

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、誤りがな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。

SH7730 グループ

CMT 設定例

要旨

この資料はコンペアマッチタイマ機能 (CMT) の設定例を掲載しています。

動作確認デバイス

SH7730

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. CMT レジスタ設定手順.....	4
4. 参考プログラムについて.....	6
5. 参考プログラム例.....	16
6. 参考ドキュメント.....	31

1. はじめに

1.1 仕様

- コンペアマッチタイマ (CMT チャンネル 0) を 500ms フリーラン動作タイマとして使用します。
- コンペアマッチ内部割り込みを使用して、500ms 間隔で LED の出力反転を行います。
- 10 回点灯が行われた際に、コンペアマッチタイマを停止します。

1.2 使用機能

- CMT チャンネル 0

1.3 適用条件

評価ボード:	アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
外付けメモリ (エリア 0):	NOR 型フラッシュメモリ 4 M バイト Spansion 製 S29AL032D70TFI04
(エリア 3):	SDR-SDRAM 32 M バイト (16 M バイト × 2 個) Samsung 製 K4S281632F-UC75
マイコン:	SH7730 (R8A77301)
動作周波数	CPU クロック: 266.66 MHz SuperHyway バスクロック: 133.33 MHz バスクロック: 66.66 MHz 周辺クロック: 33.33 MHz
エリア 0 バス幅:	16 ビット固定 (MD3 端子 = Low レベル)
クロック動作モード:	モード 2 (MD0 端子 = Low レベル, MD1 端子 = High レベル)
エンディアン:	ビッグエンディアン (MD5 端子 = Low レベル)
ツールチェーン:	ルネサス テクノロジ製 SuperH RISC engine Standard Toolchain Ver.9.2.0.0
コンパイルオプション:	High-performance Embedded Workshop でのデフォルト設定 <code>(-cpu=sh4a -include="\$(PROJDIR)¥inc"</code> <code>-object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -optimize=0</code> <code>-gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all</code> <code>-infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)</code>

1.4 関連アプリケーションノート

本資料の参考プログラムは、「SH7730 グループ アプリケーションノート SH7730 初期設定例 (RJJ06B0864)」の設定条件で動作確認しています。そちらもあわせてご参照ください。

2. 応用例の説明

本参考プログラムではコンペアマッチタイマ (CMT チャンネル 0) を使用し、500ms 間隔でコンペアマッチを発生させ、コンペアマッチ時の内部割り込み処理で LED の出力反転処理を行います。

2.1 使用機能の動作概要

コンペアマッチタイマは、コンペアマッチタイマカウンタ (CMCNT) をカウントアップさせるクロック (CMCSR.CKS の選択) でカウントアップさせ、あらかじめ設定しておいたコンペアマッチコンスタントレジスタ (CMCOR) の値と一致する (コンペアマッチ) までの時間で時間計測を行います。

CMCNT をカウントアップさせるクロックは、 $P\phi$ (周辺クロック) とコンペアマッチタイマコントロール/ステータスレジスタ (CMCSR) の CKS[2:0] の設定値により決まります。

CMCNT の値が CMCOR の値と一致すると、CMCNT は H'00000000 にクリアされ、CMCSR の CMF ビットが 1 にセットされます。このとき、CMCSR レジスタのコンペアマッチリクエストビット 1, 0 (CMR[1:0]) に B'10 が設定されている場合、内部割り込みが発生します。

表 1 に CMT の概要を示します。

CMT 機能の詳細については、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT)」を参照ください。

表 1 CMT の概要

項目	概要
チャンネル数	5 チャンネル
カウンタサイズ	16 ビットカウンタ or 32 ビットカウンタ 詳細は、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT) 19.4.2 カウンタサイズ」を参照ください。
クロックソース	$P\phi/8$ or $P\phi/32$ or $P\phi/128$ $P\phi$: 周辺クロック
コンペアマッチモード	ワンショット動作 or フリーラン動作 詳細は、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT) 19.4.1 カウンタ動作」を参照ください。
コンペアマッチフラグ	CMCOR と CMCNT が一致するとコンペアマッチ信号が発生し、CMCSR の CMF ビットが 1 にセットされます。 詳細は、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT) 19.4.5 コンペアマッチフラグのセットタイミング」を参照ください。
割り込み要求	DMA 転送要求 or 内部割り込み要求 詳細は、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT) 19.4.4 DMA 転送要求と CPU への内部割り込み要求」を参照ください。

3. CMT レジスタ設定手順

CMT 開始時のレジスタ設定フローについては図 1 ,CMT 停止時のレジスタ設定フローについては図 2 に記載します。

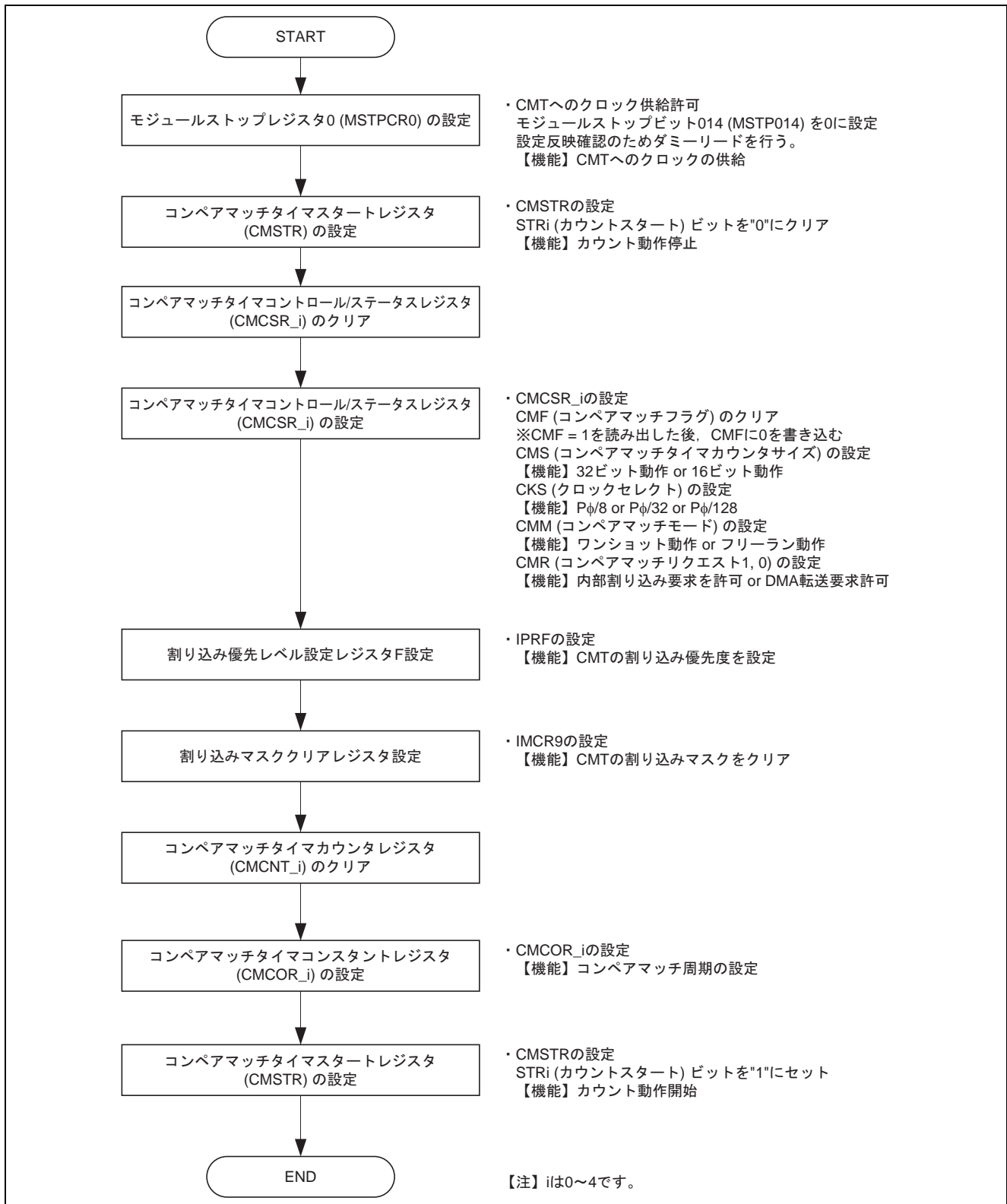


図 1 CMT 開始時レジスタ設定フロー

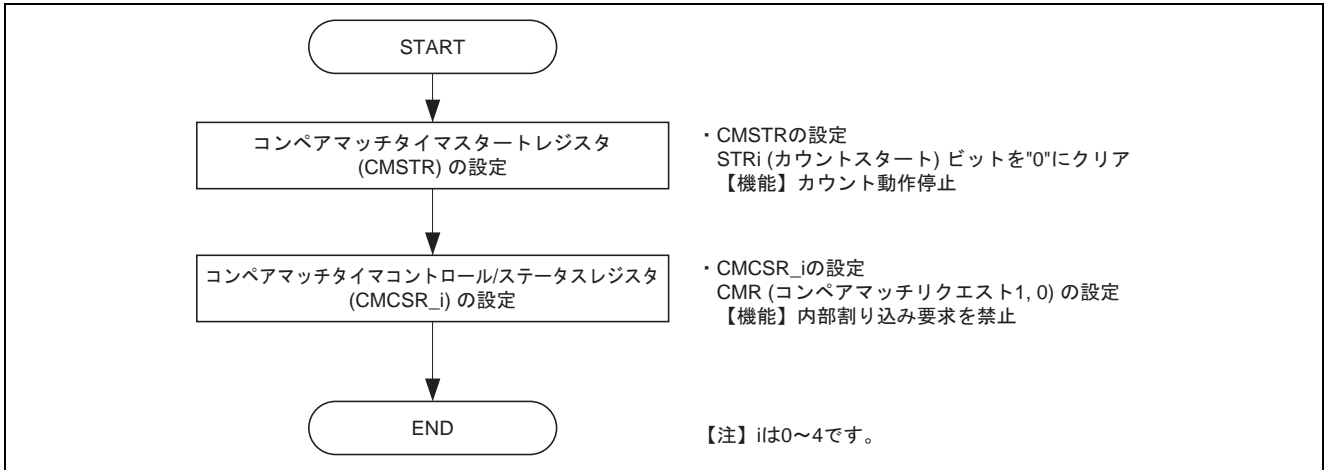


図 2 CMT 停止時レジスタ設定フロー

4. 参考プログラムについて

参考プログラムの動作は、500ms ごとに LED を点滅させ、10 回の LED 点灯後、LED 点滅を停止します。本参考プログラムでは、コンペアマッチタイマ (CMT0) を 500ms のフリーラン動作タイマとして使用し、コンペアマッチ内部割り込みで、コンペアマッチフラグ (CMCSR.CMF) のクリアと LED の出力反転処理をしています。10 回の LED 点灯後、タイマの停止を行っています。

4.1 参考プログラムの動作仕様

表 2 参考プログラム動作仕様

項目	概要
使用チャンネル	0 チャンネル
カウンタサイズ	32 ビットカウンタ
クロックソース	Pφ/128
コンペアマッチモード	フリーラン動作
オーバフローフラグ	使用せず
コンペアマッチリクエスト	内部割り込み要求を許可
CMT 割り込み優先レベル	1
タイマ設定時間	500ms

4.2 参考プログラムのインタフェース

参考プログラムでは、CMT0 を動作させるために、以下のインタフェースを作成します。

(1) CMT0 タイマ生成処理 (cmt0_create())

【概要】

CMT0 タイマの生成処理を行います。このインタフェースでは、パラメータの設定値により、CMT0 のコンペアマッチタイマカウンタサイズ (CMCSR.CMS)、クロックセレクト (CMCSR.CKS)、割り込み優先レベル (INTC.IPRF) をレジスタに設定します。コンペアマッチモードについては、フリーラン動作固定とします。

【パラメータ】

型	変数	内容
T_CMT_TimerTYPE	mCountSize	CMCSR.CMS の設定値
	mClockSelect	CMCSR.CKS の設定値
	mIntcPri	INTC.IPRF の設定値

(2) CMT0 タイマ開始処理 (cmt0_start())

【概要】

CMT0 タイマを起動します。このインタフェースでは、パラメータのタイマ設定時間よりコンペアマッチ周期を計算し、CMCOR に値を設定します。また、コンペアマッチが発生した際 (タイムアウト時) にコールされる関数の登録を行います。

【パラメータ】

型	変数	内容
T_CMT_SETCMCOR_INFO	mTimeValue	タイマ設定時間 (ms) 本応用例での制限事項 ・ ms 単位のみ指定可能 ・ 1ms ~ 1000ms の範囲指定可能
CMT_TIMEOUT_CALLBACK	void (*CMT_TIMEOUT_CALLBACK)(void)	コンペアマッチ発生時 (タイムアウト時) のコールバック関数登録用

(3) CMT0 タイマ停止処理 (cmt0_stop())

【概要】

CMT0 タイマを停止します。

【パラメータ】

なし。

4.3 参考プログラムの処理フロー

4.3.1 メイン処理フロー

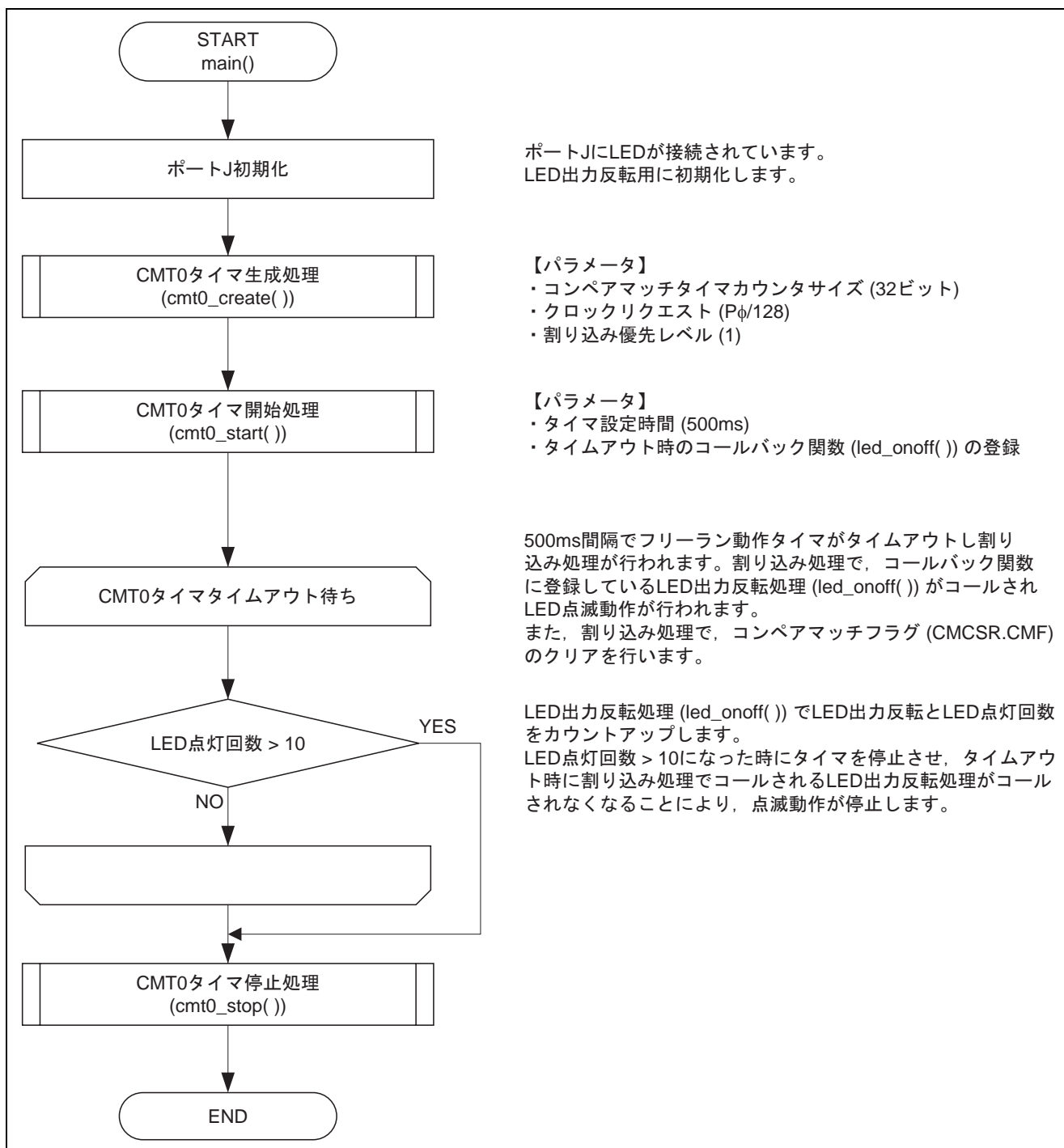


図3 CMT メイン処理フロー

4.3.2 CMT0 タイマ生成処理

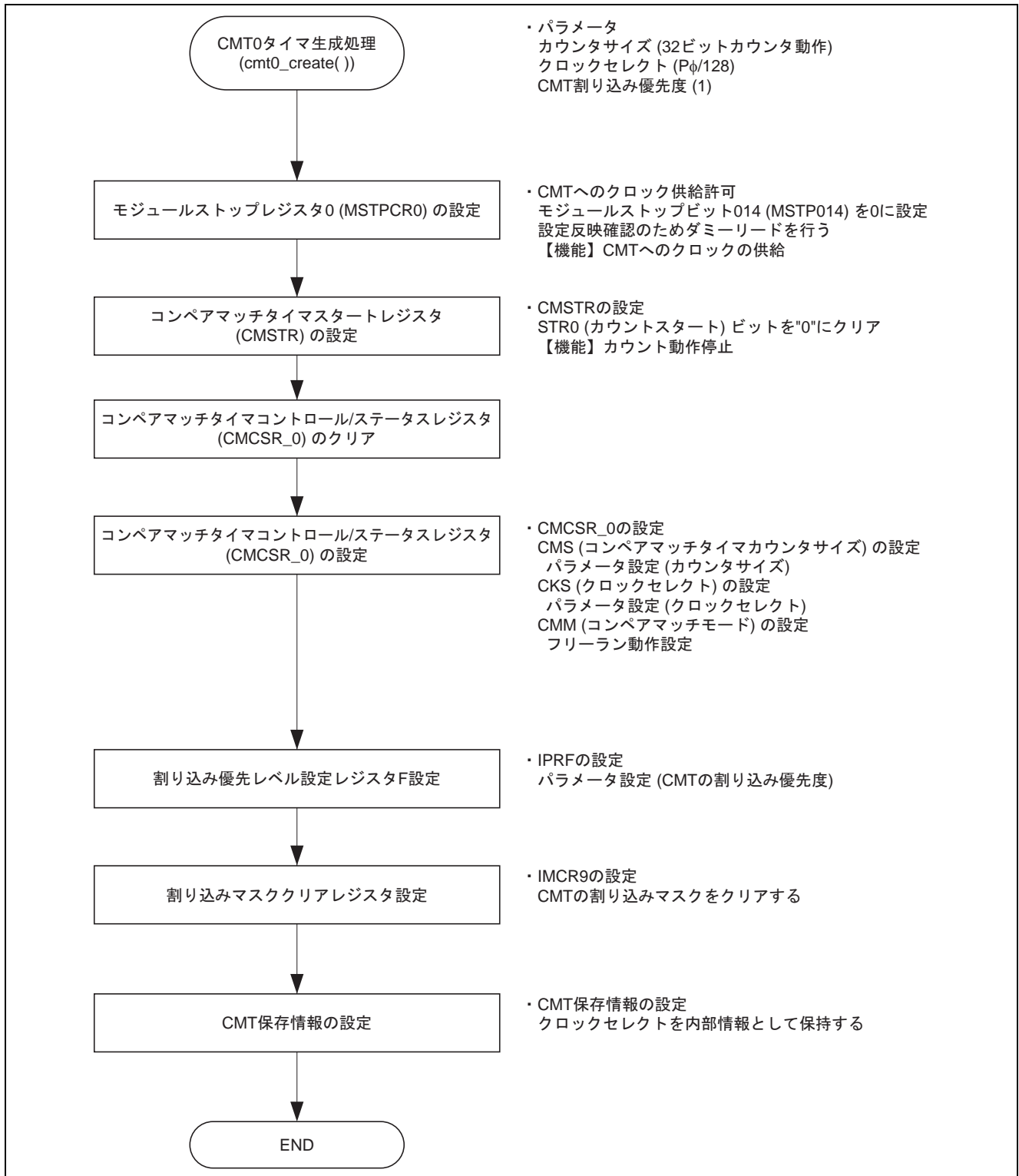


図 4 CMT0 タイマ生成処理フロー

4.3.3 CMT0 タイマ開始処理

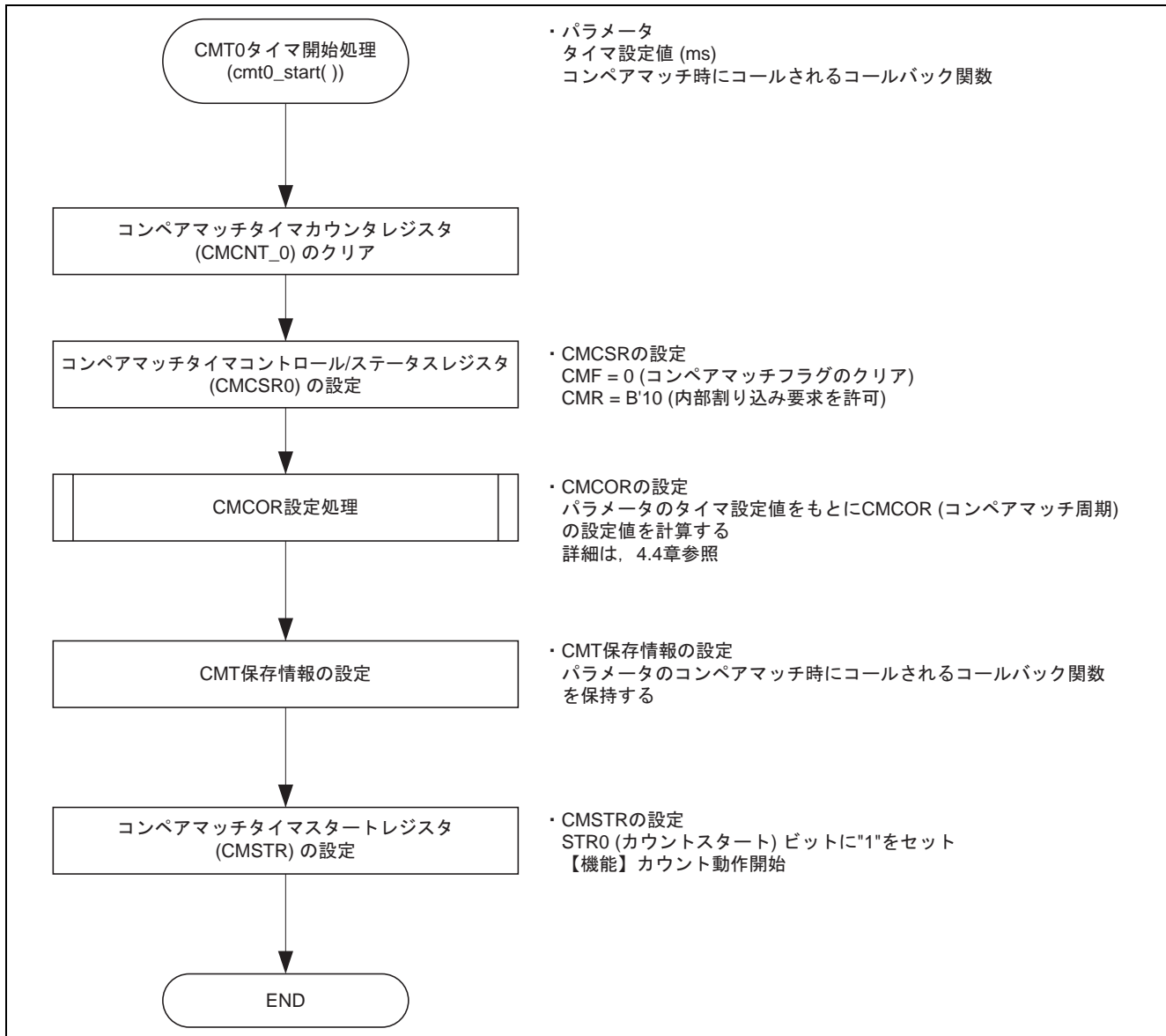


図 5 CMT0 タイマ開始処理フロー

4.3.4 CMT0 タイマ停止処理

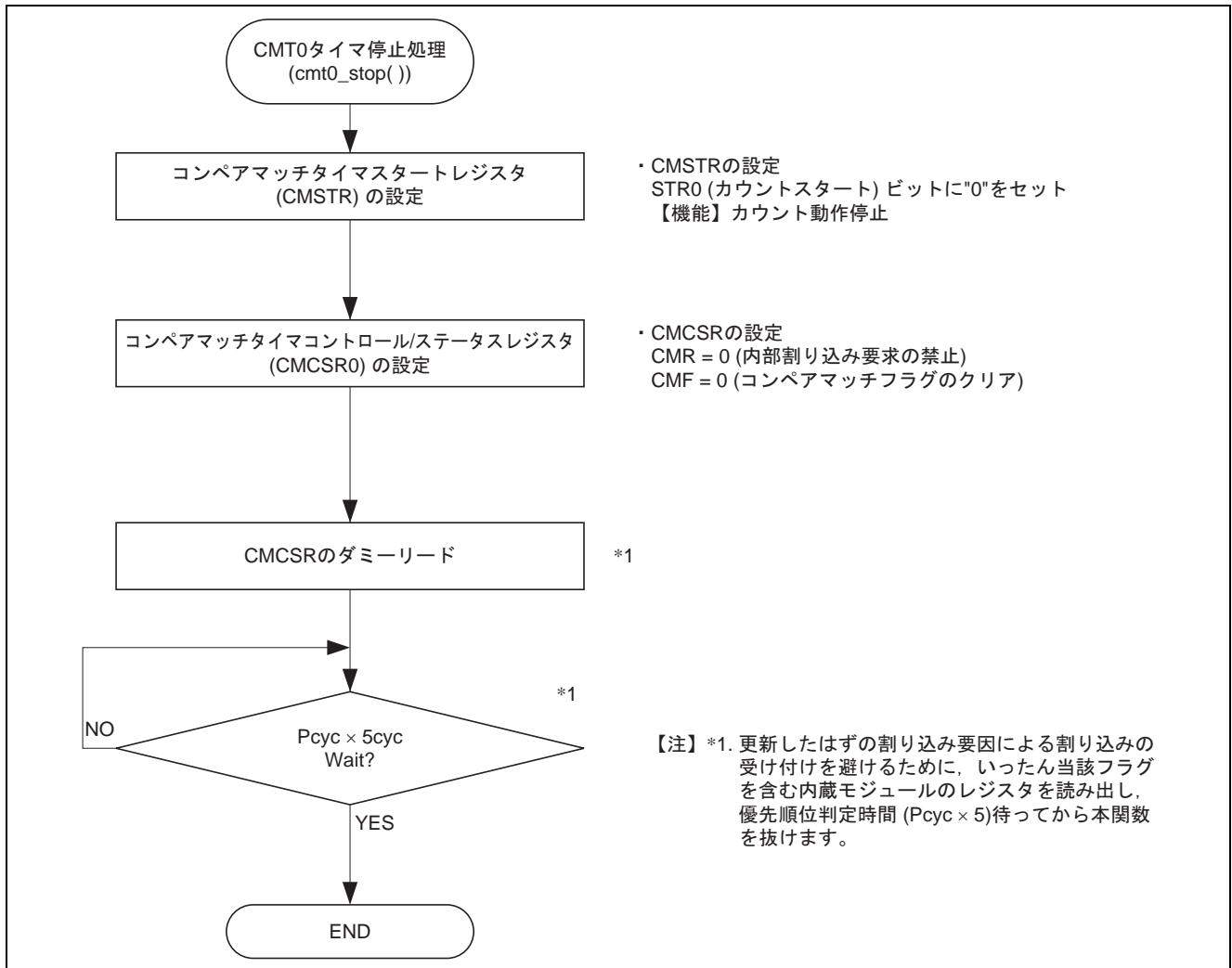


図 6 CMT0 タイマ停止処理フロー

4.3.5 割り込み時の処理フロー

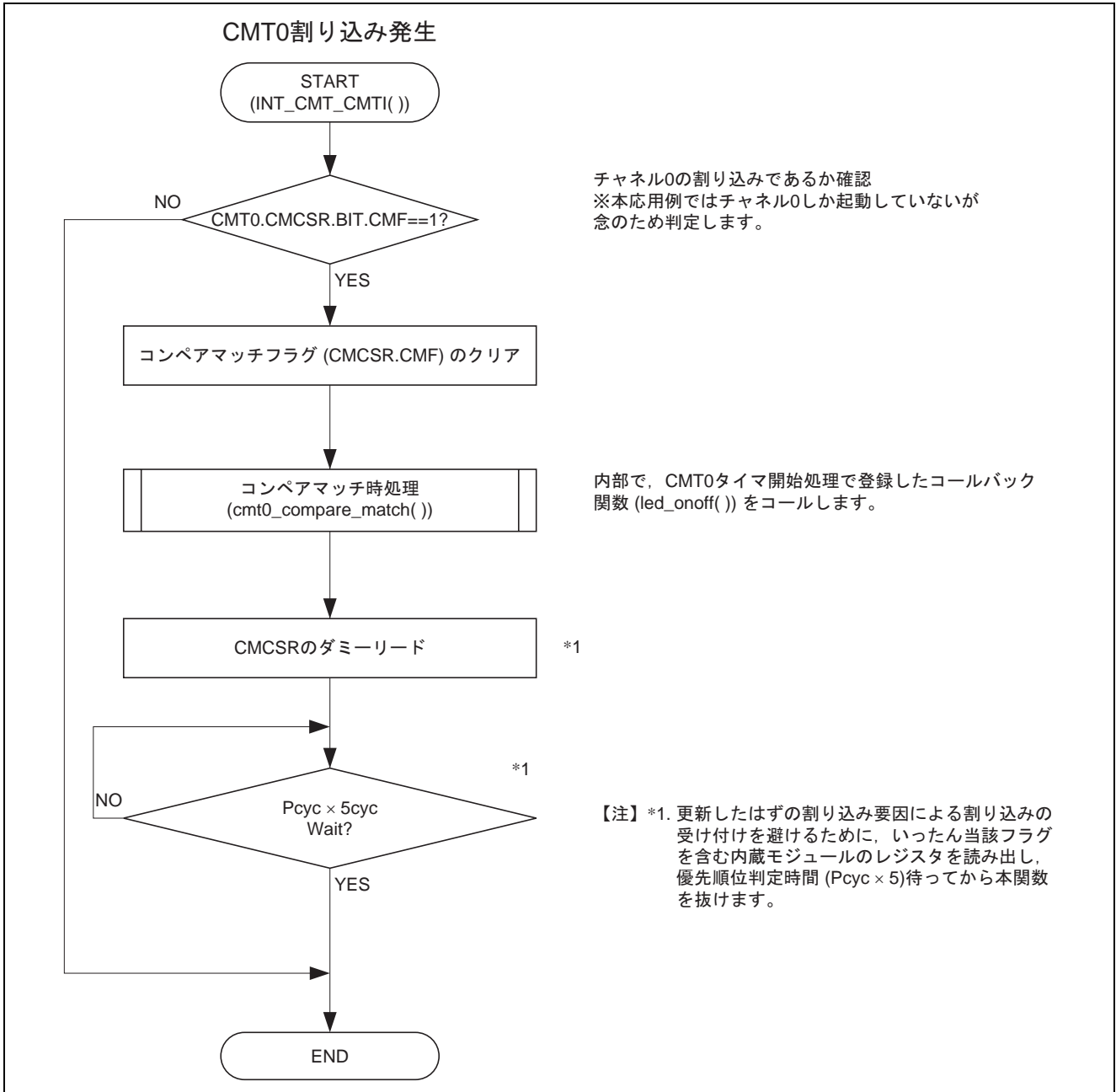


図 7 CMT 割り込み処理フロー

4.3.6 コールバック登録関数 (led_onoff()) の処理フロー

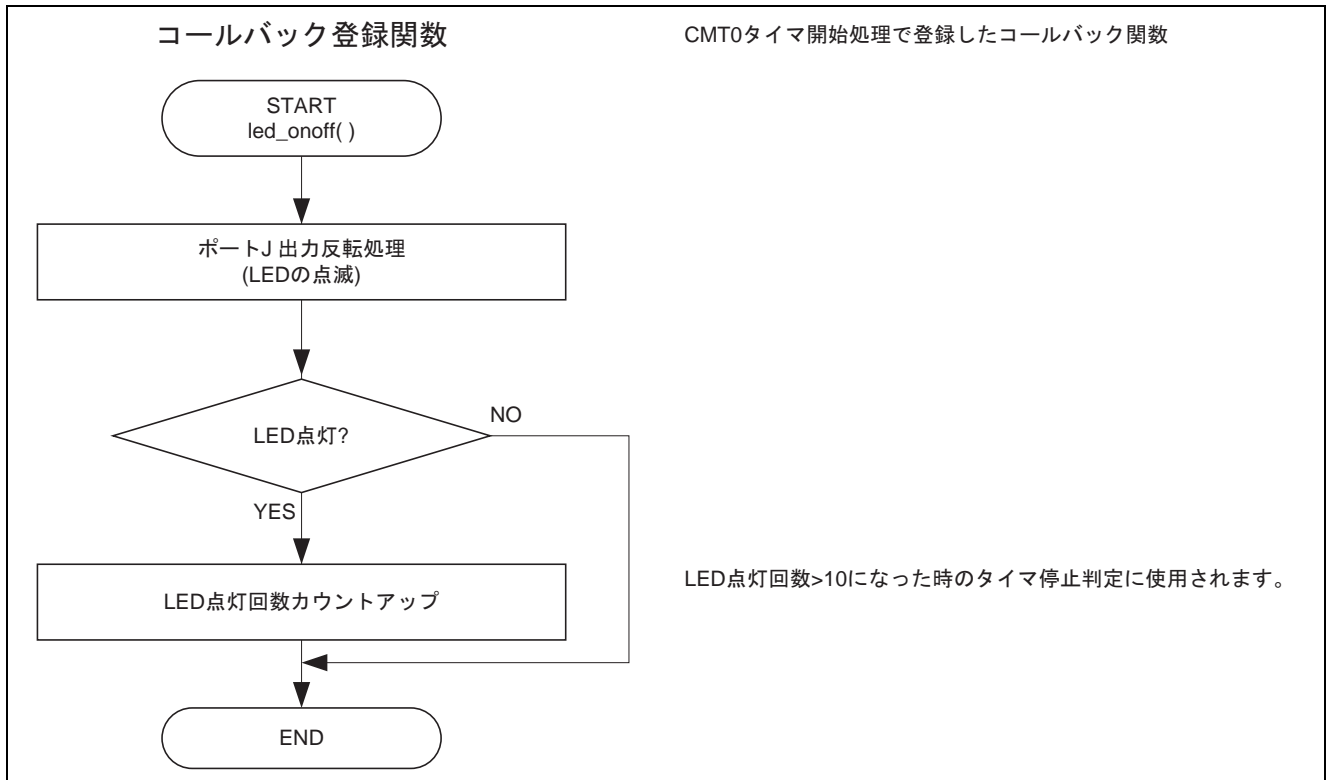


図 8 コールバック登録関数 (led_onoff())

4.4 参考プログラムのレジスタ設定

表 3 に参考プログラムのレジスタ設定を示します。

表 3 参考プログラムのレジスタ設定

レジスタ名	アドレス	設定値	機能と設定値
コンペアマッチタイマ スタートレジスタ (CMSTR)	H'A44A0000	H'0000	<ul style="list-style-type: none"> タイマ生成時 タイマ停止時 STR[4:0] = B'0: タイマカウント停止
		H'0001	<ul style="list-style-type: none"> タイマ開始時 STR[4:0] = B'1: タイマカウント開始
コンペアマッチタイマ コントロール/ステータス レジスタ_0 (CMCSR_0)	H'A44A0010	H'0102	<ul style="list-style-type: none"> タイマ生成時 CMS = 0: 32 ビットカウンタ動作 CKS[2:0] = B'010: Pφ/128 CMM = 1: フリーラン動作
		H'0122	<ul style="list-style-type: none"> タイマ開始時 CMF = 0: コンペアマッチフラグのクリア CMR[1:0] = B'10: 内部割り込み要求を許可
		H'0102	<ul style="list-style-type: none"> タイマ停止時 CMR[1:0] = B'00: 内部割り込み要求の禁止 CMF = 0: コンペアマッチフラグのクリア
コンペアマッチタイマ カウンタ_0 (CMCNT_0)	H'A44A0014	H'00000000	<ul style="list-style-type: none"> タイマ開始時 タイマカウントクリア
コンペアマッチタイマ コンスタントレジスタ_0 (CMCOR_0)	H'A44A0018	H'0001FC9F	<ul style="list-style-type: none"> タイマ開始時 130207 (H'1FC9F) 回: 約 500ms 詳細は、下記【CMCOR 設定値】を参照ください
割り込み優先レベル設定 レジスタ F (IPRF)	H'A4080014	H'0001	<ul style="list-style-type: none"> タイマ生成時 IPR3n (ビット名) = B'0001: 優先レベル 1
割り込みマスククリア レジスタ (IMCR9)	H'A40800E4	H'0100	<ul style="list-style-type: none"> タイマ生成時 IMCRn3 (ビット名) = B'1: 割り込みマスク クリア

【CMCOR 設定値】

Pφ = 33.33MHz, クロックセレクト = Pφ/128 として, タイマカウンタ開始後 500ms 後にコンペアマッチさせる CMCOR の設定値は以下ようになります。

Pφ = 33.33MHz の場合の 1 クロック時間: 約 30 (ns)
 CMCNT が 1 カウントアップする時間: 30 (ns) () × 128 = 3840 (ns)
 500ms を ns に変換: 1000000 × 500 = 500000000 (ns)
 500ms でのカウント数: 500000000 () ÷ 3840 () ≈ 130208 (H'1FCA0) 回

コンペアマッチフラグのセットタイミングは, CMCOR を N と設定した場合に CMCNT==N にするカウンタ用クロックを N クロックとした場合に, 次のクロック (N+1 クロック) のタイミングでセットされます。詳細は, 「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 19 章 コンペアマッチタイマ (CMT) 19.4.5 コンペアマッチフラグのセットタイミング」を参照ください。

そのため, 実際に割り込みが発生するのは, N+1 クロックのタイミングとなりますので, 上記 で求めた値から-1 した値を CMCOR に設定します。

CMCOR 設定値 = 130208 - 1 = 130207 (H'1FC9F)

【コンペアマッチ周期最大時間】

$P\phi = 33.33\text{MHz}$, クロックセレクト = $P\phi/128$ として、コンペアマッチ周期最大値を計算すると以下のようになります。

CMCOR 設定最大値: 4294967295 (H'FFFFFFFF)

CMCNT が 1 カウントアップする時間: 上記 より , 3840ns

コンペアマッチ周期最大時間: 4294967295×3840 (ns) = 16492674412800 (ns) \approx 16492 (秒)

つまり、 $P\phi = 33.33\text{MHz}$, クロックセレクト = $P\phi/128$ として CMCOR に設定できるタイマの最大値は、約 16492 (秒) となります。それ以上のタイマを設定する必要がある場合、例えば、20000 秒のタイマ設定する場合は、1 秒のタイマを 20000 回行う等の対応が必要となります。

4.5 参考プログラムのマクロ定義

表 4 に参考プログラムのマクロ定義を示します。

表 4 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
D_CMT_P_FREQ	H'001E	$P\phi$ の 1 クロック時間 (ns) $P\phi = 33.33\text{MHz}$ の場合の値を設定しています。 $P\phi$ を変更する場合は、この設定値を変更してください。
D_CMT_32BITCOUNTER	H'0000	32 ビットカウンタ動作
D_CMT_16BITCOUNTER	H'0001	16 ビットカウンタ動作 本参考プログラムでは使用を想定しません。
D_CMT_ONESHOT	H'0000	ワンショット動作
D_CMT_FREERUN	H'0001	フリーラン動作
D_CMT_ALLSTOP	H'0000	DMA 転送要求および内部割込み要求の禁止
D_CMT_DMA_PERMIT	H'0001	DMA 転送要求を許可 本参考プログラムでは使用を想定しません。
D_CMT_INTC_PERMIT	H'0002	内部割込み要求を許可
D_CMT_DIV_8	H'0000	$P\phi/8$
D_CMT_DIV_32	H'0001	$P\phi/32$
D_CMT_DIV_128	H'0002	$P\phi/128$

5. 参考プログラム例

(1) サンプルプログラムリスト "sh7730.c"

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : sh7730.c
32 * Abstract : SH7730 CMT 設定例 Sample Program
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 CMT 設定例のサンプルプログラムです。
40 * :
41 * Operation :
42 * Limitation :
43 * :
44 *****/
45 * History : 03.DEC.2008 Ver. 1.00 First Release
46 /*"FILE COMMENT END"***** */
47
48 #include <machine.h>
49 #include "iodefine.h"
50 #include "cmt.h"
51
52 /* プロトタイプ宣言 */
    
```

```

53 void main(void);
54 void led_onoff(void);
55
56 /* LED 出力カウンタ */
57 unsigned long g_LedOutCount = 0;
58
59 /*"FUNC COMMENT"*****
60 * ID :
61 * Outline : コンペアマッチタイマサンプル処理
62 * Include :
63 * Declaration : void main(void)
64 * Description : コンペアマッチタイマの使用例を示します。
65 * :
66 * Argument : none
67 * Return Value : none
68 * Calling Functions :
69 *"FUNC COMMENT END"*****/
70 void main(void)
71 {
72     T_CMT_TimerType TimerType;
73     T_CMT_SetCmcorInfo SetCmcorInfo;
74
75     g_LedOutCount = 0;
76     memset(&TimerType, 0x00, sizeof(TimerType));
77     memset(&SetCmcorInfo, 0x00, sizeof(SetCmcorInfo));
78
79     /* ポート J 初期化 (Led 出力反転用) */
80     PFC.PJCR.WORD = (PFC.PJCR.WORD & 0xCFFF) | 0x1000; /* PJ6 出力設定 */
81
82     /* ==== タイマ生成用構造体の設定 ==== */
83     TimerType.mCountSize = D_CMT_32BITCOUNTER; /* 32 ビットカウンタ動作 */
84     TimerType.mClockSelect = D_CMT_DIV_128; /* Pφ/128 */
85     TimerType.mIntcPri = D_CMT_PRI_1; /* 割り込みレベル 1 */
86
87     /* ==== タイマ生成処理 ==== */
88     cmt0_create(&TimerType);
89
90     /* CMCOR 設定情報設定 */
91     SetCmcorInfo.mTimeValue = 500; /* タイマ設定値 (500ms) */
92
93     /* ==== CMT タイマ開始処理 ==== */
94     cmt0_start(&SetCmcorInfo, led_onoff);
95
96     /* led 点灯まで待つ */
97     while(1)
98     {
99         /* led が 10 回点灯したらループを抜ける */
100         if(g_LedOutCount == 10)
101         {
102             cmt0_stop();
103             break;
104         }
105     }
106 }
107
108 while(1)
109 {

```

```

110     }
111
112 }
113
114 /*"FUNC COMMENT"*****
115 * ID           :
116 * Outline      : Led 出力反転処理
117 * Include      :
118 * Declaration  : void led_onoff(void)
119 * Description  : Led 出力反転処理をします。
120 *             :
121 * Argument     : none
122 * Return Value : none
123 * Calling Functions :
124 *"FUNC COMMENT END"*****/
125 void led_onoff(void)
126 {
127
128     /* ==== ポート J 出力反転処理 ==== */
129     PORT.PJDR.BYTE ^= 0x40;          /* LED1 点滅 (ポート J: H   L ) */
130
131     /* 出力回数のカウント */
132     if( PORT.PJDR.BIT.B6 == 1 )
133     {
134         g_LedOutCount++;
135     }
136
137 }
138 /* End of File */

```

(2) サンプルプログラムリスト "cmt.c"

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name   : cmt.c
32 * Abstract    : SH7730 CMT 設定例 Sample Program
33 * Version     : Ver 1.00
34 * Device      : SH7730
35 * Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01.001)
36 *             : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS          : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description  : SH7730 CMT 設定例のサンプルプログラムです。
40 *             :
41 * Operation   :
42 * Limitation  :
43 *             :
44 *****/
45 * History     : 03.DEC.2008 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47
48 #include <machine.h>
49 #include "iodefine.h"
50 #include "cmt.h"
51
52 /* CMT 保存情報 */
53 static T_CMT_SAVE_INFO          g_CMT_SAVE_INFO;
54

```

```

55  /* クロックセレクト対応テーブル */
56  static const T_CMT_CLOCK_SELECT    gc_Clock_Match_tbl[] =
57  {
58      /* クロックセレクト */ /* CKS 設定値 */
59      {D_CMT_DIV_8,          D_CMT_DIVValue_8},
60      {D_CMT_DIV_32,         D_CMT_DIVValue_32},
61      {D_CMT_DIV_128,        D_CMT_DIVValue_128}
62  };
63
64  /* ローカル関数宣言 */
65  static void cmt_set_CMCOR_Type_ms(T_CMT_SETCMCOR_INFO *i_pCmcorInfo);
66  static T_CMT_PClockDiv cmt_get_ClockDivValue(void);
67  static void cmt0_set_g_CMT_SAVE_INFO(T_CMT_SAVE_INFO *i_pCMT_SAVE_INFO);
68  static void cmt0_get_g_CMT_SAVE_INFO(T_CMT_SAVE_INFO *o_pCMT_SAVE_INFO);
69
70  /*"FUNC COMMENT"*****
71  * ID                :
72  * Outline           : CMT0 タイマ生成処理
73  * Include           :
74  * Declaration       : void cmt0_create
75  *                   : (T_CMT_TimerTYPE *i_pTimerType)
76  * Description       : CMT0 の初期設定をする。
77  *                   :
78  * Argument          : T_CMT_TimerTYPE *i_pTimerType :
79  *                   :   タイマ生成用構造体アドレス
80  * Return Value      : none
81  * Calling Functions :
82  *"FUNC COMMENT END"*****/
83  void cmt0_create(T_CMT_TimerTYPE *i_pTimerType)
84  {
85      T_CMT_SAVE_INFO    ncmt_save_info;
86      unsigned long      dummy;
87
88      memset(&ncmt_save_info, 0x00, sizeof(ncmt_save_info));
89
90      /* ==== モジュールストップレジスタの設定 ==== */
91      LOWP.MSTPCR0 &= ~0x00004000;
92
93      dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
94
95      /* ==== コンペアマッチタイマスタートレジスタの設定 ==== */
96      CMT0.CMSTR.BIT.STR0 = 0; /* カウント停止 */
97
98      /* ==== コンペアマッチタイマコントロール/ステータスレジスタ (CMCSR0) の設定 ==== */
99      CMT0.CMCSR.WORD = 0x0000; /* 初期化 */
100
101      CMT0.CMCSR.BIT.CMS = i_pTimerType->mCountSize; /* カウンタサイズ設定 */
102      CMT0.CMCSR.BIT.CKS = i_pTimerType->mClockSelect; /* クロックセレクト設定 */
103      CMT0.CMCSR.BIT.CMM = D_CMT_FREERUN; /* フリーラン動作設定 */
104
105      /* CMT 保存情報にクロックセレクト設定 */
106      ncmt_save_info.mClockSelect = i_pTimerType->mClockSelect;
107
108      /* ==== 割り込み関連設定 ==== */
109      /* 割り込み優先レベル設定レジスタ F 設定 (CMT 有効) */
110      INTC0.IPRF = i_pTimerType->mIntcPri;
111

```

```

112     /* 割り込みマスククリアレジスタ */
113     INTC0.IMCR9 = 0x10;
114
115     /* CMT 保存情報に設定 */
116     cmt0_set_g_CMT_SAVE_INFO(&ncmt_save_info);
117
118 }
119
120 /***** FUNC COMMENT *****/
121 * ID :
122 * Outline : CMT0 タイマ開始処理
123 * Include :
124 * Declaration : void cmt0_start
125 * : (T_CMT_SETCMCOR_INFO *i_pCmcorInfo,
126 * : CMT_TIMEOUT_CALLBACK i_func)
127 * Description : CMT0 タイマを開始する。
128 * :
129 * Limitation : 以下の条件を前提としていますので、
130 * : 条件に合わない場合は修正してください。
131 * : (1)本関数は引数(T_CMT_SETCMCOR_INFO)が
132 * : ms の値のみ対応しています。
133 * : (2)本関数は P が 33.33MHz として対応しています。
134 * : 変更する場合は、#define D_CMT_P_FREQ
135 * : 変更してください。
136 * : (3)タイマ設定値は 1ms ~ 1000ms のみ対応して
137 * : います。
138 * : 範囲外の場合は、1ms として扱います。
139 * :
140 * Argument : T_CMT_SETCMCOR_INFO *i_pCmcorInfo:
141 * : CMCOR 設定情報構造体アドレス
142 * : CMT_TIMEOUT_CALLBACK i_func:
143 * : タイムアウト時コールバック関数
144 * Return Value : none
145 * Calling Functions :
146 *****/
147 void cmt0_start(T_CMT_SETCMCOR_INFO *i_pCmcorInfo, CMT_TIMEOUT_CALLBACK i_func)
148 {
149     T_CMT_SAVE_INFO ncmt_save_info;
150
151     memset(&ncmt_save_info, 0x00, sizeof(ncmt_save_info));
152
153     cmt0_get_g_CMT_SAVE_INFO(&ncmt_save_info);
154
155     /* ==== コンペアマッチタイマカウンタ (CMCNT0) の設定 ==== */
156     CMT0.CMCNT = 0x0000; /* タイマカウンタクリア */
157
158     CMT0.CMCSR.BIT.CMF = 0; /* コンペアマッチフラグのクリア */
159     CMT0.CMCSR.BIT.CMR = D_CMT_INTC_PERMIT; /* 内部割り込み要求を許可 */
160
161     /* タイムアウト時コールバック関数登録 */
162     ncmt_save_info.mfunc = i_func;
163
164     /* CMCOR 設定処理 */
165     cmt_set_CMCOR_Type_ms(i_pCmcorInfo);
166
167     /* CMT 保存情報に設定 */
168     cmt0_set_g_CMT_SAVE_INFO(&ncmt_save_info);
    
```

```

169
170     /* ==== コンペアマッチタイマスタートレジスタ (CMSTR) の設定 ==== */
171     CMT.CMSTR.BIT.STR0 = 1; /* カウント開始 */
172
173 }
174
175 /*"FUNC COMMENT"*****
176 * ID :
177 * Outline : CMT0 タイマ停止処理
178 * Include :
179 * Declaration : void cmt0_stop(void)
180 * Description : CMT0 タイマを停止する。
181 * :
182 * Argument : none
183 * Return Value : none
184 * Calling Functions :
185 *"FUNC COMMENT END"*****/
186 void cmt0_stop(void)
187 {
188     /* ==== コンペアマッチタイマスタートレジスタ (CMSTR) の設定 ==== */
189     CMT.CMSTR.BIT.STR0 = 0; /* カウント停止 */
190
191     CMT0.CMCSR.BIT.CMR = D_CMT_ALLSTOP; /* 内部割り込み禁止 */
192     CMT0.CMCSR.BIT.CMF = 0; /* コンペアマッチフラグのクリア*/
193
194     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
195     int_responstime_wait_for_cmt();
196
197 }
198
199 /*"FUNC COMMENT"*****
200 * ID :
201 * Outline : CMT0 タイマ削除処理
202 * Include :
203 * Declaration : void cmt0_delete(void)
204 * Description : CMT0 タイマを削除する。
205 * :
206 * Argument : none
207 * Return Value : none
208 * Calling Functions :
209 *"FUNC COMMENT END"*****/
210 void cmt0_delete(void)
211 {
212     T_CMT_SAVE_INFO ncmt_save_info;
213
214     /* CMT0 タイマ停止処理 */
215     cmt0_stop();
216
217     memset(&ncmt_save_info, 0x00, sizeof(ncmt_save_info));
218
219     /* CMT 保存情報の初期化 */
220     cmt0_set_g_CMT_SAVE_INFO(&ncmt_save_info);
221
222 }
223
224 /*"FUNC COMMENT"*****
225 * ID :
    
```



```

226 * Outline          : コンペアマッチ時処理
227 * Include          :
228 * Declaration      : void cmt0_compare_match(void)
229 * Description      : コンペアマッチ時にコールされる。
230 *                  : intprg.c ファイルの INT_CMT_CMTI() より
231 *                  : コールされる。
232 *                  :
233 * Argument          : none
234 * Return Value     : none
235 * Calling Functions :
236 * "FUNC COMMENT END"*****/
237 void cmt0_compare_match(void)
238 {
239     T_CMT_SAVE_INFO    ncmt_save_info;
240
241     memset(&ncmt_save_info, 0x00, sizeof(ncmt_save_info));
242
243     /* CMT 保存情報の取得 */
244     cmt0_get_g_CMT_SAVE_INFO(&ncmt_save_info);
245
246     /* タイムアウト時コールバック関数のコール */
247     if(ncmt_save_info.mfunc != NULL)
248     {
249         ncmt_save_info.mfunc();
250     }
251
252 }
253
254 /*"FUNC COMMENT"*****
255 * ID                :
256 * Outline           : CMCOR 設定処理(ms 版)
257 * Include           :
258 * Declaration       : void cmt_set_CMCOR_Type_ms
259 *                   : (T_CMT_SETCMCOR_INFO *i_pCmcorInfo)
260 * Description       : 引数(ms)情報より CMCOR 設定値を計算し
261 *                   : CMCOR に設定する。
262 *                   :
263 * Limitation        : 以下の条件を前提としていますので、
264 *                   : 条件に合わない場合は修正してください。
265 *                   : (1)本関数は引数(T_CMT_SETCMCOR_INFO)が
266 *                   :   ms の値のみ対応しています。
267 *                   : (2)本関数は P が 33.33MHz として対応しています。
268 *                   :   変更する場合は、#define D_CMT_P_FREQ
269 *                   :   変更してください。
270 *                   : (3)タイマ設定値は 1ms ~ 1000ms のみ対応して
271 *                   :   います。
272 *                   :   範囲外の場合は、1ms として扱います。
273 *                   :
274 * Argument          : T_CMT_SETCMCOR_INFO *i_pCmcorInfo:
275 *                   :   タイマ設定値 (ms)
276 * Return Value      : none
277 * Calling Functions :
278 * "FUNC COMMENT END"*****/
279 void cmt_set_CMCOR_Type_ms(T_CMT_SETCMCOR_INFO *i_pCmcorInfo)
280 {
281     unsigned long  Cmcor = 0;
282     unsigned long  timelclock = 0;
    
```

```

283
284     /* タイマ設定値が条件を満たさない場合は, 1 とする */
285     if(i_pCmcorInfo->mTimeValue <= 0 || i_pCmcorInfo->mTimeValue > 1000)
286     {
287         i_pCmcorInfo->mTimeValue = 1;
288     }
289
290     /* ms 単位入力値を ns 単位に変更 */
291     Cmcor = i_pCmcorInfo->mTimeValue * 1000000;
292
293     /* CMCNT の 1 カウント時間(ns 単位)を計算 */
294     /* P 1 クロック時間(33.33MHz の場合: 30ns) × P 分割値 */
295     /* P 分割値とは CMCSR_0.CKS で設定した P /n の n の値を示します */
296     timelclock = D_CMT_P_FREQ * cmt_get_ClockDivValue();
297
298     /* CMCOR 設定値計算 */
299     Cmcor = Cmcor / timelclock;
300
301     if(Cmcor > 0)
302     {
303         /* コンペアマッチ信号, 割り込み信号は Cmcor と CMCNT が一致した
304            次のカウンタ用クロックの立ち上がりで有効になるため,
305            CMCOR 設定値は, - 1 します。
306            詳細は, SH7730 ハードウェアマニュアルのコンペアマッチ
307            フラグのセットタイミングを参照してください。
308            */
309         Cmcor= Cmcor - 1;
310     }
311
312     /* ==== コンペアマッチタイマコンスタントレジスタ (CMCOR0) の設定 ==== */
313     CMT0.CMCOR = Cmcor;
314
315 }
316
317 /*"FUNC COMMENT"*****
318 * ID :
319 * Outline : P 分割値取得処理
320 * Include :
321 * Declaration : T_CMT_PClockDiv cmt_get_ClockDivValue(void)
322 * Description : クロックセレクト対応テーブルより,
323 * : P 分割値を取得する。
324 * :
325 * Argument : none
326 * Return Value : T_CMT_PClockDiv : P 分割値
327 * Calling Functions :
328 *"FUNC COMMENT END"*****/
329 T_CMT_PClockDiv cmt_get_ClockDivValue(void)
330 {
331     T_CMT_SAVE_INFO ncmt_save_info;
332     int i;
333     int nsize;
334
335     memset(&ncmt_save_info, 0x00, sizeof(ncmt_save_info));
336
337     cmt0_get_g_CMT_SAVE_INFO(&ncmt_save_info);
338
339     nsize = sizeof(gc_Clock_Match_tbl) / sizeof(T_CMT_CLOCK_SELECT);
    
```

```

340
341     for(i = 0; i < nsize; i++)
342     {
343         if(gc_Clock_Match_tbl[i].mClockSetting == ncmt_save_info.mClockSelect)
344         {
345             return gc_Clock_Match_tbl[i].mClockDivValue;
346         }
347     }
348 }
349
350 /* エラーケース */
351 return D_CMT_DIVValue_128;
352 }
353
354 /*"FUNC COMMENT"*****
355 * ID          :
356 * Outline     : CMT 保存情報設定処理
357 * Include     :
358 * Declaration : void cmt0_set_g_CMT_SAVE_INFO
359 *             : (T_CMT_SAVE_INFO *i_pCMT_SAVE_INFO)
360 * Description : CMT 保存情報を設定する。
361 *             :
362 * Argument    : T_CMT_SAVE_INFO *i_pCMT_SAVE_INFO
363 *             :             : CMT 保存情報
364 * Return Value : none
365 * Calling Functions :
366 *"FUNC COMMENT END"*****/
367 void cmt0_set_g_CMT_SAVE_INFO(T_CMT_SAVE_INFO *i_pCMT_SAVE_INFO)
368 {
369     memcpy(&g_CMT_SAVE_INFO, i_pCMT_SAVE_INFO, sizeof(T_CMT_SAVE_INFO));
370 }
371
372 /*"FUNC COMMENT"*****
373 * ID          :
374 * Outline     : CMT 保存情報取得処理
375 * Include     :
376 * Declaration : void cmt0_get_g_CMT_SAVE_INFO
377 *             : (T_CMT_SAVE_INFO *o_pCMT_SAVE_INFO)
378 * Description : CMT 保存情報を取得する。
379 *             :
380 * Argument    : T_CMT_SAVE_INFO *o_pCMT_SAVE_INFO
381 *             :             : CMT 保存情報
382 * Return Value : none
383 * Calling Functions :
384 *"FUNC COMMENT END"*****/
385 void cmt0_get_g_CMT_SAVE_INFO(T_CMT_SAVE_INFO *o_pCMT_SAVE_INFO)
386 {
387     memcpy(o_pCMT_SAVE_INFO, &g_CMT_SAVE_INFO, sizeof(g_CMT_SAVE_INFO));
388 }
389 }
390
391 /* End of File */
    
```

(3) サンプルプログラムリスト "cmt.h"

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name   : cmt.h
32 * Abstract    : SH7730 CMT 設定例 Sample Program
33 * Version     : Ver 1.00
34 * Device      : SH7730
35 * Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01.001)
36 *             : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS          : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description  : SH7730 CMT 設定例のサンプルプログラムです。
40 *             :
41 * Operation   :
42 * Limitation  :
43 *             :
44 *****/
45 * History     : 03.DEC.2008 Ver. 1.00 First Release
46 /*"FILE COMMENT END"*****
47 #include <stdio.h>
48
49 /* ==== マクロ定義 ==== */
50 #define D_CMT_P_FREQ 30 /* P 1 クロック ns 単位時間(33MHz の場合 : 30ns) */
51
52 /* コンペアマッチタイマカウンタサイズ */
53 typedef enum {
54

```

```

55     D_CMT_32BITCOUNTER = 0,      /* 32 ビットカウンタ動作 */
56     D_CMT_16BITCOUNTER      /* 16 ビットカウンタ動作 */
57
58     }T_CMT_CountSize;
59
60     /* コンペアマッチモード */
61     typedef enum {
62
63         D_CMT_ONESHOT = 0,      /* ワンショット動作 */
64         D_CMT_FREERUN          /* フリーラン動作 */
65
66     }T_CMT_MatchMode;
67
68     /* コンペアマッチリクエスト */
69     typedef enum {
70
71         D_CMT_ALLSTOP = 0,     /* DMA 転送要求および内部割込み要求の禁止 */
72         D_CMT_DMA_PERMIT,     /* DMA 転送要求を許可 */
73         D_CMT_INTC_PERMIT     /* 内部割込み要求を許可 */
74
75     }T_CMT_Request;
76
77     /* クロックセレクト */
78     typedef enum {
79
80         D_CMT_DIV_8 = 0,      /* Pφ/8 */
81         D_CMT_DIV_32,        /* Pφ/32 */
82         D_CMT_DIV_128        /* Pφ/128 */
83
84     }T_CMT_ClockSelect;
85
86     /* CMT_割り込み優先レベル */
87     typedef enum {
88
89         D_CMT_PRI_0 = 0,     /* 優先レベル 0(割り込みマスク) */
90         D_CMT_PRI_1,        /* 優先レベル 1 */
91         D_CMT_PRI_2,        /* 優先レベル 2 */
92         D_CMT_PRI_3,        /* 優先レベル 3 */
93         D_CMT_PRI_4,        /* 優先レベル 4 */
94         D_CMT_PRI_5,        /* 優先レベル 5 */
95         D_CMT_PRI_6,        /* 優先レベル 6 */
96         D_CMT_PRI_7,        /* 優先レベル 7 */
97         D_CMT_PRI_8,        /* 優先レベル 8 */
98         D_CMT_PRI_9,        /* 優先レベル 9 */
99         D_CMT_PRI_10,       /* 優先レベル 10 */
100        D_CMT_PRI_11,       /* 優先レベル 11 */
101        D_CMT_PRI_12,       /* 優先レベル 12 */
102        D_CMT_PRI_13,       /* 優先レベル 13 */
103        D_CMT_PRI_14,       /* 優先レベル 14 */
104        D_CMT_PRI_15,       /* 優先レベル 15 */
105
106    }T_CMT_IntcPri;
107
108    /* P 分割値 */
109    typedef enum {
110
111        D_CMT_DIVValue_8     = 8,      /* Pφ/8 */
    
```

```

112     D_CMT_DIVValue_32  = 32,      /* Pφ/32 */
113     D_CMT_DIVValue_128 = 128     /* Pφ/128 */
114
115 }T_CMT_PClockDiv;
116
117 /* タイムアウト時コールバック関数 */
118 typedef void (*CMT_TIMEOUT_CALLBACK)(void);
119
120 /* ==== 構造体定義 ==== */
121 /* タイマ生成用構造体 */
122 typedef struct {
123
124     T_CMT_CountSize      mCountSize;      /* コンペアマッチタイマカウンタサイズ */
125     T_CMT_ClockSelect    mClockSelect; /* クロックセレクト */
126     T_CMT_IntcPri       mIntcPri;       /* CMT_割り込み優先レベル */
127
128 } T_CMT_TimerTYPE;
129
130 /* CMCOR 設定情報構造体 */
131 typedef struct {
132
133     unsigned long      mTimeValue;      /* タイマ設定値 */
134
135 } T_CMT_SETCMCOR_INFO;
136
137 /* CMT 保存情報構造体 */
138 typedef struct {
139
140     T_CMT_ClockSelect    mClockSelect; /* クロックセレクト */
141     CMT_TIMEOUT_CALLBACK mfunc;      /* タイムアウト時コールバック関数 */
142
143 } T_CMT_SAVE_INFO;
144
145 /* クロックセレクト対応構造体 */
146 typedef struct {
147
148     T_CMT_ClockSelect    mClockSetting;
149     T_CMT_PClockDiv      mClockDivValue;
150
151 } T_CMT_CLOCK_SELECT;
152
153 /* ==== 関数宣言 ==== */
154 void cmt0_create(T_CMT_TimerTYPE *i_pTimerType);
155 void cmt0_start(T_CMT_SETCMCOR_INFO *i_pCmcorInfo, CMT_TIMEOUT_CALLBACK i_func);
156 void cmt0_stop(void);
157 void cmt0_delete(void);
158 void cmt0_compare_match(void);
    
```

- (4) サンプルプログラムリスト "intprg.c"
 CMT に関連する割り込み処理を実装しています。

```

1 void int_responstime_wait_for_cmt(void);
2
3 /*
4 以下の INTC_RESPONSEWAIT は、周辺モジュールの優先順位判定時間待ち用の値です。
5 int_responstime_wait() にパラメータ指定して使用ください。
6 int_responstime_wait() については、common.src を参照ください。
7
8 周辺モジュールの優先順位判定時間は、5Pcyc となります。
9 int_responstime_wait() では、Icyc を 3cyc 実行する処理を、指定されたパラメータの回数
10 実行することで wait 処理を行います。
11
12 int_responstime_wait() のパラメータ指定値計算
13 H'0E >= (1/Pcyc*5cyc)/(1/Icyc*3cyc)
14 念のため 1 回多く実行します。
15
16 注)
17 I =266.66MHz、P =33.33MHz 動作時の設定となっています。
18 これらを変更する場合はこちらの設定値も見直してください。
19 */
20 #define INTC_RESPONSEWAIT (0x0000000E)
21
22
23 ...途中省略...
24
25 /* H'F00 compare match timer interrupt */
26 void INT_CMT_CMTI(void)
27 {
28     /* チャンネル 0 の割り込みの場合 */
29     if(CMT0.CMCSR.BIT.CMF == 1)
30     {
31         /* コンペアマッチフラグのクリア*/
32         CMT0.CMCSR.BIT.CMF = 0;
33
34         /* コンペアーマッチ時コール処理 */
35         cmt0_compare_match();
36
37         /* 優先順位判定時間分待ち処理 */
38         int_responstime_wait_for_cmt();
39     }
40 }
41
42 ...途中省略...
43
44 void int_responstime_wait_for_cmt(void)
45 {
46     unsigned short dummy;
47
48     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
49     dummy = CMT0.CMCSR.WORD;
50     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間分待ち */
51 }
    
```

(5) サンプルプログラムリスト "vecttbl.src"

CMT 割り込み実行時の割り込み優先度を設定しています。

CMT 割り込みの優先度を 1 に設定しているため、CMT 割り込み中に新たな CMT 割り込みが発生しないように優先度に 1 を設定しています。

```

1    ...途中省略...
2
3    ;CMT
4        ;H'F00    CMT
5        .data.b    H'10
6
7    ...途中省略...
```


6. 参考ドキュメント

- ソフトウェアマニュアル
SH-4A ソフトウェアマニュアル (RJJ09B0090)
(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。)
- ハードウェアマニュアル
SH7730 グループ ハードウェアマニュアル (RJJ09B0339)
(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。)

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	2009.05.28	—	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事務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、海底中継用の機器・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ような機器・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。
 - 1 1. 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。
 - 1 2. 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。
 - 1 3. 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