

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。

H8/300L Super Low Power シリーズ

イベントカウンタ機能によるパルスの周波数測定

要旨

タイマCの8ビットイベントカウンタ機能を使用して、タイマCイベント入力端子 (TMIC) から入力したパルスの周波数を測定します。

動作確認デバイス

H8/38024

目次

1. 仕様	2
2. 使用機能説明	2
3. 動作説明	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. タイマ C の 8 ビットイベントカウンタ機能を使用して、タイマ C イベント入力端子 (TMIC) から入力したパルスの周波数を測定します。
2. TMIC 入力端子から入力したパルスの立ち上がりエッジ検出の回数を 1s 間カウントアップし、1s 間のカウント数を RAM に格納します。
3. 1s の測定時間は、タイマ A 時計用タイムベース機能を使用して計測します。
4. タイマ C のカウンタ (TCC) は、UD 端子を GND に接続し、ハードウェア制御によるアップカウンタに設定します。

2. 使用機能説明

1. 本タスク例では、タイマ C 8 ビットイベントカウンタ機能を使用して、TMIC 入力端子に入力されたパルスの周波数測定を行います。
 - a. 図 1 にタイマ C 8 ビットイベントカウンタ機能のブロック図を示します。以下にタイマ C 8 ビットイベントカウンタ機能のブロック図について説明します。
 - タイマモードレジスタ C (TMC) は、8 ビットのリード/ライト可能なレジスタで、インターバル機能の選択、タイマカウンタ C (TCC) のアップ/ダウン制御、および入力クロックの選択を行います。TCC のアップ/ダウン制御は、UD 端子入力によるハードウェア制御とするか、ソフトウェア制御によるアップカウンタまたはダウンカウンタとするかを選択します。
 - タイマカウンタ C (TCC) は、8 ビットのリード可能なカウンタで、入力する内部クロック/外部イベントにより、カウントアップ/ダウンされます。入力するクロックはシステムクロックの 8192, 2048, 512, 64, 16, 4 分周, サブクロックの 4 分周, および外部クロックの計 8 種類のクロックから選択可能です。本タスク例では、TCC をハードウェア制御によるアップカウンタに設定し、TCC の入力クロックに TMIC 入力端子のエッジ検出を選択しています。
 - タイマ C 割り込み要求フラグ (IRRTC) は、TCC がオーバフローすることにより"1"にセットされます。IRRTC が"1"にセットされていて、割り込み許可レジスタ 2 (IENR2) のタイマ C 割り込みイネーブル (IENTC) が"1"で、かつコンディションコードレジスタ (CCR) の I ビットが"0"にクリアされている場合にタイマ C 割り込みが受け付けられ、タイマ C 割り込み処理を開始します。
 - タイマ C イベント入力端子 (TMIC) は、周波数測定を行うパルスの入力端子として機能します。

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、タイマ C 8 ビットイベントカウンタ機能による周波数測定を行います。

表 1 機能割り付け

機能	機能割り付け
PSW	32.768kHz を 4 分周したクロックを入力とする 5 ビットのカウンタ
TMA	PSW の選択, および TCA オーバフロー周期の設定
TCA	32.768kHz を 128 分周したクロックを入力とする 8 ビットのカウンタ
TMC	インターバル機能の設定, TCC をハードウェア制御によるアップカウンタに設定, および TCC 入力クロックを TMIC 入力端子のエッジ検出に設定
TCC	TMIC 入力端子のエッジ検出を入力とする 8 ビットのカウンタ
TLC	TCC オーバフロー時のリロード値を H'00 に設定
IRQ1	PB3/AN3/IRQ1 端子を TMIC 入力端子に設定
IEG1	TMIC 端子の入力センスを立ち上がりエッジ検出に設定
IENTA	タイマ A 割り込み要求を許可
IENTC	タイマ C 割り込み要求を許可
IRRTA	タイマ A 割り込み要求の有無を反映
IRRTC	タイマ C 割り込み要求の有無を反映
TMIC	周波数測定を行うパルスの入力端子
UD 端子	TCC をアップカウンタに設定する入力端子

3. 動作説明

図2に動作説明を示します。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C 8ビットイベントカウンタ機能によるパルスの周波数測定を行います。

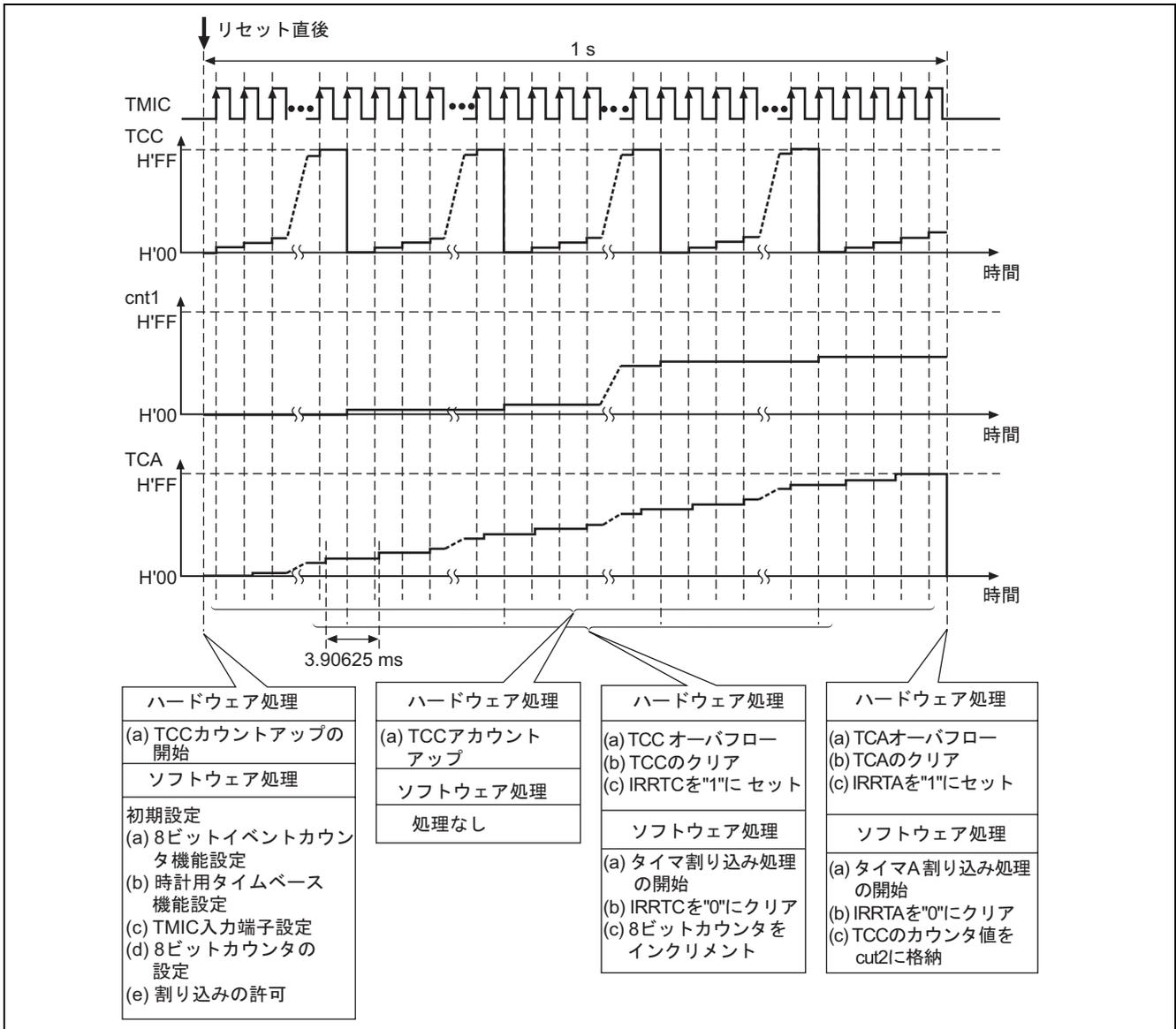


図2 タイマC 8ビットイベントカウンタ機能による周波数測定の動作説明

4. ソフトウェア説明

4.1 モジュール説明

本タスク例のモジュールを表 2 に示します。

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 C 8 ビットイベントカウンタ機能の設定, 時計用タイムベース機能の設定, 8 ビットカウンタの設定, 割り込みの許可, 測定終了後にタイマ C の初期化を行う
1s 経過	taint	タイマ A 割り込み処理で, 1s の経過により割り込みを禁止し, TCC のカウンタ値を cnt2 に格納する
8 ビットカウンタ	tcint	タイマ C 割り込み処理で, 8 ビットカウンタのインクリメント, cnt1 がオーバーフローしたときの処理を行う

4.2 引数の説明

本タスク例の引数を表 3 に示します。

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
cnt1	1s 経過時点の 8 ビットカウンタのカウンタ値を格納	8 ビットカウンタ	1 バイト	出力
cnt2	1s 経過時点の TCC のカウンタ値を格納	1s 経過	1 バイト	出力

4.3 内部レジスタの説明

本タスク例の使用内部レジスタを表 4 に示します。

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
TMA		タイマモードレジスタ A : TMA = H'18 のとき, タイマ A 機能を時計用タイムベース機能に, TCA 入力クロックソースを PSW に, TCA オーバフロー周期を 1s に設定	H'FFB0	H'18
TCA		タイマカウンタ A : 32.768kHz の 128 分周のクロックを入力とする 8 ビットのアップカウンタ	H'FFB1	H'00
TMC		タイマモードレジスタ C : TMC = H'7F のとき, タイマ C 機能をインターバル機能に, TCC をハードウェア制御によるアップカウンタに, TCC 入力クロックを TMIC 端子の入力エッジ検出に設定	H'FFB4	H'7F
TCC		タイマカウンタ C : TMIC 端子の入力エッジ検出を入力とする 8 ビットのアップカウンタ	H'FFB5	H'00
TLC		タイマロードレジスタ C : TLC = H'00 に設定すると, TCC は H'00 からカウントアップを開始し, TCC がオーバフローすると TCC に H'00 がロードされる	H'FFB5	H'00
PMR3	UD	ポートモードレジスタ 3 (P30/UD 端子切り替え) : UD = "0" のとき, P30 入出力端子として機能 : UD = "1" のとき, UD 入力端子として機能	H'FFCA ビット 0	1
PMRB	IRQ1	ポートモードレジスタ B (PB3/AN3/IRQ1 端子切り替え) : IRQ1 = "0" のとき, PB3/AN3 入力端子として機能 : IRQ1 = "1" のとき, $\overline{\text{IRQ1}}$ /TMIC 入力端子として機能	H'FFEE ビット 3	1
IEGR	IEG1	IRQ エッジセレクトレジスタ (IRQ1 エッジセレクト) : IEG1 = 0 のとき, TMIC 入力端子を立ち下がりエッジ検出に設定 : IEG1 = 1 のとき, TMIC 入力端子を立ち上がりエッジ検出に設定	H'FFF2 ビット 1	1
IENR1	IEN TA	割り込み許可レジスタ 1 (タイマ A 割り込みイネーブル) : IEN TA = 0 のとき, タイマ A 割り込み要求を禁止 : IEN TA = 1 のとき, タイマ A 割り込み要求を許可	H'FFF3 ビット 7	1
IENR2	IEN TC	割り込み許可レジスタ 2 (タイマ C 割り込みイネーブル) : IEN TC = 0 のとき, タイマ C 割り込み要求を禁止 : IEN TC = 1 のとき, タイマ C 割り込み要求を許可	H'FFF4 ビット 1	1
IRR1	IRR TA	割り込み要求レジスタ 1 (タイマ A 割り込み要求フラグ) : IRR TA = 0 のとき, タイマ A 割り込みが要求されていない : IRR TA = 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 7	0
IRR2	IRR TC	割り込み要求レジスタ 2 (タイマ C 割り込み要求フラグ) : IRR TC = 0 のとき, タイマ C 割り込みが要求されていない : IRR TC = 1 のとき, タイマ C 割り込みが要求されている	H'FFF7 ビット 1	0

4.4 使用 RAM 説明

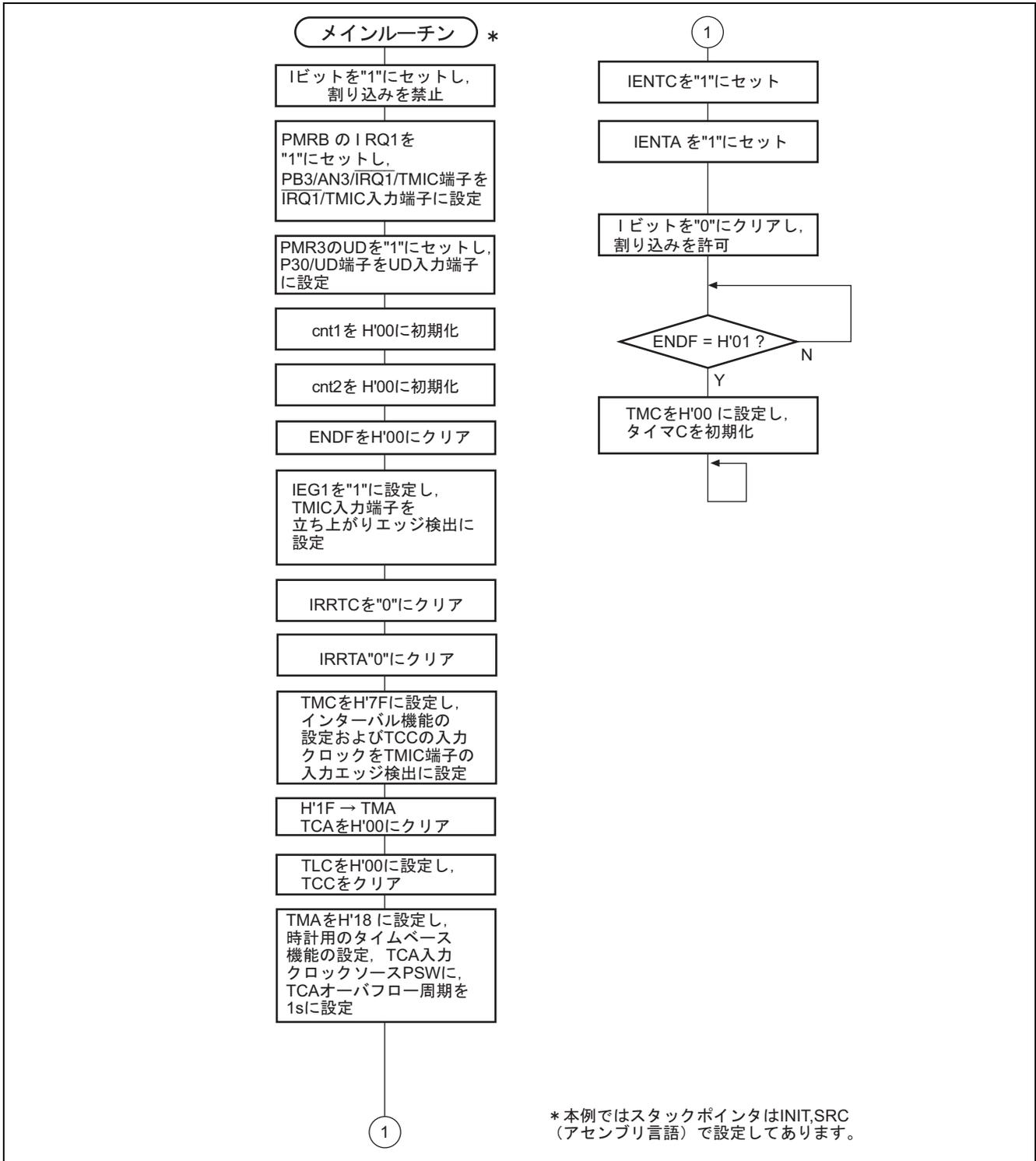
本タスク例の使用 RAM を表 5 に示します。

表 5 使用 RAM 説明

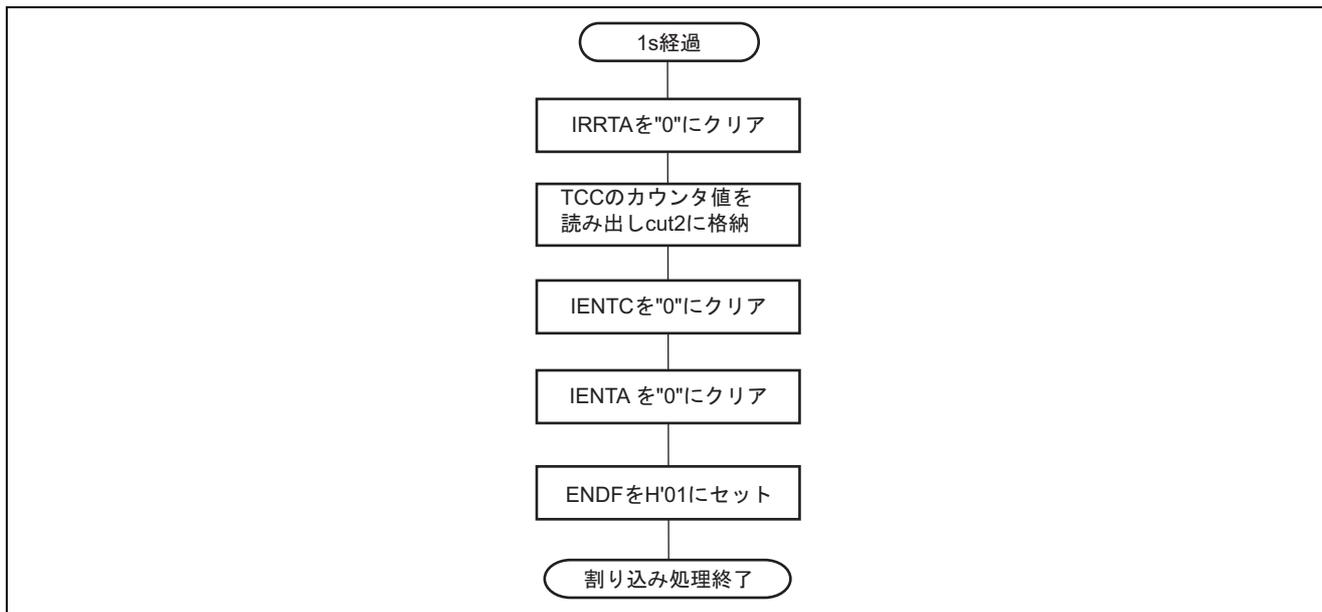
ラベル名	機能	アドレス	使用モジュール名
cnt1	1s 経過時点の 8 ビットカウンタのカウンタ値を格納	H'FB80	8 ビットカウンタ
cnt2	1s 経過時点の TCC のカウンタ値を格納	H'FB81	1s 経過
ENDF	入力パルスの周波数測定が終了したか否かを判定するデータ測定時間が 1s 未滿かつ, cnt1 < H'FF のとき, ENDF に H'00 を格納, 処理時間が 1s 経過した, または cnt1 == H'FF のとき, ENDF = H'01 を格納	H'FB82	メインルーチン 8 ビットカウンタ 1s 経過

5. フローチャート

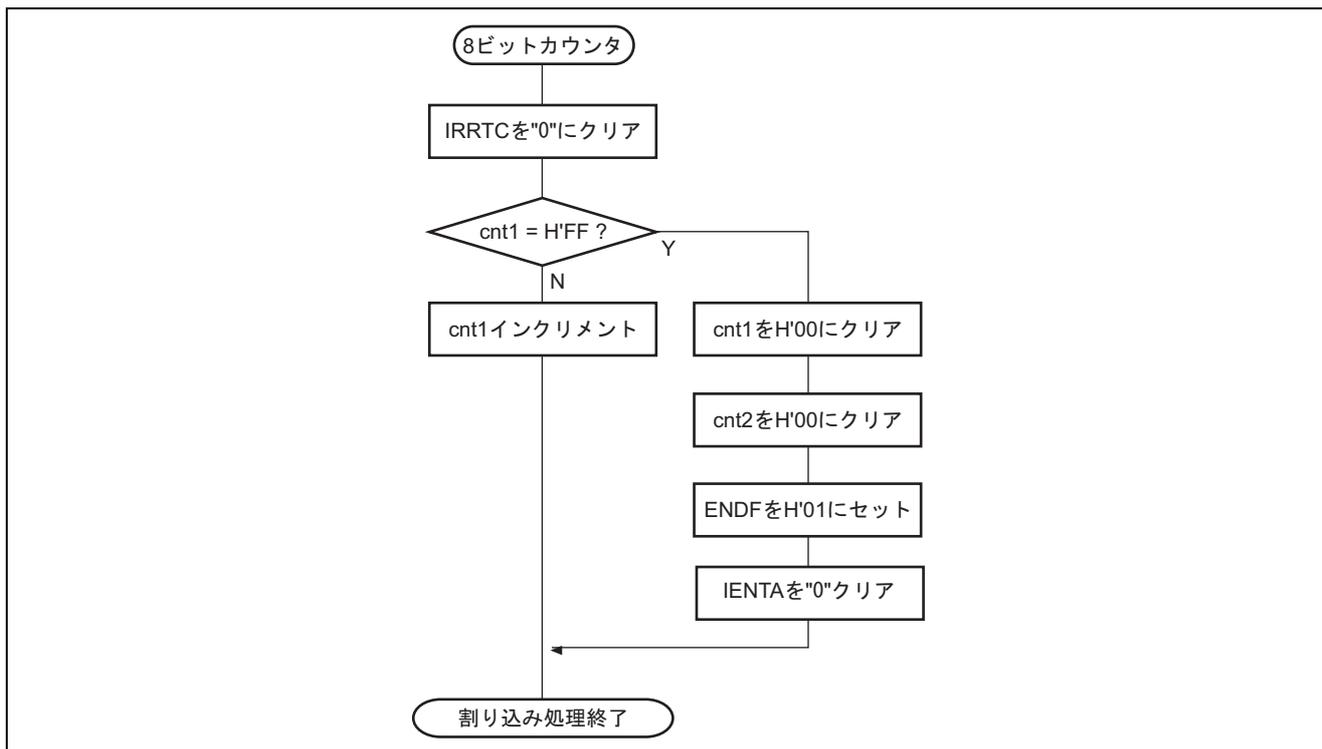
1. メインルーチン



2. タイマ A 割り込み処理ルーチン



3. タイマ C 割り込み処理ルーチン



6. プログラムリスト

6.1 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

```

```

/*****
/*
/* H8/300L Super Low Power Series
/*   -H8/38024 Series-
/* Application Note
/*
/* 'Pulse Frequency Measurement by Event
/* Counter Function'
/*
/* Function
/* :Timer C 8bit Event Counter
/*
/* External Clock :   10MHz
/* Internal Clock  :   5MHz
/* 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 *)0xFFB0    /* Timer Mode Register A */
#define TCA        *(volatile unsigned char *)0xFFB1    /* Timer Counter A */
#define TMC        *(volatile unsigned char *)0xFFB4    /* Timer Mode Register C */
#define TCC        *(volatile unsigned char *)0xFFB5    /* Timer Counter C */
#define TLC        *(volatile unsigned char *)0xFFB5    /* Timer Load Register C */
#define PMR3_BIT   (*(struct BIT *)0xFFCA)              /* Port Mode Register 3 */
#define UD         PMR3_BIT.b0                          /* Port Mode Register 3 bit0 */
#define PMRB_BIT   (*(struct BIT *)0xFFEE)              /* Port Mode Register B */

```

```

#define      IRQ1      PMRB_BIT.b3          /* Port Mode Register B bit3      */
#define      IEGR_BIT  (*(struct BIT *)0xFFF2) /* Interrupt Edge Select Register 1 */
#define      IEGR1     IEGR_BIT.b1         /* IRQ1 Edge Select                */
#define      IENR1_BIT (*(struct BIT *)0xFFF3) /* Interrupt Enable Register 1     */
#define      IENTA     IENR1_BIT.b7        /* Timer A Interrupt Enable        */
#define      IENR2_BIT (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 2     */
#define      IENTC     IENR2_BIT.b1        /* Timer C Interrupt Enable        */
#define      IRR1_BIT  (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1    */
#define      IRRTA     IRR1_BIT.b7         /* Timer A Interrupt Request Flag   */
#define      IRR2_BIT  (*(struct BIT *)0xFFF7) /* Interrupt Request Register 2    */
#define      IRRTC     IRR2_BIT.b1         /* Timer C Interrupt Request Flag   */

#pragma interrupt      (taint)
#pragma interrupt      (tcint)
/*****
/* Function define
/*****
extern void INIT ( void );          /* SP Set
void      main ( void );
void      taint ( void );
void      tcint ( void );

/*****
/* RAM define
/*****
unsigned char  cnt1;                /* 8bit Counter
unsigned char  cnt2;                /* 8bit Counter
unsigned char  ENDF;                /* End Data

/*****
/* 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) = {
    taint                /* 16 Timer A Interrupt
};
#pragma section      V3              /* VECTOR SECTOIN SET
void (*const VEC_TBL3[])(void) = {
    tcint                /* 1A Timer C Interrupt
};

#pragma section                      /* P

```

```

/*****
/* Main Program
/*****
void main ( void )
{
    set_imask_ccr(1);          /* Interrupt Disable          */

    IRQ1 = 1;                 /* TMIC Input Select          */
    UD   = 1;                 /* UD Input Select            */

    cnt1 = 0;                 /* Initialize 8bit Counter    */
    cnt2 = 0;                 /* Initialize 8bit Counter    */
    ENDF = 0;                 /* Clear END Flag             */

    IEG1 = 1;                 /* TMIC Select                */
    IRRTC = 0;                 /* Clear IRRTC                 */
    IRRTA = 0;                 /* Clear IRRTC                 */

    TMC = 0x7F;               /* Initialize 8bit Event Counter Function */
    TMA = 0x1F;               /* Initialize Timer Counter A   */
    TLC = 0x00;               /* Clear TCC                    */
    TMA = 0x18;               /* Initialize Clock Time Base Function */

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

    set_imask_ccr(0);        /* Interrupt Enable            */

    while (ENDF==0);

    TMC = 0;                  /* Initialize Timer C Function  */

    while (1) {
        ;
    }
}

/*****
/* Timer A Interrupt
/*****
void taint ( void )
{
    IRRTA = 0;                /* Clear IRRTA                 */

    cnt2 = TCC;               /* Store TCC                    */

    IENTC = 0;                /* Timer C Interrupt Disable    */

    IENTA = 0;                /* Timer A Interrupt Disable    */

    ENDF = 1;                 /* Set ENDF                     */
}

```

```

/*****
/* Timer C Interrupt
/*****
void tcint ( void )
{
    IRRTC = 0;                /* Clear IRRTC
                               */

    if ( cnt1 == 0xff ) {    /* 8bit Counter = 0xff?
                               */
        cnt1 = 0;           /* Clear cnt1
                               */
        cnt2 = 0;           /* Clear cnt2
                               */
        ENDF = 1;          /* ENDF
                               */
        IENTA = 0;         /* Timer A Interrupt Disable
                               */
    }
    else {
        cnt1++;             /* cnt1 Increment
                               */
    }
}

```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0016
CV3	H'001A
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
1.00	2003.12.19	—	初版発行
2.00	2006.09.15	5, 7, 9, 10	内容変更

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。