
RX210、RX21A、RX220 グループ

R01AN1706JJ0100

Rev.1.00

CAC を使用した HOCO キャリブレーションの方法

2014.10.10

要旨

本アプリケーションノートでは、RX210、RX21A、RX220 グループのクロック周波数精度測定回路(以下、CAC)を使用して、高速オンチップオシレータ(以下、HOCO)の周波数を調整しながら動作をする方法について説明します。

対象デバイス

RX210、RX21A、RX220 グループ

注. RX210、RX220 グループの 48 ピン版は対象外です。

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様.....	3
2. 動作確認条件	4
3. 関連アプリケーションノート	4
4. ハードウェア説明	5
4.1 使用端子一覧	5
5. ソフトウェア説明	5
5.1 動作概要	5
5.2 ファイル構成	7
5.3 オプション設定メモリ	7
5.4 定数一覧	8
5.5 変数一覧	10
5.6 関数一覧	10
5.7 関数仕様	11
5.8 フローチャート.....	14
5.8.1 メイン処理	14
5.8.2 TMR の初期設定	15
5.8.3 サブクロックの発振設定	16
5.8.4 RTC の初期設定	17
5.8.5 RTC の周期割り込み.....	18
5.8.6 CMT0 によるウェイト	19
5.8.7 CAC の初期設定.....	20
5.8.8 CAC の測定完了割り込み	21
6. RX21A、RX220 グループ 初期設定例 アプリケーションノートとの組み合わせ方.....	22
7. サンプルコード.....	23
8. 参考ドキュメント	23

1. 仕様

HOCO は、周辺温度など外的要因により、工場出荷時の発振周波数から離れることがあり、周波数の誤差を補正するためにはキャリブレーションが必要です。この HOCO の周波数を調整するために高速オンチップオシレータトリミングレジスタ(以下、HOCOTRRn (n=0~3)レジスタ)があります。HOCOTRR0、HOCOTRR1、HOCOTRR2、HOCOTRR3 レジスタはそれぞれ、32 MHz、36.864 MHz、40 MHz、50 MHz の発振周波数に対応しています。HOCO 32MHz の場合には、一定周期で HOCOTRR0 レジスタの値を調整することで、HOCO の周波数の誤差を補正できます。

本アプリケーションノートでは、1 秒間隔で HOCO の発振周波数を CAC で測定し、得られた結果により HOCOTRR0 レジスタの値を調整します。また、TMO3 端子から HOCO を 4 分周した波形を出力します。

表 1.1に使用する周辺機能と用途を、図 1.1にブロック図を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
CAC	サブクロックを基準に HOCO の周波数を測定
RTC	1 秒の周期カウンタ
TMR	HOCO を 4 分周した波形を出力

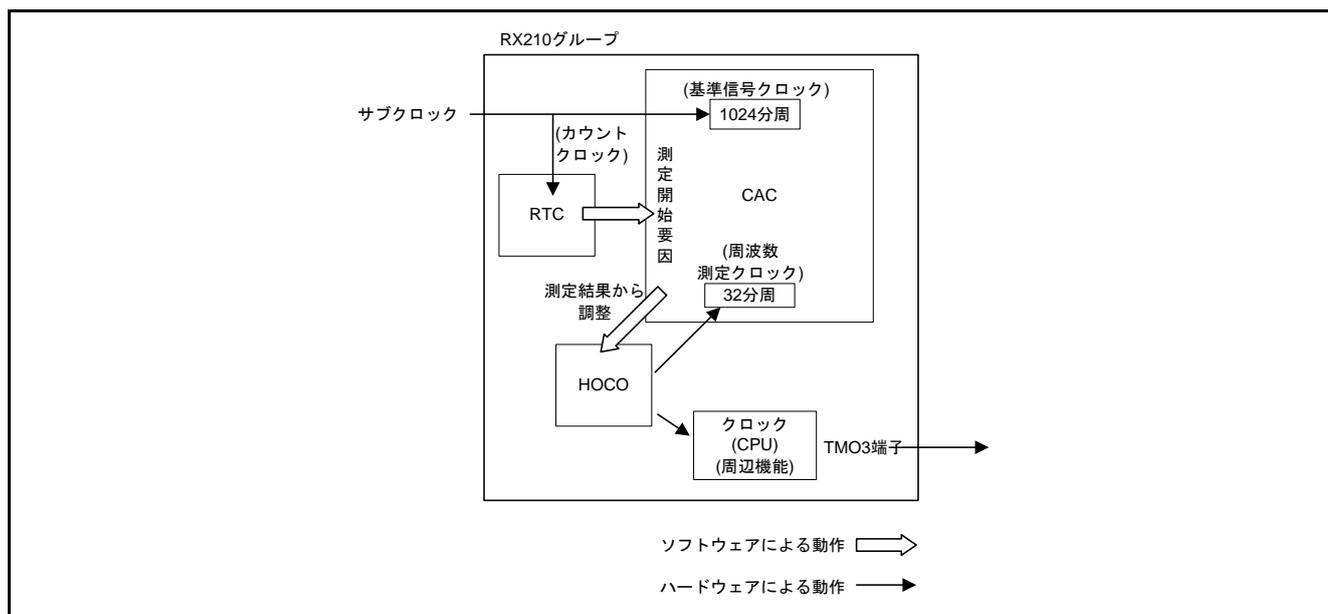


図1.1 ブロック図

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件

項目	内容
使用マイコン	R5F5210BBDFP(RX210 グループ)
動作周波数	<ul style="list-style-type: none"> ・ HOCO : 32MHz ・ サブクロック発振器(32.768kHz) ・ システムクロック(ICLK) : 32MHz(HOCO の 1 分周)
動作電圧	5.0V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.1.02 Release 01 コンパイルオプション -cpu=rx200 -output=obj="\$\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo (統合開発環境のデフォルト設定を使用しています)
iodefine.h のバージョン	Version 1.50
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit for RX210 (製品型名 : R0K505210S003BE)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

RX210 グループ 初期設定例 Rev.2.20 (R01AN1002JJ0220_RX210)
 RX21A グループ 初期設定例 Rev.1.10 (R01AN1486JJ0110_RX21A)
 RX220 グループ 初期設定例 Rev.1.10 (R01AN1494JJ0110_RX220)

上記アプリケーションノートの初期設定関数を、本アプリケーションノートのサンプルコードで使用しています。Rev は本アプリケーションノート作成時点のものです。

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

4. ハードウェア説明

4.1 使用端子一覧

表 4.1に使用端子と機能を示します。

表4.1 使用端子と機能

端子名	入出力	内容
P32/TMO3	出力	HOCO を 4 分周した波形を出力

5. ソフトウェア説明

5.1 動作概要

本アプリケーションノートでは、CAC の周波数測定クロックに HOCO(32MHz)の 32 分周を、基準信号生成クロックにサブクロック(32.768kHz)の 1024 分周を設定します。CAC は、サブクロックの立ち上がりエッジから次の立ち上がりエッジまでの期間に、HOCO の有効エッジをカウントします。

測定は、リアルタイムクロック(以下、RTC)の 1 秒周期割り込みで開始し、4 回連続で測定します。

測定した 4 回の平均値を測定結果とし、理論値との差分が±256 カウント以上であれば、HOCOTRR0 レジスタの値を加算または減算します。理論値との差分が±256 カウント未満であれば、HOCOTRR0 レジスタの値は変更しません。

表 5.1、図 5.1にレジスタ値の調整仕様、図 5.2にキャリブレーションのタイミング図を示します。

表5.1 レジスタ値の調整仕様

パターン	条件	レジスタ値の変更
1	$X \geq (Z+L)$	-1
2	$X \leq (Z-L)$	+1
3	$(Z-L) < X < (Z+L)$	変更しない

X: 測定結果、Z: 理論値、L: 許容範囲

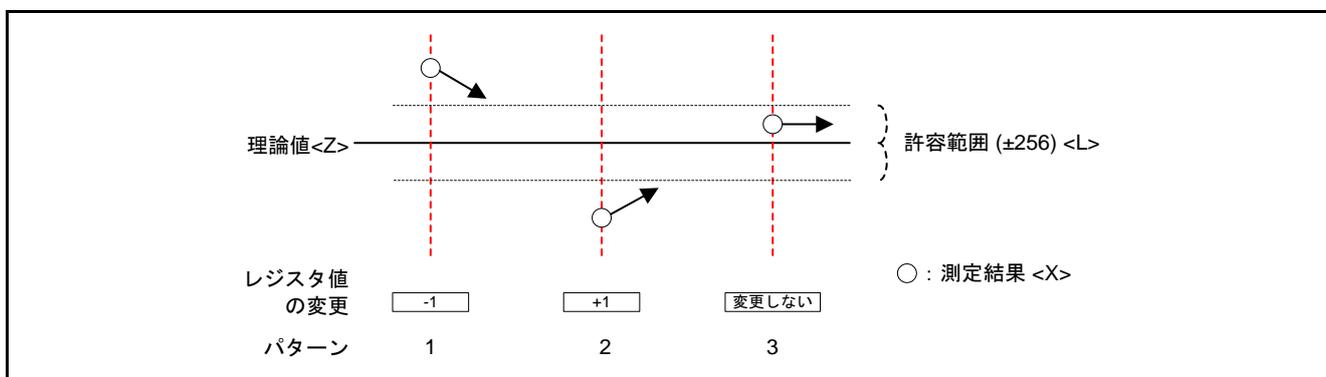


図5.1 レジスタ値の調整仕様

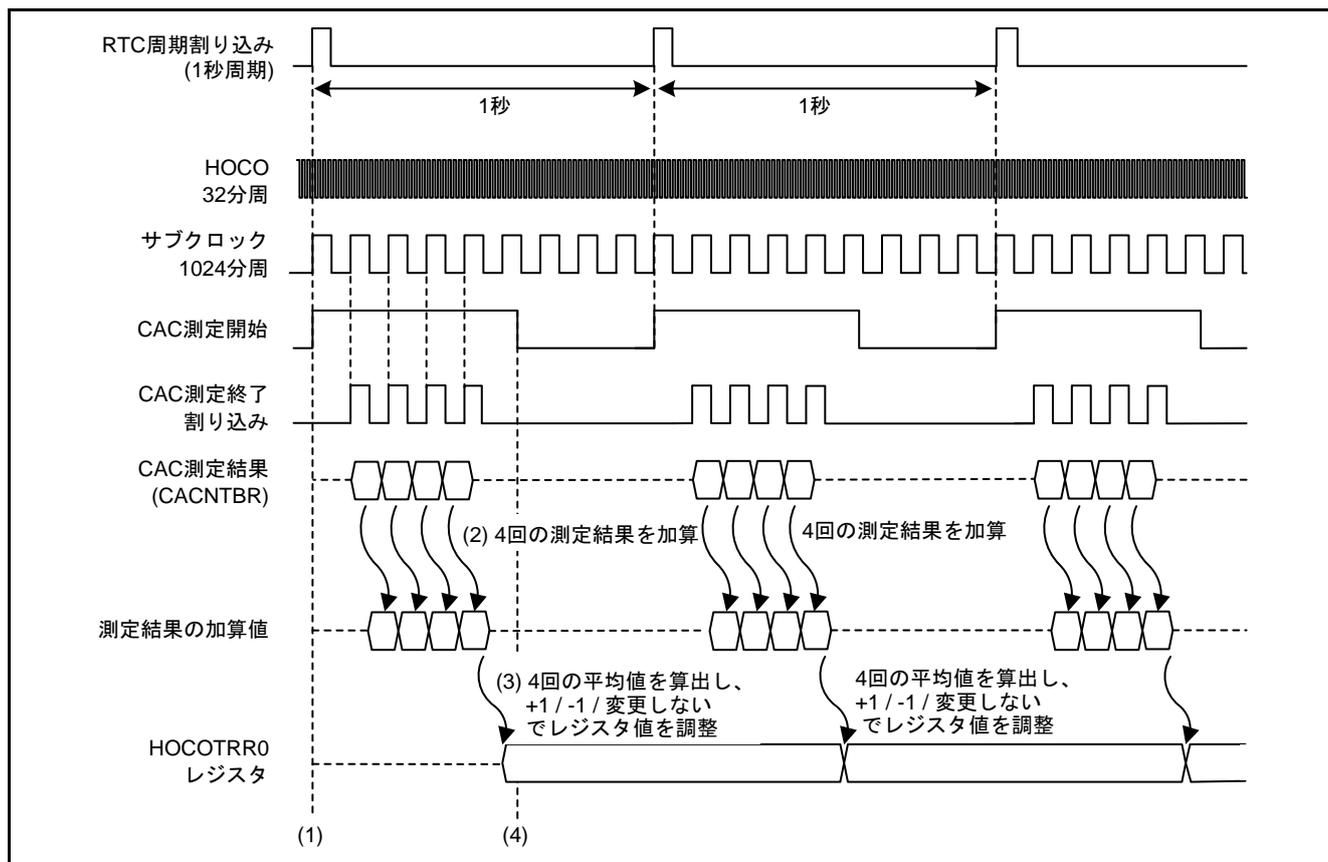


図5.2 キャリブレーションのタイミング図

- (1) RTC の 1 秒周期割り込みで CAC の起動設定と測定開始を行います。
- (2) サブクロックの 1024 分周の立ち上がりエッジから立ち上がりエッジまでの期間に、HOCO の 32 分周の有効エッジをカウントします。測定完了時の割り込みで、測定結果を取得します。測定結果は 4 回 1 セットで扱います。
- (3) 1 セット分の測定結果から平均値を算出し、その値(注)によって HOCOTRR0 レジスタの値を調整します。
- (4) HOCO の発振周波数の調整が終わると、CAC の測定を停止させます。次の RTC の 1 秒周期割り込みが発生するまで、ICLK など CPU や周辺機能に HOCO を供給しながら、HOCO の発振安定待ちをします。

注 HOCO と、サブクロックに使用されている水晶振動子の誤差、および理論値に対する許容範囲(±256 カウント)によって、以下の誤差が生じます。

- ・ HOCO 32MHz のとき、最大±0.9%
- ・ HOCO 36.864MHz のとき、最大±0.8%
- ・ HOCO 40MHz のとき、最大±0.7%
- ・ HOCO 50MHz のとき、最大±0.6%

5.2 ファイル構成

表 5.2 にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表5.2 サンプルコードで使用するファイル

ファイル名	概要	備考
main.c	メイン処理	
r_init_stop_module.c	リセット後に動作している周辺機能の停止	
r_init_stop_module.h	r_init_stop_module.c のヘッダファイル	
r_init_non_existent_port.c	存在しないポートの初期設定	
r_init_non_existent_port.h	r_init_non_existent_port.c のヘッダファイル	
r_init_clock_an1706.c	クロック初期設定	
r_init_clock.h	r_init_clock.c のヘッダファイル	
rtc_func.c	RTC の初期設定と 1 秒周期割り込み処理	
rtc_func.h	rtc_func.c のヘッダファイル	
cac_func.c	CAC の初期設定と測定完了割り込み処理	
cac_func.h	cac_func.c のヘッダファイル	
tmr_func.c	TMR の PCLK 出力初期設定	

5.3 オプション設定メモリ

表 5.3 にサンプルコードで使用するオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。

表5.3 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh~FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh~FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 リセット後、HOCO 発振が無効
MDES	FFFF FF83h~FFFF FF80h	FFFF FFFFh	リトルエンディアン

5.4 定数一覧

表 5.4、表 5.5、表 5.6にサンプルコードで使用する定数を示します。

表5.4 サンプルコードで使用する定数(初期設定例から変更した定数)

定数名	設定値	内容
SEL_MAIN	B_NOT_USE	メインクロックの発振/停止選択 使用しない (メインクロック停止)
SEL_SUB	B_USE	サブクロック使用選択(システムクロックとして使用) 使用する(注 1)
SEL_RTC	B_USE	サブクロック使用選択(RTC のカウントソースとして使用) 使用する
SEL_PLL	B_NOT_USE	PLL クロックの発振/停止選択 使用しない (PLL クロック停止)
SEL_HOCO	B_USE	HOCO クロックの発振/停止選択 使用する (HOCO クロック発振)
SEL_SYSCLK	CLK_HOCO	システムクロックのクロックソース選択 HOCO

注 1 :クロックの設定は、初期設定例の「表 1.3 クロックの設定例」にある No.4 を使用しています。ただし
“SEL_SUB” は、サブクロックを CAC に使用するため “使用する” を設定しています。

表5.5 サンプルコードで使用する定数(1/2)

定数名	設定値	内容
CHECK_CNT	4	測定回数
MEND_FINISH	0	周波数測定カウンタ : 測定停止
MEND_START	1	周波数測定カウンタ : 測定開始
MEND_CHECK_FINISH	CHECK_CNT	周波数測定カウンタ : 測定完了
ACCEPTABLE_RANGE	256	測定結果に対する許容範囲値
MAIN_CLOCK_CYCLE	(1,000,000,000L / MAIN_CLOCK_Hz)	メインクロックの周期(ns)
SUB_CLOCK_CYCLE	(1,000,000,000L / SUB_CLOCK_Hz)	サブクロックの周期(ns)
FOR_CMT0_TIME	232727	発振安定待機時間待ち用タイマ(CMT0)の カウント周期(ns)= 1/LOCO(137.5kHz) × 32 (LOCO=137.5kHz(max)、PCLKB の 32 分周)
REG_HOCOTRR (HOCO=32MHz 動作時)	SYSTEM.HOCOTRR0	HOCOTRR0 レジスタ
TYP_CAC_RESULT (HOCO=32MHz 動作時)	31250	HOCO を 32MHz で動作させた場合の測定理論値
REG_HOCOTRR (HOCO=36.864MHz 動作時)	SYSTEM.HOCOTRR1	HOCOTRR1 レジスタ
TYP_CAC_RESULT (HOCO=36.864MHz 動作時)	36000	HOCO を 36.864MHz で動作させた場合の測定理論値

表5.6 サンプルコードで使用する定数(2/2)

定数名	設定値	内容
REG_HOCOTRR (HOCO=40MHz 動作時)	SYSTEM.HOCOTRR2	HOCOTRR2 レジスタ
TYP_CAC_RESULT (HOCO=40MHz 動作時)	39062	HOCO を 40MHz で動作させた場合の測定理論値
REG_HOCOTRR (HOCO=50MHz 動作時)	SYSTEM.HOCOTRR3	HOCOTRR3 レジスタ
TYP_CAC_RESULT (HOCO=50MHz 動作時)	48828	HOCO を 50MHz で動作させた場合の測定理論値

5.5 変数一覧

表 5.7にグローバル変数を示します。

表5.7 グローバル変数

型	変数名	内容	使用関数
static uint8_t	buffer_counter	周波数測定カウンタ	INIT_CAC Excep_CAC_MENDF
static uint32_t	result_buffer	周波数測定結果格納バッファ	INIT_CAC Excep_CAC_MENDF
static int32_t	result_diff	結果の格納バッファ	INIT_CAC Excep_CAC_MENDF

5.6 関数一覧

表 5.8に関数を示します。

表5.8 関数

関数名	概要
main	メイン処理
R_INIT_StopModule	リセット後に動作している周辺機能の停止
R_INIT_NonExistentPort	存在しないポートの初期設定
R_INIT_Clock	クロック初期設定
tmr_init	TMR の初期設定
CGC_oscillation_sub_an1706	サブクロックの発振設定
enable_RTC_an1706	RTC の初期設定
Excep_RTC_PRD	RTC の周期割り込み処理
cmt0_wait_16bit	CMT0 によるウェイト処理
INIT_CAC	CAC の初期設定
Excep_CAC_MENDF	CAC の測定完了割り込み処理

5.7 関数仕様

サンプルコードの関数仕様を示します。

main	
概要	メイン処理
ヘッダ	なし
宣言	void main (void)
説明	リセット後に動作している周辺機能の停止設定関数、存在しないポートの初期設定関数、クロックの初期設定関数および TMR の初期設定を呼び出します。
引数	なし
リターン値	なし

R_INIT_StopModule	
概要	リセット後に動作している周辺機能の停止
ヘッダ	r_init_stop_module.h
宣言	void R_INIT_StopModule(void)
説明	モジュールストップ状態へ遷移する設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、モジュールストップ状態への遷移は行っていません。 本関数の詳細は、各グループのアプリケーションノート「初期設定例」を参照してください。

R_INIT_NonExistentPort	
概要	存在しないポートの初期設定
ヘッダ	r_init_non_existent_port.h
宣言	void R_INIT_NonExistentPort(void)
説明	144 ピン未満の製品に対して、存在しないポートの端子に対応するポート方向レジスタの初期設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、100 ピン版(PIN_SIZE=100)に設定しています。 本関数をコールした後に、存在しないポートを含む PDR、PODR レジスタへバイト単位で書き込む場合、存在しないポートの方向制御ビットには“1”、ポート出力データ格納ビットには“0”を設定してください。 本関数の詳細は、各グループのアプリケーションノート「初期設定例」を参照してください。

R_INIT_Clock	
概要	クロック初期設定
ヘッダ	r_init_clock.h
宣言	void R_INIT_Clock(void)
説明	クロックの初期設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、システムクロックを HOCO とし、サブクロックを RTC と CAC に使用する処理を選択しています。 本関数の詳細は、各グループのアプリケーションノート「初期設定例」を参照してください。

tmr_init	
概要	TMR の初期設定
ヘッダ	tmr_func.h
宣言	void tmr_init(void)
説明	TMR のチャンネル 3 に PCLK を 2 分周で出力する設定を行います。
引数	なし
リターン値	なし

CGC_oscillation_sub_an1706	
概要	サブクロックの発振設定
ヘッダ	r_init_clock.h
宣言	void CGC_oscillation_sub_an1706 (void)
説明	サブクロックをシステムクロック、RTC のカウントソースのどちらか、または両方に使用する場合の設定を行います。
引数	なし
リターン値	なし

enable_RTC_an1706	
概要	RTC の初期設定
ヘッダ	rtc_func.h
宣言	void enable_RTC_an1706 (void)
説明	RTC の初期設定(クロック供給の設定と RTC 設定、RTC ソフトウェアリセット)を行います。
引数	なし
リターン値	なし

Excep_RTC_PRD	
概要	RTC の周期割り込み処理
ヘッダ	rtc_func.h
宣言	static void Excep_RTC_PRD(void)
説明	RTC の周期割り込みで、CAC の初期設定処理を行います。
引数	なし
リターン値	なし

cmt0_wait_16bit	
概要	CMT0 によるウェイト処理
ヘッダ	なし
宣言	static void cmt0_wait_16bit(uint16_t cnt)
説明	サブクロック 6 クロック分の RTC クロック供給待ちを行う際に使用します。
引数	uint16_t cnt: 発振安定待機時間 cnt = 発振安定待機時間(ns)(注 1) ÷ FOR_CMT0_TIME (注 2)
リターン値	なし
備考	注1. 発振安定待機時間は発振子によって異なります。各グループのアプリケーションノート「初期設定例」の算出方法に基づいて設定してください。 注2. FOR_CMT0_TIME は LOCO = 137.5kHz(max)で算出します。実際の待ち時間は LOCO の周波数により異なります。

INIT_CAC	
概要	CAC の初期設定
ヘッダ	cac_func.h
宣言	void INIT_CAC(void)
説明	CAC の初期設定を行います。
引数	なし
リターン値	なし

Excep_CAC_MENDF	
概要	CAC の測定完了割り込み処理
ヘッダ	cac_func.h
宣言	static void Excep_CAC_MENDF(void)
説明	測定結果をもとに HOCO のキャリブレーションを行います。
引数	なし
リターン値	なし

5.8 フローチャート

5.8.1 メイン処理

図 5.3にメイン処理のフローチャートを示します。

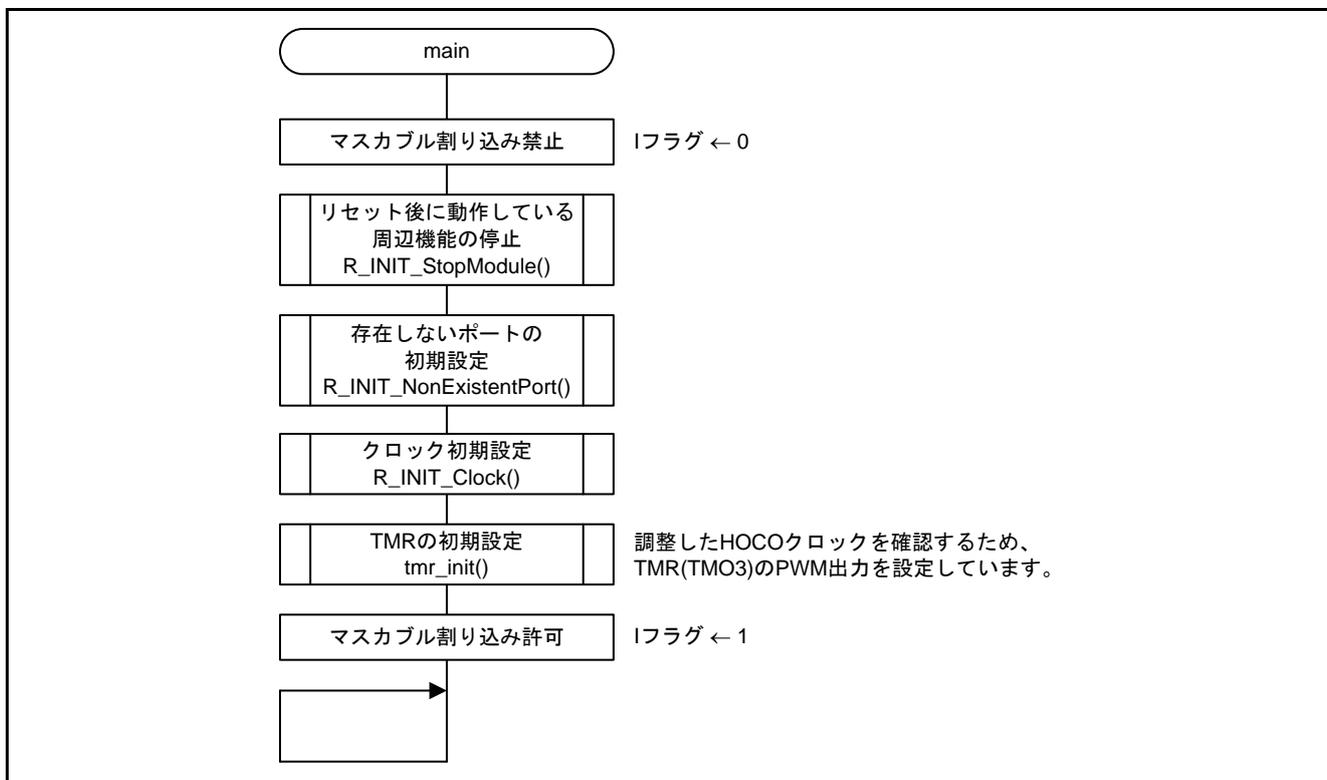


図5.3 メイン処理

5.8.2 TMR の初期設定

図 5.4にTMR の初期設定処理のフローチャートを示します。

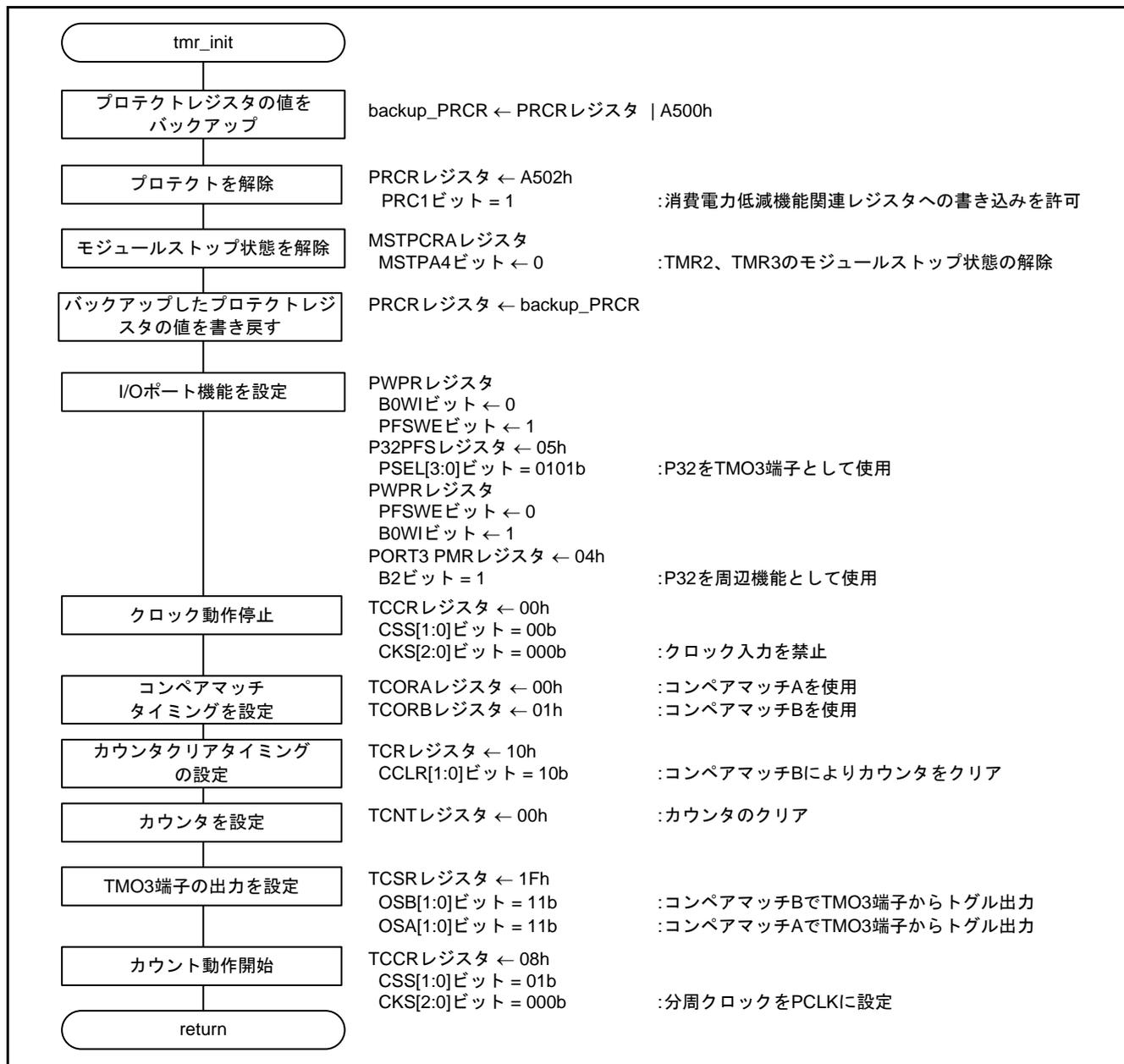


図5.4 TMR の初期設定処理

5.8.3 サブクロックの発振設定

図 5.5にサブクロックの発振設定処理のフローチャートを示します。

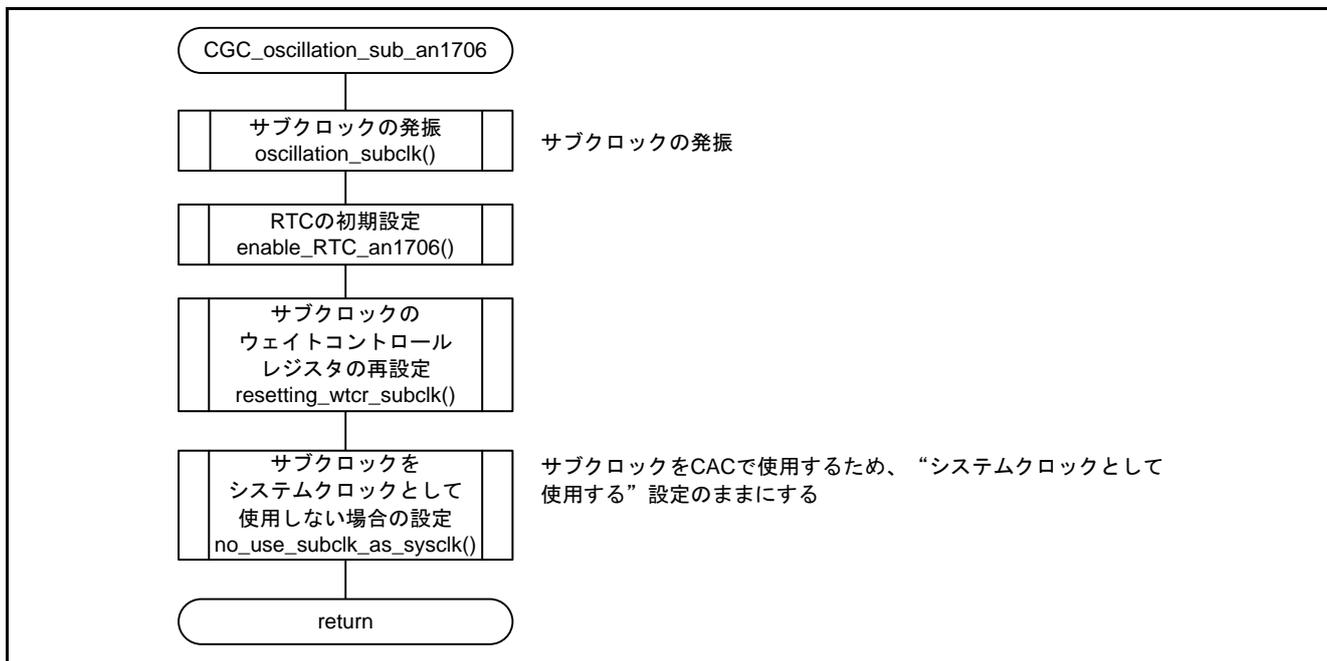


図5.5 サブクロックの発振設定処理

5.8.4 RTC の初期設定

図 5.6にRTC の初期設定処理のフローチャートを示します。

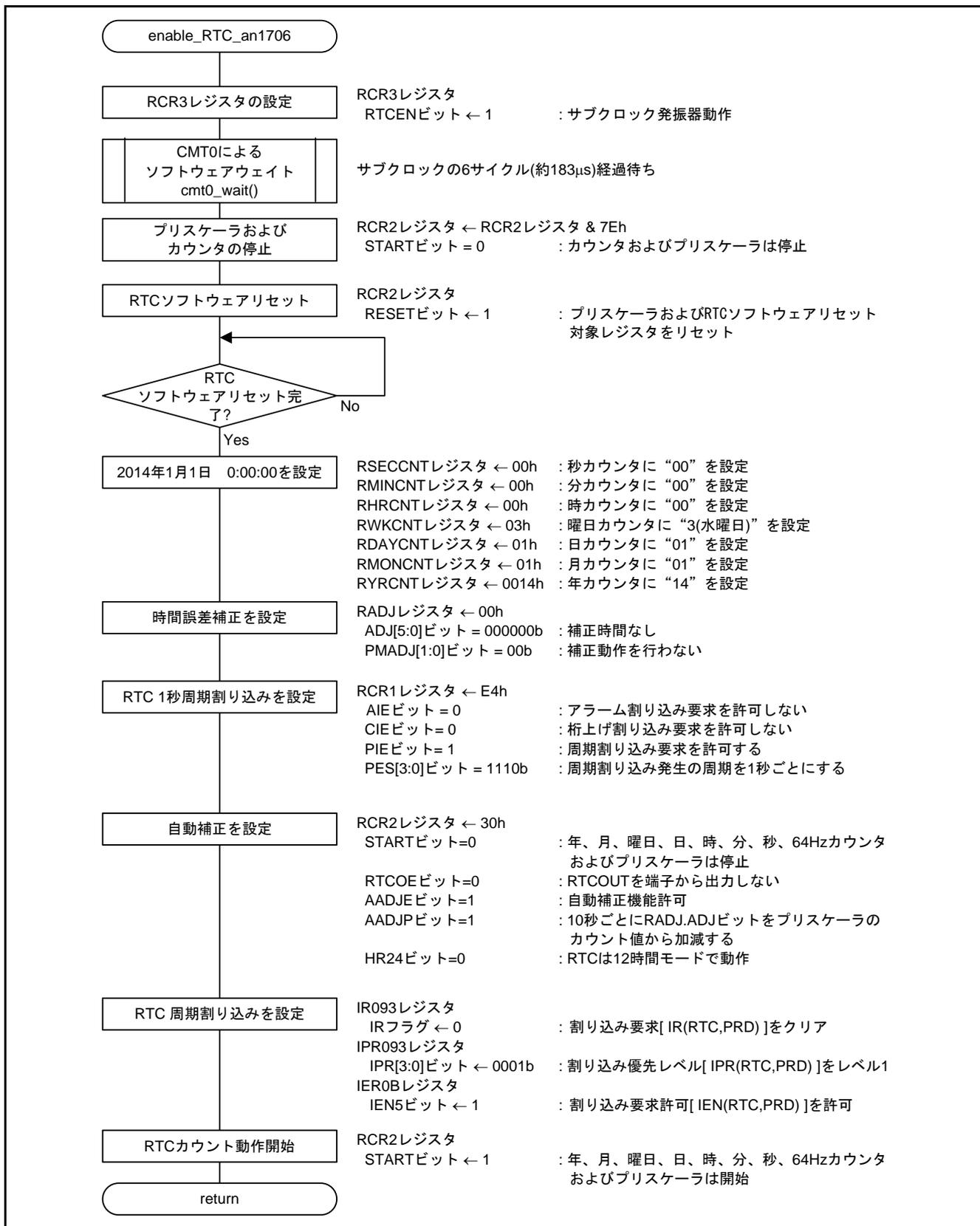


図5.6 RTC の初期設定処理

5.8.5 RTC の周期割り込み

図 5.7にRTC の周期割り込み処理のフローチャートを示します。

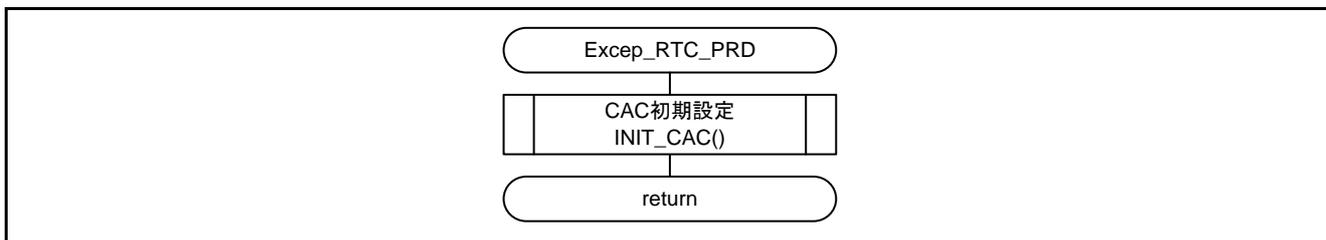


図5.7 RTC の周期割り込み処理

5.8.6 CMT0 によるウェイト

図 5.8にCMT0 によるウェイト処理のフローチャートを示します。

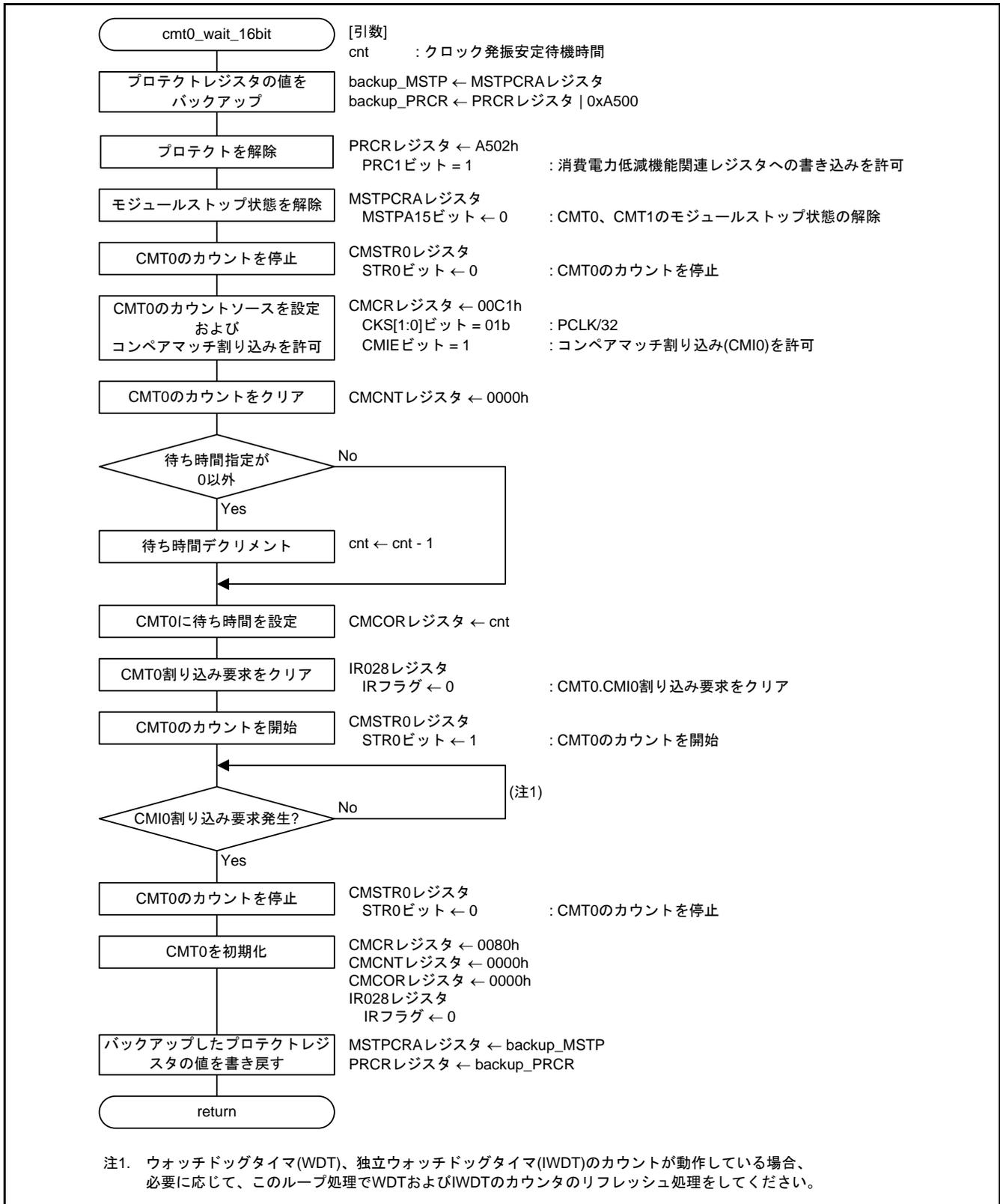


図5.8 CMT0 によるウェイト処理

5.8.7 CAC の初期設定

図 5.9にCAC の初期設定処理のフローチャートを示します。

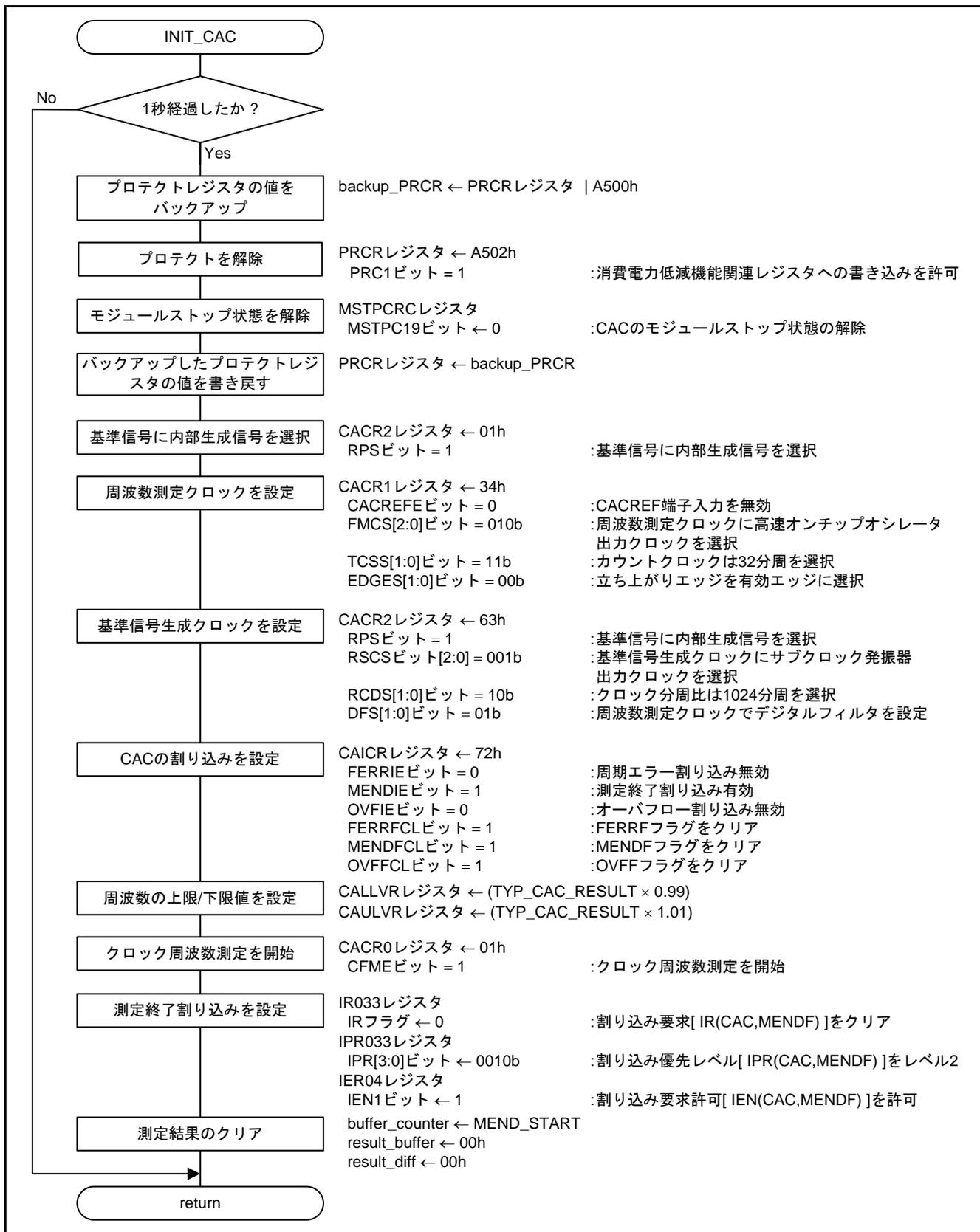


図5.9 CAC の初期設定処理

5.8.8 CAC の測定完了割り込み

図 5.10にCAC の測定完了割り込み処理のフローチャートを示します。

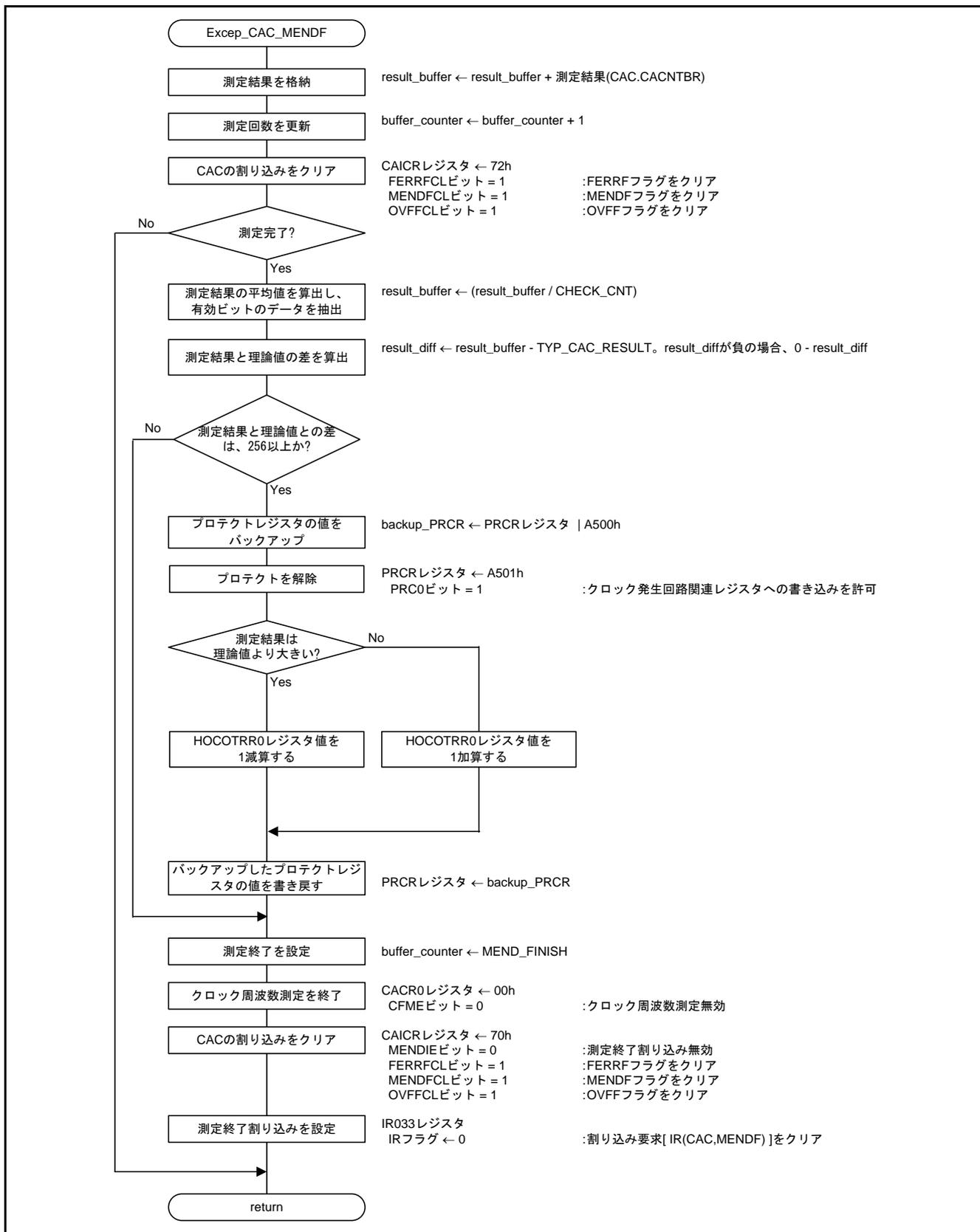


図5.10 CAC の測定完了割り込み処理

6. RX21A、RX220 グループ 初期設定例 アプリケーションノートとの組み合わせ方

本アプリケーションノートのサンプルコードは、RX210 グループで動作することを確認しています。RX21A グループや RX220 グループで動作させるには、それぞれの初期設定例のアプリケーションノートと組み合わせてください。

手順は、初期設定例のアプリケーションノート「5. RX210 グループのアプリケーションノートを RX21A グループに適用する方法」、「4. RX210 グループのアプリケーションノートを RX220 グループに適用する方法」を参照ください。

注：本アプリケーションノートでは `r_init_clock.h`、`r_init_clock.c` の設定を変更しています。

RX21A、RX220 グループの初期設定例からコピーした `r_init_clock.h` と `r_init_clock.c` は、本アプリケーションノートの設定と同じ設定にしてください。

7. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX210 グループ ユーザーズマニュアル ハードウェア編 Rev.1.50 (R01UH0037JJ)

RX21A グループ ユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0251JJ)

RX220 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10 (R01UH0292JJ)

(最新版をルネサス エレクトロニクス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 エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

RX ファミリー C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00 (R20UT0570JJ)

(最新版をルネサス エレクトロニクス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 エレクトロニクスホームページ

<http://japan.renesas.com>

お問い合わせ先

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

改訂記録	RX210、RX21A、RX220 グループ アプリケーションノート CAC を使用した HOCO キャリブレーションの方法
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.10.10	—	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。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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いません。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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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い合わせください。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。
9. 本資料に記載されている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・システムに使用することはできません。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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きを行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>