

## 2.9 インพุットキャプチャ機能によるパルスの周期測定

インพุットキャプチャ機能によるパルスの周期測定	使用機能	タイマW：インพุットキャプチャ機能
--------------------------	------	--------------------

### 仕様

- (1) タイマWインพุットキャプチャタイマ機能を使用して、インพุットキャプチャA端子 (FT10A) に入力されたパルスの周期を測定します。
- (2) 入力パルスの立ち上がりエッジ間のタイマカウンタ (TCNT) のカウンタ値をRAMに格納し、その値より入力パルスの周期を測定します。
- (3) 測定可能なパルスの最大の周期は32.768msで、精度は0.5μsです。

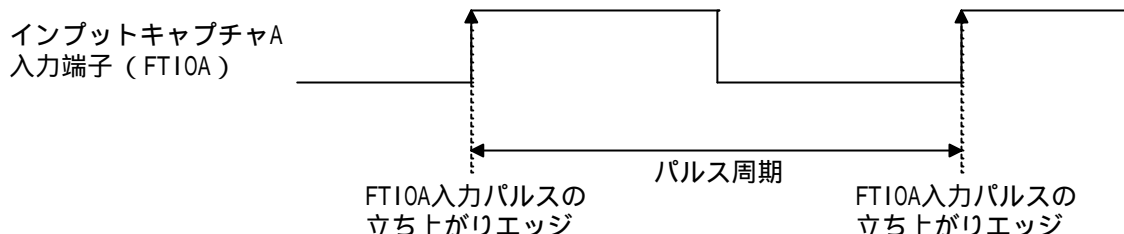


図1 入力パルスの周期測定

### 使用機能説明

- (1) 本タスク例では、タイマWインพุットキャプチャ機能を使用して、FT10A端子に入力したパルスの周期の測定を行いません。
- (a) 図2にタイマWインพุットキャプチャ機能のブロック図を示します。以下にタイマWインพุットキャプチャ機能のブロック図について説明します。
  - ・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。
  - ・プリスケラ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。
  - ・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、8分周、32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。
  - ・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、TCNTの入力クロックの選択、およびインพุットキャプチャ入力エッジの選択を行いません。
  - ・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、各割り込み要求信号の制御を行いません。
  - ・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。
  - ・タイマモードレジスタW (TMRW) は TCNTのカウンタスタート命令を行います。
  - ・タイマI/Oコントロールレジスタ0 (TIOR0) は、8ビットのリード/ライト可能なレジスタで、インพุットキャプチャレジスタの設定、FT10A入力エッジを立ち上がりエッジ設定を行います。
  - ・ジェネラルレジスタA (GRA) は、16ビットのリード専用のレジスタで、インพุットキャプチャ端子Aの入力パルスの立ち上がりエッジが検出されると、そのときのTCNTの値がGRAに転送され、TSRWのIMFAが"1"にセットされます。このとき、TIERWのIMFAが"1"ならばCPUに割り込みを要求します。
  - ・インพุットキャプチャA端子 (FT10A) より、周波数測定対象パルスを入力します。
  - ・以下に本タスク例における入力パルスの周期の計算方法を示します。ただし、TCNTがオーバーフローすると、周期測定が正確に行なえないので、入力パルスの周期はTCNTのオーバーフロー周期 (32.768ms) より小さくしなければなりません。
  - ・1番目の入力パルスの立ち上がりエッジをキャプチャした後、TCNTがオーバーフローすると、PDRHLにH'FFFFを格納します。

入力パルスの周期 = (PDRHLに格納されたTCNTのカウンタ値) × (TCNT入力クロック周期)

= (PDRHLに格納されたTCNTのカウンタ値) × 0.5μs (プリスケラS:8 × システムクロック:62.5ns)

使用機能説明

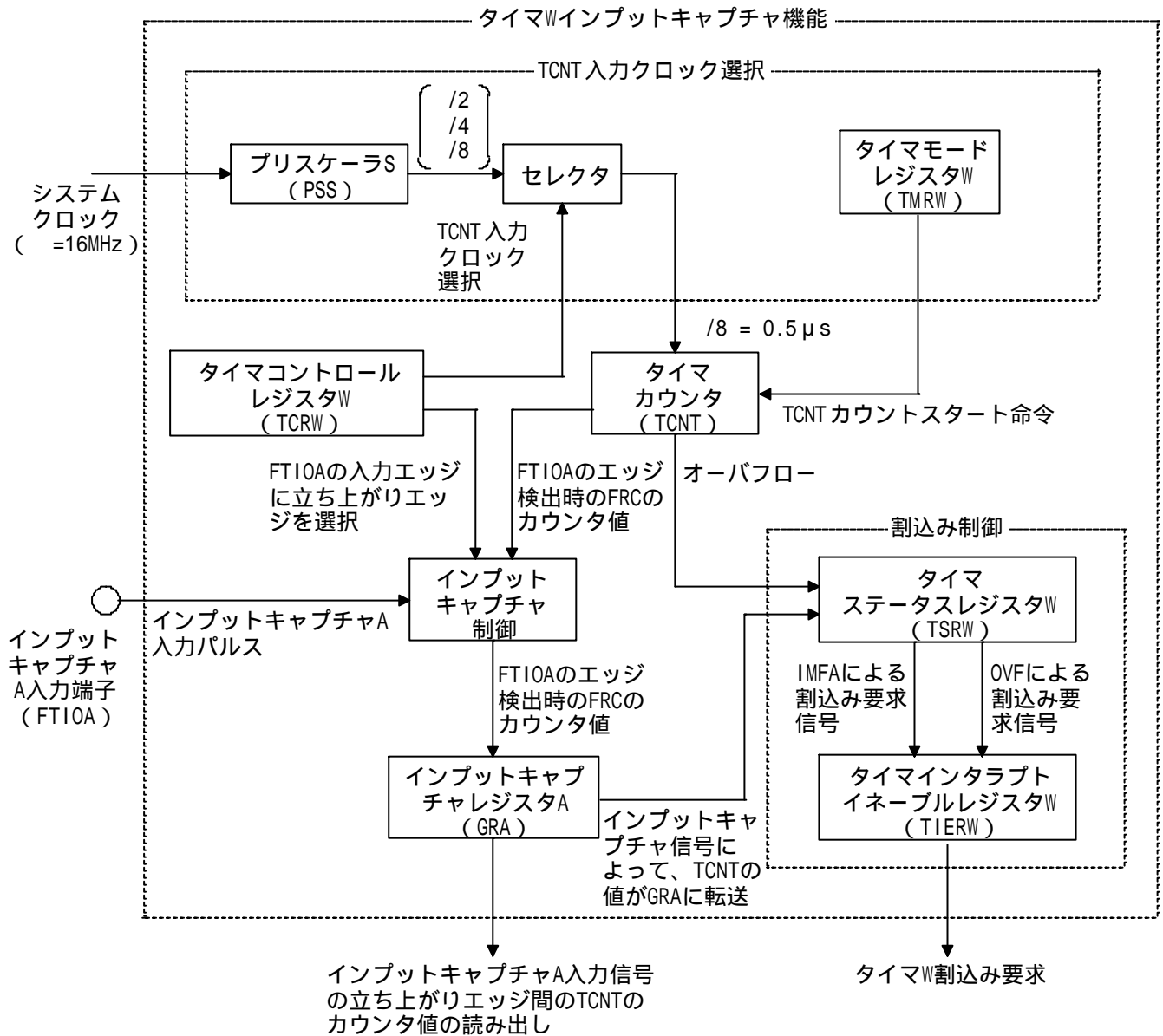


図2 タイマWインプットキャプチャ機能ブロック図

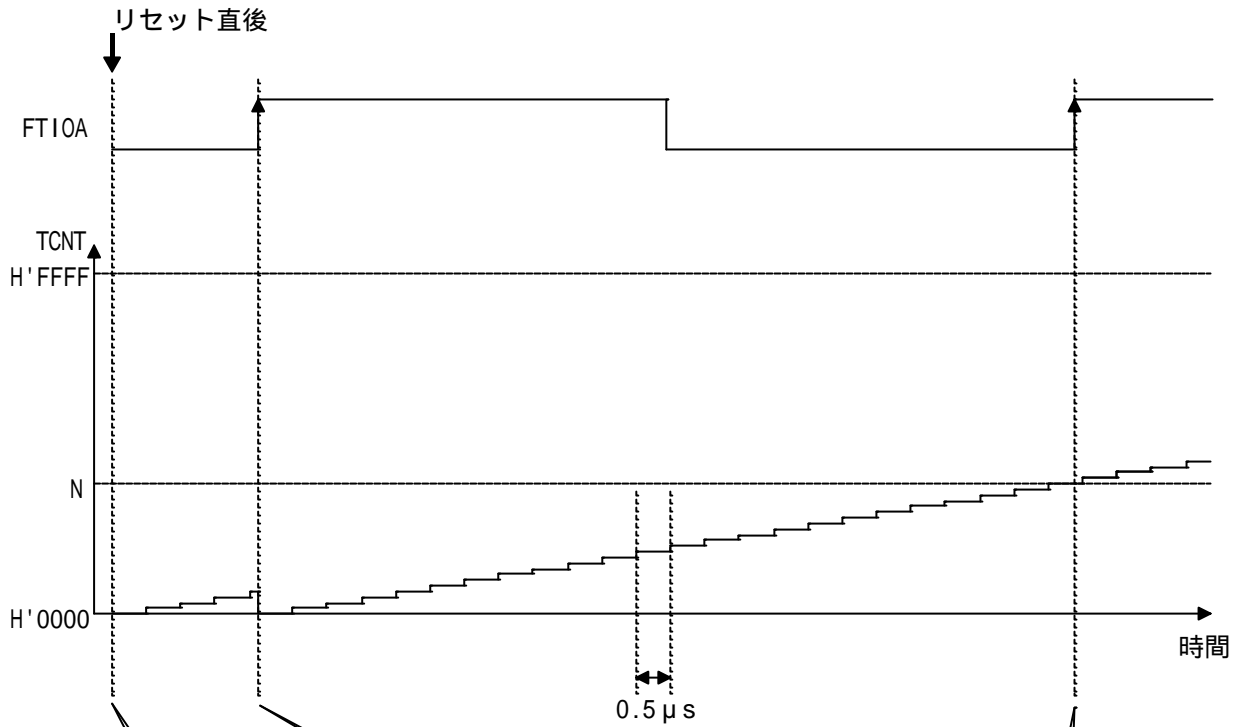
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマWインプットキャプチャ機能による周波数測定を行ないます。

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
TIERW	インプットキャプチャ信号によるTCNTのカウンタ値の転送、およびTCNTのオーバーフローによる割込みの許可
TSRW	インプットキャプチャ信号によるTCNTのカウンタ値の転送、およびTCNTのオーバーフローによる割込みの制御
TCNT	システムクロックを8分周したクロックを入力とする16ビットのアップカウンタ
GRA	FT10A入力パルスの立ち上がりエッジの検出により、その時のTCNTのカウンタ値を格納
TCRW	TCNT入力クロックの設定
TMRW	TCNTカウントスタート設定
TIOR0	アウトプットコンペアレジスタの設定、アウトプットコンペア出力の設定、およびFT10A入力エッジを立ち上がりエッジ設定

動作原理

(1) 図3に動作原理を示します。図3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インプットキャプチャ機能によるパルスの周期測定を行ないます。



【注】N : FT10A入力パルスの立ち上がりエッジ間のTCNTのカウンタ値

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)インプットキャプチャ機能の設定	
(b)TCNTカウンタスタートの許可	
(c)割込み許可	

ハードウェア処理	
(a)TCNTのカウンタ値をGRAに転送	
(b)IMFAを"1"にセット	
ソフトウェア処理	
(a)インプットキャプチャ入力信号の立ち上がりエッジの検出による割込み処理の開始	
(b)IMFAを"0"にクリア	
(c)TCNTをクリア	
(d)SRTFを"1"にセット	

ハードウェア処理	
(a)TCNTのカウンタ値をGRAに転送	
(b)IMFAを"1"にセット	
ソフトウェア処理	
(a)インプットキャプチャ入力信号の立ち上がりエッジの検出による割込み処理の開始	
(b)IMFAを"0"にクリア	
(c)GRAの値をRAMに転送	
(d)ENDFを"1"にセット	
(d)タイマW割込み要求の禁止	

図4 タイマWインプットキャプチャ機能によるパルスの周期測定の動作原理

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	インプットキャプチャ機能の設定、割込みの許可を行なう
周期測定終了	twint	タイマW割込み処理ルーチンで、1回目のIMFAによる割込みであればTCNTをH'0000にイニシャライズし、2回目のIMFAによる割込みであればGRAの内容をRAMに格納し、タイマW割込み要求の禁止を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
PRDHL	FTIOA入力パルスの立ち上がりエッジ間のTCNTのカウント値	メインルーチン	2バイト	出力

(3) 使用内部レジスタ説明

表4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4 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	CCLR タイマコントロールレジスタW(カウンタクリア) : CCLR="0"のとき、フリーランニングカウンタとして動作します。	H'FF81 ビット7	0
	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の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込みイネーブル) : OVIE="0"のとき、OVFによる割込み要求を禁止 : OVIE="1"のとき、OVFによる割込み要求を許可	H'FF82 ビット7	1
	IMIEA タイマインタラプトイネーブルレジスタW (インプットキャプチャ割込みAイネーブル) : IMIEA="1"のとき、IMFAによる割込みを許可	H'FF82 ビット0	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
	IMFA タイマステータスレジスタW (インプットキャプチャフラグA) : IMFA="1"のとき、インプットキャプチャ信号によって、TCNTの値がGRAに転送されたことを示す : IMFA="0"のとき、インプットキャプチャ信号が入力されていないことを示す	H'FF83 ビット0	0

ソフトウェア説明

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
T1OR0	IOA2 タイマI/Oコントロールレジスタ0 (I/OコントロールA2) : IOA2="1"のとき、インプットキャプチャレジスタとして機能	H'FF84 ビット2	1
	IOA1 IOA0 タイマI/Oコントロールレジスタ0 (I/OコントロールA1~0) : IOA1="0"、IOA0="0"のとき、FT10A端子の立ち上がりエッジで GRAへインプットキャプチャに設定	H'FF84 ビット1 ビット0	IOA1="0" IOA0="0"
TCNT	タイマカウンタ : システムクロックの2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'0000
GRA	ジェネラルレジスタA : インプットキャプチャ入力Aの入力信号の立ち上がりエッジ 検出により、TCNTのカウンタ値の16ビットを格納	H'FF88	H'0000

(4) 使用RAM説明

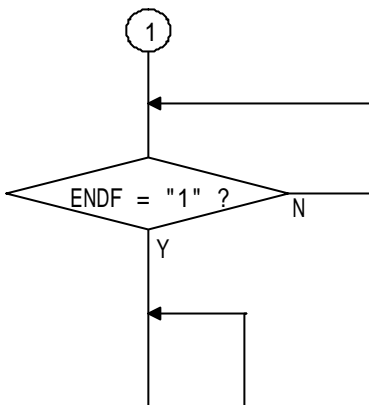
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
Dummy	Dummy Read	H'FB80	メインルーチン
PRDHL	FT1A入力端子の立ち上がりエッジ間のTCNTのカウンタ 値の16ビットを格納	H'FB82	周期測定終了
USRF	STRF 2回目のタイマW割込みか否かを判定するフラグ	H'FB84 ビット1	周期測定終了
	ENDF 周期測定が終了したか否かを判定するフラグ	H'FB84 ビット0	メインルーチン 周期測定終了

フローチャート

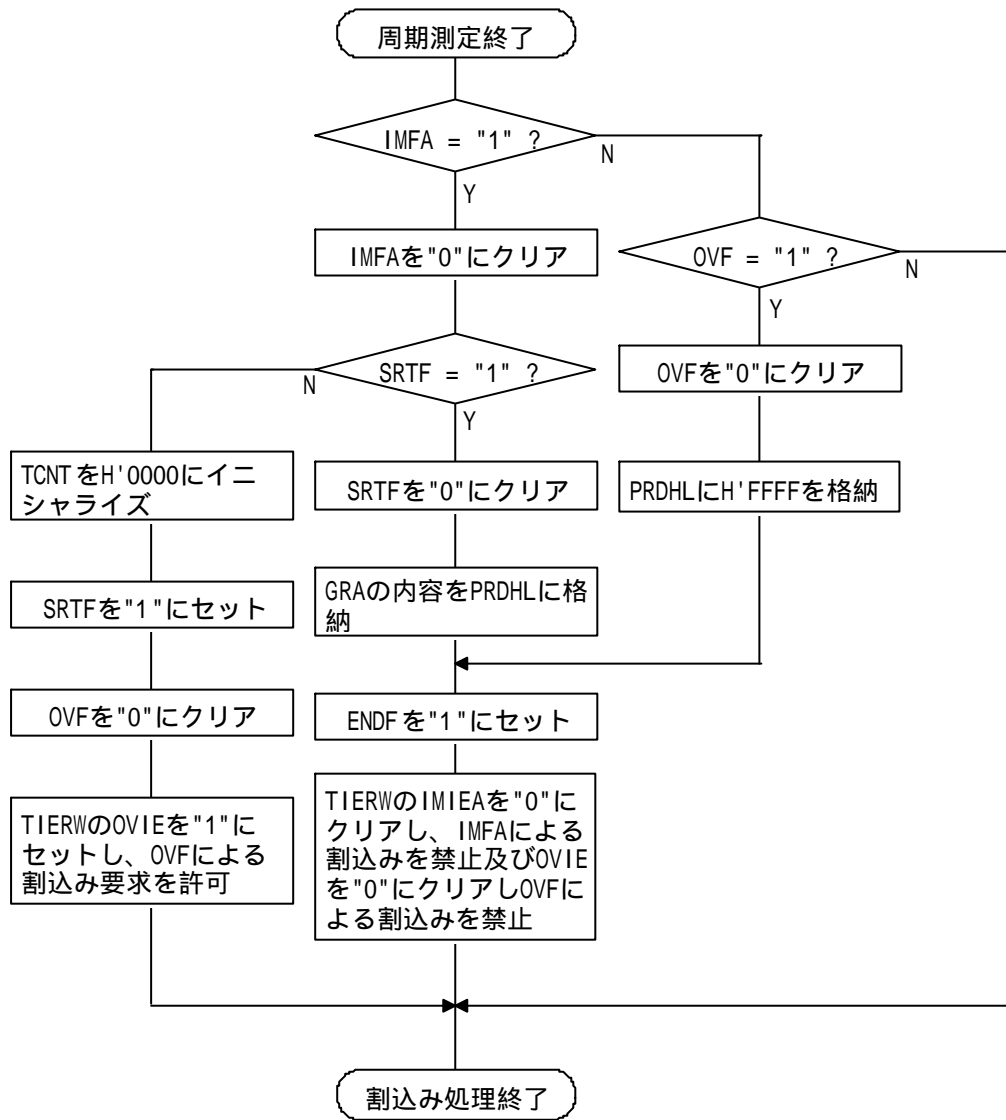
(a) メインルーチン



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W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 #H'FF80,R7
LDC.B #B'10000000,CCR
JMP @_main
;
.END

/*****
/*
/* H8/300H Tiny Series -H8/3664-
/* Application Note
/*
/* 'Pulse Period Measurement by Input
/* Caputure Function'
/*
/* Function
/* : Timer W Input Capture
/*
/* 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 CCLR TCRW_BIT.b7 /* Counter Clear A */
#define CKS1 TCRW_BIT.b5 /* Clock Select 1 */
#define CKS0 TCRW_BIT.b4 /* Clock Select 0 */
#define TOB TCRW_BIT.b1 /* Timer Output Level B */
#define TOA TCRW_BIT.b0 /* Timer Output Level A */
#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 IMIEA TIERW_BIT.b0 /* Input Capture Interrupt A 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 IMFA TSRW_BIT.b0 /* Input Capture Flag A */
#define TIORO *(volatile unsigned char *)0xFF84 /* Timer I/O Control Register 0 */
#define TIORO_BIT (*(struct BIT *)0xFF84) /* Timer I/O Control Register 0 */
#define IOA2 TIORO_BIT.b2 /* Input Edge Select A2 */
#define IOA1 TIORO_BIT.b1 /* Input Edge Select A1 */
#define IOA0 TIORO_BIT.b0 /* Input Edge Select A0 */
#define TCNT *(volatile unsigned int *)0xFF86 /* Timer Counter 16bit */
#define GRA *(volatile unsigned int *)0xFF88 /* Input Capture Register A */
#define TMA *(volatile unsigned char *)0xFFA6 /* Timer Mode Register A */
#define TCA *(volatile unsigned char *)0xFFA7 /* Timer Counter A */

```



プログラムリスト

```

#define PDR8      *(volatile unsigned char *)0xFFDB /* Port Data Register 8 */
#define PDR8_BIT (*(struct BIT *)0xFFDB) /* Port Data Register 8 */
#define P81      PDR8_BIT.b1 /* Port Data Register 8 bit1 */
#define PCR8      *(volatile unsigned char *)0xFFEB /* Port Control Register 8 */
#define PCR8_BIT (*(struct BIT *)0xFFEB) /* Port Control Register 8 */
#define PCR81    PCR8_BIT.b1 /* Port Control Register 8 bit1 */
#define IENR1_BIT (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1 */
#define IENTA    IENR1_BIT.b6 /* Timer A Interrupt Enable */
#define IRR1_BIT (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1 */
#define IRRTA    IRR1_BIT.b6 /* Timer A Interrupt Request Flag */

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

/*****
/* RAM define */
*****/
unsigned int Dummy;
unsigned int PRDHL; /* Period Higher&Lower */
unsigned char USRF; /* User Flag Area */

#define USRF_BIT (*(struct BIT *)&USRF)
#define SRTF    USRF_BIT.b1 /* Start Flag */
#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 */
/*****
/* Main Program */
*****/
void main ( void )
{
set_imask_ccr(1); /* Interrupt Disable */

P81 = 0; /* Clear Port Data Register 8 bit1 */
PCR81 = 0; /* Initialize Port Control Register 8 bit1 */

TIOR0 = 0x8C; /* Initialize FTIA Input Edge */
TCRW = 0x30; /* Initialize Input Clock Period */
Dummy = TSRW;
TSRW = 0x70; /* Initialize Input Capture Flag A */
TIERW = 0xF1; /* IMIEA&OVIE Interrupt Enable */

SRTF = 0; /* Clear SRTF */

ENDF = 0; /* Clear ENDF */

TCNT = 0; /* Clear TCNT */

GRA = 0; /* Clear GRA */

PRDHL = 0; /* Clear PRDHL */

```

プログラムリスト

```

TMRW = 0xC8; /* Initialize timer Mode Register */

set_imask_ccr(0); /* Interrupt Enable */

while(ENDF != 1){ /* ENDF = 1 ? */
    ;
}

while(1) {
    ;
}

}

/*****
/* Timer W Interrupt */
*****/
void twint ( void )
{
    if ( IMFA == 1 ){ /* IMFA = "1" ? */
        IMFA = 0; /* Clear IMFA */
        if ( SRTF == 1 ){ /* SRTF = "1" ? */
            SRTF = 0; /* Clear STRF */
            PRDHL = GRA; /* Store ICRAL */
            ENDF = 1; /* Set ENDF */
            IMIEA = 0; /* IMIEA Interrupt Disable */
            OVIE = 0; /* OVIE Interrupt Disable */
        }
        else{
            TCNT = 0; /* Initialize TCNT */
            SRTF = 1; /* Set SRTF */
            OVF = 0; /* Clear OVF */
            OVIE = 1; /* OVIE Interrupt Enable */
        }
    }
    else{
        if ( OVF == 1 ){ /* OVF = 1 ? */
            OVF = 0; /* Clear OVF */
            PRDHL = 0xFFFF; /* TCNT(FRC) Overflow Process */
            ENDF = 1; /* Set ENDF */
            IMIEA = 0; /* IMIEA Interrupt Disable */
            OVIE = 0; /* OVIE 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