバータ	動作	動作	リセット	リセット	リセット

【注】 * サブアクティブモードではレジスタのリード/ライトが可能です。

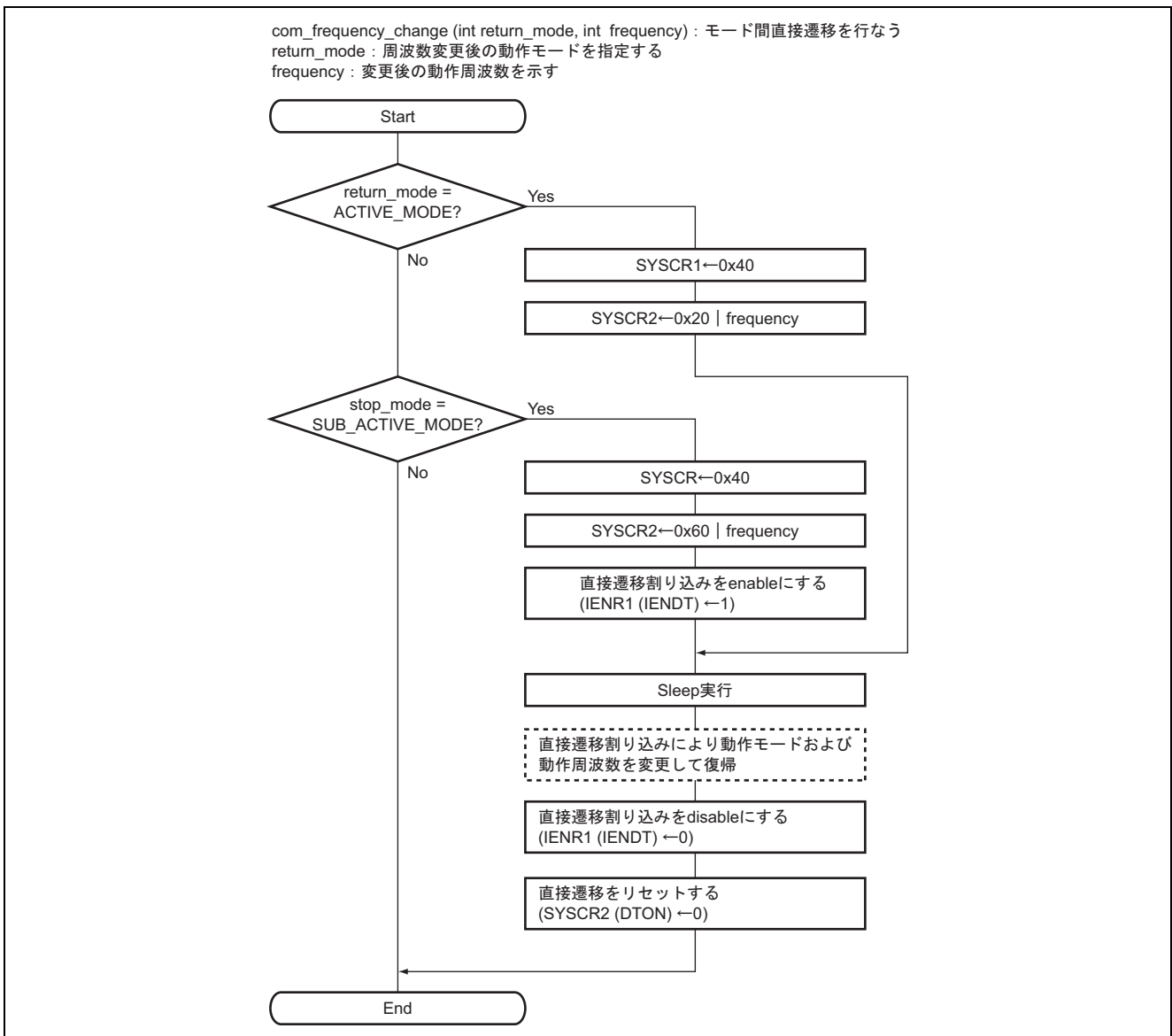
3.6 使用レジスタ一覧

本サンプルプログラムで使用する H8 マイコンの内部レジスタの一覧を示します。内容の詳細は、H8/3664 グループハードウェアマニュアルを参照してください。

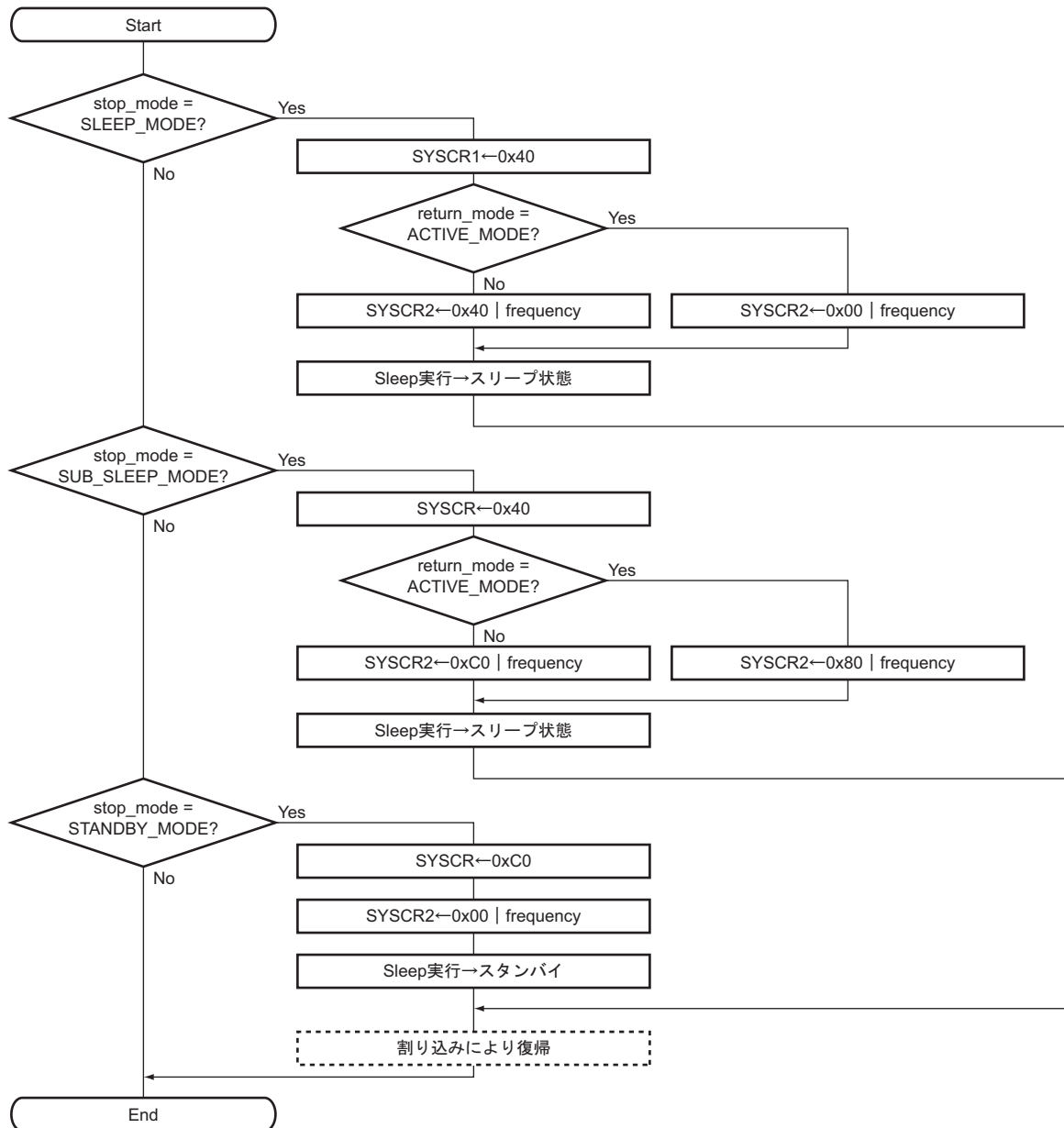
名称	概要
システムコントロールレジスタ 1 (SYSCR1)	低消費電力モードの制御
システムコントロールレジスタ 2 (SYSCR2)	低消費電力モードの制御

3.7 フローチャート

プログラムの処理フローを以下に示します。



com_mode_change (int stop_mode, int return_mode, int frequency) : モード間遷移を行なう
 stop_mode : 休止モードを指定する
 return_mode : 周波数変更後の動作モードを指定する
 frequency : 変更後の動作周波数を示す



3.8 プログラムリスト

```

/* ----- */
/* ----- */
/* 1 . サンプルプログラム 11-A define 定義----- */
/* ----- */
/* ----- */

/*****
/* 周波数変更, モード変更用 */
/*****

#define OSC_PER_1      0x00
#define OSC_PER_8      0x10
#define OSC_PER_16     0x14
#define OSC_PER_32     0x18
#define OSC_PER_64     0x1C
#define OSC_PER_W8     0x00
#define OSC_PER_W4     0x01
#define OSC_PER_W2     0x02

#define ACTIVE_MODE    0
#define SUB_ACTIVE_MODE 1

#define SLEEP_MODE     0
#define SUB_SLEEP_MODE 1
#define STANDBY_MODE   2

/* ----- */
/* ----- */
/* 2 . サンプルプログラム 11-B プロトタイプ宣言----- */
/* ----- */
/* ----- */

/* モード変更処理 */
void com_mode_change (int stop_mode , int return_mode , int frequency) ;
void com_frequency_change (int return_mode , int frequency ) ;

```

```

/* ----- */
/* ----- */
/* 4 . サンプルプログラム 11-c 共通処理ソースコード----- */
/* ----- */
/* ----- */

/*****
/*****
/*****
/*
/*
/*          低消費電力モード
/*
/*
/*****
/*****
/*****

/*****
/* 1. モジュール名称 : com_mode_change
/* 2. 機能概要 : モード間遷移を行なう
/*   パラメータの組み合わせ
/* stop_mode      return_mode      frequency
/* SLEEP_MODE     ACTIVE_MODE      OSC_PER_1/8/16/32/64
/* SLEEP_MODE     SUB_ACTIVE_MODE   OSC_PER_W8/4/2
/* SUB_SLEEP_MODE ACTIVE_MODE      OSC_PER_1/8/16/32/64
/* SUB_SLEEP_MODE SUB_ACTIVE_MODE   OSC_PER_W8/4/2
/* STANDBY_MODE   ACTIVE_MODE      OSC_PER_1/8/16/32/64
/*
/*****

void com_mode_change (int stop_mode , int return_mode , int frequency )
{

switch (stop_mode){
case SLEEP_MODE :
/*****
/*   SYSCR1 の設定 :
/*   SSBSY   = 0 sleep 実行時スリープモードに遷移
/*   STS2:0  = 100 スタンバイ スリープ以降時のクロック安定までの待ち時間 8.2ms に設定
/*   NESEL   = 0 ; ノイズ除去サンプリング周波数を 8MHzk の 16 分周
/*****
SYSCR1.BYTE   = 0x40 ;

/*****
/*   SYSCR2 の設定 :
/*   SMSEL   = 0 スリープモード
/*   LSON    = 0/1
/*   DTON    = 0 sleep 命令実行時の直接遷移を抑制
/*   MA2:0   = 000 メインクロック動作周波数選択(引数 frequency で設定)
/*   SA1:0   = 00 サブクロック動作周波数選択(引数 frequency で設定)
/*****
if (return_mode == ACTIVE_MODE){
    SYSCR2.BYTE   = 0x00 | frequency ;
}
else{
    SYSCR2.BYTE   = 0x40 | frequency ;
}
}
}

```

```

/*****/
/*  sleep させ, スリープモードにする */
/*****/
sleep();

/* ===== */
/* === 割り込みによりここに戻ってくる === */
/* ===== */

break;

case SUB_SLEEP_MODE :
/*****/
/*  SYSCR1 の設定 : */
/*      SSBSY      = 0  sleep 実行時スリープモードに遷移 */
/*      STS2:0     = 100 スタンバイ スリープ以降時のクロック安定までの待ち時間 8.2ms に設定 */
/*      NESEL      = 0 ; ノイズ除去サンプリング周波数を 8MHzk の 16 分周 */
/*****/
SYSCR1.BYTE      = 0x40 ;

/*****/
/*  SYSCR2 の設定 : */
/*      SMSEL      = 1 サブスリープモード */
/*      LSON       = 1/0 */
/*      DTON       = 0  sleep 命令実行時の直接遷移を抑制 */
/*      MA2:0      = 000 メインクロック動作周波数選択(引数 frequency で設定) */
/*      SA1:0      = 00 サブクロック動作周波数選択(引数 frequency で設定) */
/*****/
if (return_mode == ACTIVE_MODE){
    SYSCR2.BYTE    = 0x80 | frequency ;
}
else{
    SYSCR2.BYTE    = 0xC0 | frequency ;
}

/*****/
/*  sleep させ, スリープモードにする */
/*****/
sleep();

/* ===== */
/* === 割り込みによりここに戻ってくる === */
/* ===== */

break;

case STANDBY_MODE :
/*****/
/*  SYSCR1 の設定 : */
/*      SSBSY      = 1 スリープモード実行時スタンバイモードに遷移 */
/*      STS2:0     = 100 スタンバイ スリープ以降時のクロック安定までの待ち時間 8.2ms に設定 */
/*      NESEL      = 0 ; ノイズ除去サンプリング周波数を 8MHzk の 16 分周 */
/*****/
SYSCR1.BYTE      = 0xC0 ;

```

```

/*****/
/*  SYSCR2 の設定      :                               */
/*      SMSEL      = x                               */
/*      LSON       = x                               */
/*      DTON       = 0 sleep 命令実行時の直接遷移を抑制                               */
/*      MA2:0      = 000 メインクロック動作周波数選択(引数 frequency で設定)           */
/*      SA1:0      = 00 サブクロック動作周波数選択(引数 frequency で設定)           */
/*****/
SYSCR2.BYTE = 0x00 | frequency ;

/*****/
/*  sleep させ, スタンバイモードにする                               */
/*****/
sleep() ;
/* ===== */
/* == 割込みによりここに戻ってくる                               == */
/* ===== */

break ;

}

}

/*****/
/*  1.モジュール名称 : com_frequency_change                               */
/*  2.機能概要 : モード間直接遷移を行なう                               */
/*  パラメータの組み合わせ                               */
/*      return_mode      frequency                               */
/*      ACTIVE_MODE      OSC_PER_1/8/16/32/64                   */
/*      SUB_ACTIVE_MODE   OSC_PER_W8/4/2                         */
/*      ===== */
/*****/
void com_frequency_change (int return_mode , int frequency )
{
    int i , j ;
    unsigned int  h8_addr;

    switch (return_mode){
        case ACTIVE_MODE :

            /*****/
            /*  SYSCR1 の設定      :                               */
            /*      SSBSY      = 0                               */
            /*      STS2:0     = 100 スタンバイ スリープ以降時のクロック安定までの待ち時間 8.2ms に設定           */
            /*      NESEL     = 0 ; ノイズ除去サンプリング周波数を 8MHzk の 16 分周                               */
            /*****/
            SYSCR1.BYTE = 0x40 ;

```

```

/*****
/*  SYSCR2 の設定 :
/*      SMSEL      = 0
/*      LSON       = 0
/*      DTON       = 1 sleep 命令実行時の直接遷移を設定
/*      MA2:0      = 000 メインクロック動作周波数選択(引数 frequency で設定)
/*      SA1:0      = 00 サブクロック動作周波数選択(引数 frequency で設定)
/*****
SYSCR2.BYTE = 0x20 | frequency ;

break ;

case SUB_ACTIVE_MODE :

/*****
/*  SYSCR1 の設定 :
/*      SSBSY      = 0
/*      STS2:0     = 100 スタンバイ スリープ以降時の加減安定までの待ち時間 8.2ms に設定
/*      NESEL      = 0 ; ノイズ除去サンプリング周波数を 8MHzk の 16 分周
/*****
SYSCR1.BYTE = 0x40 ;

/*****
/*  SYSCR2 の設定 :
/*      SMSEL      = 0 未使用
/*      LSON       = 1 ロースピードフラグ
/*      DTON       = 1 sleep 命令実行時の直接遷移を設定
/*      MA2:0      = 000 メインクロック動作周波数選択(引数 frequency で設定)
/*      SA1:0      = 00 サブクロック動作周波数選択(引数 frequency で設定)
/*****
SYSCR2.BYTE = 0x60 | frequency ;

break ;

}

/*  直接遷移割込みを enable にする
IENR1.BIT.IENDT = 1 ;

/*****
/*  sleep させ、直接遷移の割込みを発生させる
/*  直接遷移用の割込みベクタールーチンを用意しておくこと
/*****
sleep() ;

/* =====
/* === ここで直接遷移の割込みが発生し、周波数が指定のものと切り替わる ===
/* =====

/*****
/*  割込み禁止解除
/*****

/*  直接遷移割込みを抑制する
IENR1.BIT.IENDT = 0 ;

/*  直接遷移をリセットする
SYSCR2.BIT.DTON = 0 ;

}

```


4. 参考文献

- H8/3664 グループ ハードウェアマニュアル (ルネサス テクノロジ発行)

ホームページとサポート窓口

ルネサステクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.09.24	—	初版発行
2.00	2006.09.01	1, 3	誤記修正

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。