

## 2.20 外部割込みによるカウントスタート

| 外部割込みによるカウントスタート | 使用機能   | 外部割込み |
|------------------|--|-------|
| 仕様               | <p>(1) <math>\overline{\text{IRQ}}_0</math>端子に接続したスイッチ入力のオンにより<math>\text{IRQ}_0</math>割込みを発生させ、16ビット汎用レジスタ(counter_sub)に設定した16ビットカウンタのカウンタアップを開始します。</p> <p>(2) <math>\text{IRQ}_0</math>割込みは、<math>\text{IRQ}_0</math>端子入力の立ち下がりエッジの検出により要求されます。</p> <p>(3) R1に設定した16ビットカウンタがオーバーフローするたびにLEDの点灯、または消灯を行ないます。</p> <p>(4) 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> <p>(5) 図1に<math>\text{IRQ}_0</math>入力端子のスイッチ接続例を示します。</p> <div data-bbox="470 459 1093 705" style="text-align: center;"> </div> <p style="text-align: center;">図1 <math>\overline{\text{IRQ}}_0</math>入力端子のスイッチ接続例</p> |       |

| 使用機能説明 | <p>(1) 本タスク例では、外部割込みを使用して<math>\text{IRQ}_0</math>外部割込みによるカウントスタートを行ないます。</p> <p>(a) 以下に外部割込みについて説明します。</p> <ul style="list-style-type: none"> <li>外部割込みには、NMI、<math>\text{IRQ}_3 \sim \text{IRQ}_0</math>、WKP割込みの6要因があります。</li> <li>NMI割込みは、NMI端子の入力信号により要求されます。NMI割込みは、立ち上がり/立ち下がりエッジセンスを割込みエッジセレクトレジスタ1 (IEGR1) のNMIEGにより指定できます。</li> <li>NMI割込み要求は最優先の割込み要求で、CCRのIビットの値に関わらず常に受け付けられます。</li> <li><math>\text{IRQ}_3 \sim \text{IRQ}_0</math>割込みは、<math>\text{IRQ}_3 \sim \text{IRQ}_0</math>端子の入力信号により要求されます。<math>\text{IRQ}_3 \sim \text{IRQ}_0</math>割込みは、立ち上がり/立ち下がりエッジセンスを割込みエッジセレクトレジスタ1 (IEGR1) のIEG3 ~ IEG0により指定できます。</li> <li>ポートモードレジスタ1 (PMR1) により、端子機能が<math>\text{IRQ}_3 \sim \text{IRQ}_0</math>端子に選択された状態で指定されたエッジが入力されると割込みフラグレジスタ1 (IRR1) のIRR13 ~ IRR10の対応するビットが"1"にセットされ、割込み要求を発生します。</li> <li>割込み要求の受け付けは、割込みイネーブルレジスタ1 (IENR1) のIEN3 ~ IEN0を"0"にクリアすることにより禁止できます。</li> <li>WKP割込みは、<math>\text{WKP}_5 \sim \text{WKP}_0</math>端子の入力信号により要求されます。WKP割込みは、立ち上がり/立ち下がりエッジセンスを割込みエッジセレクトレジスタ2 (IEGR2) のWPEG5 ~ WPEG0により指定できます。</li> <li><math>\text{WKP}_5 \sim \text{WKP}_0</math>端子は、ポート5との兼用端子です。本端子を<math>\text{WKP}_5 \sim \text{WKP}_0</math>入力端子として使用する場合はポートモードレジスタ5 (PMR5) の<math>\text{WKP}_5 \sim \text{WKP}_0</math>ビット端子を"1"に設定します。</li> <li>ポートモードレジスタ5 (PMR5) により、端子機能が<math>\text{WKP}_5 \sim \text{WKP}_0</math>端子に選択された状態で指定されたエッジが入力されると、ウェイクアップ割込みフラグレジスタ (IWPR) のIWPF5 ~ IWPF0の対応するビットが"1"にセットされ、割込み要求を発生します。</li> <li>割込み要求の受け付けは、割込みイネーブルレジスタ1 (IENR1) のIENWPを"0"にクリアすることにより禁止できます。</li> <li>コンディションコードレジスタ (CCR) のIビットを"1"にすることによりすべての割込みを禁止できます。</li> <li>以下に割込みの動作を示します。</li> </ul> <p>(1) 割込みイネーブルレジスタの対応するビットが"1"にセットされている状態で、割込み要因が発生したとき、割込みコントローラに対して割込み要求信号が送られます。</p> <p>(2) 割込みコントローラに割込み要求信号が送られると、割込み要求フラグがセットされます。</p> <p>(3) 割込み要求フラグが"1"にセットされている割込みの中で、優先順位に従って最高位の割込み要求が選択され、その他は保留となります。</p> <p>(4) CCRのIビットを参照し、Iビットが"0"にクリアされている場合は、割込み要求は受け付けられませんが、Iビットが"1"にセットされている場合は割込み要求は保留となります。</p> <p>(5) 割込みが受け付けられると、そのとき実行中の命令の処理が終了した後プログラムカウンタ (PC) とCCRがスタック領域に退避されます。スタックされるPCはリターン後に実行する最初のアドレスを示しています。</p> <p>(6) CCRのIビットが"1"にセットされます。これにより、すべての割込みはマスクされます。</p> <p>(7) 受け付けた割込みに対応するベクタアドレスを生成し、そのアドレスの内容によって示されるアドレスから、割込み処理ルーチンの実行を開始します。</p> <ul style="list-style-type: none"> <li>割込みイネーブルレジスタ1をクリアすることにより割込みをディスエーブルにする場合、または割込みフラグレジスタ1をクリアする場合は、必ず割込みをマスクした状態 (I="1") で行ないます。I="0"の状態で行なうと、当該操作命令の実行と当該割込みの発生が競合した場合には、当該操作命令の実行終了時に発生した割込みに対応する例外処理を実行します。</li> </ul> |  |
|--------|--|--|
|--------|--|--|

**使用機能説明**

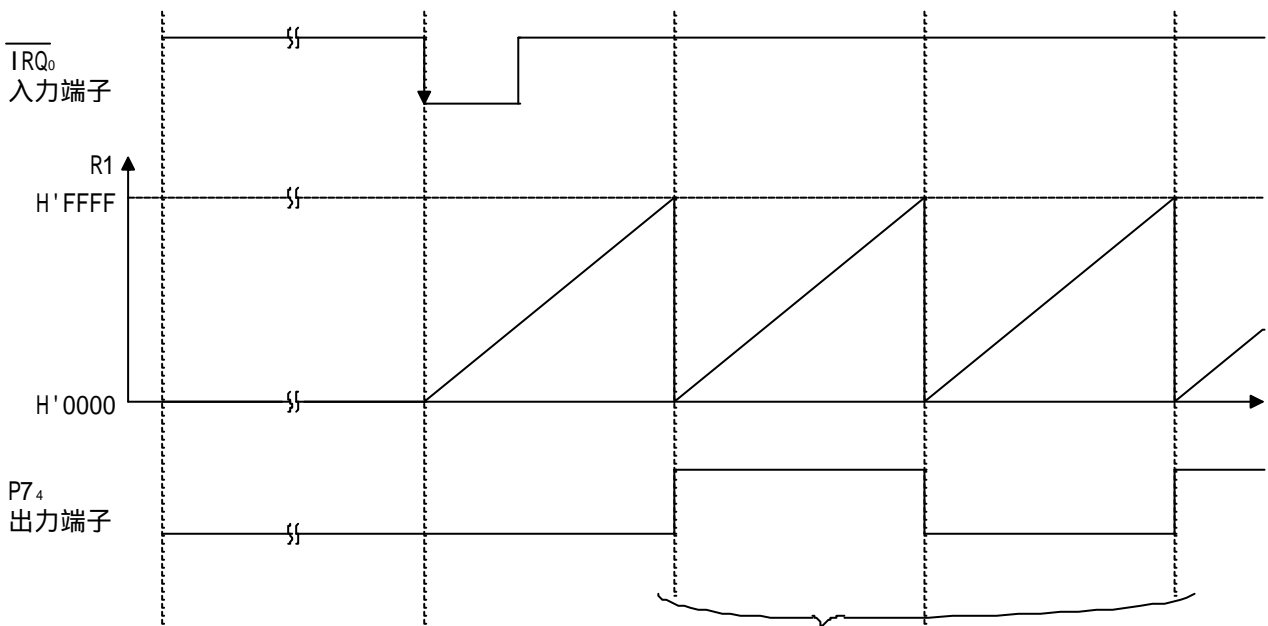
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、I/Oポートに接続したLEDの点滅動作を行ないます。

表1 機能割付け

| 機能               | 機能割付け                        |
|------------------|------------------------------|
| IRR10            | IRQ <sub>0</sub> 割込みの有無を反映   |
| IEN0             | IRQ <sub>0</sub> 端子の割込み要求を許可 |
| IEG0             | IRQ <sub>0</sub> 端子の入力エッジの選択 |
| TRQ <sub>0</sub> | スイッチ入力の入力端子                  |
| PCR7             | P7 <sub>4</sub> 出力端子機能の設定    |
| PDR7             | P7 <sub>4</sub> 出力端子のデータの格納  |
| P7 <sub>4</sub>  | LED出力の出力端子                   |

**動作説明**

(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外部割込みによるカウントスタート動作を行ないます。



|                                  |
|----------------------------------|
| ハードウェア処理                         |
| 処理なし                             |
| ソフトウェア処理                         |
| 初期設定                             |
| (a) IRQ <sub>0</sub> 入力端子の設定     |
| (b) counter_subに設定する16ビットカウンタの設定 |
| (c) P7 <sub>4</sub> 出力端子の設定      |
| (d) 割込み許可                        |

|  |
|--|
| ハードウェア処理                                     |
| (a) IRR10を"1"にセット                            |
| ソフトウェア処理                                     |
| (a) IRQ <sub>0</sub> 割込み処理の開始                |
| (b) IRR10を"0"にクリア                            |
| (c) counter_subに設定した16ビットカウンタのカウンタのカウントアップ開始 |
| (d) IEN0を"0"にクリア                             |

|                                       |
|---------------------------------------|
| ハードウェア処理                              |
| 処理なし                                  |
| ソフトウェア処理                              |
| (a) counter_subに設定した16ビットカウンタのオーバーフロー |
| (b) P7 <sub>4</sub> を"1"にセット          |

|                                       |
|---------------------------------------|
| ハードウェア処理                              |
| 処理なし                                  |
| ソフトウェア処理                              |
| (a) counter_subに設定した16ビットカウンタのオーバーフロー |
| (b) P7 <sub>4</sub> を"0"にクリア          |

|                  |      |       |
|------------------|------|-------|
| 外部割込みによるカウントスタート | 使用機能 | 外部割込み |
|------------------|------|-------|

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 | IRQ <sub>0</sub> 割込みの設定、LED出力端子の設定、割込みの許可、16ビットカウンタのインクリメント、およびLED出力を行なう |
| スイッチオン  | IRQ0 | IRQ <sub>0</sub> 割込み処理ルーチンで、SWONFを"1"にセット                                |

(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 |
|---------------------------|--|----------------|-----|
| PDR7<br>P7 <sub>4</sub>   | ポートデータレジスタ7 (ポートデータレジスタ7 <sub>4</sub> )<br>: P7 <sub>4</sub> =0のとき、P7 <sub>4</sub> 端子の出力レベルは"Low"<br>: P7 <sub>4</sub> =1のとき、P7 <sub>4</sub> 端子の出力レベルは"High" | H'FFDA<br>ビット4 | 0   |
| PCR7<br>PCR7 <sub>4</sub> | ポートコントロールレジスタ7<br>(ポートコントロールレジスタ7 <sub>4</sub> )<br>: PCR7 <sub>4</sub> ="1"のとき、P7 <sub>4</sub> 入出力端子は、P7 <sub>4</sub> 出力端子として機能                             | H'FFEA<br>ビット4 | 1   |
| IEGR1<br>IEG0             | 割込みエッジセレクトレジスタ1 (IRQ <sub>0</sub> エッジセレクト)<br>: IEG0="0"のとき、IRQ <sub>0</sub> 端子入力の立ち上がりエッジを検出  | H'FFF2<br>ビット0 | 0   |
| IENR1<br>IEN0             | 割込みイネーブルレジスタ1 (IRQ <sub>0</sub> 割込みイネーブル)<br>: IEN0="1"のとき、IRQ <sub>0</sub> 端子の割込み要求を許可  | H'FFF4<br>ビット0 | 1   |
| IRR1<br>IRR10             | 割込みフラグレジスタ1 (IRQ <sub>0</sub> 割込み要求フラグ)<br>: IRR10="0"のとき、IRQ <sub>0</sub> 割込みが要求されていない<br>: IRR10="1"のとき、IRQ <sub>0</sub> 割込みが要求されている                       | H'FFF6<br>ビット0 | 0   |

(4) 使用RAM説明

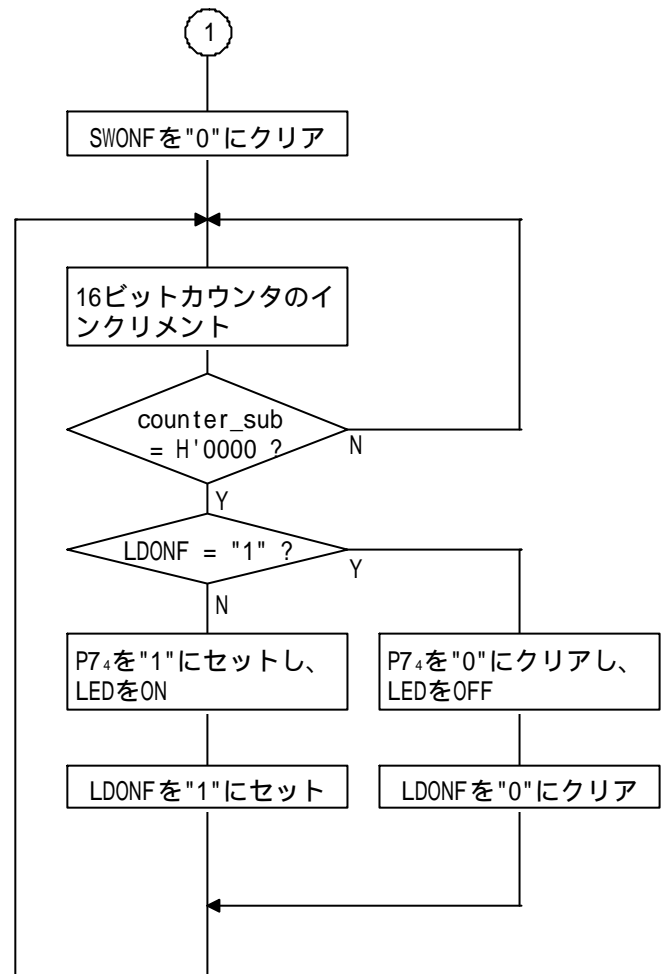
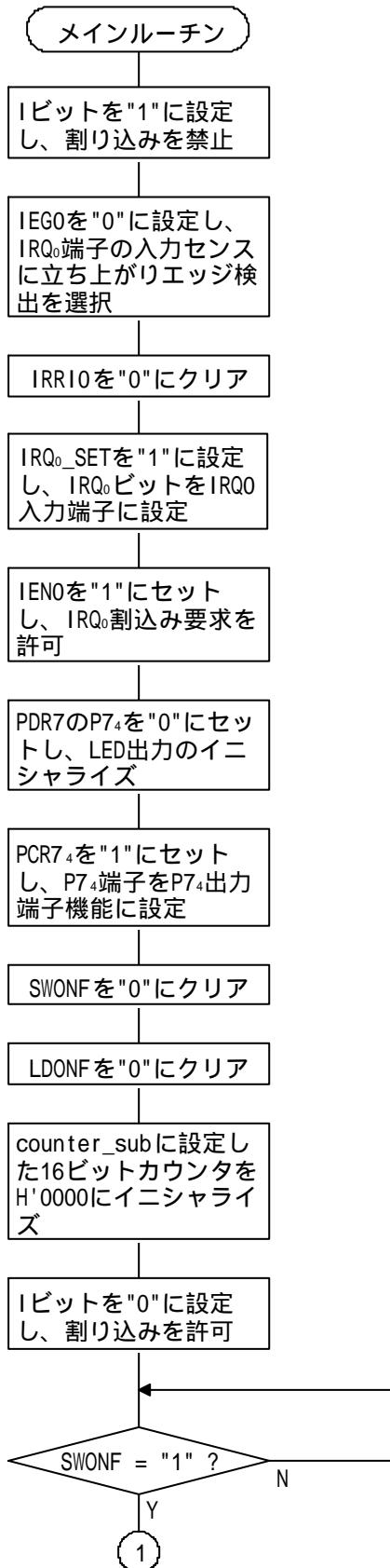
表4に本タスク例で使用するRAMの説明を示します。

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_sub | オーバフローする度にLED点灯・消灯を行う16ビットアップカウンタ | H'FB80         | メインルーチン           |
| USRF        | SWONF<br>スイッチ入力のON/OFFを判定するフラグ    | H'FB82<br>ビット0 | メインルーチン<br>スイッチオン |
|             | LDONF<br>LEDのON/OFFを判定するフラグ       | H'FB82<br>ビット1 | メインルーチン           |

## フローチャート

## (a) メインルーチン



## (b) IRQ0割り込み処理ルーチン



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。

## 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
/*
/*   'Count Start by External Interrupt'
/*
/*   Function
/*   : External Interrupt
/*
/*   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 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 bit4 */
#define IEGR1_BIT   (*(struct BIT *)0xFF2)  /* Interrupt Edge Select Register 1 */
#define I EGO       IEGR1_BIT.b0          /* IRQ0 Edge Select */
#define IENR1_BIT   (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1 */
#define IENO        IENR1_BIT.b0          /* IRQ0 Interrupt Enable */
#define IRR1_BIT    (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1 */
#define IRR10       IRR1_BIT.b0           /* IRQ0 Interrupt Request Flag */
#define PMR1_BIT    (*(struct BIT *)0xFFE0) /* Port Mode Register 1 */
#define IRQ0_SET    PMR1_BIT.b4           /* Port Mode Register 1 bit4 */

#pragma interrupt (IRQ0)
/*****
/*   関数定義
*****/
extern void INIT ( void ); /* SP Set */
void main ( void );
void IRQ0 ( void );
void wait ( void );

/*****
/*   RAM define
*****/

```

| 外部割込みによるカウントスタート   | 使用機能 | 外部割込み |
|--|------|-------|
| プログラムリスト   |      |       |
| <pre> unsigned int    counter_sub; unsigned char   USRF;  #define USRF_BIT (*(struct BIT *)&amp;USRF) #define SWONF    USRF_BIT.b0 /* Switch On Flag #define LDONF    USRF_BIT.b1 /* 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) = {     IRQ0 /* IRQ0 Interrupt }; #pragma section /* P /***** /*  Main Program /***** void main ( void ) {     set_imask_ccr(1); /* Interrupt Disable     IEGO = 1; /* Initialize IRQ0 Terminal Input Edge     IRRIO = 0; /* Initialize IRRIO     IRQ0_SET = 1; /* Initialize Input TerminalIRQ0     IENO = 1; /* IRQ0 Interrupt Request Enable     P74 = 0; /* Initialzie LED Output     PCR74 = 1; /* Initialize P74 Input-Output Terminal Function     SWONF = 0; /* Initialize SWONF     LDONF = 0; /* Initialize LDONF     counter_sub = 0x0000; /* Initialize 16bit Counter     set_imask_ccr(0); /* Interrupt Enable     while(SWONF != 1){         ;     }     SWONF = 0; /* Clear SWONF     while(1){         do{             counter_sub++; /* Increment 16bit Counter         }while(counter_sub != 0x0000); /* 16bit Counter = H'0000 ?         if(LDONF == 1){             P74 = 0; /* Turn Off LED             LDONF = 0; /* Clear LDONF         }         else{             P74 = 1; /* Turn On LED             LDONF = 1; /* Set LDONF         }     } } </pre> |      |       |

## プログラムリスト

```

/*****
/*   IRQ0 Interrupt                               */
/*****
void   IRQ0 ( void )
{

    IRR10 = 0;                /* Clear   IRR10           */

    SWONF = 1;               /* Set SWONF              */

    IENO = 0;                /* IRQ0 Interrupt Disable */

}

```

## リンクアドレス指定

| セクション名 | アドレス   |
|--------|--------|
| CV1    | H'0000 |
| CV2    | H'001C |
| P      | H'0100 |
| B      | H'FB80 |