

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。

M16C/64 グループ

ウォッチドッグタイマ

1. 要約

ウォッチドッグタイマは、プログラムの暴走を検知するための機能です。

ウォッチドッグタイマは 15 ビットのカウンタを持ち、カウントソース保護モードの有効、無効を選択できます。

表 1 にウォッチドッグタイマの仕様を示します。

表 1. ウォッチドッグタイマの仕様

項目	カウントソース保護モード無効時	カウントソース保護モード有効時
カウントソース	CPU クロック	125kHz オンチップオシレータクロック
カウント動作	ダウンカウント	
カウント開始条件	<ul style="list-style-type: none"> ・ OFS1 番地の WDTON ビット="0" のとき リセット後、自動的にウォッチドッグタイマのカウントを開始 ・ OFS1 番地の WDTON ビット="1" のとき WDTN レジスタへの書き込みによりウォッチドッグタイマのカウントを開始 	
カウント停止条件	ストップモード、ウェイトモード、ホールド状態	なし
アンダフロー時の動作	<ul style="list-style-type: none"> ・ PM1 レジスタの PM12 ビット="0" のとき ウォッチドッグタイマ割り込み ・ PM1 レジスタの PM12 ビット="1" のとき ウォッチドッグタイマリセット 	ウォッチドッグタイマリセット

ウォッチドッグタイマ周期は、CPU クロック、プリスケアラの分周比、カウントソース保護モードが有効か無効かで異なります。各設定によるウォッチドッグタイマの周期を表 2 に示します。

表 2. ウォッチドッグタイマ周期

CSPR レジスタの CSPRO ビット(注 1,注 2)	CM0 レジスタの CM07 ビット	WDC レジスタの WDC7 ビット	ウォッチドッグタイマ周期
"0"(カウントソース保護 モード無効)	"0"(CPU クロック は、メインクロック、 PLL クロック、または、 125kHz オンチップオシレータ)	"0"(16 分周)	$\frac{16(\text{プリスケアラの分周比}) \times 32768(\text{WDT のカウント値})}{\text{CPU クロック}}$
		"1"(128 分周)	$\frac{128(\text{プリスケアラの分周比}) \times 32768(\text{WDT のカウント値})}{\text{CPU クロック}}$
	"1"(CPU クロック は、サブクロック)	—	$\frac{2(\text{プリスケアラの分周比}) \times 32768(\text{WDT のカウント値})}{\text{CPU クロック}}$
"1"(カウントソース保護 モード有効)	—	—	$\frac{4096(\text{WDT のカウント値})}{125\text{kHz オンチップオシレータ}}$

— : "0"でも"1"でもよい。

WDT : ウォッチドッグタイマ

(注 1) : OFS1 番地の CSPROINI ビットが"0" (リセット後、カウントソース保護モード有効) のとき、リセット後の CSPRO ビットは"1" (カウントソース保護モード有効) になります。

(注 2) : CSPRO ビットを"1"にするためには、プログラムで"0"を書いた後、続けて"1"を書いてください。プログラムでは CSPRO ビットを"0" (カウントソース保護モード無効) にできません。

2. はじめに

この資料で説明する応用例は、次のマイコンでの利用に適用されます。

- ・マイコン：M16C/64 グループ

M16C/64グループと同様のSFR(周辺機能制御レジスタ)を持つ他のM16Cファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等に変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 応用例

ここでは、ウォッチドッグタイマのカウントソースに CPU クロックを使用して、ウォッチドッグタイマ割り込みまたはウォッチドッグタイマリセットを発生させる方法について示します。

3.1 応用例の説明

- (1) WDTR レジスタに“00h”を書いて、続いて“FFh”を書くと、ウォッチドッグタイマは“7FFFh”に初期化(注1)されます。
- (2) WDTS レジスタに書き込みを行うと、ウォッチドッグタイマはダウンカウントを開始します。
- (3) カウント中に再度、WDTR レジスタへ書き込みを行うと、ウォッチドッグタイマは“7FFFh”に初期化(注1)され、カウントを継続して行います。
- (4) ウェイトモードやストップモード、ホールド状態のとき、ウォッチドッグタイマはカウント中の値を保持して停止します。ウェイトモードやストップモードから復帰したあと、保持した値からカウントを再開します。(注2)
- (5) ウォッチドッグタイマ割り込みを使用する場合：ウォッチドッグタイマがアンダフローすると、ウォッチドッグタイマ割り込みが発生します。この時、ウォッチドッグタイマは初期化されないので、割り込み処理内で、(1)の WDTR レジスタへの書き込みを行ってください。カウント動作は継続して行います。
ウォッチドッグタイマリセットを使用する場合：ウォッチドッグタイマがアンダフローすると、端子、CPU、SFR を初期化して、リセットベクタで示されるアドレスからプログラムを実行します。

注1. CSPR レジスタの CSPRO ビットが“1”(カウントソース保護モード有効)の場合、“0FFFh”に初期化されます。

注2. CSPR レジスタの CSPRO ビットが“1”(カウントソース保護モード有効)の場合、ウェイトモード、ストップモードおよび、ホールド状態の時でもウォッチドッグタイマは停止しません。

図1に動作タイミングを示します。

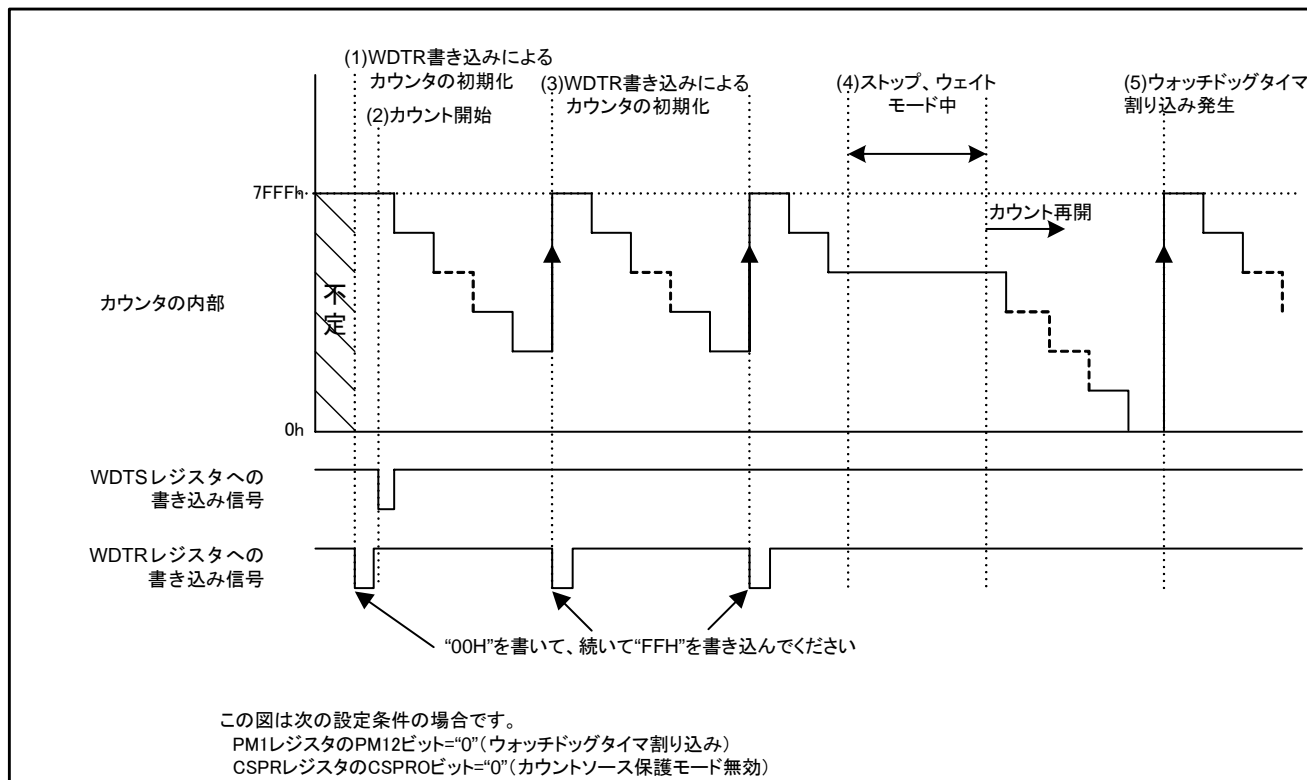
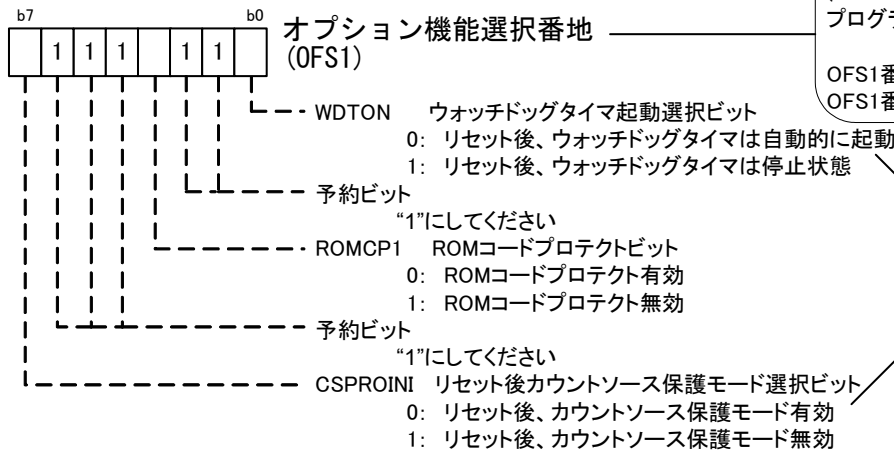


図1. ウォッチドッグタイマの動作例

4. 設定方法

「3. 応用例」の動作を実現するための設定手順と設定値を示します。各レジスタの詳細は M16C/64 グループのハードウェアマニュアルを参照ください。

(1) オプション機能選択番地の設定

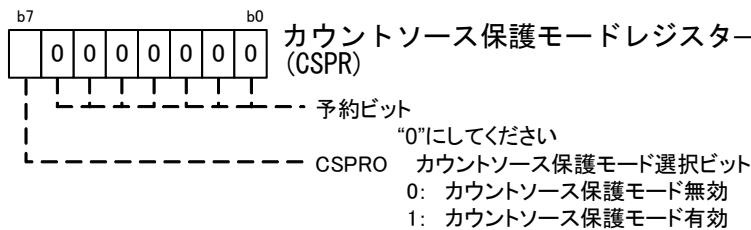


OFS1番地はフラッシュメモリ上にあります。(FFFFh番地) プログラムと一緒に書き込んでください。

OFS1番地を含むブロックを消去すると、OFS1番地は"FFh"になります

CSPROINIビットを"0"にするときは、WDTONビットも"0"にしてください。

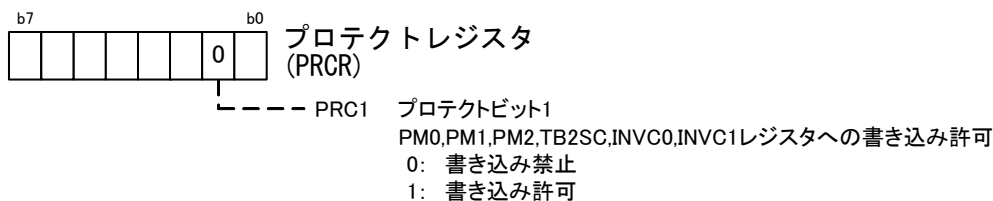
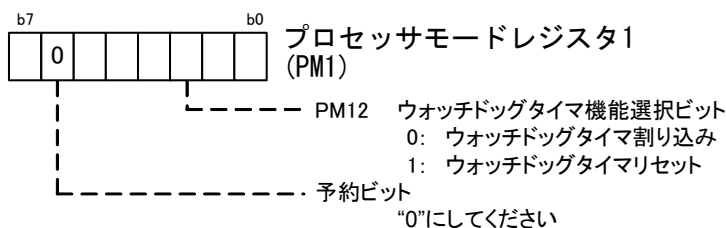
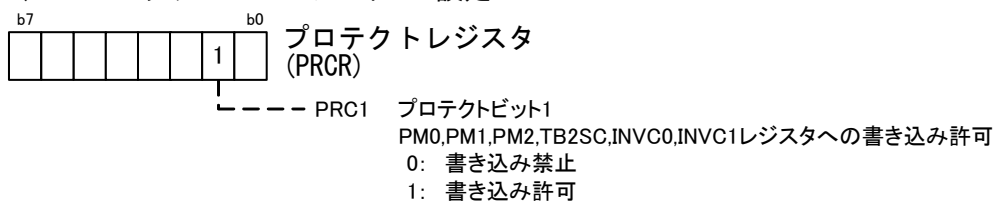
(2) カウントソース保護モードレジスタの設定



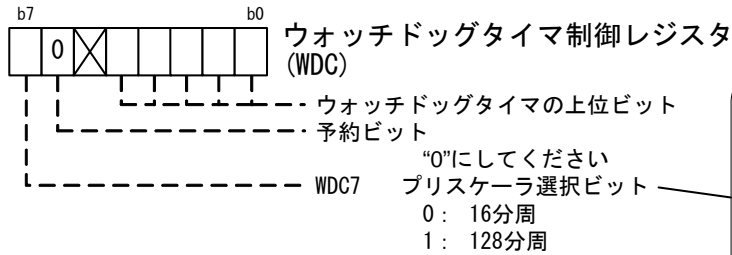
OFS1番地のCSPROINIビットに"0"を書いたとき、リセット後の値は"10000000b"になります。

CSPROビットを"1"にするためには、"0"を書いた後で、続いて"1"を書いてください。プログラムでは"0"にできません。

(3) プロセッサモードレジスタ1の設定



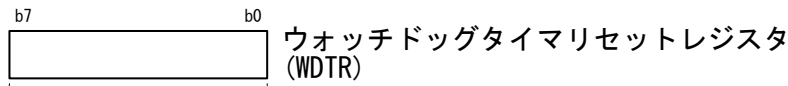
(4) ウォッチドッグタイマ制御レジスタの設定



CSPROビットが“1”(カウントソース保護モード有効)の場合、カウントソースはオンチップオシレータとなり、WDC7の値にかかわらず、プリスケールによる分周を行いません。

CSPROビットが“0”(カウントソース保護モード無効)かつCPUクロックにサブクロックを選択している場合、WDC7の値にかかわらず、プリスケールの分周は2分周となります。

(5) ウォッチドッグタイマリセットレジスタの設定

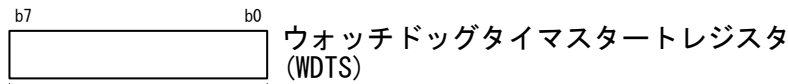


“00h”を書いて、続いて“FFh”を書くと、ウォッチドッグタイマは初期化されます。
ウォッチドッグタイマの初期値

- ・カウントソース保護モード無効時: “7FFFh”
- ・カウントソース保護モード有効時: “0FFFh”

“00h”の書き込みと、“FFh”の書き込みの間に、割り込み、DMA転送が入らないようにしてください。
ウォッチドッグタイマ割り込み発生後は、WDTRレジスタでウォッチドッグタイマを初期化してください。

(6) ウォッチドッグタイマスタートレジスタの設定



WDTSレジスタに対する書き込み命令で、ウォッチドッグタイマはスタートします。

5. 参考プログラム例

5.1 OFS1 番地への書き込み

OFS1 番地はフラッシュメモリ上にあるためプログラムと一緒に書き込みます。

図 2 に C 言語で as30 の拡張指示命令“.OFSREG”を使用した OFS1 番地への設定方法を示します。

```
#if __WATCH_DOG__ != 0
    _asm(“.ofsreg OFEH"); /* WATCH DOG TIMER START When on Reset */
#else
    _asm(“.ofsreg OFFH"); /* WATCH DOG TIMER STOP When on Reset */
#endif
```

図 2. OFS1 番地への設定方法

5.2 ウォッチドッグタイマ割り込みを使用する場合

本例では、CPU クロックに PLL クロック(24MHz)を使用します。また、OFS1 番地の WDTON ビット、CSPROINI ビットは“1”を設定しています。

WDTR レジスタへの書き込みを行いながら、ポート P10 の表示を 500ms ごとにインクリメントします。ポート P10 の出力が“40h”になると、WDTR レジスタへの書き込みを止め、ポート P10 の表示の更新を停止します。

ウォッチドッグタイマ割り込みが発生すると、ウォッチドッグタイマ割り込み処理にて、WDTR レジスタへの書き込みを行いながら、ポート P10 の表示を 500ms ごとにデクリメントします。ポート P10 の出力が“00h”になるとポート P10 の表示の更新を停止します。

```

/*****/
/*
/* M16C/64 Group Program Collection
/*
/* File name : rjj05b1289_int_src.c
/* CPU : M16C/64 Group
/* Function : Operation of Watchdog Timer
/* Version : 1.00 (2008-05-21) Initial
/*
/* Copyright(C)2008, Renesas Technology Corp. , All rights reserved.
/*
/*****/
/* Refer to the corresponding application notes for program specifications. */

#include "sfr64.h"

void main(void);
void mcu_init(void);

/*****/
/* DEFINE
/*****/
void wdt_int(void);

/*****/
/* RAM
/*****/

/*****/
/* ROM
/*****/

```



```

/*****
/*   Pragma
*****/

/*****
/*   Main Program
*****/
void main(void)
{
    unsigned int i;

    mcu_init();          /* MCU initialize */

    p10 = 0;
    pd10 = 0xff;

    ta0mr = 0x40;        /* Selection of timer mode
                          Pulse output function select bit (0:Pulse is not output)
                          Gate function select bit (00:Gate fuction not available)
                          Count source (01:f8) */
    ta0 = 4000-1;        /* Setting counter value (1msec @24MHz, f8) */

    cpsrf = 0;          /* Setting clock prescaler reset flag (0:No effect) */

    ta0ic = 0x00;       /* Setting interrupt priority levels in timer A0 */

    ta0s = 1;           /* TimerA1 count start */

    prcr = 0x02;

    pm12 = 0;           /* Watchdog Timer Function Select Bit(0 : Watchdog timer interrupt)
                          */
    prcr = 0x00;

    wdc = 0;            /* Setting watchdog timer control register
                          Prescaler select bit is set to 0 (0:Divided be 16) */
    wdtr = 0x00;        /* Setting watchdog timer reset register
                          Watchdog timer initialize */
    wdtr = 0xff;        /* Setting watchdog timer value (0x7FFF) */

    i = 0;

    wdts = 1;           /* Setting watchdog timer start register */

    while (1) {
        while (!ir_ta0ic) {}
        ta0ic = 0x00;
        i++;
        if ( i >= 500 ) {
            i = 0;
            p10++;
        }
        if ( p10 >= 64 ) {
            p10 = 64;
        } else {
            wdtr = 0x00;    /* Setting watchdog timer reset register
                              Watchdog timer initialize */
            wdtr = 0xff;    /* Setting watchdog timer value (0x7FFF) */
        }
    }
}

/*****
/*   MCU Initialize
*****/
void mcu_init()
{
    unsigned int i;
    prcr = 0x03;        /* CM0, CM1, CM2, PLC0, PCLKR register protect off */
    pm0 = 0x00;         /* PM0, PM1, PM2, TB2SC, INVCO, INVC1 register protect off */
    pm1 = 0x08;         /* Processor mode: Single-chip mode */
    cm2 = 0x00;         /* Watchdog timer function: Watchdog timer interrupt */
    cm1 = 0x20;         /* Internal reserved area: The entire area usable */
    cm0 = 0x08;         /* Wait: No wait state */
    cm2 = 0x00;         /* System clock select: Main clock */
    cm1 = 0x20;         /* Main clock division select: No division mode */
    cm0 = 0x08;         /* Main clock division select: CM16 and CM17 enabled */
}

```

```

pm20 = 0;          /* SFR 2waits when PLL on          */
plc0 = 0x14;       /* PLL clock Multiply by 4          */
                    /* PLL multiplying factor select: Multiply by 8 */
                    /* Reference frequency counter: Divide by 2   */
plc07 = 1;        /* PLL operation enable: PLL ON     */
for (i = 0; i < 20000; i++); /* Wait until the PLL clock becomes stable (tsu(PLL)) */
cm11 = 1;        /* System clock select: PLL clock   */
prcr = 0x00;     /* Protect on                       */
}

/*****
 * Watchdog timer interrupt routine
 *****/
#pragma interrupt wdt_int()
void wdt_int() {
    unsigned int i = 0;

    wdtr = 0x00;    /* Set the WDTS register in the beginning of the */
    wdtr = 0xff;   /* watchdog timer interrupt routine */

    while (1) {
        while (!ir_ta0ic) {}
        ta0ic = 0x00;
        i++;

        wdtr = 0x00;
        wdtr = 0xff;

        if(i == 500) {
            i = 0;
            if ( p10 != 0 ) {
                p10--;
            }
        }
    }
}
}

```

5.3 ウォッチドッグタイマのアンダフローによりリセットする場合

本例では、CPUクロックにPLLクロック(24MHz)を使用します。また、OFS1番地のWDTONビット、CSPROINIビットは“1”を設定しています。

WDTRレジスタへの書き込みを行いながら、ポートP10の表示を500msごとにインクリメントします。ポートP10の出力が“40h”になると、WDTRレジスタへの書き込みを止め、ポートP10の表示の更新を停止します。ウォッチドッグタイマのアンダフローが発生すると、リセットされます。

```

/*****
 */
/* M16C/64 Group Program Collection
 */
/* File name   : rjj05b1289_reset_src.c
 */ CPU       : M16C/64 Group
 */ Function  : Operation of Watchdog Timer
 */ Version   : 1.00 (2008-05-21) Initial
 */
/* Copyright(C)2008, Renesas Technology Corp. , All rights reserved.
 */
/*****
 */ Refer to the corresponding application notes for program specifications.
 */

#include "sfr64.h"

void main(void);
void mcu_init(void);

/*****
 */ DEFINE
 *****/
void wdt_int(void);

/*****
 */ RAM
 *****/

```

```

/*****/
/* ROM
/*****/

/*****/
/* Pragma
/*****/

/*****/
/* Main Program
/*****/
void main(void)
{
    unsigned int i;

    mcu_init();          /* MCU initialize */

    p10 = 0;
    pd10 = 0xff;

    ta0mr = 0x40;        /* Selection of timer mode
                          Pulse output function select bit (0:Pulse is not output)
                          Gate function select bit (00:Gate function not available)
                          Count source (01:f8) */
    ta0 = 4000-1;        /* Setting counter value (1msec @24MHz, f8) */

    cpsrf = 0;          /* Setting clock prescaler reset flag (0:No effect) */

    ta0ic = 0x00;        /* Setting interrupt priority levels in timer A0 */

    ta0s = 1;           /* TimerA1 count start */

    prcr = 0x02;

    pm12 = 1;           /* Watchdog Timer Function Select Bit(1 : Watchdog timer reset)
                          */
    prcr = 0x00;

    wdc = 0;            /* Setting watchdog timer control register
                          Prescaler select bit is set to 0 (0:Divided be 16) */
    wdtr = 0x00;        /* Setting watchdog timer reset register
                          Watchdog timer initialize */
    wdtr = 0xff;        /* Setting watchdog timer value (0x7FFF) */

    i = 0;

    wdts = 1;           /* Setting watchdog timer start register */

    while (1) {
        while (!ir_ta0ic) {}
        ta0ic = 0x00;
        i++;
        if ( i >= 500 ) {
            i = 0;
            p10++;
        }
        if ( p10 >= 64 ) {
            p10 = 64;
        } else {
            wdtr = 0x00;          /* Setting watchdog timer reset register
                                  Watchdog timer initialize */
            wdtr = 0xff;          /* Setting watchdog timer value (0x7FFF) */
        }
    }
}

/*****/
/* MCU Initialize
/*****/
void mcu_init()
{
    unsigned int i;
    prcr = 0x03;          /* CMO, CM1, CM2, PLC0, PCLKR register protect off */
    pm0 = 0x00;          /* PM0, PM1, PM2, TB2SC, INVCO, INVCI register protect off */
    pm1 = 0x08;          /* Processor mode: Single-chip mode */
                          /* Watchdog timer function: Watchdog timer interrupt */
                          /* Internal reserved area: The entire area usable */
}

```

```

    cm2 = 0x00;          /* Wait: No wait state */
    cm1 = 0x20;          /* System clock select: Main clock */
    cm0 = 0x08;          /* Main clock division select: No division mode */
    pm20 = 0;           /* Main clock division select: CM16 and CM17 enabled */
    plc0 = 0x14;         /* SFR 2waits when PLL on */
                        /* PLL clock Multiply by 4 */
                        /* PLL multiplying factor select: Multiply by 8 */
                        /* Reference frequency counter: Divide by 2 */
    plc07 = 1;          /* PLL operation enable: PLL ON */
    for (i = 0; i < 20000; i++); /* Wait until the PLL clock becomes stable (tsu(PLL)) */
    cm11 = 1;           /* System clock select: PLL clock */
    prcr = 0x00;        /* Protect on */
}

/*****
 * Watchdog timer interrupt routine
 *****/
#pragma interrupt wdt_int()
void wdt_int() {
    unsigned int i = 0;

    wdtr = 0x00;        /* Set the WDTS register in the beginning of the */
    wdtr = 0xff;        /* watchdog timer interrupt routine */

    while (1) {
        while (!ir_ta0ic) {}
        ta0ic = 0x00;
        i++;

        wdtr = 0x00;
        wdtr = 0xff;

        if(i == 500){
            i = 0;
            if ( p10 != 0 ) {
                p10--;
            }
        }
    }
}

```

6. 参考ドキュメント

ハードウェアマニュアル

M16C/64 グループハードウェアマニュアル

(最新版をルネサス テクノロジホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス テクノロジホームページから入手してください。)

ホームページとサポート窓口

ルネサス テクノロジM16Cホームページ

<http://japan.renesas.com/m16c>

ルネサス製品全般に関するお問い合わせ先

<http://japan.renesas.com/inquiry>

E-mail : csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.05.21	-	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事情報の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ (<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。
5. 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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのある機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444