

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。

アプリケーション・ノート

78K0R/Kx3

サンプル・プログラム（周波数補正）

高速内蔵発振クロックのキャリブレーション編

この資料は、サンプル・プログラムの動作概要や使用方法、および高速内蔵発振器トリミング・レジスタ（HIOTRM）の設定方法や活用方法について説明したものです。サンプル・プログラムは、サブシステム・クロックを使用し、高速内蔵発振クロックの周波数自身を高速内蔵発振器トリミング・レジスタ（HIOTRM）の調整で、より8MHz近くに合わせこむプログラムです。約5秒毎にキャリブレーション開始処理を呼び出し、キャリブレーションを行います。

対象デバイス

78K0R/KE3マイクロコントローラ
 78K0R/KF3マイクロコントローラ
 78K0R/KG3マイクロコントローラ
 78K0R/KH3マイクロコントローラ
 78K0R/KJ3マイクロコントローラ

目次

第1章 概要 ...	3
第2章 回路図 ...	6
2.1 回路図 ...	6
2.2 周辺ハードウェア ...	6
第3章 ソフトウェアについて ...	7
3.1 ファイル構成 ...	7
3.2 使用する内蔵周辺機能 ...	8
3.3 使用する周辺の初期設定と動作概要 ...	9
3.4 フロー・チャート ...	10
第4章 設定方法について ...	12
4.1 高速内蔵発振器トリミング・レジスタの設定 ...	12
4.2 使用する周辺の初期設定 ...	14
4.3 メイン処理 ...	20
4.4 キャリブレーション開始処理 ...	22
4.5 INTTM02割り込み処理 ...	24
第5章 関連資料 ...	29
付録A プログラム・リスト ...	30
付録B 改版履歴 ...	50

- 本資料に記載されている内容は2008年12月現在のもので、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品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、一切その責を負いません。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。
- 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、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。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。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないように、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を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「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。

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

特別水準：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

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、生命維持のための装置またはシステム等

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。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社が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う。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

第1章 概 要

このサンプル・プログラムは、水晶振動子による周波数精度の良いサブシステム・クロックを使用することにより、高速内蔵発振クロックの周波数を高速内蔵発振器トリミング・レジスタ (HIOTRM) を調整し、より8MHz 近くに補正を行うプログラムです。キャリブレーション開始処理の呼び出しにより、タイマ・アレイ・ユニットのインターバル・タイマ・モードにてサブシステム・クロックを用い精度の良い7.8125ms(0.5s/2⁶)の時間を計測し、キャプチャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値が理想値の範囲内でないときは、高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行い、再度、キャリブレーションを行い、理想値の範囲内になるまで行います。

本サンプル・プログラムでは、約5秒毎にキャリブレーション開始処理を呼び出し、キャリブレーションを行います。

(1) 使用する周辺の初期設定の主な内容

使用する周辺の初期設定の主な内容は、次のとおりです。

割り込みの禁止

CPU / 周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定

タイマ・アレイ・ユニット0 (TAU0) の設定

- ・チャンネル0を高速内蔵発振クロックのカウント計測のためキャプチャ・モードに設定
- ・チャンネル1を約5秒毎にキャリブレーションを行うため、5.0s周期のインターバル・タイマ・モードに設定し、動作開始 (サブシステム・クロックの発振安定待ちも兼ねます。)
- ・チャンネル2をサブシステム・クロックを用い精度の良い7.8125ms(0.5s/2⁶)の時間計測を行うため、カウント・クロックをサブシステム・クロックの4分周に設定し、7.8125ms(0.5s/2⁶)周期のインターバル・タイマ・モードに設定

割り込みの許可

注意 本サンプル・プログラムで設定している時間、キャリブレーションの動作等は一例です。本サンプル・プログラムでは、フローの簡略化、理解の容易さを考慮し、固定周期でキャリブレーションを行うものとしています。キャリブレーションの必要性に応じて、キャリブレーション開始処理の呼び出しタイミングを調整してください。特に周期的に呼び出す必要はありませんが、固定周期で呼び出す場合は約5秒である必要はありません。

本サンプル・プログラムのシステム上、計測時間は、長いほど精度が高くなりますが、繰返して計測を行うため、短い時間のほうが望ましいです。精度とトレードオフし、計測時間は誤差の少ない7.8125ms(0.5s/2⁶)としています。

(2) メイン処理動作の内容

タイマ・アレイ・ユニット0 (TAU0) のチャンネル1によるインターバル・タイマ・モードにて、約5秒毎にキャリブレーション開始処理を呼び出します。

注意 このサンプル・プログラムでは約5秒毎にキャリブレーションを行っていますが、高速内蔵発振クロックの精度が必要な処理の前にキャリブレーションするなど、セットにおいて必要な精度を満たせる十分な評価のもと調整してください。

(3) キャリブレーション開始処理の内容

キャリブレーション開始処理の主な内容は、次のとおりです。

- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル2のインターバル動作開始
カウント・クロックを水晶振動子によるサブシステム・クロックの4分周に設定し、精度の良い7.8125ms(0.5s/2⁶)を計測します。
- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル0のキャプチャ動作開始
7.8125ms(0.5s/2⁶)間の高速内蔵発振クロックでのカウント数を計測します。

注意 タイマ・アレイ・ユニット0(TAU0)のチャンネル2のインターバル動作開始時の割り込みを許可し、最初の割り込みが発生した時点でチャンネル0のキャプチャ動作を開始することで、7.8125ms(0.5s/2⁶)間のカウント数を精度良く計測します。

(4) 割り込み処理動作の内容

キャリブレーション開始処理の呼び出しから7.8125ms(0.5s/2⁶)後に発生する割り込み処理です。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル0のキャプチャ・モードで計測したタイマ・カウント値から高速内蔵発振器トリミング・レジスタ (HIOTRM) の調整を行います。タイマ・カウント値が、高速内蔵発振クロック周波数を正確な8MHzとして計算した理想値の-0.16% ~ +0.16%の範囲内であれば、キャリブレーションを終了します。タイマ・カウント値が理想値の-0.16%以下のときは、速度が遅いということですので、高速内蔵発振器トリミング・レジスタ (HIOTRM) を増加させることで速くし、理想値の+0.16%より大きくなるときは速度が速いということですので、高速内蔵発振器トリミング・レジスタ (HIOTRM) を減少させることで遅くし、再度、キャリブレーションを行い、理想値の-0.16 ~ +0.16%になるまで繰り返します。

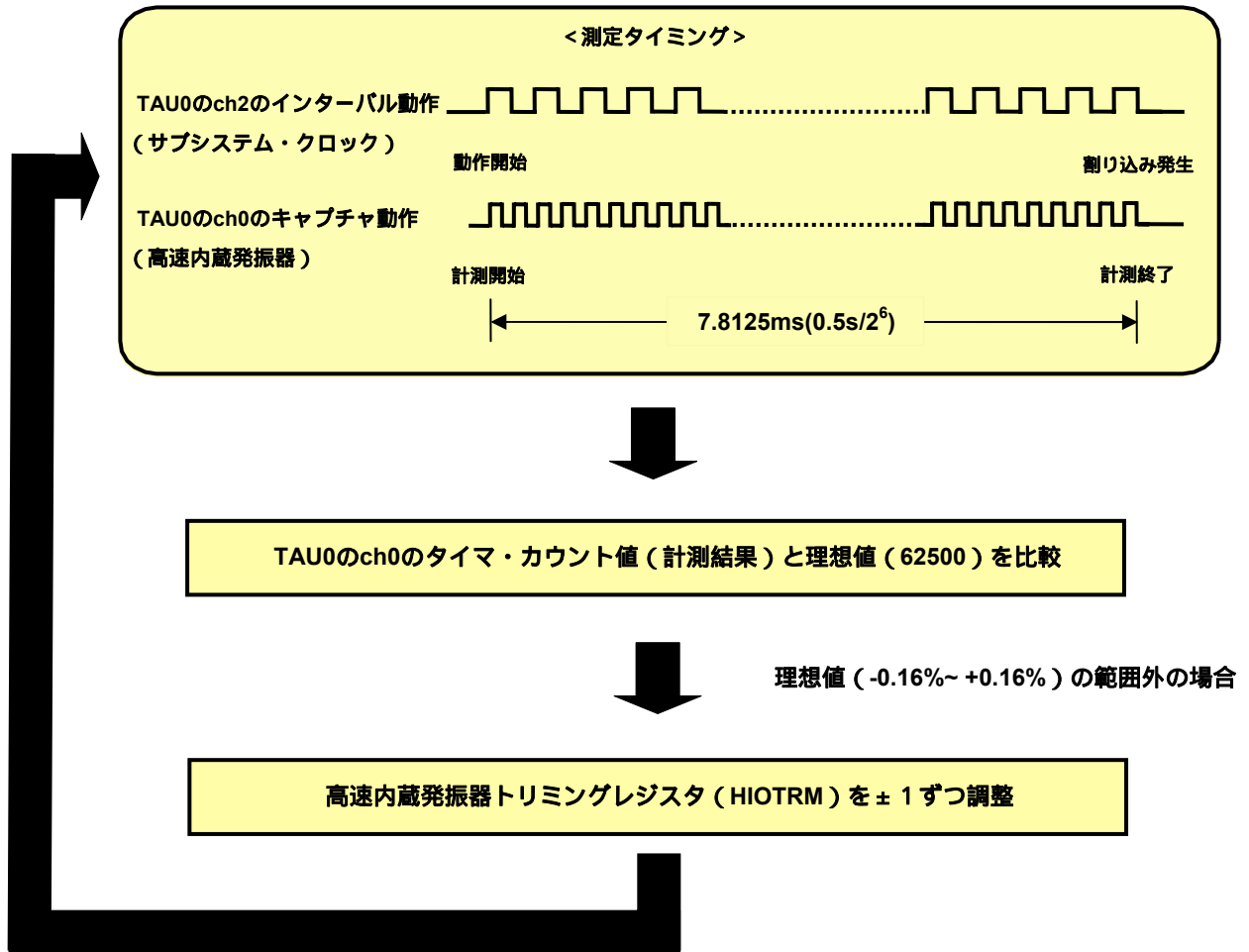
本サンプル・プログラムでは、フローの簡略化、理解の容易さを考慮し、高速内蔵発振器トリミング・レジスタ (HIOTRM) を±1ずつ変化させます。

注意 本サンプル・プログラムでは、キャリブレーションを繰り返す過程で、高速内蔵発振器トリミング・レジスタ (HIOTRM) の調整が遅くする-1 速くする+1 遅くする-1の値調整、もしくは速くする+1 遅くする-1 速くする+1の調整に遷移した場合は、理想値に近づいたと判断しキャリブレーションを終了します。また、高速内蔵発振器トリミング・レジスタ (HIOTRM) の値が最大値、最小値に到達した場合もキャリブレーションを終了します。

【補正方法】

● キャリブレーション

サブシステム・クロックによる割り込み発生までの、高速内蔵発振クロックカウント数を計測



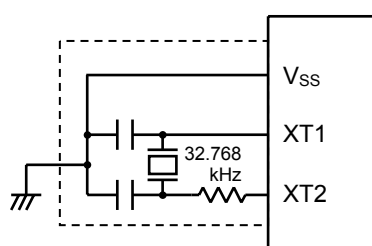
理想値 (-0.16%~ +0.16%) になるまで
再キャリブレーションを実行

第2章 回路図

2.1 回路図

回路図を次に示します。

XT1発振回路の外付け回路例（水晶発振）



注意 XT1発振回路およびXT2発振回路を使用する場合は、配線容量などの影響を避けるために、図の破線の部分を次のように配線してください。

- ・配線は極力短くする。
- ・他の信号線と交差させない。また、変化する大電流が流れる線と接近させない。
- ・発振回路のコンデンサの接地点は、常に V_{SS} と同電位となるようにする。大電流が流れるグランド・パターンに接地しない。
- ・発振回路から信号を取り出さない。

特に、XT1発振回路は、低消費電力にするために増幅度の低い回路になっていますのでご注意ください。

備考 発振子の選択および発振回路定数についてはお客様において発振評価していただくか、発振子メーカーに評価を依頼してください。

2.2 周辺ハードウェア

使用する周辺ハードウェアを次に示します。



- ・XT1, XT2に32.768 kHzの発振子を接続します。

第3章 ソフトウェアについて

この章では、ダウンロードする圧縮ファイルのファイル構成，使用するマイコンの内蔵周辺機能，サンプル・プログラムの使用する周辺の初期設定と動作概要，およびフロー・チャートを説明します。

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は，次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
main.asm (アセンブリ言語版) ----- main.c (C言語版)	マイコンのハードウェア初期化処理，メイン処理と割り込み処理のソース・ファイル	注	注
HIOTRM_SubClock.prw	統合開発環境 PM+用ワーク・スペース・ファイル		
HIOTRM_SubClock.prj	統合開発環境 PM+用プロジェクト・ファイル		

注. アセンブリ言語版には「main.asm」，C言語版には「main.c」が同封されています。

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+で使用するファイルを同封

3.2 使用する内蔵周辺機能

このサンプル・プログラムでは、マイコンに内蔵する次の周辺機能を使用します。

- ・高速内蔵発振クロックのカウント数の計測用：

タイマ・アレイ・ユニット0 (TAU0) のチャンネル0をキャプチャ・モードで使用

- ・約5秒周期でのキャリブレーション開始処理の呼び出し用：

タイマ・アレイ・ユニット0 (TAU0) のチャンネル1をインターバル・タイマ・モードで使用

- ・サブシステム・クロックでの7.8125ms($0.5s/2^6$)の時間計測用：

タイマ・アレイ・ユニット0 (TAU0) のチャンネル2をインターバル・タイマ・モードで使用

- ・高速内蔵発振器の精度補正：

高速内蔵発振器トリミング・レジスタ (HIOTRM)

3.3 使用する周辺の初期設定と動作概要

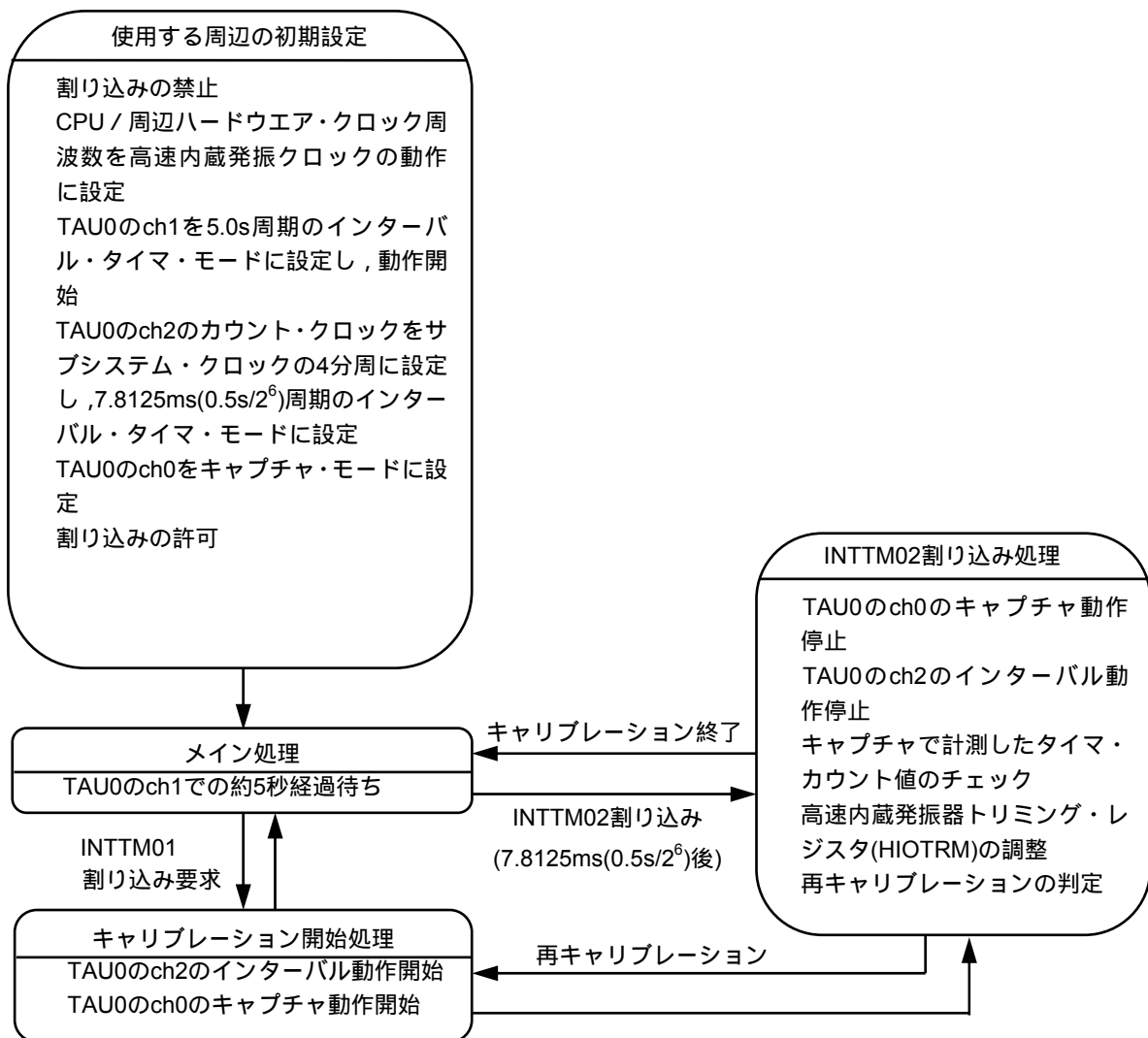
このサンプル・プログラムでは、使用する周辺の初期設定にて、クロック周波数の選択や、タイマ・アレイ・ユニットのインターバル・タイマ・モードおよびキャプチャ・モード設定などを行います。

使用する周辺の初期設定完了後、メイン処理にてタイマ・アレイ・ユニット0 (TAU0) のチャンネル1のインターバル・タイマ・モードにて約5秒毎にキャリブレーション開始処理を呼び出します。このサンプル・プログラムでは約5秒毎にキャリブレーションを行っていますが、高速内蔵発振クロックの精度が必要な処理の前にキャリブレーションするなど、セットにおいて必要な精度を満たせる十分な評価のもと調整してください。

キャリブレーション開始処理では、タイマ・アレイ・ユニット0 (TAU0) のチャンネル2のインターバル・タイマ・モードにてサブシステム・クロックを用い精度の良い7.8125ms(0.5s/2⁶)で、割り込みを発生させます。また、その7.8125ms(0.5s/2⁶)間の高速内蔵発振クロックのカウント数を、タイマ・アレイ・ユニット0 (TAU0) のチャンネル0のキャプチャ・モードにて計測します。

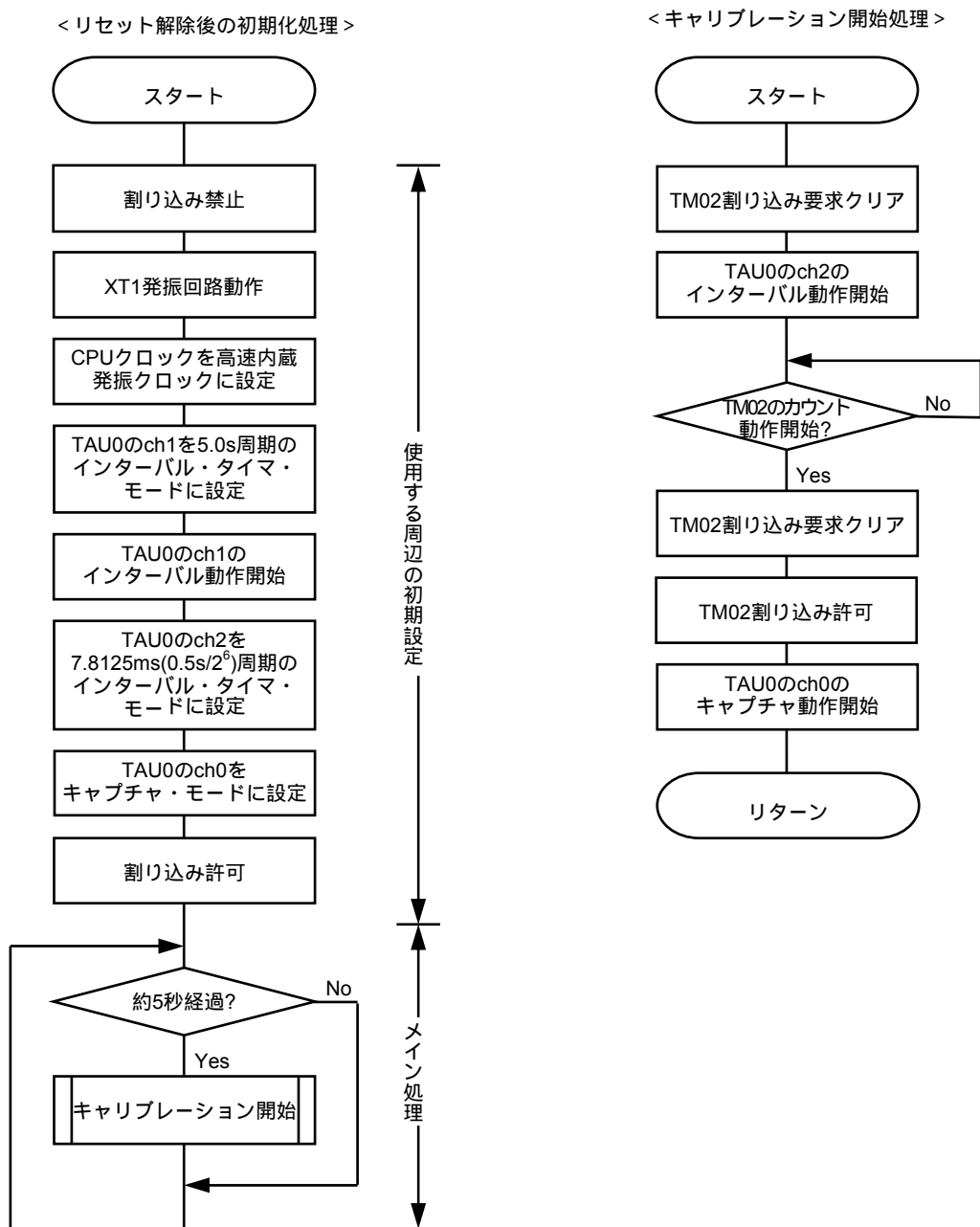
キャリブレーション開始処理で開始した7.8125ms(0.5s/2⁶)後の割り込み処理では、タイマ・アレイ・ユニット0 (TAU0) のチャンネル0のキャプチャ・モードで計測したカウント値をもとに高速内蔵発振器トリミング・レジスタ (HIOTRM) の調整を行うことで、高速内蔵発振クロックの周波数の補正を行います。

詳細については、次の状態遷移図 (ステート・チャート) に示します。



3.4 フロー・チャート

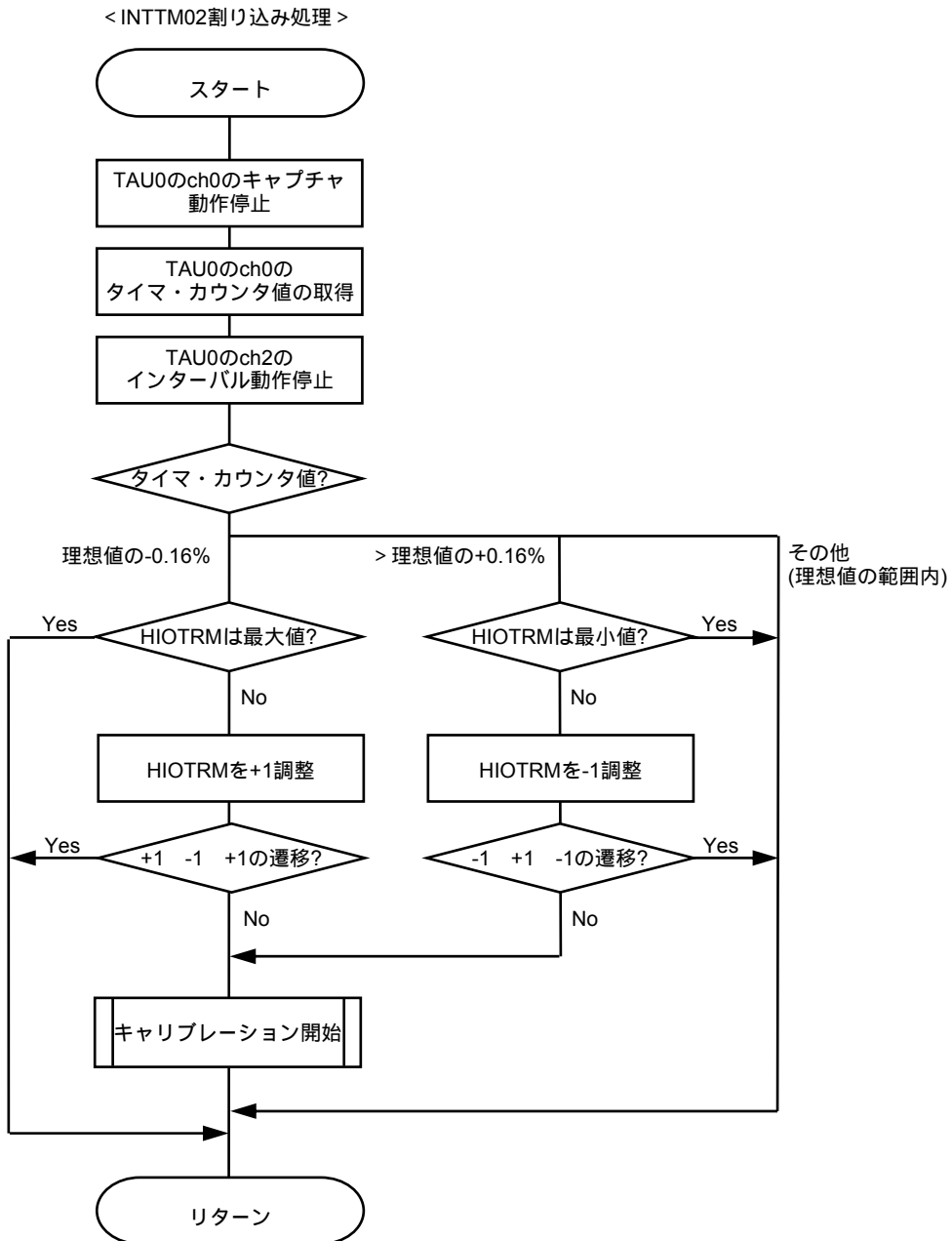
このサンプル・プログラムのフロー・チャートを次に示します。



注意 オプション・バイトは、RA78K0Rのリンカ・オプションにて設定してください。設定の仕方については、RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアルを参照してください。

オプション・バイトは、次の内容が設定されます。

- ・ウォッチドッグ・タイマの動作
- ・リセット解除時（電源立ち上げ時）のLVIの設定
- ・オンチップ・デバッグの動作制御



第4章 設定方法について

この章では、高速内蔵発振器トリミング・レジスタ（HIOTRM）の設定、初期設定、メイン処理、キャリブレーション開始処理、INTTM02割り込み処理について説明します。

初期設定方法の詳細については、78K0R/Kx3 サンプル・プログラム（初期設定） LED点灯のスイッチ制御編 アプリケーション・ノートを参照してください。

レジスタ設定方法の詳細については、各製品のユーザズ・マニュアル（78K0R/KE3、78K0R/KF3、78K0R/KG3、78K0R/KH3、78K0R/KJ3）を参照してください。

アセンブラ命令については、78K0Rマイクロコントローラ 命令編 ユーザズ・マニュアルを参照してください。

4.1 高速内蔵発振器トリミング・レジスタ（HIOTRM）の設定

高速内蔵発振器の精度補正をするレジスタです。

本サンプル・プログラムでは、水晶振動子による周波数精度の良いサブシステム・クロックを使用することにより、高速内蔵発振クロックの周波数を高速内蔵発振器トリミング・レジスタ（HIOTRM）を調整し、より8MHz近くに補正を行います。

高速内蔵発振器トリミング・レジスタ（HIOTRM）は、8ビット・メモリ操作命令で設定します。

リセット信号の発生により、10Hになります。

注意 精度補正後に温度、 V_{DD} 端子電圧に変化があった場合、周波数は変動します。また、高速内蔵発振器トリミング・レジスタ（HIOTRM）に初期値（10H）以外を設定した場合は、その後の温度、 V_{DD} 電圧変動、または高速内蔵発振器トリミング・レジスタ（HIOTRM）の設定値によって高速内蔵発振クロックの発振精度が $8\text{MHz} \pm 5\%$ を越える可能性があります。温度、 V_{DD} 電圧が変動する場合は、周波数の精度が必要になる前または定期的に補正を実行する必要があります。

本サンプル・プログラムでは、一例として約5秒毎に調整を行っています。

図4 - 1 - 1 高速内蔵発振器トリミング・レジスタ (HIOTRM) のフォーマット

アドレス : F00F2H

0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0
---	---	---	-------	-------	-------	-------	-------

クロック補正值 (ターゲット)

(2.7V V_{DD} 5.5V)

TTRM4-0	MIN.	TYP.	MAX.
0 0 0 0 0	- 5.54%	- 4.88%	- 4.02%
0 0 0 0 1	- 5.28%	- 4.62%	- 3.76%
0 0 0 1 0	- 4.99%	- 4.33%	- 3.47%
0 0 0 1 1	- 4.69%	- 4.03%	- 3.17%
0 0 1 0 0	- 4.39%	- 3.73%	- 2.87%
0 0 1 0 1	- 4.09%	- 3.43%	- 2.57%
0 0 1 1 0	- 3.79%	- 3.13%	- 2.27%
0 0 1 1 1	- 3.49%	- 2.83%	- 1.97%
0 1 0 0 0	- 3.19%	- 2.53%	- 1.67%
0 1 0 0 1	- 2.88%	- 2.22%	- 1.36%
0 1 0 1 0	- 2.23%	- 1.91%	- 1.31%
0 1 0 1 1	- 1.92%	- 1.60%	- 1.28%
0 1 1 0 0	- 1.60%	- 1.28%	- 0.96%
0 1 1 0 1	- 1.28%	- 0.96%	- 0.64%
0 1 1 1 0	- 0.96%	- 0.64%	- 0.32%
0 1 1 1 1	- 0.64%	- 0.32%	±0%
1 0 0 0 0	±0% (デフォルト)		
1 0 0 0 1	±0%	+ 0.32%	+ 0.64%
1 0 0 1 0	+ 0.33%	+ 0.65%	+ 0.97%
1 0 0 1 1	+ 0.66%	+ 0.98%	+ 1.30%
1 0 1 0 0	+ 0.99%	+ 1.31%	+ 1.63%
1 0 1 0 1	+ 1.32%	+ 1.64%	+ 1.96%
1 0 1 1 0	+ 1.38%	+ 1.98%	+ 2.30%
1 0 1 1 1	+ 1.46%	+ 2.32%	+ 2.98%
1 1 0 0 0	+ 1.80%	+ 2.66%	+ 3.32%
1 1 0 0 1	+ 2.14%	+ 3.00%	+ 3.66%
1 1 0 1 0	+ 2.48%	+ 3.34%	+ 4.00%
1 1 0 1 1	+ 2.83%	+ 3.69%	+ 4.35%
1 1 1 0 0	+ 3.18%	+ 4.04%	+ 4.70%
1 1 1 0 1	+ 3.53%	+ 4.39%	+ 5.05%
1 1 1 1 1	+ 3.88%	+ 4.74%	+ 5.40%
1 1 1 1 1	+ 4.24%	+ 5.10%	+ 5.76%

注意 高速内蔵発振の周波数は、HIOTRMの値をある値より大きくすることにより速くなり、小さくすることにより遅くなります。大きくすることにより周波数が遅くなったり、小さくすることにより速くなるような逆転は起こりません。

4.2 使用する周辺の初期設定

アセンブリ言語の使用する周辺の初期設定では、次の動作を行います。

割り込みを禁止します。

レジスタバンクの設定を行います。

スタック・ポインタの設定を行います。

メイン・システム・クロックの設定 / サブシステム・クロックの発振開始等のクロック設定を行います。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル1を5.0s周期のインターバル・タイマ・モードに設定します。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル2を7.8125ms(0.5s/2⁶)周期のインターバル・タイマ・モードに設定します。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル0をキャプチャ・モードに設定します。

割り込みを許可します。

メインループ処理へ続きます。

```

;*****
;
;      使用する周辺の初期設定
;
;*****
XMAIN   CSEG      UNIT
RESET_START:

-----
DI                                     ; 割り込み禁止
;-----
;      レジスタバンク設定
;-----
SEL      RB0
;-----
;      スタック・ポインタの設定
;-----
MOVW     SP,      #LOWW STACKTOP      ; スタック・ポインタを設定
;-----
;      クロック周波数の設定
;-----
;      高速内蔵発振クロックで動作が行えるように設定します。
;-----
MOV      CMC,     #00010000B          ; クロック動作モード
; |||||+----- AMPH: 2MHz fMX 10MHz
; |||+----- <000>
; ||+----- OSCSELS: XT1発振モード
; |+----- <0>
; +----- EXCLK/OSCSEL: 入力ポート・モード

MOV      CSC,     #10000000B          ; クロック動作ステータス制御
; |||||+----- HIOSTOP: 高速内蔵発振回路動作
; ||+++++----- <00000>
; |+----- XTSTOP: XT1発振回路動作
; +----- MSTOP: X1発振回路停止

MOV      OSMC,    #00000000B          ; 動作スピード・モード
; |||||+----- FSEL: 10MHz以下の周波数で動作
; +----- <00000>

```



```

MOV     CKC,    #00001000B          ;クロック選択
;|||||++++----- MDIV2-0: CPU周辺ハードウェア・クロック (fCLK)=fIH
;|||||+----- <1>
;|||+----- MCM0: 高速内蔵発振クロック (fIH)
;||+----- <R>
;|+----- CSS: メイン・システム・クロック (fMAIN)=fCLK
;+----- <R>
;-----
;
; インターバル・タイマの設定
;-----
;
; タイマ・アレイ・ユニット0のチャンネル1を5.0s周期のインターバル・タイマ・
; モードに設定します。
;
; キャリブレーション処理を約5秒毎に呼び出すために使用します。
;-----
SET1    !TAU0EN                      ;TAU0使用許可
MOV     TPS0L,  #10100000B          ;タイマクロック選択
;|||||++++----- PRS003-000(CK00): fCLK (キャプチャ動作で使用)
;++++----- PRS013-010(CK01): fCLK/2^10
;                                           (インターバル動作で使用)
;-----
MOVW    AX,    #10000000000000000B;動作モード設定
;|||||||||||||++++- MD013-010: インターバル・タイマ・モード
;|||||||||||||+----- <00>
;|||||++----- CIS011-010: 未使用
;|||+----- STS012-010: ソフトウェア・トリガ・スタート
;||+----- MASTER01: 単体動作 (奇数チャンネル)
;|+----- CCS01: CKS01ビットで指定した動作クロックMCK
;|+----- <00>
;+----- CKS01: PRSで選択した動作クロックCK01(MCK)
MOVW    TMR01,  AX
MOVW    TDR01,  #39063-1           ;5.0sインターバル: (fCLK/2^10)*5.0(s)=39062.5
;-----
CLR1    TMIF01                      ;割り込み要求クリア
SET1    TMMK01                      ;割り込み禁止
;-----
SET1    TS0L.1                       ;TAU0チャンネル1のインターバル動作開始(トリガ動作)
;-----
;
; キャリブレーション用の設定
;-----
;
; 精度の良い7.8125ms (0.5s/2^6)を計測するため、TAU0のチャンネル2をサブシステ
; ム・クロックの4分周をカウントクロックとし、インターバル・タイマ・モード
; に設定します。
;
; また、その7.8125ms (0.5s/2^6)間の高速内蔵発振クロックでのカウント数を計測
; するため、チャンネル0をキャプチャモードに設定します。
;-----
;
; インターバル割り込みの設定
;-----
;
; 7.8125ms (0.5s/2^6)に一度のインターバル
;-----
SET1    TIS0.2                       ;チャンネル2で使用するタイマをサブシステム・クロックの
;                                       4分周とする (fXT/4)
;-----
MOVW    AX,    #00010000000000001B;動作モード設定
;|||||||||||||++++- MD023-020: インターバル・タイマ・モード
;|||||||||||||          (カウント開始時に割り込み発生)
;|||||||||||||+----- <00>
;|||||++----- CIS021-020: 未使用
;|||+----- STS022-020: ソフトウェア・トリガ・スタート
;||+----- MASTER02: 単体動作
;|+----- CCS02: カウントクロック=TI02端子からの入力信号の
;   有効エッジ
;|+----- <00>
;+----- CKS02: CK00(エッジ検出部に使用)

```

```

MOVW    TMR02,  AX
MOVW    TDR02,  #64-1          ;7.8125ms (0.5s/2^6) インターバル:
                                   ; (fXT/4) *7.8125 (ms)=64

CLR1    TMIF02          ;割り込み要求クリア
SET1    TMMK02         ;割り込み禁止 (後ほど許可)

;-----
; キャプチャの設定
;-----
; 7.8125ms (0.5s/2^6)間の高速内蔵発振クロ
; ックのカウンタ数の計測
;-----
;済 MOV    TPS0L,  #10100000B      ;タイマクロック選択 (チャンネル1で設定済み)
                                   ; |||++++----- PRS003-000 (CK00): fCLK (キャプチャ動作で使用)
                                   ;++++----- PRS013-010 (CK01): fCLK/2^10
                                   ;                                     (インターバル動作で使用)

MOVW    AX,            #0000000000000100B;動作モード設定
                                   ; ||| ||| ||| ||| |||++++ MD003-000: キャプチャ・モード, アップ・カウン
                                   ; ||| ||| ||| ||| |||++----- <00>
                                   ; ||| ||| ||| ||| |||++----- CIS001-000: 未使用
                                   ; ||| ||| ||| ||| |||++----- STS002-000: ソフトウェア・トリガ・スタート
                                   ; ||| ||| ||| ||| |||+----- MASTER00: 単体動作
                                   ; ||| ||| ||| ||| |||+----- CCS00: CKS00ビットで指定した動作クロックMCK
                                   ; ||| ||| ||| ||| |||+----- <00>
                                   ; ||| ||| ||| ||| |||+----- CKS00: PRSで選択した動作クロックCK00 (MCK)

MOVW    TMR00,  AX

CLR1    TMIF00          ;割り込み要求クリア
SET1    TMMK00         ;割り込み禁止

EI                                     ;割り込み許可

```

C言語の使用する周辺の初期設定も、アセンブリ言語と同様な動作を行います。

C言語では、hdwinit 関数を作成することにより、初期設定のタイミングを速くすることができます。

hdwinit 関数は、周辺装置 (sfr) の初期設定をする関数としてユーザが必要に応じて作成する関数です。

```

/*****
使用する周辺の初期設定

*****/
void hdwinit(void)
{
    DI(); /* 割り込み禁止 */
/-----
    クロック周波数の設定
/-----
    高速内蔵発振クロックで動作が行えるように設定します。
/-----*/
    CMC = 0b00010000; /* クロック動作モード */
    /* ||| ||| | +----- AMPH: 2MHz fMX 10MHz */
    /* ||| | +----- <000> */
    /* || | +----- OSCSELS: XT1発振モード */
    /* | | +----- <0> */
    /* ++----- EXCLK/OSCSEL: 入力ポート・モード */

    CSC = 0b10000000; /* クロック動作ステータス制御 */
    /* ||| ||| | +----- HIOSTOP: 高速内蔵発振回路動作 */
    /* | | +----- <00000> */
    /* +----- XTSTOP: XT1発振回路動作 */
    /* +----- MSTOP: X1発振回路動作 */

    OSMC = 0b00000000; /* 動作スピード・モード */
    /* ||| ||| | +----- FSEL: 10MHz以下の周波数で動作 */
    /* +----- <00000> */

    CKC = 0b00001000; /* クロック選択 */
    /* ||| ||| | +----- MDIV2-0: CPU周辺ハードウェア・クロック */
    /* ||| ||| | (fCLK)=fIH */
    /* ||| | +----- <1> */
    /* || | +----- MCM0: 高速内蔵発振クロック (fIH) */
    /* | | +----- <R> */
    /* +----- CSS: メイン・システム・クロック (fMAIN)=fCLK */
    /* +----- <R> */

/-----
    インターバル・タイマの設定
/-----
    タイマ・アレイ・ユニット0のチャンネル1を5.0s周期のインターバル・タイマ・
    モードに設定します。
    キャリブレーション処理を約5秒毎に呼び出すために使用します。
/-----*/
    TAU0EN= 1; /* TAU0使用許可 */
    TPSOL = 0b10100000; /* タイマクロック選択 */
    /* ||| | +----- PRS003-000 (CK00): fCLK (キャプチャ動作で使用) */
    /* +----- PRS013-010 (CK01): fCLK/2^10 */
    /* (インターバル動作で使用) */

```



```
TMR00 = 0b000000000000000100;    /* 動作モード設定 */
/* | | | | | | | | | | | | | | | | | | +++++----- MD003-000: キャプチャ・モード、アップ・カウント */
/* | | | | | | | | | | | | | | | | | | ++----- <00> */
/* | | | | | | | | | | | | | | | | | | ++----- CIS001-000: 未使用 */
/* | | | | | | | | | | | | | | | | | | +++----- STS002-000: ソフトウェア・トリガ・スタート */
/* | | | | | | | | | | | | | | | | | | +----- MASTER00: 単体動作 */
/* | | | | | | | | | | | | | | | | | | ++----- CCS00: CKS00ビットで指定した動作クロックMCK */
/* | | | | | | | | | | | | | | | | | | ++----- <00> */
/* | | | | | | | | | | | | | | | | | | ++----- CKS00: PRSで選択した動作クロックCK00 (MCK) */

TMIF00 = 0;                        /* 割り込み要求クリア */
TMMK00 = 1;                        /* 割り込み禁止 */

EI ();                             /* 割り込み許可 */

}
```

4.3 メイン処理

アセンブリ言語のメイン処理では、次の動作を行います。

割り込み発生周期の確認を行います。

割り込み要求をクリアします。

タイマ・アレイ・ユニット0 (TAU0) のインターバル動作を停止します。

キャリブレーション開始処理を呼び出します。

の確認により、一定 (5.0s) 周期での処理が行えます。割り込みが発生していない場合は、何も行いません。により、次のインターバル割り込みの発生 (5.0s周期後) を確認できるようにします。は、キャリブレーション開始処理の2度呼びを防ぎます。の処理呼び出しにより、キャリブレーションを開始します。

```

;*****
;
;      メイン処理
;
;*****
MAIN_LOOP:

;-----
; 約5秒毎の処理
;-----
; 約5秒毎にキャリブレーションを行います。
;-----
..... BF      TMIF01,  $LMAIN500      ;5.0s経過? No,
..... CLR1   TMIF01                  ;割り込み要求クリア
..... SET1   TT0L, 1                  ;TAU0チャンネル1のインターバル動作停止(トリガ動作)
..... CALL  !!SCALIBSTT              ;キャリブレーション開始
LMAIN500:

;-----
; 各種メイン処理
;-----

; ここで各種メイン処理があれば呼び出します。

BR      MAIN_LOOP                    ; MAIN_LOOPへ

```

C言語のメイン処理も、アセンブリ言語と同様な動作を行います。

```

/*****
    メイン処理
*****/
void main(void)
{
    while (1){
        /*-----
        約5秒毎の処理
        -----
        約5秒毎にキャリブレーションを行います。
        -----*/
        if(TMIF01){
            /* 5.0s経過 */
            TMIF01 = 0; /* 割り込み要求クリア */
            TTOL.1 = 1; /* TAU0チャンネル1のインターバル動作停止(トリガ動作) */
            fn_CalibrationStart(); /* キャリブレーション開始 */
        }

        /*-----
        各種メイン処理
        -----*/

        /* ここで各種メイン処理があれば呼び出します。 */
    }
}

```

4.4 キャリブレーション開始処理

アセンブリ言語のキャリブレーション開始処理では、次の動作を行います。

キャリブレーションの回数を初期化します。

(速くする調整+1 / 遅くする調整-1の遷移を調べるために使用します。)

タイマ・アレイ・ユニット0 (TAU0) のチャンネル2を7.8125ms(0.5s/2⁶)周期のインターバル割り込みとして動作を開始します。

1周期目のインターバル動作は、カウント開始まで最大でカウントクロックの1クロック分の遅れが生じるので、精度良く7.8125ms(0.5s/2⁶)を計測するためにカウント動作開始の割り込みを待ちます。

により、インターバル動作のより正確な開始時点から、タイマ・アレイ・ユニット0 (TAU0) のチャンネル0のキャプチャ動作も開始します。

```

;*****
;
;   キャリブレーション開始処理
;
;-----
;   TAU0のチャンネル2を7.8125ms(0.5s/2^6)周期のインターバル割り込みとして動作
;   を開始します。それと共に、その間の高速内蔵発振クロックでのカウント数を計
;   測するため、チャンネル0をキャプチャモードにて動作を開始します。
;*****
SCALIBSTT:
-----
MOV      RCALCNT, #0          ;キャリブレーションの回数の初期化
SRECALIBSTT:
;-----
;   インターバル割り込み動作開始
;-----
CLR1     TMIF02              ;割り込み要求クリア
SET1     TS0L.2              ;TAU0のチャンネル2のインターバル動作開始(トリガ動作)

;MD020(TMR02.0)の設定により、インターバル動作のカウント開始時の割り込みを発生させ、
;同時にキャプチャを開始することで、精度良く基準時間内のカウント数を計測します。
JSCALIB300:
-----
NOP      ;(割り込み要求の保留対策)
BF       TMIF02, $JSCALIB300 ;カウント開始? No,
;-----
;   キャプチャ開始
;-----
CLR1     TMIF02
CLR1     TMMK02              ;TAU0のチャンネル2のインターバル割り込み許可
SET1     TS0L.0              ;TAU0のチャンネル0のキャプチャ動作開始(トリガ動作)

RET

```


C言語のキャリブレーション開始処理も、アセンブリ言語と同様な動作を行います。

```

/*****
      キャリブレーション開始処理
-----
TAU0のチャンネル2を7.8125ms(0.5s/2^6)周期のインターバル割り込みとして動作
を開始します。それと共に、その間の高速内蔵発振クロックでのカウント数を計
測するため、チャンネル0をキャプチャモードにて動作を開始します。
*****/
static void fn_CalibrationStart(void)
{
    ucCalibrationCount = 0;          /* キャリブレーション回数の初期化 */
    fn_ReCalibrationStart();
}

static void fn_ReCalibrationStart(void)
{
    /*-----*/
    インターバル割り込み動作開始
    -----*/
    TMIF02 = 0;                      /* 割り込み要求クリア */
    TSOL.2 = 1;                      /* TAU0のチャンネル2のインターバル動作開始(トリガ動作) */

    /* MD020(TMR02.0)の設定により、インターバル動作のカウント開始時の割り込みを発生させ、
       同時にキャプチャを開始することで、精度良く基準時間内のカウント数を計測します。 */
    while(!TMIF02){}                /* カウント開始タイミングを待つ */

    /*-----*/
    キャプチャ開始
    -----*/
    TMIF02 = 0;
    TMMK02 = 0;                      /* TAU0のチャンネル2のインターバル割り込み許可 */
    TSOL.0 = 1;                      /* TAU0のチャンネル0のキャプチャ動作開始(トリガ動作) */
}

```

4.5 INTTM02割り込み処理

アセンブリ言語のINTTM02割り込み処理では、次の動作を行います。

割り込み処理で汎用レジスタを使用するため、レジスタバンクを切り替えます。

タイマ・アレイ・ユニット0 (TAU0) チャンネル0のキャプチャ動作を停止します。

7.8125ms(0.5s/2⁶)間的高速内蔵発振クロックのタイマ・カウント値を取得します。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル2のインターバル動作を停止します。

キャリブレーション回数を更新します。

(速くする調整+1 / 遅くする調整-1の遷移を調べるために使用します。)

タイマ・カウント値が理想値の-0.16 ~ +0.16%の範囲であるかのチェックを行います。

範囲内であれば 以下の処理は行いません。

タイマ・カウント値が理想値の+0.16%より大きいとき、つまり速度が速いときの処理です。高速内蔵発振器トリミング・レジスタ(HIOTRM)を-1に調整することで遅くし、再度、キャリブレーションを行うように分岐します。既に一番遅い設定となっている場合は、キャリブレーションを終了します。

速くする調整+1 / 遅くする調整-1の遷移の状態情報を残すため、今回の遷移状態の"遅くする調整-1"の情報を残します。直近の3回の遷移が遅くする調整-1 速くする調整+1 遅くする調整-1と遷移している場合は、理想値8MHzに近づいたと判断し、キャリブレーションを終了します。

タイマ・カウント値が理想値の-0.16%以下のとき、つまり速度が遅いときの処理です。高速内蔵発振器トリミング・レジスタ(HIOTRM)を+1に調整することで速くし、再度、キャリブレーションを行うように分岐します。既に一番速い設定となっている場合は、キャリブレーションを終了します。

速くする調整+1 / 遅くする調整-1の遷移の状態情報を残すため、今回の遷移状態の"速くする調整-1"の情報を残します。直近の3回の遷移が速くする調整-1 遅くする調整+1 速くする調整-1と遷移している場合は、理想値8MHzに近づいたと判断し、キャリブレーションを終了します。

再度、キャリブレーションを行います。

キャリブレーション開始処理の呼び出し前で停止させたタイマ・アレイ・ユニット0 (TAU0) のインターバル動作を開始します。

```

;*****
;
;
;   INTTM02割り込み
;   (INTTM02使用, 7.8125ms (0.5s/2^6) に一度のインターバル割り込み)
;
;-----
;
;   キャリブレーション開始処理の呼び出しから7.8125ms (0.5s/2^6) 後に発生する
;   インターバル割り込み処理です。TAU0のチャンネル0のキャプチャ・モードで計測
;   した, 7.8125ms (0.5s/2^6) 間的高速内蔵発振クロックのタイマ・カウント値から
;   高速内蔵発振器トリミング・レジスタ (HIOTRM) の調整を行います。
;*****
INT_TM:
-----
SEL      RB1
-----
SET1     TT0L.0           ;TAU0のチャンネル0のキャプチャ動作停止(トリガ動作)
-----
MOVW     AX,      TCR00   ;7.8125ms (0.5s/2^6) 間的高速内蔵発振クロックでの
; タイマ・カウント値を取得
-----
SET1     TT0L.2           ;TAU0のチャンネル2のインターバル動作停止(トリガ動作)
SET1     TMMK02           ; 割り込み禁止
-----
INC      RCALCNT         ;キャリブレーション回数の更新
-----
;理想値(62500)の確認
;理想値の-0.16~+0.16%(62400~62600)の範囲であれば補正なし
CMPW     AX,      #62400   ;~-0.16%の範囲(遅い)?
BC       $HITM500         ; Yes, (補正あり)
BZ       $HITM500         ; Yes, (補正あり)
CMPW     AX,      #62600   ;-0.16 ~ +0.16%の範囲?
BZ       $HITMRET        ; Yes, (補正なし)
BC       $HITMRET        ; Yes, (補正なし)
; ;
BR       HITM300         ; No, (補正あり)
-----
;理想値の+0.16%より大きい範囲(速い)
; ;HITM300:
CMP      HIOTRM, #00000000B ;既に一番遅い設定?
BZ       $HITMRET        ; Yes, (補正不可)
DEC      HIOTRM          ;遅くする
CMP      RCALCNT, #1      ;キャリブレーションは初回?
BNZ     $HITM350         ; No,
MOV      RCALSTS, #00000000B ; Yes, (過去情報の初期化)
BR       HITM700
-----
HITM350:
MOV      A,      RCALSTS   ;キャリブレーションの過去情報を更新
CLR1     CY             ;HIOTRMの-1調整
ROL      A,      1
MOV      RCALSTS, A
AND      A,      #00000111B
CMP      A,      #0000010B ;-1 +1 -1の遷移?
BZ       $HITMRET        ; Yes, (補正終了)
BR       HITM700
-----

```

速くする調整+1 / 遅くする調整-1の遷移の状態情報の初期化を判断するために使用

キャリブレーションが初回なら、遷移情報を初期化

遷移情報を残すため、これまでの遷移情報を1ビットずらし、最下位ビットには遅くする調整-1を表す0を入れる

直近の3回のキャリブレーションが遅くする調整-1 速くする調整+1 遅くする調整-1の遷移状態となっているか?

```

;理想値の-0.16%以下の範囲(遅い)
HITM500:
  CMP    HIOTRM, #00011111B    ;既に一番速い設定?
  BZ     $HITMRET              ; Yes, (補正不可)
  INC    HIOTRM                ;速くする
  CMP    RCALCNT, #1           ;キャリブレーションは初回?
  BNZ    $HITM550              ; No,
  MOV    RCALSTS, #11111111B   ; Yes, (過去情報の初期化)
  BR     HITM700

HITM550:
  MOV    A,    RCALSTS          ;キャリブレーションの過去情報を更新
  SET1   CY,   HIOTRMの+1調整
  ROLC  A,    1
  MOV    RCALSTS, A
  AND   A,    #00000111B
  CMP   A,    #00000101B      ;+1 -1 +1の遷移?
  BZ     $HITMRET              ; Yes, (補正終了)

HITM700:
;再キャリブレーション開始
CALL    !!SRECALIBSTT
BR      HITMRETI

HITMRET:
CLR1    TMIF01                ;割り込み要求クリア
SET1    TS0L.1                ;TAU0チャンネル1のインターバル動作開始(トリガ動作)

HITMRETI:
RETI

```

キャリブレーションが初回
なら、遷移情報を初期化

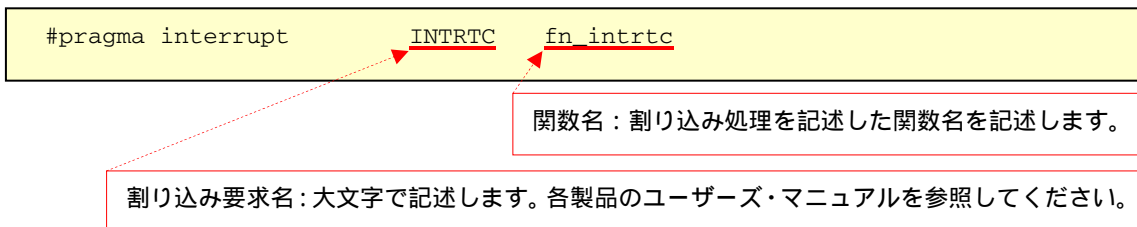
遷移情報を残すため、これ
までの遷移情報を1ビット
ずらし、最下位ビットには
速くする調整+1を表す1を
入れる

直近の3回のキャリブレーションが速くする調整+1
遅くする調整-1 速くする調整+1の遷移状態とな
っているか?

C言語での割り込み処理の設定

C言語での割り込み処理の設定は、`pragma`指令により、指定された割り込み要求名に対応する割り込みベクタ・テーブルに登録します。記述の仕方を以下に示します。

- ・`#pragma`指令による割り込みベクタ・テーブルへの登録（Cソースの先頭）



- ・割り込み関数修飾子によるハードウェア割り込み関数であることの宣言（Cソース部分）

```
__interrupt void fn_intrtc(void)
{
    ... (略) ...
}
```

詳細に関しては、ユーザーズ・マニュアル CC78K0R コンパイラ・パッケージ 言語編 を参照ください。

C言語の割り込み処理も、アセンブリ言語と同様な動作を行います。

```

/*****
INTTM02割り込み
(INTTM02使用、7.8125ms(0.5s/2^6)に一度のインターバル割り込み)

-----

キャリブレーション開始処理の呼び出しから7.8125ms(0.5s/2^6)後に発生する
インターバル割り込み処理です。TAU0のチャンネル0のキャプチャ・モードで計測
した、7.8125ms(0.5s/2^6)間的高速内蔵発振クロックのタイマ・カウント値から
高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行います。
*****/
__interrupt void fn_inttm02(void)
{
    register unsigned short ushnTimeCnt;          /* タイマ・カウント値 */

    TT0L.0= 1;                                     /* TAU0のチャンネル0のキャプチャ動作停止(トリガ動作) */

    ushnTimeCnt = TCR00;                           /* 7.8125ms(0.5s/2^6)間的高速内蔵発振クロックでの
                                                    タイマ・カウント値を取得 */

    TT0L.2 = 1;                                     /* TAU0のチャンネル2のインターバル動作停止(トリガ動作) */
    TMMK02 = 1;                                     /* 割り込み禁止 */

    ucCalibrationCountr ++;                        /* キャリブレーション回数の更新 */
}

```

```

    if( ushnTimeCnt <= 62400 ){
        /* 理想値の+0.16%より大きい範囲(遅い) 速くする */
        if( HIOTRM != 0b00011111 ){
            /* 既に一番速い設定? No, */
            HIOTRM ++; /* 速くする */
            if( ucCalibrationCounter == 1 ){
                ucCalibrationStatus = 0b11111111; /* +1調整時の初期値 */
                fn_ReCalibrationStart(); /* 再キャリブレーション開始 */
                return;
            }
            else{
                /* キャリブレーションの過去情報を更新 */
                ucCalibrationStatus =
                (ucCalibrationStatus<<1)|0b00000001; /* +1調整 */
                if((ucCalibrationStatus & 0b00000111 )!= 0b00000101){
                    /* +1 -1 +1の遷移? No, */
                    fn_ReCalibrationStart();
                    /* 再キャリブレーション開始 */
                    return;
                }
            }
        }
    }
}
else if( ushnTimeCnt > 62600 ){
    /* 理想値の-0.16%以下の範囲(速い) 遅くする */
    if( HIOTRM != 0b00000000 ){
        /* 既に一番遅い設定? No, */
        HIOTRM --; /* 遅くする */
        if( ucCalibrationCounter == 1 ){
            ucCalibrationStatus = 0b00000000; /* -1調整時の初期値 */
            fn_ReCalibrationStart(); /* 再キャリブレーション開始 */
            return;
        }
        else{
            /* キャリブレーションの過去情報を更新 */
            ucCalibrationStatus =
            (ucCalibrationStatus<<1)&0b11111110; /* -1調整 */
            if((ucCalibrationStatus & 0b00000111 )!= 0b00000010){
                /* -1 +1 -1の遷移? No, */
                fn_ReCalibrationStart();
                /* 再キャリブレーション開始 */
                return;
            }
        }
    }
}

TMIF01 = 0; /* 割り込み要求クリア */
TSOL.1 = 1; /* TAU0チャンネル1のインターバル動作開始(トリガ動作) */
}

```

第5章 関連資料

資料名		和文 / 英文
78K0R/KE3 ユーザーズ・マニュアル		PDF
78K0R/KF3 ユーザーズ・マニュアル		PDF
78K0R/KG3 ユーザーズ・マニュアル		PDF
78K0R/KH3 ユーザーズ・マニュアル		PDF
78K0R/KJ3 ユーザーズ・マニュアル		PDF
78K0Rマイクロコントローラ 命令編 ユーザーズ・マニュアル		PDF
RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
CC78K0R Cコンパイラ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0R/KG3マイクロコントローラのソース・プログラムを次に示します。

```
main.asm (アセンブリ言語版)
;*****
;
; NEC Electronics      78K0R/KG3マイクロコントローラ
;
;*****
; 78K0R/KG3マイクロコントローラ   サンプル・プログラム
;*****
; 周波数補正 (高速内蔵発振クロックのキャリブレーション)
;*****
; 【履歴】
; 2007.11.--      新規作成
;*****
;
; 【概要】
;
; このサンプル・プログラムは、水晶振動子による周波数精度の良いサブシステム・クロックを使用することにより、高速内蔵発振クロックの周波数を高速内蔵発振器トリミング・レジスタ (HIOTRM) を調整し、より8MHz近くに補正を行うプログラムです。キャリブレーション開始処理の呼び出しにより、タイマ・アレイ・ユニットのインターバル・タイマ・モードにてサブシステム・クロックを用い精度の良い7.8125ms(0.5s/26)の時間を計測し、キャプチャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値が理想値の範囲内にはないときは、高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行い、再度、キャリブレーションを行い、理想値の範囲内になるまで行います。
; 本サンプル・プログラムでは、約5秒毎にキャリブレーション開始処理を呼び出し、キャリブレーションを行います。
;
;
; <使用する周辺の初期設定の主な設定内容>
;
; ・ベクタ・テーブルの設定
; ・スタック・ポインタの設定
; ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
; ・5.0sインターバルに使用するTAU0のチャンネル1の設定
; ・7.8125ms(0.5s/2^6)インターバル割り込みに使用するTAU0のチャンネル2の設定
```



```

;   ・高速内蔵発振クロックのカウント計測のためキャプチャ動作に使用するTAU0のチャネ
;   ル0の設定
;
;
;
;   < キャリブレーション開始処理の設定内容 >
;
;   ・TAU0のチャネル2を7.8125ms(0.5s/2^6)インターバル割り込み動作開始
;   ・高速内蔵発振クロックのカウント計測のTAU0のチャネル0をキャプチャ開始
;
;
;   < INTTM02割り込み
;   (INTTM02使用, 7.8125ms(0.5s/2^6)に一度のインターバル割り込み) >
;
;   ・キャプチャ動作とインターバル割り込み動作の停止
;   ・計測したキャプチャ値を理想値の範囲であるかチェックを行う
;   ・理想値の範囲外であれば, 高速内蔵発振器トリミング・レジスタ(HIOTRM)の+1/-1の
;   調整を行い, 再キャリブレーション開始。ただし, 調整の変化が+1 -1 +1, もしく
;   は-1 +1 -1と変化した場合は調整終了とする。
;
;
;
;*****

```

```

;=====

```

```

;   ベクタ・テーブルの設定
;
;
;=====

```

```

TVCT1 CSEG   AT      000000H
      DW     RESET_START      ;(00)  RESET入力,POC,LVI,WDT,TRAP
TVCT2 CSEG   AT      000004H
      DW     RESET_START      ;(04)  INTWDTI
      DW     RESET_START      ;(06)  INTLVI
      DW     RESET_START      ;(08)  INTP0
      DW     RESET_START      ;(0A)  INTP1
      DW     RESET_START      ;(0C)  INTP2
      DW     RESET_START      ;(0E)  INTP3
      DW     RESET_START      ;(10)  INTP4
      DW     RESET_START      ;(12)  INTP5
      DW     RESET_START      ;(14)  INTST3
      DW     RESET_START      ;(16)  INTSR3
      DW     RESET_START      ;(18)  INTSRE3
      DW     RESET_START      ;(1A)  INTDMA0

```

```

DW    RESET_START                ;(1C)  INTDMA1
DW    RESET_START                ;(1E)  INTST0/INTCS100
DW    RESET_START                ;(20)  INTSR0/INTCS101
DW    RESET_START                ;(22)  INTSRE0
DW    RESET_START                ;(24)  INTST1/INTCS110/INTIIC1
DW    RESET_START                ;(26)  INTSR1
DW    RESET_START                ;(28)  INTSRE1
DW    RESET_START                ;(2A)  INTIIC0
DW    RESET_START                ;(2C)  INTTM00
DW    RESET_START                ;(2E)  INTTM01
DW    INT_TM                      ;(30)  INTTM02
DW    RESET_START                ;(32)  INTTM03
DW    RESET_START                ;(34)  INTAD
DW    RESET_START                ;(36)  INTRTC
DW    RESET_START                ;(38)  INTRTCI
DW    RESET_START                ;(3A)  INTKR
DW    RESET_START                ;(3C)  INTST2/INTCS120/INTIIC20
DW    RESET_START                ;(3E)  INTSR2
DW    RESET_START                ;(40)  INTSRE2
DW    RESET_START                ;(42)  INTTM04
DW    RESET_START                ;(44)  INTTM05
DW    RESET_START                ;(46)  INTTM06
DW    RESET_START                ;(48)  INTTM07
DW    RESET_START                ;(4A)  INTP6
DW    RESET_START                ;(4C)  INTP7
DW    RESET_START                ;(4E)  INTP8
DW    RESET_START                ;(50)  INTP9
DW    RESET_START                ;(52)  INTP10
DW    RESET_START                ;(54)  INTP11

TBRK  CSEG    AT      00007EH
      DW    RESET_START                ;(7E)  BRK

;=====
;
;   スタック領域の確保
;
;=====

DSTK  DSEG    AT      OFFEC0H
STACKEND:
      DS      20H                      ;スタック領域を32バイト確保
STACKTOP:                             ;スタック領域の先頭アドレス = FEFE0H

```

```

;=====
;
;   R A M
;
;=====
DMAIN DSEG   SADDR
RCALSTS:
    DS      1                ;キャリブレーションの過去情報
                                ; ビット毎にHIOTRMを+1方向に調整(1)/-1方向に調整(0)
                                ; bit7: 7回前の調整(最も古い過去の情報)
                                ; bit6: 6回前の調整
                                ;   :
                                ; bit1: 1回前の調整
                                ; bit0: 最新の調整(最も新しい現在の情報)
RCALCNT:
    DS      1                ;キャリブレーションの回数

;*****
;
;   使用する周辺の初期設定
;
;*****
XMAIN CSEG   UNIT
RESET_START:

    DI                ;割り込み禁止
;-----
;   レジスタバンク設定
;-----
    SEL      RBO
;-----
;   スタック・ポインタの設定
;-----
    MOVW    SP,      #LOWW STACKTOP ;スタック・ポインタを設定
;-----
;   クロック周波数の設定
;-----
;   高速内蔵発振クロックで動作が行えるように設定します。
;-----

```

```

MOV    CMC,    #00010000B          ;クロック動作モード
        ;|||||+----- AMPH: 2MHz fMX 10MHz
        ;|||+++----- <000>
        ;||+----- OSCSELS: XT1発振モード
        ;||+----- <0>
        ;++----- EXCLK/OSCSEL: 入力ポート・モード

MOV    CSC,    #10000000B          ;クロック動作ステータス制御
        ;|||||+----- HIOSTOP: 高速内蔵発振回路動作
        ;||++++----- <00000>
        ;|+----- XTSTOP: XT1発振回路動作
        ;+----- MSTOP: X1発振回路停止

MOV    OSMC,   #00000000B          ;動作スピード・モード
        ;|||||+----- FSEL: 10MHz以下の周波数で動作
        ;+++++----- <00000>

MOV    CKC,    #00001000B          ;クロック選択
        ;||||+++----- MDIV2-0: CPU周辺ハードウェア・クロック(fCLK)=fIH
        ;|||+----- <1>
        ;||+----- MCMO: 高速内蔵発振クロック(fIH)
        ;||+----- <R>
        ;|+----- CSS: メイン・システム・クロック(fMAIN)=fCLK
        ;+----- <R>

;-----
; インターバル・タイマの設定
;-----
; タイマ・アレイ・ユニット0のチャンネル1を5.0s周期のインターバル・タイマ・
; モードに設定します。
; キャリブレーション処理を約5秒毎に呼び出すために使用します。
;-----

SET1   !TAU0EN          ;TAU0使用許可
MOV    TPSOL, #10100000B          ;タイマクロック選択
        ;||||++++----- PRS003-000(CK00): fCLK (キャプチャ動作で使用)
        ;++++----- PRS013-010(CK01): fCLK/2^10(インターバル動作で使用)

```

```

MOVW  AX,      #1000000000000000B      ;動作モード設定
      ;|||||||||++++----- MD013-010: インターバル・タイマ・モード
      ;|||||||||++----- <00>
      ;|||||||++----- CIS011-010: 未使用
      ;|||||++++----- STS012-010: ソフトウェア・トリガ・スタート
      ;|||+----- MASTER01: 単体動作(奇数チャンネル)
      ;||+----- CCS01: CKS01ビットで指定した動作クロックMCK
      ;|++----- <00>
      ;+----- CKS01: PRSで選択した動作クロックCK01(MCK)

MOVW  TMR01,  AX
MOVW  TDR01,  #39063-1                  ;5.0sインターバル : (fCLK/2^10)*5.0(s)=39062.5

CLR1  TMIF01                      ;割り込み要求クリア
SET1  TMMK01                      ;割り込み禁止

SET1  TSOL.1                        ;TAU0チャンネル1のインターバル動作開始(トリガ動作)

;-----
;   キャリブレーション用の設定
;-----
;   精度の良い7.8125ms(0.5s/2^6)を計測するため、TAU0のチャンネル2をサブシステ
;   ム・クロックの4分周をカウントクロックとし、インターバル・タイマ・モード
;   に設定します。
;   また、その7.8125ms(0.5s/2^6)間的高速内蔵発振クロックでのカウント数を計測
;   するため、チャンネル0をキャプチャモードに設定します。
;-----
;-----
;   インターバル割り込みの設定
;-----
;   7.8125ms(0.5s/2^6)に一度のインターバル
;-----
SET1  TIS0.2                        ;チャンネル2で使用するタイマをサブシステム・クロックの4分周とする(fXT/4)

```

```

MOVW  AX,    #0001000000000001B    ;動作モード設定
      ;|||||||||++++----- MD023-020: インターバル・タイマ・モード
      ;|||||||||                (カウント開始時に割り込み発生)
      ;|||||||||++----- <00>
      ;|||||++----- CIS021-020: 未使用
      ;|||||++++----- STS022-020: ソフトウェア・トリガ・スタート
      ;|||+----- MASTER02: 単体動作
      ;||+----- CCS02: カウントクロック=TI02端子からの入力信号の有効エ
      ;||                ッジ
      ;|++----- <00>
      ;+----- CKS02: CK00(エッジ検出部に使用)

MOVW  TMR02, AX
MOVW  TDR02, #64-1                ;7.8125ms(0.5s/2^6)インターバル:
      ;(fXT/4)*7.8125(ms)=64

CLR1  TMIF02                    ;割り込み要求クリア
SET1  TMMK02                    ;割り込み禁止(後ほど許可)

;-----
; キャプチャの設定
;-----
; 7.8125ms(0.5s/2^6)間的高速内蔵発振クロ
; ックのカウンタ数の計測
;-----
;済 MOV  TPSOL, #10100000B        ;タイムクロック選択(チャンネル1で設定済み)
      ;||||++++----- PRS003-000(CK00): fCLK (キャプチャ動作で使用)
      ;++++----- PRS013-010(CK01): fCLK/2^10(インターバル動作で使用)

MOVW  AX,    #0000000000000100B    ;動作モード設定
      ;|||||||||++++----- MD003-000: キャプチャ・モード, アップ・カウント
      ;|||||||||++----- <00>
      ;|||||++----- CIS001-000: 未使用
      ;|||||++++----- STS002-000: ソフトウェア・トリガ・スタート
      ;|||+----- MASTER00: 単体動作
      ;||+----- CCS00: CKS00ビットで指定した動作クロックMCK
      ;|++----- <00>
      ;+----- CKS00: PRSで選択した動作クロックCK00(MCK)

MOVW  TMR00, AX

CLR1  TMIF00                    ;割り込み要求クリア
SET1  TMMK00                    ;割り込み禁止

```

```

EI                                     ;割り込み許可

;*****
;
;
;   メイン処理
;
;*****
MAIN_LOOP:

;-----
;   ; 約5秒毎の処理
;-----
;   ; 約5秒毎にキャリブレーションを行います。
;-----

BF      TMIF01, $LMAIN500             ;5.0s経過? No,
CLR1    TMIF01                       ;割り込み要求クリア
SET1    TTOL.1                       ;TAU0チャンネル1のインターバル動作停止(トリガ動作)
CALL    !!SCALIBSTT                  ;キャリブレーション開始

LMAIN500:

;-----
;   ; 各種メイン処理
;-----

;   ; ここで各種メイン処理があれば呼び出します。

BR      MAIN_LOOP                    ; MAIN_LOOPへ

;*****
;
;
;   キャリブレーション開始処理
;
;-----
;   ; TAU0のチャンネル2を7.8125ms(0.5s/2^6)周期のインターバル割り込みとして動作
;   ; を開始します。それと共に、その間の高速内蔵発振クロックでのカウント数を計
;   ; 測するため、チャンネル0をキャプチャモードにて動作を開始します。
;*****
SCALIBSTT:
MOV     RCALCNT,#0                   ;キャリブレーションの回数の初期化

SRECALIBSTT:

;-----
;   ; インターバル割り込み動作開始
;-----

```

```
CLR1    TMIF02                ;割り込み要求クリア
SET1    TSOL.2                ;TAU0のチャンネル2のインターバル動作開始(トリガ動作)
```

```
;MD020(TMR02.0)の設定により、インターバル動作のカウント開始時の割り込みを発生させ、
;同時にキャプチャを開始することで、精度良く基準時間内のカウント数を計測します。
```

JSCALIB300:

```
NOP                ;(割り込み要求の保留対策)
BF      TMIF02, $JSCALIB300    ;カウント開始? No,
;-----
; キャプチャ開始
;-----
CLR1    TMIF02
CLR1    TMMK02            ;TAU0のチャンネル2のインターバル割り込み許可
SET1    TSOL.0            ;TAU0のチャンネル0のキャプチャ動作開始(トリガ動作)

RET
```

```
;*****
```

```
;
;
;   INTTM02割り込み
;   (INTTM02使用, 7.8125ms(0.5s/2^6)に一度のインターバル割り込み)
;
;-----
```

```
;   キャリブレーション開始処理の呼び出しから7.8125ms(0.5s/2^6)後に発生する
;   インターバル割り込み処理です。TAU0のチャンネル0のキャプチャ・モードで計測
;   した,7.8125ms(0.5s/2^6)間的高速内蔵発振クロックのタイマ・カウント値から
;   高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行います。
```

```
;*****
```

INT_TM:

```
SEL     RB1

SET1    TTOL.0            ;TAU0のチャンネル0のキャプチャ動作停止(トリガ動作)

MOVW   AX,    TCR00        ;7.8125ms(0.5s/2^6)間的高速内蔵発振クロックでのタイマ・
```

カウント値を取得

```
SET1    TTOL.2            ;TAU0のチャンネル2のインターバル動作停止(トリガ動作)
SET1    TMMK02            ;割り込み禁止
```

```
INC     RCALCNT            ;キャリブレーション回数の更新
```

```
;理想値(62500)の確認
```


;理想値の-0.16~+0.16%(62400~62600)の範囲であれば補正なし

```

CMPW  AX,    #62400                ;~-0.16%の範囲(遅い)?
BC     $HITM500                    ; Yes,(補正あり)
BZ     $HITM500                    ; Yes,(補正あり)
CMPW  AX,    #62600                ;-0.16 ~ +0.16%の範囲?
BZ     $HITMRET                    ; Yes,(補正なし)
BC     $HITMRET                    ; Yes,(補正なし)
;;   BR     HITM300                ; No, (補正あり)

```

;理想値の+0.16%より大きい範囲(速い)

;;HITM300:

```

CMP    HIOTRM, #00000000B          ;既に一番遅い設定?
BZ     $HITMRET                    ; Yes,(補正不可)
DEC    HIOTRM                      ;遅くする
CMP    RCALCNT,#1                  ;キャリブレーションは初回?
BNZ    $HITM350                    ; No,
MOV    RCALSTS,#00000000B         ; Yes,(過去情報の初期化)
BR     HITM700

```

HITM350:

```

MOV    A,    RCALSTS              ;キャリブレーションの過去情報を更新
CLR1   CY
ROLC   A,    1
MOV    RCALSTS,A
AND    A,    #00000111B
CMP    A,    #00000010B           ;-1 +1 -1の遷移?
BZ     $HITMRET                    ; Yes,(補正終了)
BR     HITM700

```

;理想値の-0.16%以下の範囲(遅い)

HITM500:

```

CMP    HIOTRM, #00011111B          ;既に一番速い設定?
BZ     $HITMRET                    ; Yes,(補正不可)
INC    HIOTRM                      ;速くする
CMP    RCALCNT,#1                  ;キャリブレーションは初回?
BNZ    $HITM550                    ; No,
MOV    RCALSTS,#11111111B         ; Yes,(過去情報の初期化)
BR     HITM700

```

HITM550:

```

MOV    A,    RCALSTS              ;キャリブレーションの過去情報を更新
SET1   CY
ROLC   A,    1
MOV    RCALSTS,A
AND    A,    #00000111B

```

```

    CMP    A,      #00000101B      ;+1 -1 +1の遷移?
    BZ     $HITMRET                ; Yes,(補正終了)
HITM700:
    ;再キャリブレーション開始
    CALL   !!SRECALIBSTT
    BR     HITMRET I
HITMRET:
    CLR1   TMIF01                 ;割り込み要求クリア
    SET1   TSOL.1                 ;TAU0チャンネル1のインターバル動作開始(トリガ動作)
HITMRETI:
    RETI
end

```

main.c (C言語版)

/******

NEC Electronics 78K0R/KG3マイクロコントローラ

78K0R/KG3マイクロコントローラ サンプル・プログラム

周波数補正 (高速内蔵発振クロックのキャリブレーション)

【履歴】

2007.11.-- 新規作成

【概要】

このサンプル・プログラムは、水晶振動子による周波数精度の良いサブシステム・クロックを使用することにより、高速内蔵発振クロックの周波数を高速内蔵発振器トリミング・レジスタ(HIOTRM)を調整し、より8MHz近くに補正を行うプログラムです。キャリブレーション開始処理の呼び出しにより、タイマ・アレイ・ユニットのインターバル・タイマ・モードにてサブシステム・クロックを用い精度の良い17.8125ms(0.5s/26)の時間を計測し、キャプチャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値が理想値の範囲内でないときは、高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行い、再度、キャリブレーションを行い、理想値の範囲内になるまで行います。

本サンプル・プログラムでは、約5秒毎にキャリブレーション開始処理を呼び出し、キャリブレーションを行います。

<使用する周辺の初期設定の主な設定内容>

- ・ベクタ・テーブルの設定
- ・スタック・ポインタの設定
- ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
- ・5.0sインターバルに使用するTAU0のチャンネル1の設定
- ・7.8125ms(0.5s/2^6)インターバル割り込みに使用するTAU0のチャンネル2の設定
- ・高速内蔵発振クロックのカウント計測のためキャプチャ動作に使用するTAU0のチャンネル0の設定

<キャリブレーション開始処理の設定内容>

- ・TAU0のチャンネル2を7.8125ms(0.5s/2^6)インターバル割り込み動作開始

- ・高速内蔵発振クロックのカウント計測のTAU0のチャンネル0をキャプチャ開始

< INTTM02割り込み

(INTTM02使用, 7.8125ms(0.5s/2^6)に一度のインターバル割り込み)>

- ・キャプチャ動作とインターバル割り込み動作の停止
- ・計測したキャプチャ値を理想値の範囲であるかチェックを行う
- ・理想値の範囲外であれば, 高速内蔵発振器トリミング・レジスタ(HIOTRM)の+1/-1の調整を行い, 再キャリブレーション開始。ただし, 調整の変化が+1 -1 +1, もしくは-1 +1 -1と変化した場合は調整終了とする。

*****/

/*=====

前処理指令 (#pragma指令)

=====*/

```
#pragma SFR /* 特殊機能レジスタ(SFR)名を記述可能にする */
#pragma DI /* DI命令を記述可能にする */
#pragma EI /* EI命令を記述可能にする */
#pragma NOP /* EI命令を記述可能にする */
#pragma interrupt INTTM02 fn_inttm02 /* 割り込み関数宣言:INTTM02 */
```

/*=====

関数プロトタイプ宣言

=====*/

```
static void fn_CalibrationStart(void);
static void fn_ReCalibrationStart(void);
```

/*=====

R A M

=====*/

```
static unsigned char ucCalibrationStatus; /* キャリブレーションの過去情報
*/
/* ビット毎にHIOTRMを+1方向に調整(1)/-1方向に調整(0) */
```

```

/* bit7: 7回前の調整(最も古い過去の情報)
*/
/* bit6: 6回前の調整 */
/* : */
*/
/* bit1: 1回前の調整 */
/* bit0: 最新の調整(最も新しい現在の情報) */

static unsigned char ucCalibrationCounter; /* キャリブレーションの調整回数 */

/*****

使用する周辺の初期設定

*****/
void hdwinit(void)
{
    DI(); /* 割り込み禁止 */
/*-----
    クロック周波数の設定
    -----
    高速内蔵発振クロックで動作が行えるように設定します。
    ----- */
    CMC = 0b00010000; /* クロック動作モード */
    /* |||+----- AMPH: 2MHz fMX 10MHz */
    /* |||+++----- <000> */
    /* |+----- OSCSELS: XT1発振モード */
    /* |+----- <0> */
    /* ++----- EXCLK/OSCSEL: 入力ポート・モード */

    CSC = 0b10000000; /* クロック動作ステータス制御 */
    /* |||+----- HIOSTOP: 高速内蔵発振回路動作 */
    /* ||++++----- <00000> */
    /* |+----- XTSTOP: XT1発振回路動作 */
    /* +----- MSTOP: X1発振回路動作 */

    OSMC = 0b00000000; /* 動作スピード・モード */
    /* |||+----- FSEL: 10MHz以下の周波数で動作 */
    /* ++++++----- <00000> */

```

```

CKC = 0b00001000;          /* クロック選択 */
/* |||+----- MDIV2-0: CPU周辺ハードウェア・クロック(fCLK)=fIH */
/* |||+----- <1> */
/* |+----- MCM0: 高速内蔵発振クロック(fIH) */
/* |+----- <R> */
/* +----- CSS: メイン・システム・クロック(fMAIN)=fCLK */
/* +----- <R> */

/*-----
インターバル・タイマの設定
-----

タイマ・アレイ・ユニット0のチャンネル1を5.0s周期のインターバル・タイマ・
モードに設定します。
キャリブレーション処理を約5秒毎に呼び出すために使用します。
-----*/

TAU0EN= 1;                /* TAU0使用許可 */
TPSOL = 0b10100000;      /* タイマクロック選択 */
/* |||+----- PRS003-000(CK00): fCLK (キャプチャ動作で使用) */
/* +----- PRS013-010(CK01): fCLK/2^10(インターバル動作で使用)

*/

TMR01 = 0b1000000000000000; /* 動作モード設定 */
/* |||+----- MD013-010: インターバル・タイマ・モード */
/* |||+----- <00> */
/* |||+----- CIS011-010: 未使用 */
/* |||+----- STS012-010: ソフトウェア・トリガ・スタート */
/* |||+----- MASTER01: 単体動作(奇数チャンネル) */
/* |+----- CCS01: CKS01ビットで指定した動作クロックMCK */
/* +----- <00> */
/* +----- CKS01: PRSで選択した動作クロックCK01(MCK) */

TDR01 = 39063-1;         /* 5.0sインターバル : (fCLK/2^10)*5.0(s)=39062.5

*/

TMIF01= 0;              /* 割り込み要求クリア */
TMMK01= 1;              /* 割り込み禁止 */

TSOL.1= 1;              /* TAUインターバル動作開始(トリガ動作) */

/*-----
キャリブレーション用の設定
-----

精度の良い17.8125ms(0.5s/2^6)を計測するため、TAU0のチャンネル2をサブシステ

```

ム・クロックの4分周をカウントクロックとし、インターバル・タイマ・モードに設定します。

また、その7.8125ms(0.5s/2^6)間的高速内蔵発振クロックでのカウント数を計測するため、チャンネル0をキャプチャモードに設定します。

```

----- */
/*-----
   インターバル割り込みの設定
-----
   7.8125ms(0.5s/2^6)に一度のインターバル
----- */
TIS0.2 = 1;                               /* チャンネル2で使用するタイマをサブシステム・クロックの
                                           4分周とする(fXT/4) */

TMR02 = 0b0001000000000001;             /* 動作モード設定 */
   /*|||||++++----- MD023-020: インターバル・タイマ・モード */
   /*|||||              (カウント開始時に割り込み発生) */
   /*|||||++----- <00> */
   /*|||||++----- CIS021-020: 未使用 */
   /*|||||+++----- STS022-020: ソフトウェア・トリガ・スタート */
   /*|||||+----- MASTER02: 単体動作 */
   /*|||+----- CCS02: カウントクロック=T102端子からの入力信号の有
   /*|||              効エッジ */
   /*|++----- <00> */
   /*+----- CKS02: CK00(エッジ検出部に使用)*/

TDR02 = 64-1;                             /* 7.8125ms(0.5s/2^6)インターバル:
                                           (fXT/4)*7.8125(ms)=64 */

TMIF02 = 0;                               /* 割り込み要求クリア */
TMMK02 = 1;                               /* 割り込み禁止(後ほど許可) */

/*-----
   キャプチャの設定
-----
   7.8125ms(0.5s/2^6)間的高速内蔵発振クロ
   ックのカウント数の計測
----- */
/*済*/ /*TPSOL = 0b10100000;             /* タイマクロック選択(チャンネル1で設定済み) */
   /*|||||++++----- PRS003-000(CK00): fCLK (キャプチャ動作で使用) */
   /*++++----- PRS013-010(CK01): fCLK/2^10(インターバル動作で使
用) */

```



```

/*****

```

キャリブレーション開始処理

```

-----

TAU0のチャンネル2を7.8125ms(0.5s/2^6)周期のインターバル割り込みとして動作
を開始します。それと共に、その間的高速内蔵発振クロックでのカウント数を計
測するため、チャンネル0をキャプチャモードにて動作を開始します。
*****/

static void fn_CalibrationStart(void)
{
    ucCalibrationCountr = 0;          /* キャリブレーション回数の初期化 */
    fn_ReCalibrationStart();
}

static void fn_ReCalibrationStart(void)
{
    /*-----
    インターバル割り込み動作開始
    -----*/
    TMIF02 = 0;                      /* 割り込み要求クリア */
    TSOL.2 = 1;                      /* TAU0のチャンネル2のインターバル動作開始(トリガ動作) */

    /* MD020(TMR02.0)の設定により、インターバル動作のカウント開始時の割り込みを発生させ、
    同時にキャプチャを開始することで、精度良く基準時間内のカウント数を計測します。 */
    while(!TMIF02){                  /* カウント開始タイミングを待つ */

    /*-----
    キャプチャ開始
    -----*/
    TMIF02 = 0;
    TMMK02 = 0;                      /* TAU0のチャンネル2のインターバル割り込み許可 */
    TSOL.0 = 1;                      /* TAU0のチャンネル0のキャプチャ動作開始(トリガ動作) */
}

/*****

```

INTTM02割り込み
(INTTM02使用、7.8125ms(0.5s/2^6)に一度のインターバル割り込み)

```

-----

キャリブレーション開始処理の呼び出しから7.8125ms(0.5s/2^6)後に発生する
インターバル割り込み処理です。TAU0のチャンネル0のキャプチャ・モードで計測

```

した,7.8125ms(0.5s/2^6)間的高速内蔵発振クロックのタイマ・カウント値から
高速内蔵発振器トリミング・レジスタ(HIOTRM)の調整を行います。

```

*****/
__interrupt void fn_inttm02(void)
{
    register unsigned short ushnTimeCnt;    /* タイマ・カウント値 */

    TTOL.0= 1;                             /* TAU0のチャンネル0のキャプチャ動作停止(トリガ動作) */

    ushnTimeCnt = TCR00;                    /* 7.8125ms(0.5s/2^6)間的高速内蔵発振クロックでの
                                           タイマ・カウント値を取得 */

    TTOL.2 = 1;                             /* TAU0のチャンネル2のインターバル動作停止(トリガ動作) */
    TMMK02 = 1;                             /* 割り込み禁止 */

    ucCalibrationCountr ++;                 /* キャリブレーション回数の更新 */

    if( ushnTimeCnt <= 62400 ){
        /* 理想値の+0.16%より大きい範囲(遅い) 速くする */
        if( HIOTRM != 0b00011111 ){
            /* 既に一番速い設定? No, */
            HIOTRM ++;                       /* 速くする */
            if( ucCalibrationCountr == 1 ){
                ucCalibrationStatus = 0b11111111; /* +1調整時の初期値 */
                fn_ReCalibrationStart();         /* 再キャリブレーション開始 */
                return;
            }
            else{
                /* キャリブレーションの過去情報を更新 */
                ucCalibrationStatus = (ucCalibrationStatus<<1)|0b00000001; /* +1調整 */
                if((ucCalibrationStatus & 0b00000111 )!= 0b00000101){
                    /* +1 -1 +1の遷移? No, */
                    fn_ReCalibrationStart();     /* 再キャリブレーション開始 */
                    return;
                }
            }
        }
    }

    else if( ushnTimeCnt > 62600 ){
        /* 理想値の-0.16%以下の範囲(速い) 遅くする */
        if( HIOTRM != 0b00000000 ){
            /* 既に一番遅い設定? No, */
            HIOTRM --;                         /* 遅くする */
        }
    }
}

```

```
if( ucCalibrationCountr == 1 ){
    ucCalibrationStatus = 0b00000000;      /* -1調整時の初期値 */
    fn_ReCalibrationStart();               /* 再キャリブレーション開始 */
    return;
}
else{
    /* キャリブレーションの過去情報を更新 */
    ucCalibrationStatus = (ucCalibrationStatus<<1)&0b11111110; /* -1調整 */
    if((ucCalibrationStatus & 0b00000111 )!= 0b00000010){
        /* -1 +1 -1の遷移? No, */
        fn_ReCalibrationStart();           /* 再キャリブレーション開始 */
        return;
    }
}
}
}
}
TMIF01 = 0; /* 割り込み要求クリア */
TSOL.1 = 1; /* TAU0チャンネル1のインターバル動作開始(トリガ動作) */
}
```

付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	December 2008	-	-

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。
