

2.25 サブスリープモードへの遷移

サブスリープモードへの遷移	使用機能	低消費電力モード：サブスリープモード
仕様	: Xは"1"、"0"どちらでも良い	
<p>(1) サブスリープモードへの遷移を行ないます。</p> <p>(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。</p> <p>(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。</p> <p>(4) サブスリープモードは、タイマA割込みにより解除しサブアクティブモードへ復帰します。</p> <p>(5) タイマA割込み処理の中で、LEDの制御とタイマA割込み発生回数のカウントを行ないます。タイマA割込みは0.5secごとに発生し、タイマA割込みが120回発生したところで、タイマA割込みを要求を禁止し、終了します。LEDは0.5secごとに点灯/消灯を繰り返します。</p> <p>(6) タイマA割込みが発生し、サブアクティブモードへ遷移すると、タイマA割込み回数の判定を行い、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割込みが120回発生するまで繰り返します。</p> <p>(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。</p>		
使用機能説明	<p>(1) 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ないます。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。</p> <ul style="list-style-type: none"> ・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、ウォッチドックタイマ、タイマW、IIC、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割込み、RES端子入力によって行われます。 ・割込みによる解除は、割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、あるいは割込みイネーブルレジスタ1により当該割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。 ・RES端子による解除は、RES端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、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。 ・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"X"、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。(発振安定時間：8.2ms) 	
<p style="text-align: right;">: Xは"1"、"0"どちらでも良い</p> <pre> graph TD subgraph "プログラム実行状態" AM[アクティブモード] SAM[サブアクティブモード] end subgraph "プログラム停止状態" SSM[サブスリープモード] end AM -- "SLEEP命令(直接遷移) : LSON='0', SSBY='X' SMSEL='X', DTON='1'" --> SAM SAM -- "SLEEP命令(直接遷移) : LSON='1', SSBY='X' SMSEL='X', DTON='1'" --> AM SAM -- "SLEEP命令 : SSBY='0', DTON='0' LSON='1', SMSEL='1'" --> SSM SSM -- "タイマA割込み" --> SAM </pre>		
<p>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</p>		

使用機能説明

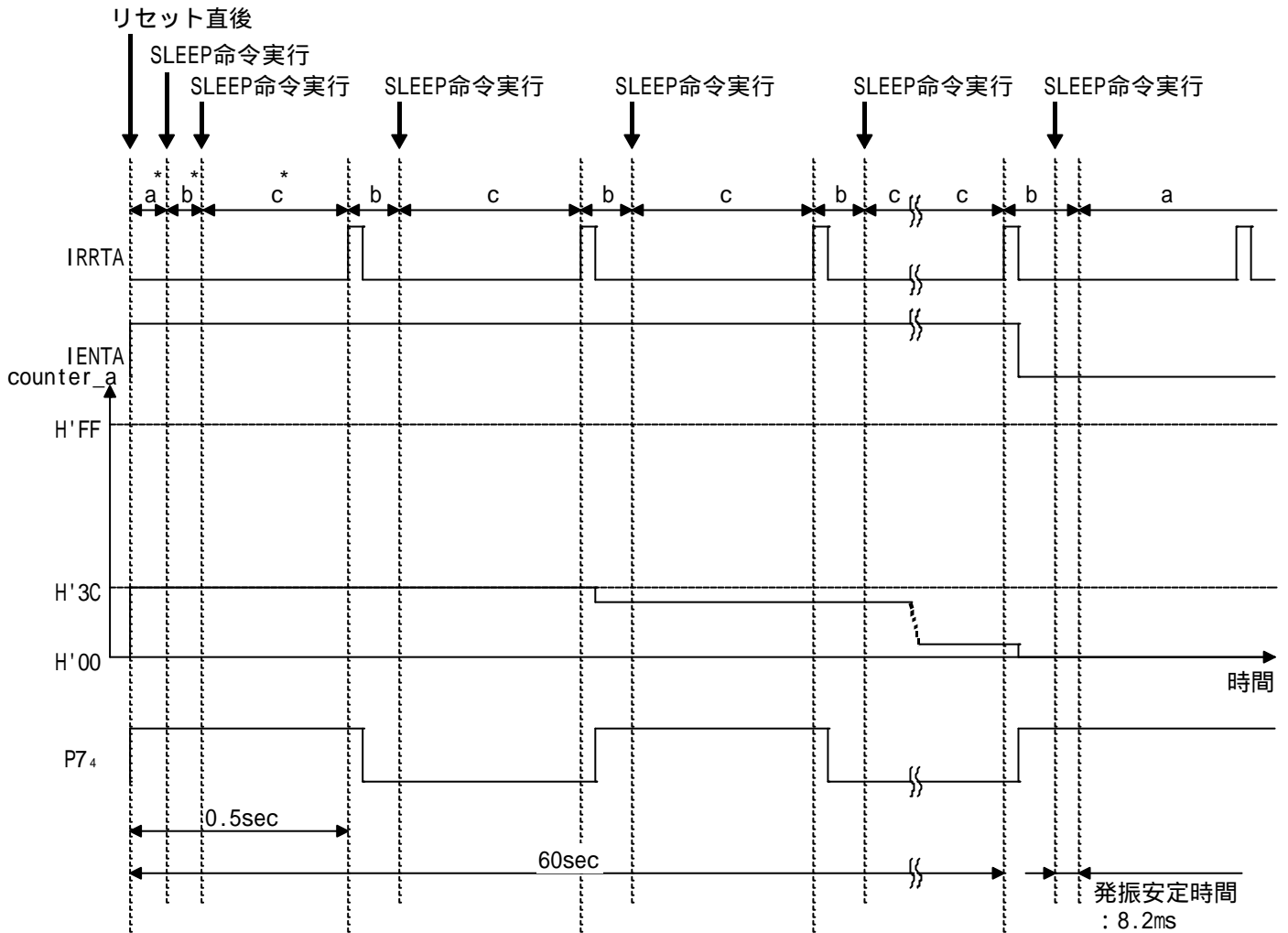
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、サブスリープモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
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	P7 ₄ 出力端子機能の設定
PDR7	P7 ₄ 出力端子のデータの格納
P7 ₄	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0.5sec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割込み要求の有無を反映
IENTA	タイマA割込み要求の許可、または禁止を設定

動作原理

(1) 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。



ハードウェア処理
(a) TCAのカウントアップ開始
ソフトウェア処理
初期設定
(a) タイマAの設定
(b) ポート7 ₄ の設定
(c) counter_aの設定
(d) 割込みの許可

ハードウェア処理
(a) IRRTAを"1"にセット
ソフトウェア処理
(a) タイマA割込み処理の起動
(b) IRRTAを"0"にクリア
(c) P7 ₄ を"0"にクリアし、LEDを消灯

ハードウェア処理
(a) IRRTAを"1"にセット
ソフトウェア処理
(a) タイマA割込み処理の起動
(b) IRRTAを"0"にクリア
(c) P7 ₄ を"1"にセットし、LEDを点灯
(d) counter_aをデクリメント

ハードウェア処理
(a) IRRTAを"1"にセット
ソフトウェア処理
(a) タイマA割込み処理の起動
(b) IRRTAを"0"にクリア
(c) P7 ₄ を"1"にセットし、LEDを点灯
(d) counter_aをデクリメント
(e) IENTAを"1"にセットし、タイマA割込み要求を禁止

【注】 * a: アクティブモード
 b: サブアクティブモード
 c: サブスリープモード

図2 サブスリープモードへの遷移の動作原理

ソフトウェア説明

(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割込みの設定、ポート7の設定、counter_aの設定、割込みの許可、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割込み処理ルーチンで、LEDの制御、8ビットタイマA割込みカウンタのデクリメント、60sec経過後、タイマA割込み要求の禁止を行なう
直接遷移	dtint	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(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のオーバーフロー周期を0.5sec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0.5secでオーバーフローする8ビットのアップカウンタ	H'FFA7	H'00
PDR7	P7 ₄ ポートデータレジスタ7 (ポートデータレジスタ7 ₄) : P7 ₄ =0のとき、P7 ₄ 端子の出力レベルは"Low" : P7 ₄ =1のとき、P7 ₄ 端子の出力レベルは"High"	H'FFDA ビット4	1
PCR7	PCR7 ₄ 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7 ₄) : PCR7 ₄ =1のとき、P7 ₄ 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状態に設定	H'FFF0 ビット6 ビット5 ビット4	STS2="1" STS1="0" STS0="0"
SYSCR2	SMSSEL システムコントロールレジスタ2 (スリープモード選択) : SMSSEL="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、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 $\omega/8$ に設定	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"

ソフトウェア説明

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
IENR1	IENDT 割込みイネーブルレジスタ1 (直接遷移割込みイネーブル) : IENDT="0"のとき、直接遷移による割込み要求を禁止 : IENDT="1"のとき、直接遷移による割込み要求を許可	H'FFF4 ビット7	1
	IENTA 割込みイネーブルレジスタ1 (タイマA割込みイネーブル) : IENTA="0"のとき、タイマA割込み要求を禁止 : IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IRR1	IRRDT 割込みフラグレジスタ1 (直接遷移割込み要求フラグ) : IRRDT="0"のとき、直接遷移による割込みが要求されていない : IRRDT="1"のとき、直接遷移による割込みが要求されている	H'FFF6 ビット7	0
	IRRTA 割込みフラグレジスタ1 (タイマA割込み要求フラグ) : IRRTA="0"のとき、タイマA割込みが要求されていない : IRRTA="1"のとき、タイマA割込みが要求されている	H'FFF6 ビット6	0

(4) 使用RAM説明

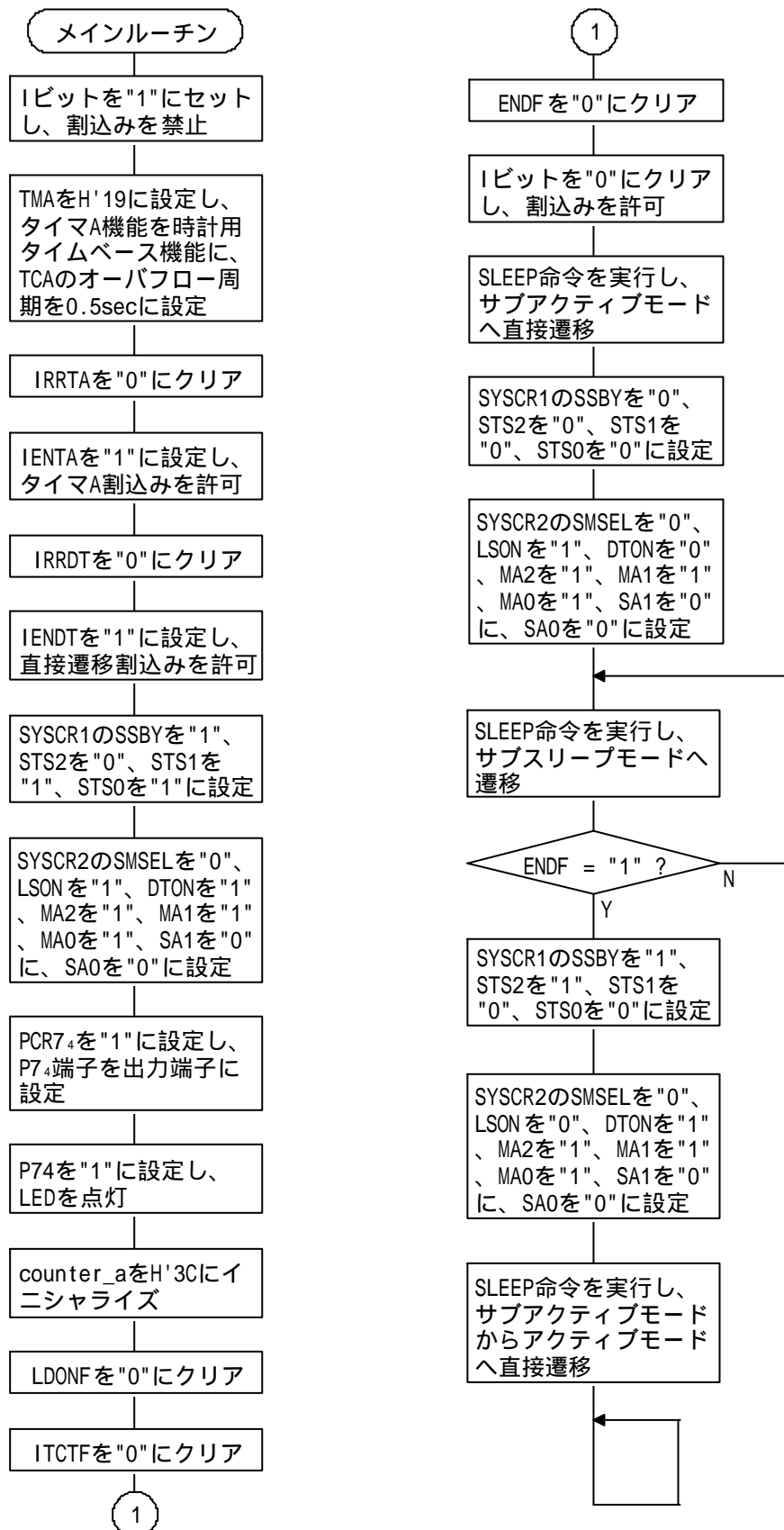
表4に本タスク例における使用RAM説明を示します。

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
USRFB	ENDF 60sec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

フローチャート

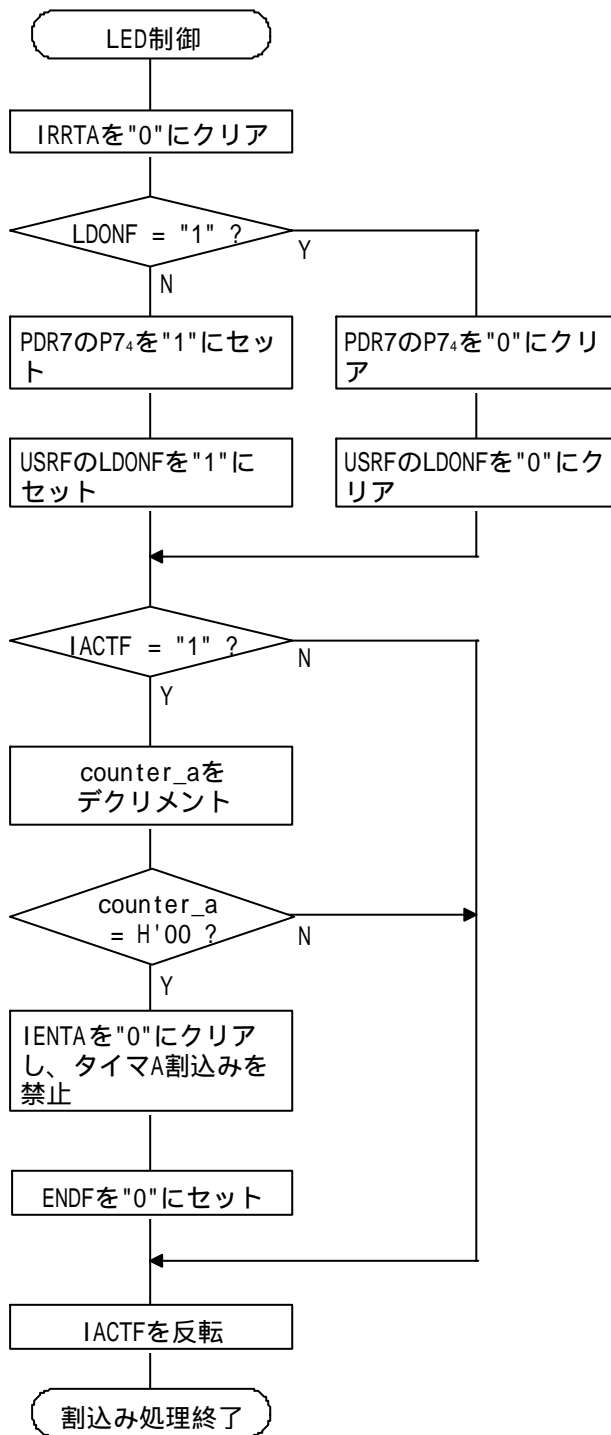
(a) メインルーチン



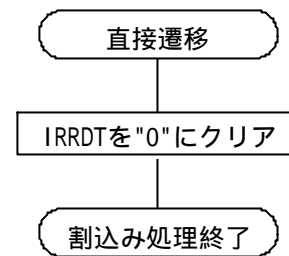
本例ではスタックポインタはINIT.SRC（アセンブリ言語）で設定してあります。

フローチャート

(b) タイマA割込み処理ルーチン



(c) 直接遷移割込み処理ルーチン



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    #'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 */

```


プログラムリスト

```

#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                         */

```

プログラムリスト

```

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 = 0x00;           /* 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){        /* LDONF = "1" ?              */
    P74 = 0;             /* Turn Off LED                */
    LDONF = 0;           /* Clear LDONF                 */
  }
  else{
    P74 = 1;             /* Turn On LED                 */
    LDONF = 1;           /* Set LDONF                   */
  }

  if(IACTF == 1){        /* IACTF = "1" ?              */
    counter_a--;         /* Decrement 8bit Timer A Interrupt Counter */
    if(counter_a == 0x00){
      IENTA = 0;         /* Timer A Interrupt Disable  */
      ENDF = 1;         /* Set ENDF                    */
    }
  }

  IACTF = ~IACTF;       /* Invert IACTF                */
}

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

```

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