

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、誤りが無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

サブスリープモードへの遷移

要旨

サブスリープモードへの遷移を行いません。

動作確認デバイス

H8/300H Tiny シリーズ –H8/3664–

目次

1. 仕様	2
2. 使用機能説明	3
3. 動作原理	5
4. ソフトウェア説明	6
5. フローチャート	9
6. プログラムリスト	11

1. 仕様

1. サブスリープモードへの遷移を行ないます。
2. アクティブモードで、SYSCR1 の SSBY が "X", SYSCR2 の LSON が "1", SMSEL が "X", DTON が "1" のとき SLEEP 命令を実行することにより、サブアクティブモードへ直接遷移します。
3. サブアクティブモードで、SYSCR1 の SSBY が "0", SYSCR2 の LSON が "1", SMSEL が "1", DTON が "0" のとき SLEEP 命令を実行することにより、サブスリープモードへ遷移します。
4. サブスリープモードは、タイマ A 割り込みにより解除しサブアクティブモードへ復帰します。
5. タイマ A 割り込み処理の中で、LED の制御とタイマ A 割り込み発生回数のカウントを行ないます。タイマ A 割り込みは 0.5s ごとに発生し、タイマ A 割り込みが 120 回発生したところで、タイマ A 割り込みの要求を禁止し、終了します。LED は 0.5s ごとに点灯/消灯を繰り返します。
6. タイマ A 割り込みが発生し、サブアクティブモードへ遷移すると、タイマ A 割り込み回数の判定を行ない、再びサブスリープモードへ遷移させます。この処理を、タイマ A 割り込みが 120 回発生するまで繰り返します。
7. LED はポート 7 の P74 出力端子に接続されているものとします。

【注】 X は 1, 0 どちらでも良い

2. 使用機能説明

- 本タスク例では、低消費電力モードのサブスリープモードへの遷移を行いません。図1にサブスリープモードへのモード遷移図を示します。以下にサブスリープモードの機能の説明を示します。
 - アクティブモードで、SYSCR1のSSBYが"X", SYSCR2のLSONが"1", SMSELが"X", DTONが"1"のとき SLEEP 命令を実行することにより、サブアクティブモードへ直接遷移し、サブアクティブモードで、SYSCR1のSSBYが"0", SYSCR2のLSONが"1", SMSELが"1", DTONが"0"のとき SLEEP 命令を実行することにより、サブスリープモードへ遷移します。
 - サブスリープモードでは、タイマ A、ウォッチドッグタイマ、タイマ W、I²C、以外の内蔵周辺機能は動作を停止します。
 - 規定の電圧が与えられている限り、CPU と一部の内蔵周辺機能の内部レジスタ、内蔵 RAM の内容は保持され、I/O ポートは遷移前の状態を保持します。
 - サブスリープモードの解除は、タイマ A、IRQ3~IRQ0、WKP5~WKP0 割り込み、 $\overline{\text{RES}}$ 端子入力によって行なわれます。
 - 割り込みによる解除は、割り込み要求が発生するとサブスリープモードは解除され、割り込み例外処理を開始します。
 - CCR の I ビットが "1" の場合、あるいは割り込みイネーブルレジスタ 1 により当該割り込みの受け付けが禁止されている場合は、サブスリープモードは解除されません。
 - $\overline{\text{RES}}$ 端子による解除は、 $\overline{\text{RES}}$ 端子を "Low" レベルにするとリセット状態に遷移し、サブスリープモードは解除されます。発振安定時間経過後、 $\overline{\text{RES}}$ 端子を "High" レベルにすると CPU はリセット例外処理を開始します。システムクロックの発振開始と同時に LSI 全体にシステムクロックが供給されます。
 - $\overline{\text{RES}}$ 端子は、必ずシステムクロックの発振が安定するまで、Low レベルを保持しなければなりません。
 - 本タスク例では、サブスリープモードの解除にタイマ A 割り込みを使用します。なお、サブスリープモードの解除後は、サブアクティブモードに遷移します。
 - サブアクティブモードで SYSCR1 の SSBY が "X", SYSCR2 の LSON が "0", SMSEL が "0", DTON が "1" のとき SLEEP 命令を実行することにより、SYSCR1 の STS2~STS0 により設定された待機時間を経過した後、アクティブモードへ遷移します。
 - サブアクティブモード解除後の発振安定の待機時間は、SYSCR1 の STS2~STS0 により設定します。
 - 本タスク例では、動作周波数に 16MHz を使用し、待機ステート数を 131,072 ステートに設定します。(発振安定の待機時間：8.2ms)

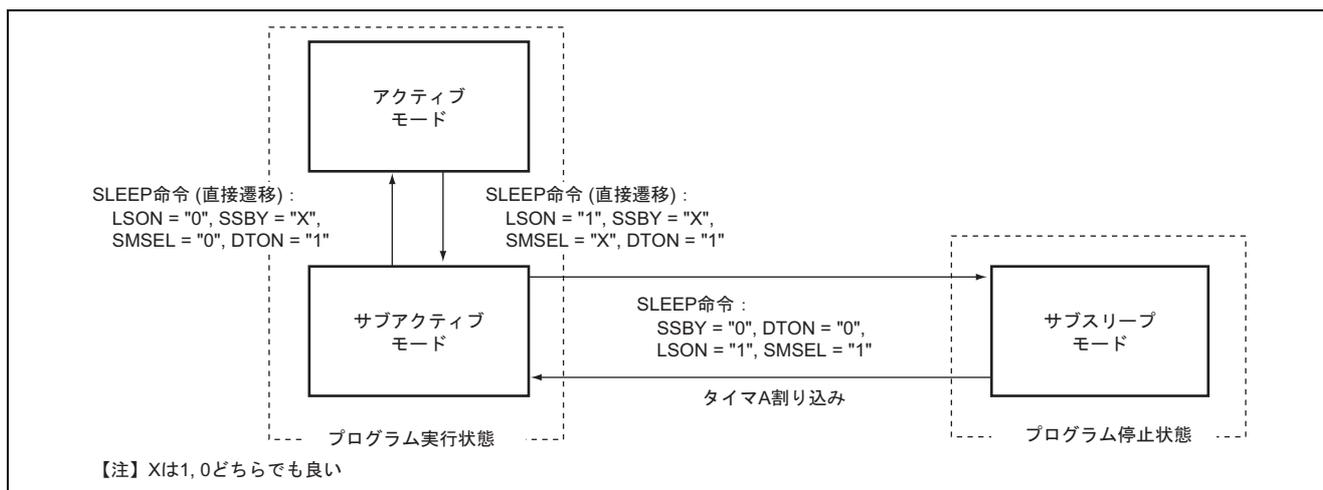


図1 サブスリープモードへの遷移におけるモード遷移図

2. 表 1 に本タスク例の機能割り付けを示します。表 1 に示すように機能を割り付け、サブスリープモードへの遷移を行いません。

表 1 機能割り付け

機能	機能割り付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P74 出力端子機能の設定
PDR7	P74 出力端子のデータの格納
P74	LED 出力
TMA	タイマ A 時計用タイムベース機能、および TCA オーバフロー周期の設定
TCA	時計用タイムベース機能により 0.5s でオーバフローする 8 ビットのアップカウンタ
IRRTA	タイマ A 割り込み要求の有無を反映
IENTA	タイマ A 割り込み要求の許可、または禁止を設定

3. 動作原理

図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりサブスリープモードへの遷移を行ないます。

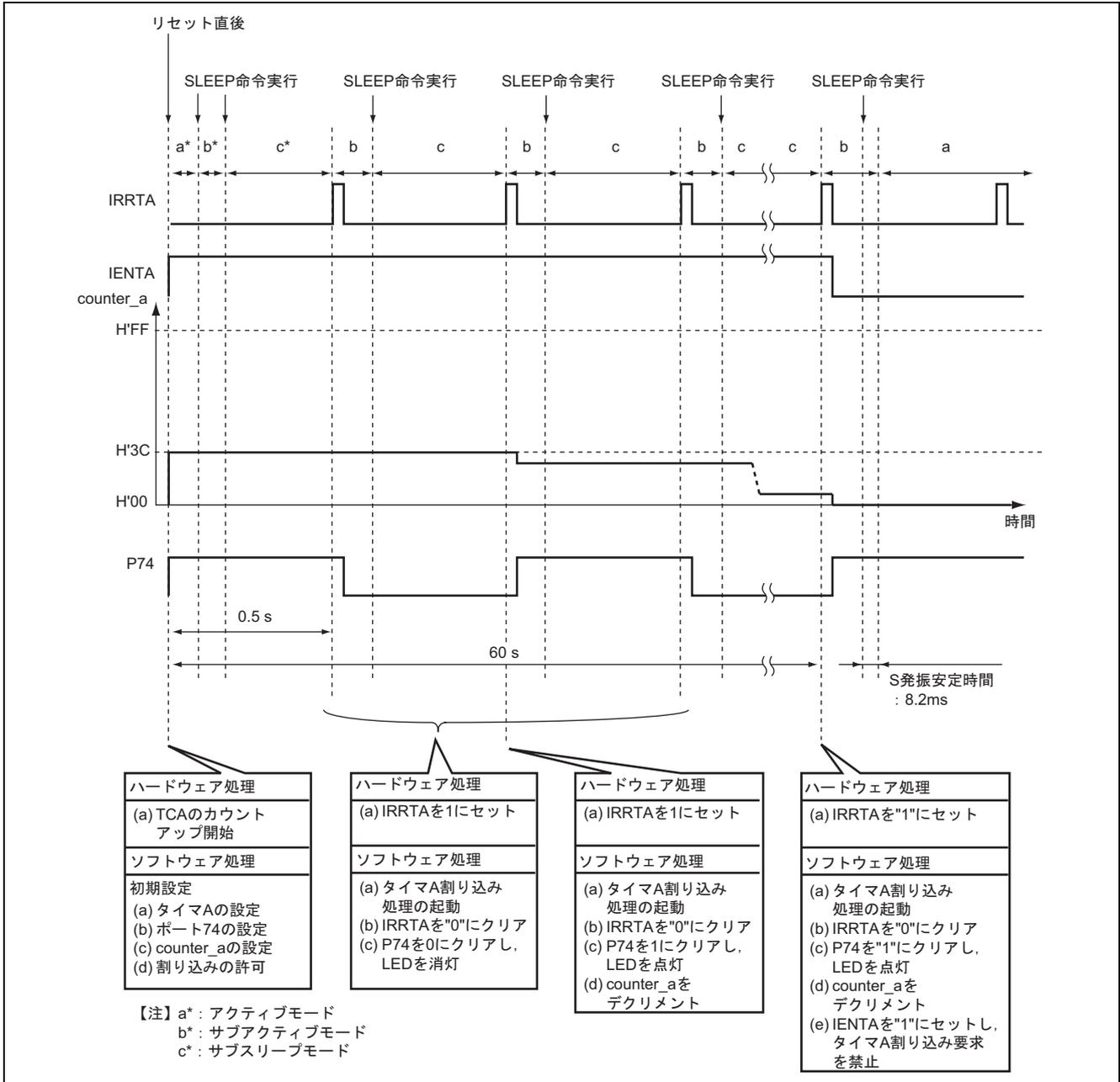


図2 サブスリープモードへの遷移の動作原理

4. ソフトウェア説明

4.1 モジュール説明

表 2 に本タスク例におけるモジュール説明を示します。

表 2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	タイマ A 割り込みの設定, ポート 7 の設定, counter_a の設定, 割り込みの許可, サブアクティブモードへの遷移, サブスリープモードへの遷移, アクティブモードへの遷移を行なう
LED 制御	taint	タイマ A 割り込み処理ルーチンで, LED の制御, 8 ビットタイマ A 割り込みカウンタのデクリメント, 60s 経過後, タイマ A 割り込み要求の禁止を行なう
直接遷移	dtint	直接遷移割り込み処理ルーチンで, 直接遷移割り込み要求フラグのクリアを行なう

4.2 引数の説明

本タスク例では, 引数は使用していません。

4.3 使用内部レジスタ説明

表 3 に本タスク例における使用内部レジスタ説明を示します。

表 3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMA	タイマモードレジスタ A : TMA = H'19 のとき, タイマ A 機能を時計用タイムベース機能に, TCA のオーバフロー周期を 0.5s に設定	H'FFA6	H'19	
TCA	タイマカウンタ A : PSW 出力クロックを入力とし, 時計用タイムベース機能により 0.5s で オーバフローする 8 ビットのアップカウンタ	H'FFA7	H'00	
PDR7	P74	ポートデータレジスタ 7 (ポートデータレジスタ 74) : P74 = 0 のとき, P74 端子の出力レベルは Low : P74 = 1 のとき, P74 端子の出力レベルは High	H'FFDA ビット 4	1
PCR7	PCR74	ポートコントロールレジスタ 7 (ポートコントロールレジスタ 74) : PCR74 = 1 のとき, P74 端子を出力端子に設定	H'FFEA ビット 4	1
SYSCR1	SSBY	システムコントロールレジスタ 1 (ソフトウェアスタンバイ) : SSBY = X のとき, サブアクティブモード, アクティブモードに直接遷移 (X は 1, 0 どちらでも良い)	H'FFF0 ビット 7	本タスク例 では 1
	STS2 STS1 STS0	システムコントロールレジスタ 1 (スタンバイタイムセレクト 2, 1, 0) : STS2 = 1, STS1 = 0, STS0 = 0 のとき, 待機状態数を 131.072 ステートに設定	H'FFF0 ビット 6 ビット 5 ビット 4	STS2 = 1 STS1 = 0 STS0 = 0
SYSCR2	SMSEL	システムコントロールレジスタ 2 (スリープモード選択) : SMSEL = 0 のとき, スリープ命令実行後, スリープモードを設定	H'FFF1 ビット 7	0
	LSON	システムコントロールレジスタ 2 (ロースピードオンフラグ) : LSON = 1 のとき, スリープ命令実行後, スリープモード, サブスリープモード, サブアクティブモード (直接遷移) の各 1 モードを設定	H'FFF1 ビット 6	1
	DTON	システムコントロールレジスタ 2 (ダイレクトトランスファオンフラグ) : DTON = 1 のとき, スリープ命令実行後, アクティブモード, サブアクティブモードの各 1 モードを設定	H'FFF1 ビット 3	1
	MA2 MA1 MA0	システムコントロールレジスタ 2 (アクティブモードクロックセレクト 2~0) : MA2 = 1, MA1 = 1, MA0 = 1, のとき, ϕ OSC/64 をクロックに設定	H'FFF1 ビット 4 ビット 3 ビット 2	MA2 = 1 MA1 = 1 MA0 = 1
	SA1 SA0	システムコントロールレジスタ 2 (サブアクティブモードクロックセレクト 1, 0) : SA1 = 0, SA0 = 0 のとき, サブアクティブモードの CPU の動作クロックを ϕ W/8 に設定	H'FFF1 ビット 1 ビット 0	SA1 = 0 SA0 = 0
IENR1	IENDT	割り込みイネーブルレジスタ 1 (直接遷移割り込みイネーブル) : IENDT = 0 のとき, 直接遷移による割り込み要求を禁止 : IENDT = 1 のとき, 直接遷移による割り込み要求を許可	H'FFF4 ビット 7	1
	IENTA	割り込みイネーブルレジスタ 1 (タイマ A 割り込みイネーブル) : IENTA = 0 のとき, タイマ A 割り込み要求を禁止 : IENTA = 1 のとき, タイマ A 割り込み要求を許可	H'FFF4 ビット 6	1

表 3 使用内部レジスタ説明 (つづき)

レジスタ名		機能	アドレス	設定値
IRR	IRRDT	割り込みフラグレジスタ 1 (直接遷移割り込み要求フラグ) : IRRDT = 0 のとき、直接遷移による割り込みが要求されていない : IRRDT = 1 のとき、直接遷移による割り込みが要求されている	H'FFF6 ビット 7	0
	IRRTA	割り込みフラグレジスタ 1 (タイマ A 割り込み要求フラグ) : IRRTA = 0 のとき、タイマ A 割り込みが要求されていない : IRRTA = 1 のとき、タイマ A 割り込みが要求されている	H'FFF6 ビット 6	

4.4 使用 RAM 説明

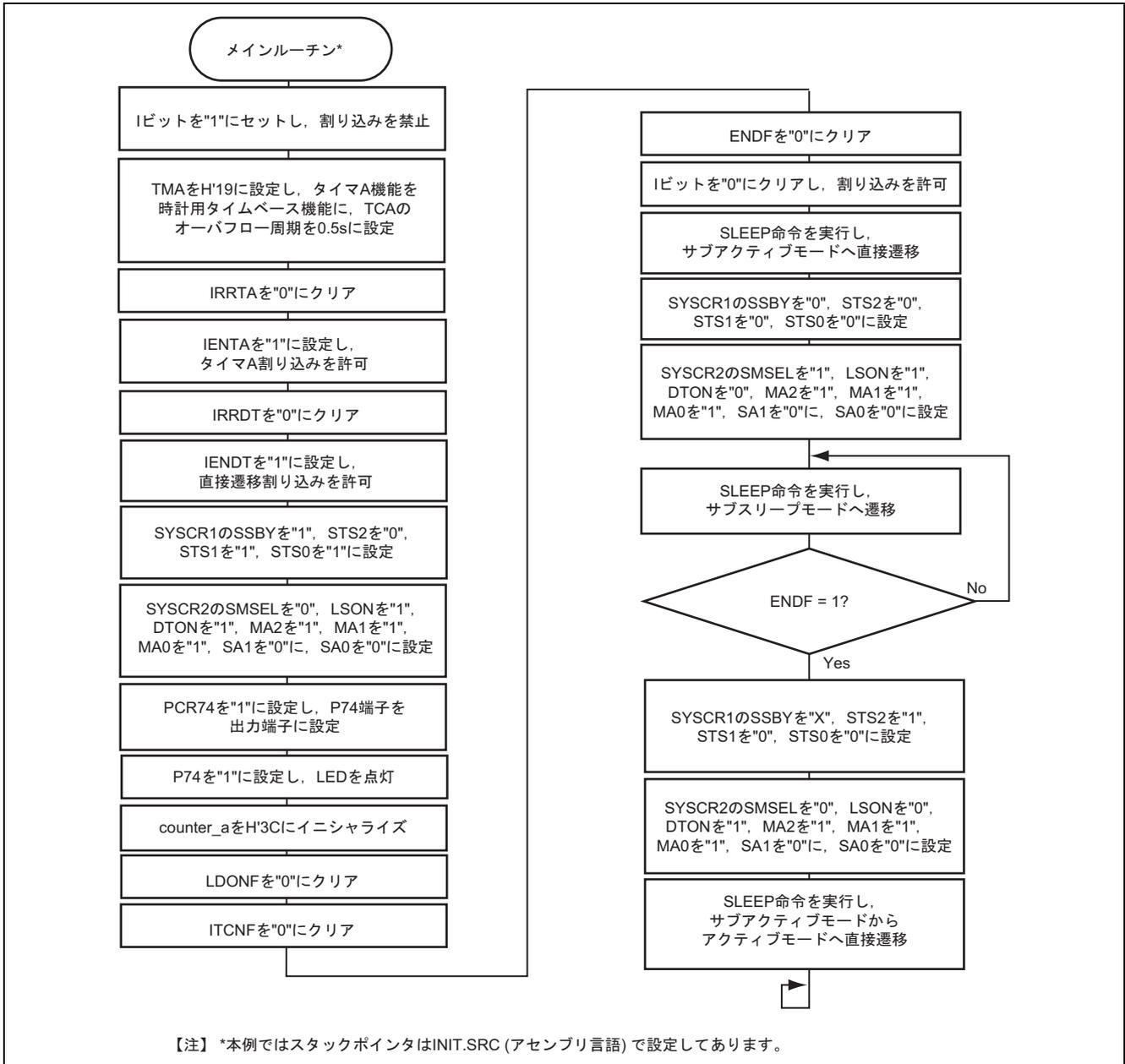
表 4 に本タスク例で使用する RAM の説明を示します。

表 4 使用 RAM 説明

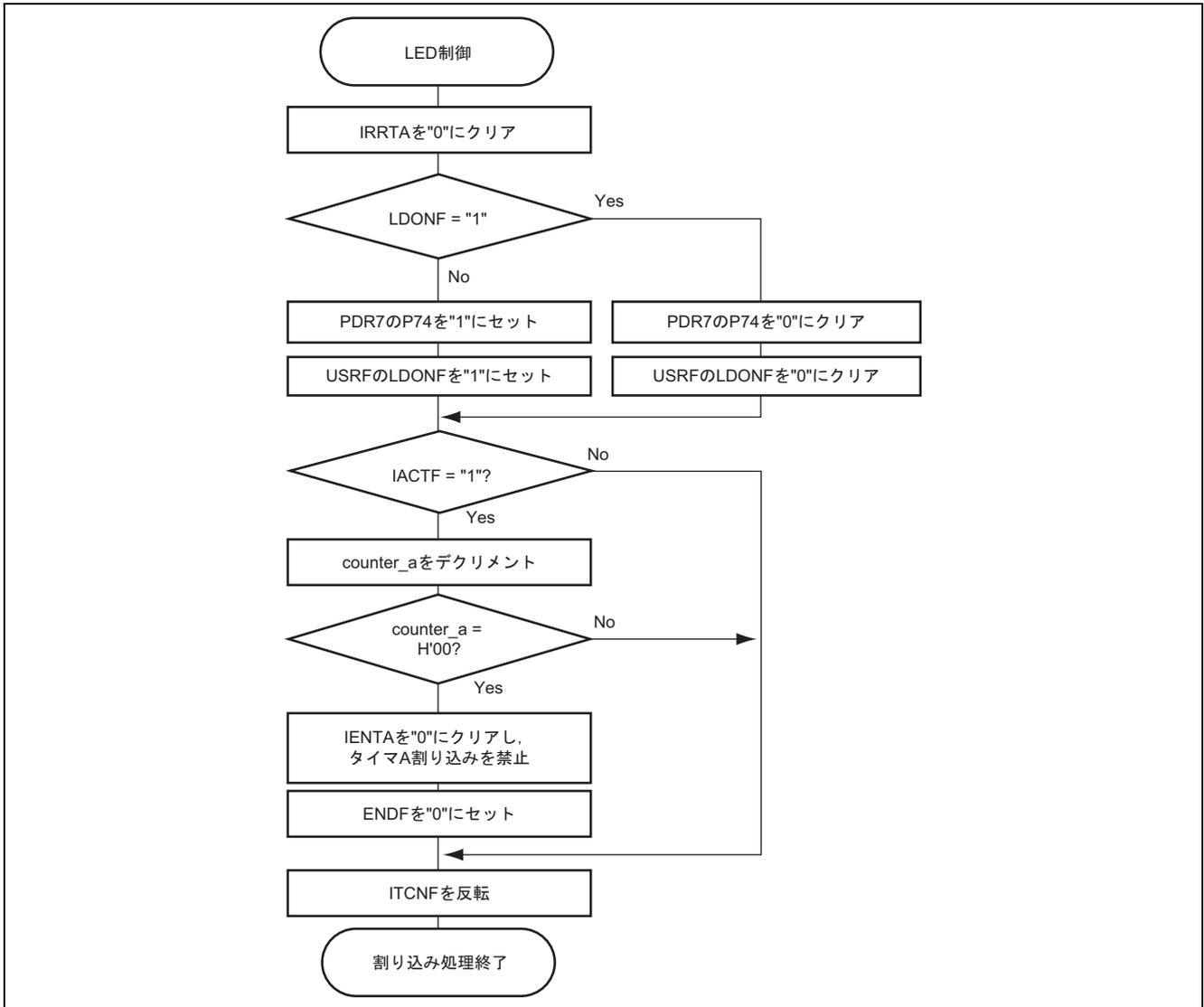
ラベル名	機能	アドレス	使用モジュール名	
counter_a	タイマ A の割り込みをカウントするダウンカウンタ	H'FB80	メインルーチン	
USRF	ENDF	60s 経過したか否かを判定するフラグ	H'FB81 ビット 2	メインルーチン
	ITCNF	タイマ A 割り込みが奇数回目か偶数回目かを判定するフラグ	H'FB81 ビット 1	LED 制御
	LDONF	LED の ON/OFF を判定するフラグ	H'FB81 ビット 0	LED 制御

5. フローチャート

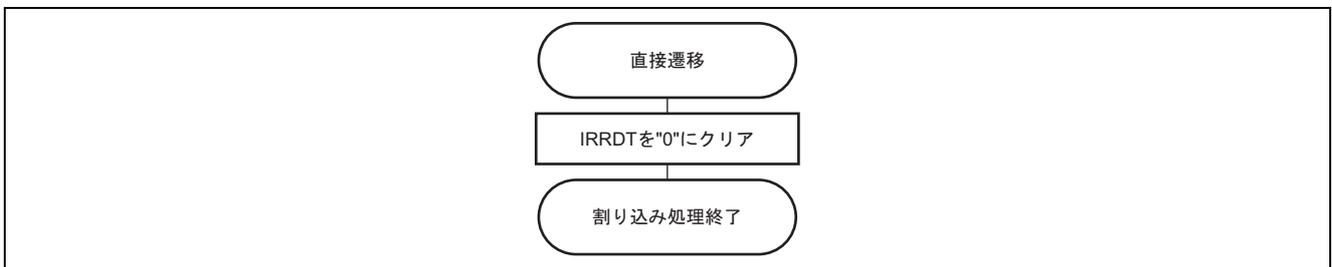
5.1 メインルーチン



5.2 タイマ A 割り込み処理ルーチン



5.3 直接遷移割り込み処理ルーチン



6. プログラムリスト

INIT.SRC (プログラムリスト)

```

        .EXPORT          _INIT
        .IMPORT          _main
;
        .SECTION        P, CODE
_INIT:
        MOV.W           #H'FF80,R7
        LDC.B           #B'10000000,CCR
        JMP              @_main
;
        .END

/*****
/*
/* H8/300H Tiny Series -H8/3664-
/* Application Note
/*
/* 'Transition to Sucsleep Mode'
/*
/*
/* Function
/* : Power-Down Mode
/*     Subsleep Mode
/*
/* External Clock : 16MHz
/* Internal Clock : 16MHz
/* Sub Clock      : 32.768kHz
/*
*****/

#include    <machine.h>

/*****
/* Symbol Definition
*****/
struct BIT {
    unsigned char    b7:1;    /* bit7
    unsigned char    b6:1;    /* bit6
    unsigned char    b5:1;    /* bit5
    unsigned char    b4:1;    /* bit4
    unsigned char    b3:1;    /* bit3
    unsigned char    b2:1;    /* bit2
    unsigned char    b1:1;    /* bit1
    unsigned char    b0:1;    /* bit0
};

#define TMA          *(volatile unsigned char *)0xFFA6    /* Timer Mode Register A
#define TCA          *(volatile unsigned char *)0xFFA7    /* Timer Counter A
#define PDR7_BIT    (*(struct BIT *)0xFFDA)              /* Port Data Register 7
#define P74         PDR7_BIT.b4                          /* Port Data Register 7 bit4
#define PCR7_BIT    (*(struct BIT *)0xFFEA)              /* Port Control Register 7
#define PCR74       PCR7_BIT.b4                          /* Port Control Register 7 bit4
#define SYSCR1      *(volatile unsigned char *)0xFFF0    /* System Control Register 1
#define SYSCR1_BIT  (*(struct BIT *)0xFFF0)              /* System Control Register 1
#define SSBY        SYSCR1_BIT.b7                        /* Software Standby
#define STS2        SYSCR1_BIT.b6                        /* Standby Timer Select 2
#define STS1        SYSCR1_BIT.b5                        /* Standby Timer Select 1
#define STS0        SYSCR1_BIT.b4                        /* Standby Timer Select 0

```

```

#define NESEL        SYSCR1_BIT.b3                /* Noise Elimination Sampling Frequency Select */
#define SYSCR2        *(volatile unsigned char *)0xFFF1 /* System Control Register 2 */
#define SYSCR2_BIT    (*(struct BIT *)0xFFF1)      /* System Control Register 2 */
#define LSON          SYSCR2_BIT.b6                /* Low Speed On Flag */
#define DTON          SYSCR2_BIT.b5                /* Direct Transfer On Flag */
#define MA1           SYSCR2_BIT.b3                /* Active Mode Clock Select 1 */
#define MA0           SYSCR2_BIT.b2                /* Active Mode Clock Select 0 */
#define SA1           SYSCR2_BIT.b1                /* Subactive Mode Clock Select 1 */
#define SA0           SYSCR2_BIT.b0                /* Subactive Mode Clock Select 0 */
#define IENR1_BIT    (*(struct BIT *)0xFFF4)      /* Interrupt Enable Register 1 */
#define IENDT         IENR1_BIT.b7                /* Direct Transfer Interrupt Enable */
#define IENTA        IENR1_BIT.b6                /* Timer A Interrupt Enable */
#define IRR1_BIT     (*(struct BIT *)0xFFF6)      /* Interrupt Request Register 1 */
#define IRRDT        IRR1_BIT.b7                /* Direct Transfer Interrupt Request Flag */
#define IRRTA        IRR1_BIT.b6                /* Timer A Interrupt Request Flag */

#pragma interrupt (dtint)
#pragma interrupt (taint)
/*****
/*      関数定義
*****/
extern void INIT ( void ); /* SP Set */
void main ( void );
void dtint ( void );
void taint ( void );
void sleep ( void );

/*****
/* RAM define
*****/
    unsigned char counter_a;
    unsigned char USRF; /* User Flag Area */

#define USRF_BIT    (*(struct BIT *)&USRF)
#define ENDF        USRF_BIT.b2                /* End Flag */
#define IACTF       USRF_BIT.b1                /* Timer A Interrupt Counter Flag */
#define LDONF       USRF_BIT.b0                /* LED On Flag */

/*****
/* Vector Address
*****/
#pragma section V1 /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
    INIT /* 00 Reset */
};
#pragma section V2 /* VECTOR SECTOIN SET */
void (*const VEC_TBL2[])(void) = {
    dtint /* Direct Transfer Interrupt */
};
#pragma section V3 /* VECTOR SECTOIN SET */
void (*const VEC_TBL3[])(void) = {
    taint /* timer A Interrupt */
};
#pragma section /* P */
/*****
/* Main Program
*****/

```

```

void main ( void )
{

    set_imask_ccr(1);                /* Interrupt Disable                */

    TMA = 0x19;                      /* Initialize Timer A Function      */

    IRRTA = 0;                       /* Clear IRRTA                     */

    IENTA = 1;                      /* Timer A Interrupt Enable        */

    IRRDT = 0;                       /* Clear IRRDT                     */

    IENDT = 1;                      /* Direct Transfer Interrupt Enable */

    SYSCR1 = 0xB0;                  /* Initialize Function of Subactive Mode 1 */
    SYSCR2 = 0x7C;                  /* Initialize Function of Subactive Mode 2 */

    P74 = 0;                        /* Initialize P74                   */
    PCR74 = 1;                      /* Initialize P74 Output Port      */

    counter_a = 0x3C;              /* Initialize 8bit Timer A Interrupt Counter */

    LDONF = 0;                     /* Initialize LDONF                */

    IACTF = 0;                     /* Initialize IACTF                */

    ENDF = 0;                      /* Initialize ENDF                 */

    set_imask_ccr(0);              /* Interrupt Enable                 */

    sleep();                        /* Transition to Subactive Mode     */

    SYSCR1 = 0x00;                  /* Initialize Function of Subsleeep Mode 1 */
    SYSCR2 = 0x5C;                  /* Initialize Function of Subsleeep Mode 2 */

    do{
        sleep();                    /* Transion to Subsleeep Mode     */
    }while(ENDF != 1);             /* ENDF = 1 ?                     */

    SYSCR1 = 0xC0;                  /* Initialize Function of Active Mode 1 */
    SYSCR2 = 0x3C;                  /* Initialize Function of Active Mode 2 */

    sleep();                        /* Transition to Active Mode       */

    while(1){
        ;
    }
}

/*****
/* Timer A Interrupt
*****/
void taint ( void )
{

    IRRTA = 0;                    /* Clear IRRTA                     */

```

```

if(LDONF == 1){
    P74 = 0;
    LDONF = 0;
}
else{
    P74 = 1;
    LDONF = 1;
}

if(IACTF == 1){
    counter_a--;
    if(counter_a == 0x00){
        IENTA = 0;
        ENDF = 1;
    }
}

IACTF = ~IACTF;

}

/*****
/* Direct Transfer Interrupt
*****/
void dtint ( void )
{
    IRRDT = 0;
}

```

6.1 リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'001A
CV3	H'0026
P	H'0100
B	H'FB80

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
2.00	2006.09.01	全頁	日立版からルネサス版へのフォーマット変更

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、慎重に制作したのですが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。