

## 2.6 16ビットフリーランニング機能による割り込み回数のカウント

16ビットフリーランニング機能による 割り込み回数のカウント	使用機能	タイマW : 16ビットフリーランニング機能
仕様		
<p>(1) タイマW16ビットフリーランニングタイマ機能を使用して、タイマW割り込み回数をカウントし、50回の割り込みをカウントしたところで終了します。</p> <p>(2) タイマカウンタ (TCNT) がオーバーフローすることにより、タイマWのオーバーフローによる割り込み要求が発生し、タイマW割り込み処理の中でRAMに設定した8ビットカウンタをデクリメントします。</p> <p>(3) RAMに設定する8ビットカウンタはタイマW割り込み処理の中でデクリメントします。そして、50回のタイマW割り込みをカウントしたところで、タイマW割り込み要求を禁止して終了します。</p> <p>(4) タイマW割り込みは、タイマカウンタ (TCNT) がオーバーフローする時間32.768msごとに発生するように設定します。</p>		

使用機能説明
<p>(1) 本タスク例では、タイマWフリーランニング機能を使用して、タイマW割り込み回数のカウントを行いません。</p> <p>(a) 図1にタイマWフリーランニング機能のブロック図を示します。以下にタイマWフリーランニング機能のブロック図について説明します。</p> <ul style="list-style-type: none"> <li>・システムクロック ( ) は、16MHzのOSCクロックで、CPUおよび周辺機能を動作させるための基準クロックです。</li> <li>・プリスケアラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。</li> <li>・タイマカウンタ (TCNT) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックはシステムクロックの2分周、4分周、8分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、TCNTの入力クロックにシステムクロックの8分周のクロックを選択しています。</li> <li>・タイマコントロールレジスタW (TCRW) は、8ビットのリード/ライト可能なレジスタで、タイマカウンタ (TCNT) の入力クロックの選択を行いません。</li> <li>・タイマステータスレジスタW (TSRW) は、8ビットのレジスタで、割り込み要求信号の制御を行いません。</li> <li>・タイマインタラプトイネーブルレジスタW (TIERW) は、8ビットのリード/ライト可能なレジスタで、各割り込み要求の許可/禁止を制御します。本タスク例では、タイマオーバーフローによる割り込み要求を許可し、それ以外の割り込みは禁止しています。</li> <li>・タイマモードレジスタW (TMRW) はタイマカウンタ (TCNT) のカウンタスタート命令を行います。</li> <li>・以下に、本タスク例におけるタイマカウンタ (TCNT) のオーバーフロー周期の計算法を示します。</li> </ul>

$$\begin{aligned} \text{TCNT オーバーフロー周期} &= \frac{1}{\text{システムクロック} / 8} \times 65536 \\ &= 32.768\text{ms} \end{aligned}$$

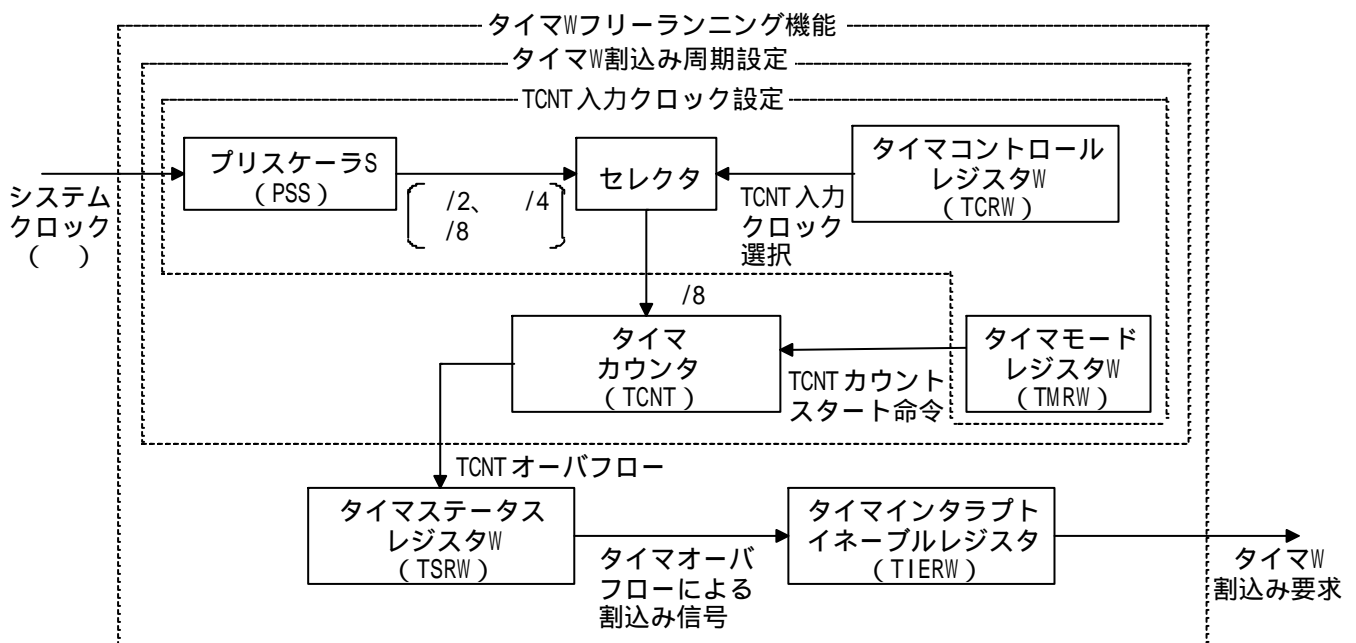


図1 フリーランニング機能ブロック図

16ビットフリーランニング機能による 割り込み回数のカウント	使用機能	タイマW : 16ビットフリーランニング機能
-----------------------------------	------	------------------------

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、タイマWフリーランニング機能による割り込み回数のカウントを行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCNT	システムクロックの8分周のクロックを入力とする16ビットのカウンタ
TCRW	TCNT 入力クロックの設定
TMRW	TCNT カウントスタート設定
TSRW	タイマオーバーフローによる割り込み要求信号の制御
TIERW	タイマオーバーフローによる割り込み要求の許可
counter_sub	タイマW割り込み回数を50回カウントする8ビットカウンタ

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タイマWフリーランニング機能による割り込み回数のカウントを行ないます。

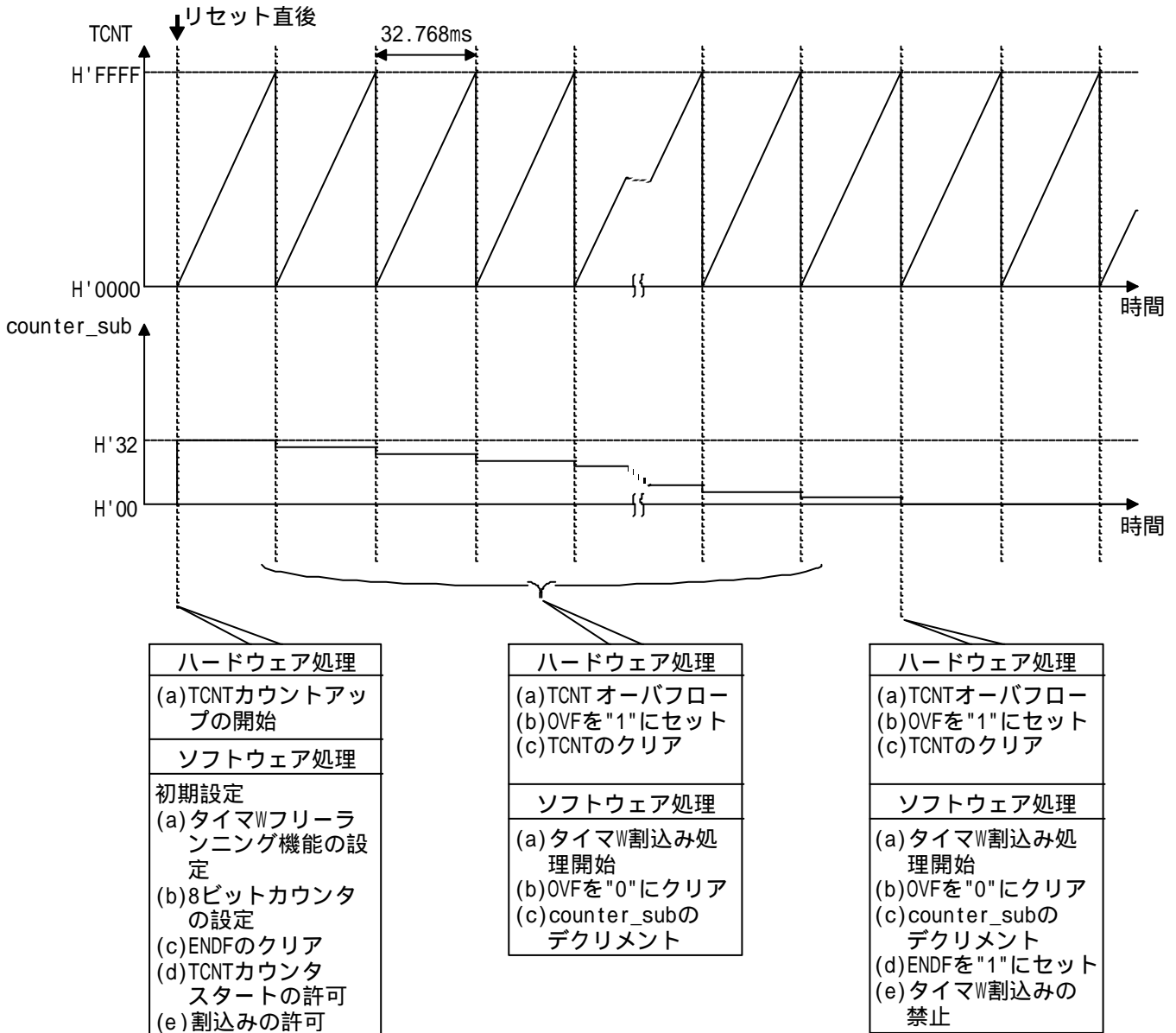


図2 タイマW16ビットフリーランニング機能による割り込み回数のカウント動作原理

16ビットフリーランニング機能による 割り込み回数のカウント	使用機能	タイマW : 16ビットフリーランニング機能
-----------------------------------	------	------------------------

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	16ビットフリーランニング機能の設定、8ビットカウンタの設定、割り込みの許可、TCNTカウンタスタート設定およびENDFが"1"にセットされた時点で終了する
割り込みカウント	twint	タイマW割り込み処理で、8ビットカウンタのデクリメント、カウンタ値がH'00の時点でENDFを"1"にセットし、タイマW割り込みを禁止する

(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
TMRW CTS	タイマモードレジスタW (タイマカウンタスタート) : CTS="1"のとき、TCNTがカウンタ開始を示す : CTS="0"のとき、TCNTがカウンタ停止を示す	H'FF80 ビット7	1
TCRW CKS2 CKS1 CKS0	タイマコントロールレジスタW (クロックセレクト1、0) : CKS2="0"、CKS1="1"、CKS0="1"のとき、TCNT入力クロックをシステムクロックの8分周のクロックに設定	H'FF81 ビット6 ビット5 ビット4	CKS2="0" CKS1="1" CKS0="1"
TIERW OVIE	タイマインタラプトイネーブルレジスタW (タイマオーバーフロー割り込みイネーブル) : OVIE="0"のとき、OVFによる割り込み要求を禁止 : OVIE="1"のとき、OVFによる割り込み要求を許可	H'FF82 ビット7	1
TSRW OVF	タイマステータスレジスタW (タイマオーバーフロー) : OVF="0"のとき、TCNTがオーバーフローしていないことを示す : OVF="1"のとき、TCNTがオーバーフローしたことを示す	H'FF83 ビット7	0
TCNT	タイマカウンタ : システムクロックの8分周のクロックを入力とする16ビットのアップカウンタ	H'FF86	H'00

(4) 使用RAM説明

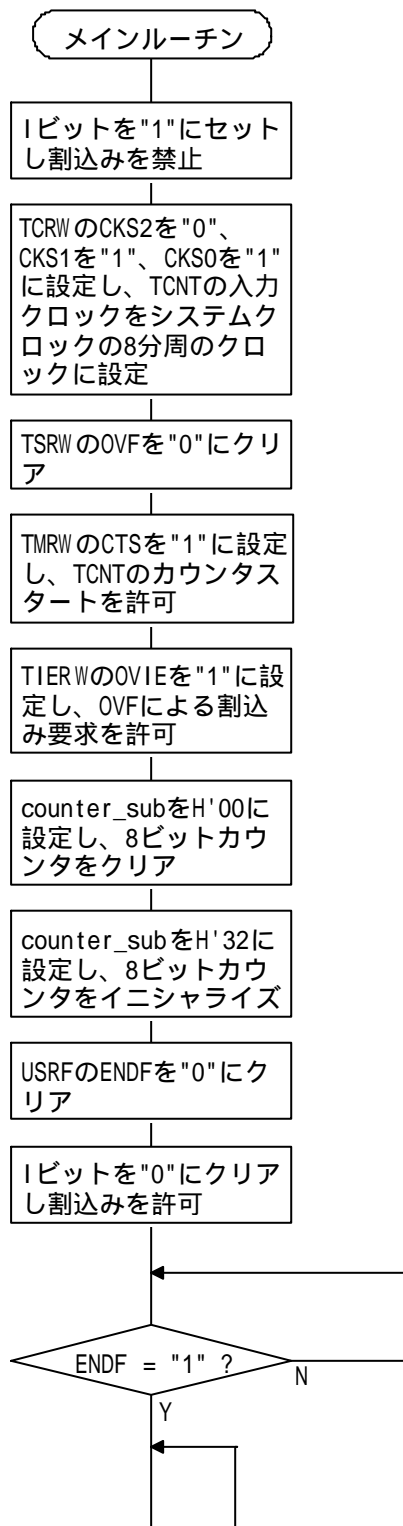
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

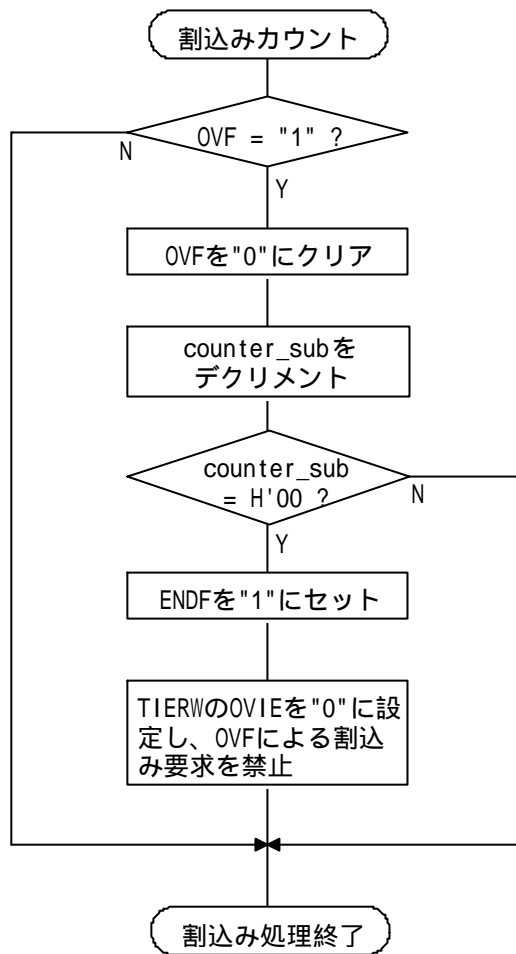
ラベル名	機能	アドレス	使用モジュール名
counter_sub	タイマW割り込みを50回カウントする8ビットカウンタ	H'FB80	メインルーチン 割り込みカウント
USRF ENDF	8ビットカウンタのカウンタ値がH'00になったかを判定するフラグ	H'FB81 ビット0	メインルーチン 割り込みカウント

フローチャート

(a) メインルーチン



(b) タイマW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
/*
/* 'Interrupt Counting by 16-bit Free Running
/* Function'
/*
/*
/* Function
/* : Timer W 16bit Free Running Timer
/*
/*
/* 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 TMRW *(volatile unsigned char *)0xFF80 /* Timer Mode Register W */
#define TCRW *(volatile unsigned char *)0xFF81 /* Timer Control Register W */
#define TCRW_BIT (*(struct BIT *)0xFF81) /* Timer Control Register W */
#define CKS1 TCRW_BIT.b5 /* Clock Select 1 */
#define CKS0 TCRW_BIT.b4 /* Clock Select 0 */
#define TIERW *(volatile unsigned char *)0xFF82 /* Timer Interrupt Enable Register */
#define TIERW_BIT (*(struct BIT *)0xFF82) /* Timer Interrupt Enable Register */
#define OVIE TIERW_BIT.b7 /* Timer Overflow Interrupt Enable */
#define TSRW *(volatile unsigned char *)0xFF83 /* Timer Status Register W */
#define TSRW_BIT (*(struct BIT *)0xFF83) /* Timer Status Register W */
#define OVF TSRW_BIT.b7 /* Timer Over flow */
#define TCNT *(volatile unsigned int *)0xFF86 /* Time Counter */

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

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

## プログラムリスト

```
unsigned char    counter_sub;
unsigned char    USRF;                                /* User Flag Area */

#define    USRF_BIT    (*(struct BIT *)&USRF)
#define    ENDF    USRF_BIT.b0                        /* End Flag */

/*****
/*    Vector Address
*****/
#pragma section    V1                                /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
    INIT                                /* 00 Reset */
};
#pragma section    V2                                /* VECTOR SECTOIN SET */
void (*const VEC_TBL2[])(void) = {
    twint                                /* 2A Timer W Interrupt */
};
#pragma section    P                                /* P */
/*****
/*    Main Program
*****/
void main ( void )
{
    set_imask_ccr(1);                            /* コンディションコードの設定 (割り込みマスクビットセット) */

    TCRW = 0x30;                                /* Initialize FRC Input Clock Period */

    TSRW = 0x00;                                /* Clear OVF */

    TMRW = 0x80;                                /* Timer Counter Count Start */

    TIERW = 0x80;                                /* OVF Interrupt Enable */

    counter_sub = 0x00;                            /* Clear 8bit Counter_sub */
    counter_sub = 0x32;                            /* Initialize 8bit Counter_sub */

    ENDF = 0;                                    /* Initialize ENDF */

    set_imask_ccr(0);                            /* コンディションコードの設定 (割り込みマスクビット解除) */

    while(ENDF != 1){
        ;
    }

    while(1) {
        ;
    }
}
```

プログラムリスト

```

/*****
/*   Timer W Interrupt   */
*****/
void twint ( void )
{
    if ( OVF == 1 )    {
        OVF = 0;           /* Clear OVF           */
        counter_sub--;    /* Decrement 8bit Counter */

        if ( counter_sub == 0x00 ){
            ENDF = 1;     /* 8bit Counter != H'00 */
            OVFIE = 0;    /* Set ENDF             */
            OVFIE = 0;    /* OVF 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'002A
P	H'0100
B	H'FB80