

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い申し上げます。

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

V850ES/Jx3-L

サンプル・プログラム（16ビット・タイマ/イベント・カウンタ（TMP, TMQ））

インターバル・タイマ・モード編

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16ビット・タイマ/イベント・カウンタP, Q (TMP, TMQ)のインターバル・タイマ機能の設定方法や活用方法を説明したものです。サンプル・プログラムでは、16ビット・タイマ/イベント・カウンタP (TMP)のインターバル・タイマ機能を使用して、LED1を一定周期で点滅させます。また、スイッチ入力回数に応じて、LED1の点滅周期を変更します。

対象デバイス

V850ES/JF3-L マイクロコントローラ

V850ES/JG3-L マイクロコントローラ

目次

第1章 概要 ... 3	
1.1 初期化 ... 4	
1.2 16ビット・タイマ/イベント・カウンタP (TMP) によるインターバル割り込み処理 ... 5	
1.3 スイッチ入力によるINTP0割り込み処理 ... 5	
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 初期設定と動作概要 ... 8	
3.4 フロー・チャート ... 10	
3.5 V850ES/JG3-L版とV850ES/JF3-L版の違い ... 13	
3.5 TMPとTMQの違い ... 13	
3.7 セキュリティIDについて ... 13	
第4章 レジスタ設定について ... 14	
4.1 16ビット・タイマ/イベント・カウンタP (TMP) の設定 ... 15	
4.2 LED1点滅周期とチャタリング検出時間の設定 ... 26	
第5章 関連資料 ... 30	
付録A プログラム・リスト ... 31	

資料番号 U19674JJ1V0AN00

発行年月 February 2009 NS

- 本資料に記載されている内容は2009年2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いません。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、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。

M8E0710J

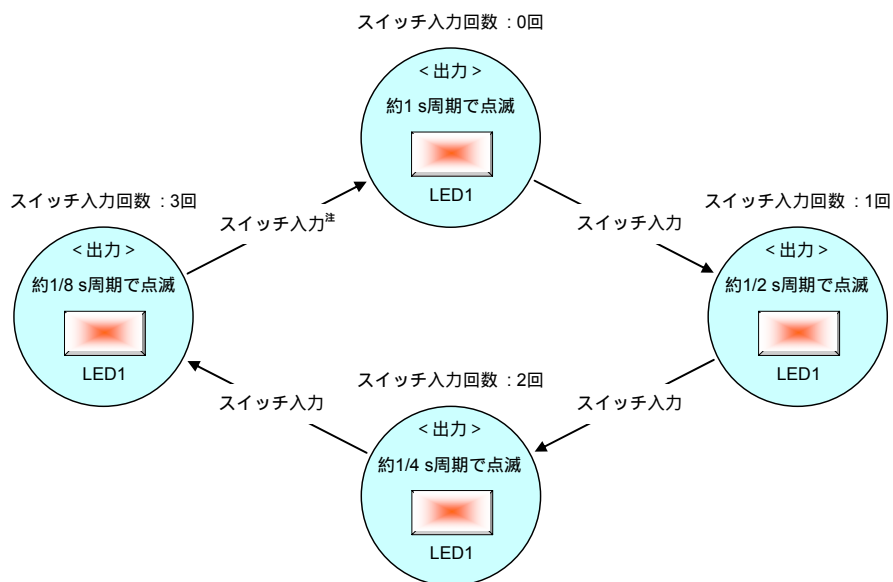
第1章 概 要

このサンプル・プログラムでは、16ビット・タイマ/イベント・カウンタP (TMP) のインターバル・タイマ機能の使用例を示しています。LED1を一定周期で点滅させ、スイッチの入力回数に応じて、LED1の点滅周期を変更します。

点滅周期は、16ビット・タイマ/イベント・カウンタP (TMP) のインターバル時間により制御し、スイッチ入力を検出した際にインターバル時間を変更することにより点滅周期を変更します。

リセット解除後に動作停止状態の周辺機能について、このサンプル・プログラムで使用しない周辺機能は、設定していません。

スイッチ入力回数と点滅周期の関係を以下に示します。



注 4回目以降は、0回目からの点滅周期の繰り返しになります。

1.1 初期化

<オプション・バイトの参照>

- ・リセット解除後の発振安定時間を参照

<内蔵周辺設定>

- ・内蔵周辺I/Oレジスタへのバス・アクセスのウェイト<ウェイト：1>設定
- ・オンチップ・デバッグ・モード<通常動作モード>の設定
- ・内蔵発振器を停止，ウォッチドッグ・タイマ停止
- ・CPUクロックを分周なしに設定
- ・PLLモードに設定し 5 MHz × 4逡倍 = 20 MHz 動作に設定

<端子設定>

- ・未使用端子の設定
- ・外部割り込み端子の設定（エッジ指定，優先度指定，マスク解除）
- ・LED1出力端子の設定

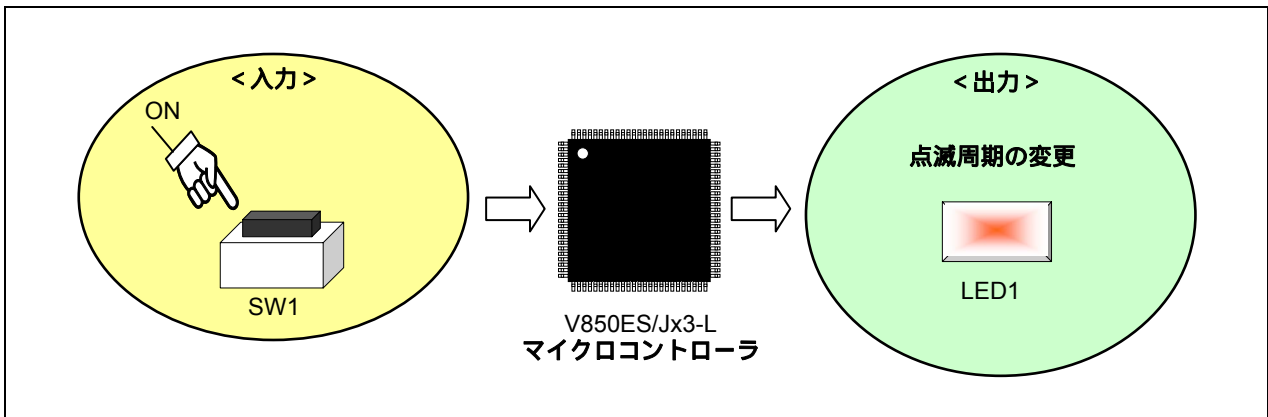
<タイマ（TMP）の設定>

- ・カウント・クロックをf_{xx}（20 MHz）に設定
- ・インターバル周期を2 ms（0.05 μs × 40000）に設定
- ・INTTP0CC0割り込みのマスク解除

1.2 16ビット・タイマ/イベント・カウンタP (TMP) によるインターバル割り込み処理

初期設定完了後は、16ビット・タイマ/イベント・カウンタP (TMP) の割り込み (INTTP0CC0) 発生を利用して、LED1を一定周期で点滅します。

16ビット・タイマ/イベント・カウンタP (TMP) の0.25 ms~2 msインターバルの割り込み回数をカウントして0.125 s~1 sの時間を算出し、その周期でLED1を点滅させます。周期はINTTP0端子に接続したスイッチにより 1 s 0.5 s 0.25 s 0.125 s 1 s (以降は繰り返し) へ変化させます。



1.3 スイッチ入力によるINTP0割り込み処理

スイッチ入力によるINTP0端子の立ち下がりエッジを検出し、割り込み処理を行います。INTP0端子の立ち下がりを検出してから、約10 ms経過後に、INTP0端子がハイ・レベル (スイッチがOFF) であった場合は、チャタリングであると判定します。エッジ検出してから約10 ms経過後に、INTP0端子がロウ・レベル (スイッチがON) であった場合は、スイッチの入力回数に応じて、LED1の点滅周期を変更します。

注意 デバイス使用上の注意事項については、各製品のユーザーズ・マニュアル (V850ES/Jx3-L) を参照してください。



【コラム】 チャタリングとは

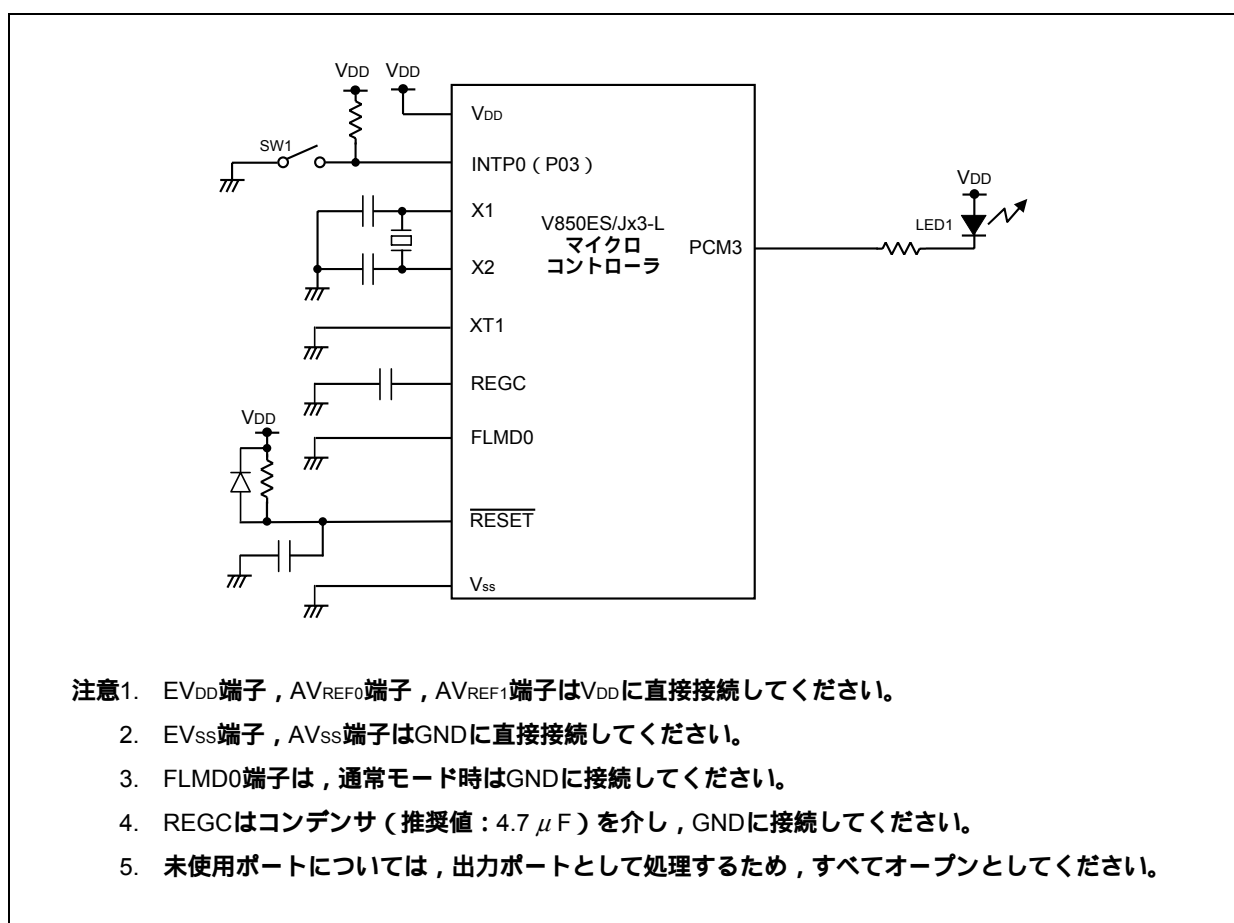
スイッチが切り替わった直後に、接点が機械的にばたつくことにより、電気信号がONとOFFを繰り返す現象のことです。

第2章 回路図

この章では、このサンプル・プログラムで使用する場合の回路図および周辺ハードウェアを説明します。

2.1 回路図

回路図を次に示します。



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。

(1) スイッチ (SW1)

LED1点滅制御用の割り込み入力として、スイッチを使用します。

(2) LED (LED1)

16ビット・タイマ/イベント・カウンタP (TMP) のインターバル・タイマ機能とスイッチ入力に対応した出力として、LED1を使用します。



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。

【C言語版】

ファイル名 (ツリー構造)	説明	同封圧縮 (*.zip) ファイル	
			
<pre> c ├── conf │ ├── crtE.s │ ├── AppNote_ITVL_TMP.dir │ ├── AppNote_ITVL_TMP.prj │ └── AppNote_ITVL_TMP.prw └── src ├── main.c ├── minicube2.s └── opt_b.s </pre>	スタートアップ・ルーチン・ファイル ^{注1}	-	
	リンク・ディレクティブ・ファイル ^{注2}		
	統合開発環境 PM+用プロジェクト・ファイル	-	
	統合開発環境 PM+用ワーク・スペース・ファイル	-	
	マイコンのハードウェア初期化処理とメイン処理を記述したC言語ソース・ファイル		
	MINICUBE2用の領域予約を行うソース・ファイル		
	オプション・バイト設定を行うソース・ファイル		

- 注1. ワークスペース新規作成時の「スタートアップ・ファイルの指定」時に、「サンプルをコピーして使用する(C)」を選択した際にコピーされるスタートアップ・ファイル(デフォルト・インストール・パスであれば、C:\Program Files\NEC Electronics Tools\CA850**使用バージョン**\lib850\r32\crtE.sのコピーとなります)。
2. ワークスペース新規作成時の「リンク・ディレクティブ・ファイルの指定」時に、「サンプルを作成して使用する(C)」を選択し、「メモリの使用方法：内蔵メモリのみ(C)」をチェックした際に、自動生成されるリンク・ディレクティブ・ファイルに、**MINICUBE2用のセグメントを追加**したもの(デフォルト・インストール・パスであれば、C:\Program Files\NEC Electronics Tools\PM+**使用バージョン**\bin\w_data\V850_i.dat が基準となります)。

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。

- ・インターバル・タイマ機能 : 16ビット・タイマ/イベント・カウンタP (TMP)
- ・外部割り込み入力 (スイッチ入力用) : INTP0 (SW1)
- ・出力ポート (LED1点灯用) : PCM3 (LED1)

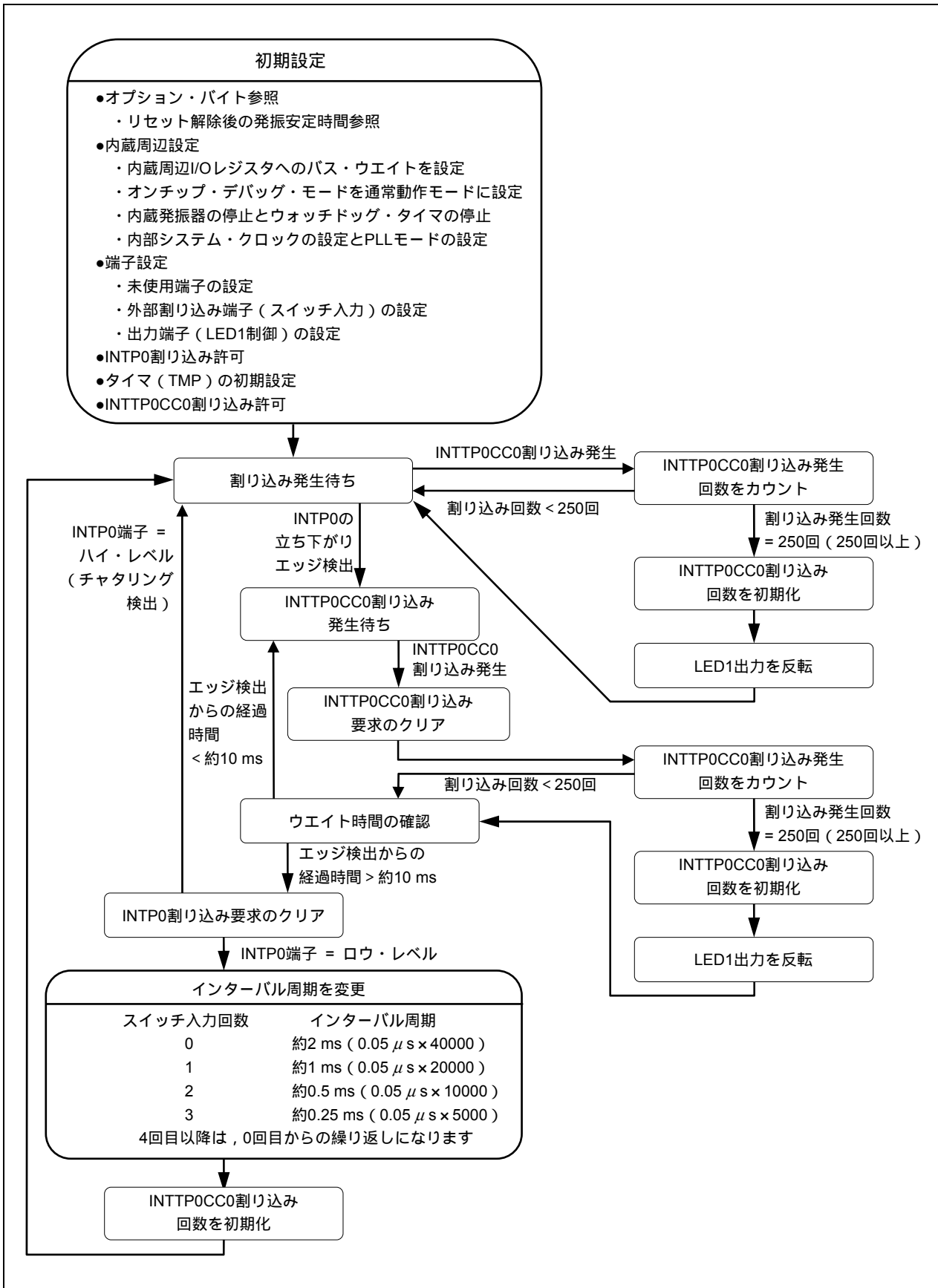
3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定で、クロック周波数の選択や、ウォッチドッグ・タイマの停止設定、入出力ポートや外部割り込みの端子設定、16ビット・タイマ/イベント・カウンタP (TMP) のインターバル・タイマ設定、割り込みの設定などを行います。

初期設定完了後は、16ビット・タイマ/イベント・カウンタP (TMP) の割り込み (INTTP0CC0) 発生を利用して、LED1を一定周期で点滅します。

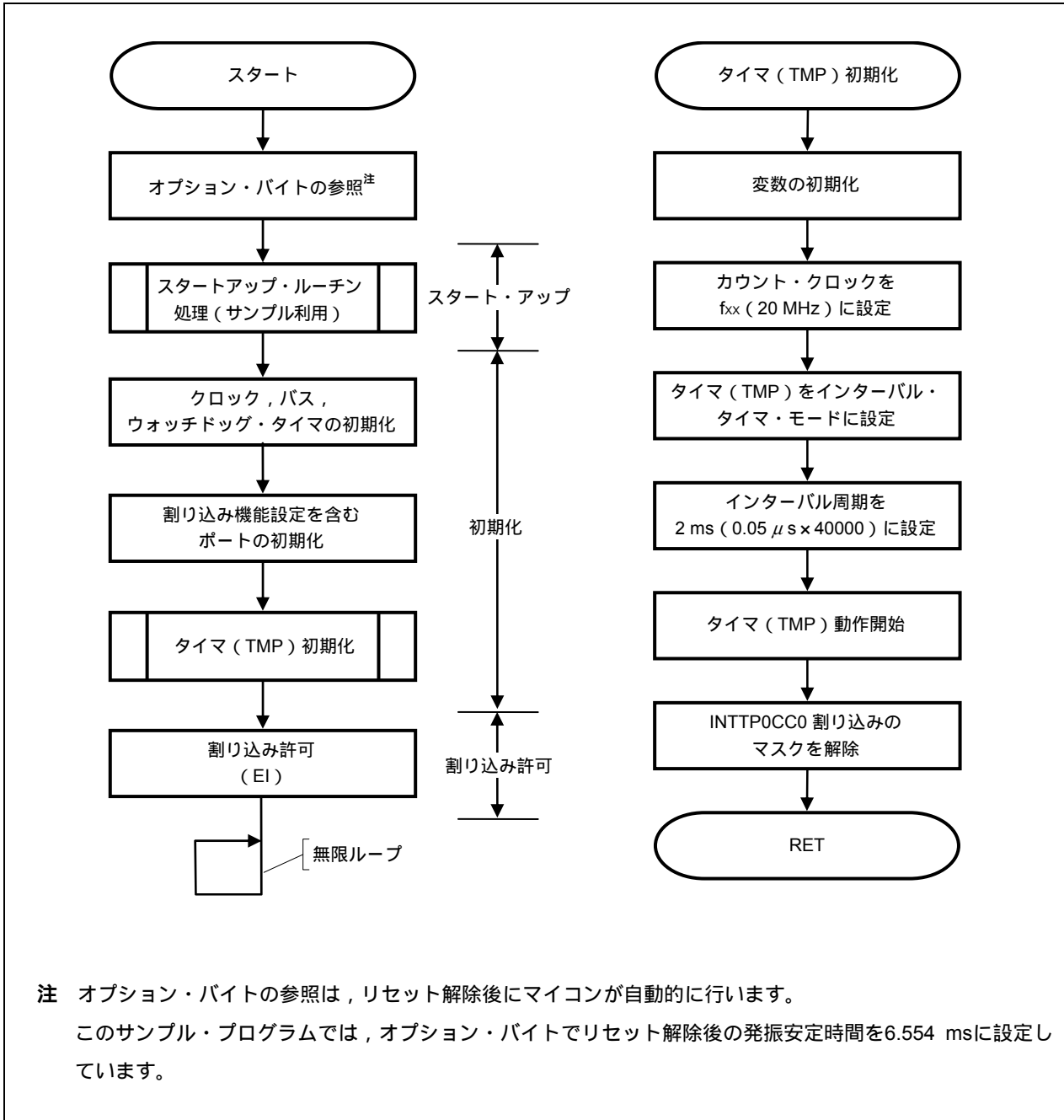
スイッチ入力によるINTP0端子の立ち下がりを検出したら、INTP0の割り込み処理を行います。INTP0端子の立ち下がりを検出してから、約10 ms経過後に、INTP0端子がハイ・レベル (スイッチがOFF) であった場合は、チャタリングであると判定します。エッジ検出してから約10 ms経過後に、INTP0端子がロウ・レベル (スイッチがON) であった場合は、スイッチの入力回数に応じて、LED1の点滅周期を変更します。

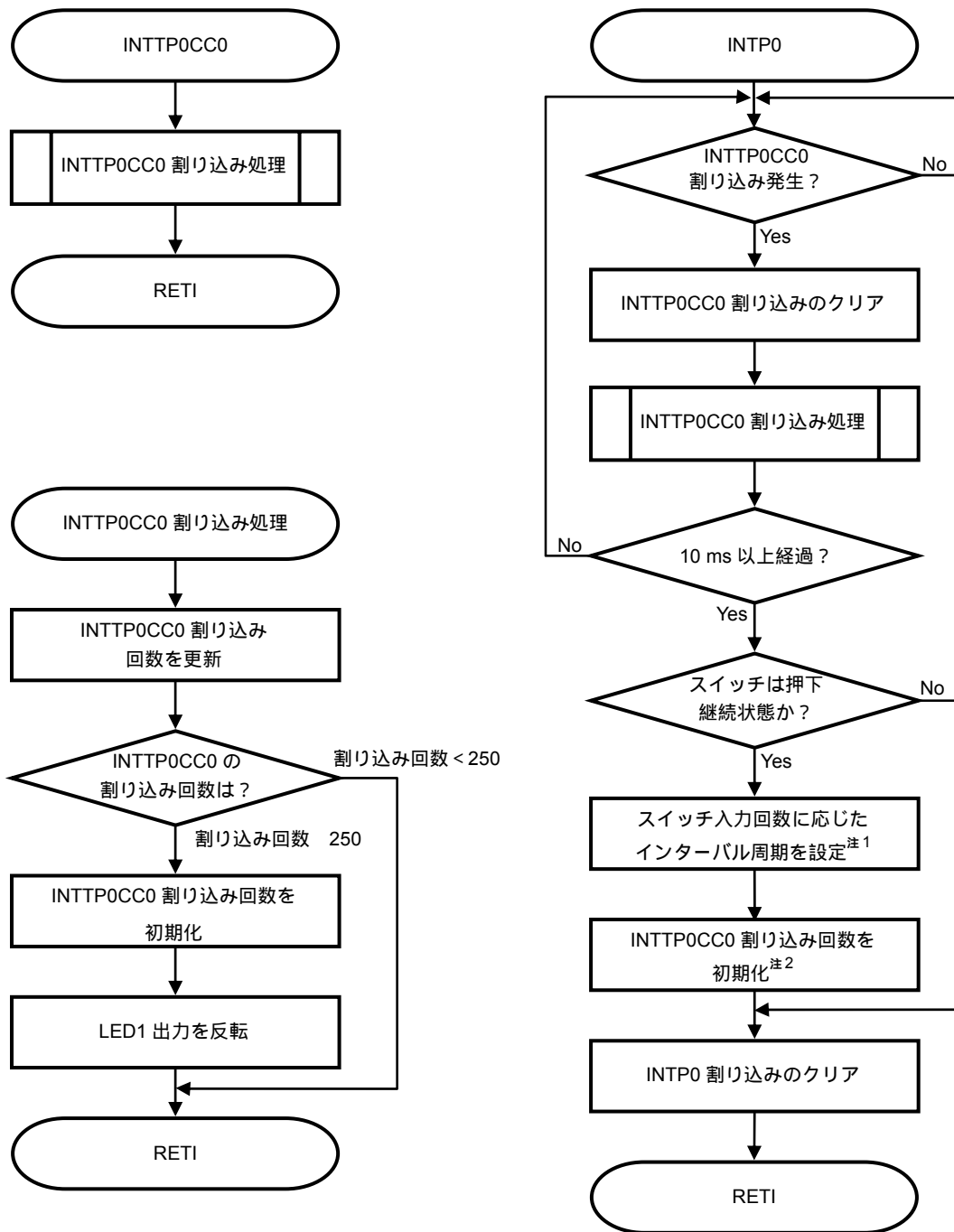
次に状態遷移図を示します。



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1. カウンタを停止せずにTPnCCR0レジスタの値を書き換える場合、TPnCCR0レジスタの値を小さい値に書き換えると、16ビット・カウンタがオーバーフローする場合がありますので注意が必要です。詳細については、ユーザズ・マニュアルを参照してください。

このサンプル・プログラムでは、INTTPnCCR0割り込みに同期して、TPnCCR0への書き込みが行われます。このため、タイマのカウンタ値がTPnCCR0の最小値になる前に書き込みが完了するので、カウンタを停止しなくてもオーバーフローは発生しません。

2. インターバル時間変更後にINTTP0CC0割り込み回数カウンタを初期化しているため、高速にスイッチの押下を繰り返した場合、その期間中にLED1の点滅が行われません。

**【コラム】スタートアップ・ルーチン処理の内容**

スタートアップ・ルーチンは、V850のリセットが解除されたあと、メイン関数を実行する前に、実行されるルーチンです。基本的にはC言語で記述されたプログラムが動作するための初期化処理を行います。具体的には、以下のことを行います。

- ・ main関数の引数領域の確保
- ・ スタック領域の確保
- ・ リセットが入った時のRESETハンドラの設定
- ・ テキスト・ポインタ (tp) の設定
- ・ グローバル・ポインタ (gp) の設定
- ・ スタック・ポインタ (sp) の設定
- ・ エlement・ポインタ (ep) の設定
- ・ マスク・レジスタ (r20, r21) へマスク値を設定
- ・ sbss領域, bss領域 のゼロ・クリア
- ・ 関数のプロローグ・エピローグ・ランタイム・ライブラリ用のCTBP値の設定
- ・ r6とr7をmain関数の引数に設定
- ・ main関数へ分岐する

3.5 V850ES/JG3-L版とV850ES/JF3-L版の違い

V850ES/JG3-Lは、V850ES/JF3-Lに対して、I/O、タイマ/カウンタ、シリアル・インタフェースなどの機能を拡張したものです。

このサンプル・プログラムにおいては、I/Oの初期化におけるP1, P3, P7, P9, PDHの初期化範囲が異なります。サンプル・プログラムの詳細については、**付録A プログラム・リスト**を参照してください。

3.6 TMPとTMQの違い

16ビット・タイマ/イベント・カウンタP (TMP) と16ビット・タイマ/イベント・カウンタQ (TMQ) は、キャプチャ・トリガ端子やタイマ出力端子、キャプチャ/コンペア・レジスタの本数などが異なります。

このサンプル・プログラムにおいては、16ビット・タイマ/イベント・カウンタP (TMP) を使用しています。

16ビット・タイマ/イベント・カウンタQ (TMQ) を使用する場合は、**第4章 レジスタ設定について**、**付録A プログラム・リスト**の内容を参考に設定してください。

3.7 セキュリティIDについて

オンチップ・デバッグ・エミュレータによるオンチップ・デバッグ時、フラッシュ・メモリの内容を第三者に読み出されることを防ぐために、10バイトのIDコードによる認証を行います。

セキュリティIDの詳細については、V850ES/Jx3-L **サンプル・プログラム (割り込み) スイッチ入力による外部割り込み編 アプリケーション・ノート**を参照してください。

第4章 レジスタ設定について

この章では、16ビット・タイマ/イベント・カウンタP(TMP)、16ビット・タイマ/イベント・カウンタQ(TMQ)機能の設定について説明します。

その他の初期設定については、V850ES/Jx3-L **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。また、割り込みについては、V850ES/Jx3-L **サンプル・プログラム(割り込み) スイッチ入力による外部割り込み編 アプリケーション・ノート**を参照してください。

なお、リセット解除後に動作停止状態の周辺機能について、このサンプル・プログラムで使用しない周辺機能は、設定していません。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアルを参照してください。

- ・ V850ES/JG3-L 32ビット・シングルチップ・マイクロコントローラ ハードウェア編
ユーザーズ・マニュアル
- ・ V850ES/JF3-L 32ビット・シングルチップ・マイクロコントローラ ハードウェア編
ユーザーズ・マニュアル

C言語の拡張記述の詳細については、CA850 **Cコンパイラ・パッケージ C言語編 ユーザーズ・マニュアル**を参照してください。

4.1 16ビット・タイマ/イベント・カウンタP (TMP) の設定

16ビット・タイマ/イベント・カウンタP (TMP) を使用する際に関係するレジスタには、次の9種類があります。

- ・ TMPn制御レジスタ0 (TPnCTL0)
- ・ TMPn制御レジスタ1 (TPnCTL1)
- ・ TMPnI/O制御レジスタ0 (TPnIOC0)
- ・ TMPnI/O制御レジスタ1 (TPnIOC1)
- ・ TMPnI/O制御レジスタ2 (TPnIOC2)
- ・ TMPnオプション・レジスタ0 (TPnOPT0)
- ・ TMPnキャプチャ/コンペア・レジスタ0 (TPnCCR0)
- ・ TMPnキャプチャ/コンペア・レジスタ1 (TPnCCR1)
- ・ TMPnカウンタ・リード・バッファ・レジスタ (TPnCNT)

備考 n = 0-5

注意 このサンプル・プログラムでは、n = 0を使用しています。

16ビット・タイマ/イベント・カウンタQ (TMQ) を使用する際に関係するレジスタには、次の11種類があります。次頁以降では、TMPの説明を行いますので、TMQを使用する場合は、以下の各レジスタと読み替えてください。

- ・ TMQ0制御レジスタ0 (TQ0CTL0)
- ・ TMQ0制御レジスタ1 (TQ0CTL1)
- ・ TMQ0I/O制御レジスタ0 (TQ0IOC0)
- ・ TMQ0I/O制御レジスタ1 (TQ0IOC1)
- ・ TMQ0I/O制御レジスタ2 (TQ0IOC2)
- ・ TMQ0オプション・レジスタ0 (TQ0OPT0)
- ・ TMQ0キャプチャ/コンペア・レジスタ0 (TQ0CCR0)
- ・ TMQ0キャプチャ/コンペア・レジスタ1 (TQ0CCR1)
- ・ TMQ0キャプチャ/コンペア・レジスタ2 (TQ0CCR2)
- ・ TMQ0キャプチャ/コンペア・レジスタ3 (TQ0CCR3)
- ・ TMQ0カウンタ・リード・バッファ・レジスタ (TQ0CNT)

4.1.1 16ビット・タイマ/イベント・カウンタP (TMP) の動作クロックの設定

TMPn制御レジスタ0 (TPnCTL0) で、16ビット・タイマ/イベント・カウンタP (TMP) のカウント・クロックの選択、および動作の制御を行います。

TPnCKS2-TPnCKS0ビットは、TPnCEビット = 0の状態を設定してください。

このサンプル・プログラムでは、初期化時に0x00を設定することで、TPnCKS2-TPnCKS0ビットをf_{xx}(20 MHz) に設定します。

16ビット・タイマ/イベント・カウンタP (TMP) の各レジスタ設定の終了後に、TPnCEビット = 1に設定します。

図4 - 1 TPnCTL0レジスタのフォーマット

TMPn制御レジスタ0 (TPnCTL0)
 アドレス : TP0CTL0 0xFFFFF590, TP1CTL0 0xFFFFF5A0,
 TP2CTL0 0xFFFFF5B0, TP3CTL0 0xFFFFF5C0,
 TP4CTL0 0xFFFFF5D0, TP5CTL0 0xFFFFF5E0

	7	6	5	4	3	2	1	0
	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	TMPnの動作の制御
0	TMPn動作禁止 (TMPnを非同期にリセット)
1	TMPn動作許可。TMPn動作開始

TPnCKS2	TPnCKS1	TPnCKS0	内部カウント・クロックの選択	
			n = 0, 2, 4	n = 1, 3, 5
0	0	0	f _{xx}	
0	0	1	f _{xx} /2	
0	1	0	f _{xx} /4	
0	1	1	f _{xx} /8	
1	0	0	f _{xx} /16	
1	0	1	f _{xx} /32	
1	1	0	f _{xx} /64	f _{xx} /256
1	1	1	f _{xx} /128	f _{xx} /512

注意 表の赤字部分がサンプル・プログラムでの設定値となります。

4.1.2 16ビット・タイマ/イベント・カウンタP (TMP) の動作モードの設定

TMPn制御レジスタ1 (TPnCTL1) で、16ビット・タイマ/イベント・カウンタP (TMP) の動作モードを設定します。

このサンプル・プログラムでは、TPnMD2-TPnMD0を000Bに設定することにより、インターバル・タイマ・モードとして設定しています。

図4 - 2 TPnCTL1レジスタのフォーマット

TMPn制御レジスタ1 (TPnCTL1)
 アドレス : TP0CTL1 0xFFFFF591, TP1CTL1 0xFFFFF5A1,
 TP2CTL1 0xFFFFF5B1, TP3CTL1 0xFFFFF5C1,
 TP4CTL1 0xFFFFF5D1, TP5CTL1 0xFFFFF5E1

	7	6	5	4	3	2	1	0
	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	ソフトウェア・トリガ制御
0	-
1	外部トリガ入力の有効な信号を作成

TPnEEE	カウント・クロックの選択
0	外部イベント・カウント入力での動作禁止
1	外部イベント・カウント入力での動作許可

TPnMD2	TPnMD1	TPnMD0	タイマ・モードの選択
0	0	0	インターバル・タイマ・モード
0	0	1	外部イベント・カウント・モード
0	1	0	外部トリガ・パルス出力モード
0	1	1	ワンショット・パルス出力モード
1	0	0	PWM出力モード
1	0	1	フリー・ランニング・タイマ・モード
1	1	0	パルス幅測定モード
1	1	1	設定禁止

注意 表の赤字部分がサンプル・プログラムでの設定値となります。

4.1.3 タイマ出力の制御

TMPnI/O制御レジスタ0 (TPnIOC0) で、タイマ出力の制御を行います。

インターバル・タイマ・モードとして動作する場合、TPnIOC0レジスタの制御は不要のため、このサンプル・プログラムでは、本レジスタの制御を行っていません。

図4 - 3 TPnIOC0レジスタのフォーマット

TMPnI/O制御レジスタ0 (TPnIOC0)

アドレス : TP0IOC0 0xFFFFF592, TP1IOC0 0xFFFFF5A2,

TP2IOC0 0xFFFFF5B2, TP3IOC0 0xFFFFF5C2,

TP4IOC0 0xFFFFF5D2, TP5IOC0 0xFFFFF5E2

7	6	5	4	3	2	1	0
0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

TPnOL1	TOPn1端子出力レベルの設定
0	TOPn1端子ハイ・レベル・スタート
1	TOPn1端子ロウ・レベル・スタート

TPnOE1	TOPn1端子出力の設定
0	タイマ出力禁止
1	タイマ出力許可

TPnOL0	TOPn0端子出力レベルの設定
0	TOPn0端子ハイ・レベル・スタート
1	TOPn0端子ロウ・レベル・スタート

TPnOE0	TOPn0端子出力の設定
0	タイマ出力禁止
1	タイマ出力許可

注意 TPnIOC0レジスタは、このサンプル・プログラムでは使用していません。

16ビット・タイマ/イベント・カウンタQ(TMQ)では、TQ0IOC0レジスタのビット7-4に、TQ0OL3、TQ0OE3、TQ0OL2、TQ0OE2ビットがそれぞれ割り当てられています。

4.1.4 キャプチャ・トリガ入力信号の有効エッジの制御

TMPnI/O制御レジスタ1 (TPnIOC1) で、キャプチャ・トリガ入力信号 (TIPn0, TIPn1端子) に対する有効エッジを制御します。

インターバル・タイマ・モードとして動作する場合、TPnIOC1レジスタの制御は不要のため、このサンプル・プログラムでは、本レジスタの制御を行っていません。

図4 - 4 TPnIOC1レジスタのフォーマット

TMPnI/O制御レジスタ1 (TPnIOC1)

アドレス : TP0IOC1 0xFFFFF593, TP1IOC1 0xFFFFF5A3,

TP2IOC1 0xFFFFF5B3, TP3IOC1 0xFFFFF5C3,

TP4IOC1 0xFFFFF5D3, TP5IOC1 0xFFFFF5E3

7	6	5	4	3	2	1	0
0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	キャプチャ・トリガ入力信号 (TIPn1端子) の有効エッジの設定
0	0	エッジ検出なし
0	1	立ち上がりエッジを検出
1	0	立ち下がりエッジを検出
1	1	両エッジを検出

TPnIS1	TPnIS0	キャプチャ・トリガ入力信号 (TIPn0端子) の有効エッジの設定
0	0	エッジ検出なし
0	1	立ち上がりエッジを検出
1	0	立ち下がりエッジを検出
1	1	両エッジを検出

注意 TPnIOC1レジスタは、このサンプル・プログラムでは使用していません。

16ビット・タイマ/イベント・カウンタQ (TMQ) では、TQ0IOC1レジスタのビット7-4にTQ0IS7, TQ0IS6, TQ0IS5, TQ0IS4ビットがそれぞれ割り当てられています。

4.1.5 外部入力信号の制御

TMPnI/O制御レジスタ2 (TPnIOC2) で、外部イベント・カウント入力信号 (TIPn0端子)、外部トリガ入力信号 (TIPn0端子) を制御します。

インターバル・タイマ・モードとして動作する場合、TPnIOC2レジスタの制御は不要のため、このサンプル・プログラムでは、本レジスタの制御を行っていません。

図4-5 TPnIOC2レジスタのフォーマット

TMPnI/O制御レジスタ2 (TPnIOC2)

アドレス : TP0IOC2 0xFFFFF594, TP1IOC2 0xFFFFF5A4,

TP2IOC2 0xFFFFF5B4, TP3IOC2 0xFFFFF5C4,

TP4IOC2 0xFFFFF5D4, TP5IOC2 0xFFFFF5E4

7	6	5	4	3	2	1	0
0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

TPnEES1	TPnEES0	外部イベント・カウント入力信号 (TIPn0端子) の有効エッジの指定
0	0	エッジ検出なし
0	1	立ち上がりエッジを検出
1	0	立ち下がりエッジを検出
1	1	両エッジを検出

TPnETS1	TPnETS0	外部トリガ入力信号 (TIPn0端子) の有効エッジの指定
0	0	エッジ検出なし
0	1	立ち上がりエッジを検出
1	0	立ち下がりエッジを検出
1	1	両エッジを検出

注意 TPnIOC2レジスタは、このサンプル・プログラムでは使用していません。

4.1.6 キャプチャ/コンペア動作の制御

TMPnオプション・レジスタ0 (TPnOPT0) で、キャプチャ/コンペア動作の設定、オーバフローの検出を制御します。

インターバル・タイマ・モードとして動作する場合、TPnOPT0レジスタの制御は不要のため、このサンプル・プログラムでは、本レジスタの制御を行っていません。

図4 - 6 TPnOPT0レジスタのフォーマット

TMPnオプション・レジスタ0 (TPnOPT0)

アドレス : TP0OPT0 0xFFFFF595, TP1OPT0 0xFFFFF5A5,

TP2OPT0 0xFFFFF5B5, TP3OPT0 0xFFFFF5C5,

TP4OPT0 0xFFFFF5D5, TP5OPT0 0xFFFFF5E5

7	6	5	4	3	2	1	0
0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1レジスタのキャプチャ/コンペア選択
0	コンペア・レジスタに選択
1	キャプチャ・レジスタに選択

TPnCCS0	TPnCCR0レジスタのキャプチャ/コンペア選択
0	コンペア・レジスタに選択
1	キャプチャ・レジスタに選択

TPnOVF	TMPnのオーバフロー検出フラグ
セット(1)	オーバフロー発生
リセット(0)	TPnOVFビットへの0ライトまたはTPnCTL0.TPnCEビット = 0

注意 TPnOPT0レジスタは、このサンプル・プログラムでは使用していません。

16ビット・タイマ/イベント・カウンタQ (TMQ) では、TQ0OPT0レジスタのビット7-4にTQ0CCS3, TQ0CCS2, TQ0CCS1, TQ0CCS0ビットがそれぞれ割り当てられています。

4.1.7 インターバル時間の制御

TMPnキャプチャ/コンペア・レジスタ0 (TPnCCR0) により、インターバル時間の制御を行うことが可能です。

タイマのカウンタ値が、TPnCCR0レジスタに設定されたカウンタ値となった場合は、INTTPnCC0割り込みが、TPnCCR1レジスタに設定されたカウンタ値となった場合は、INTTPnCC1割り込みが発生します。

このサンプル・プログラムでは、インターバル時間としてLED1の点滅時間を計測する必要があり、TPnCCR0レジスタを使用しています。TPnCCR1レジスタについては使用していません。

$$\cdot \text{インターバル時間} = (N + 1) / f_{CNT}$$

備考 N : TPnCCR0/TPnCCR1レジスタの設定値

f_{CNT} : 16ビット・タイマ/イベント・カウンタのカウンタ・クロック周波数

図4 - 7 TPnCCR0レジスタのフォーマット

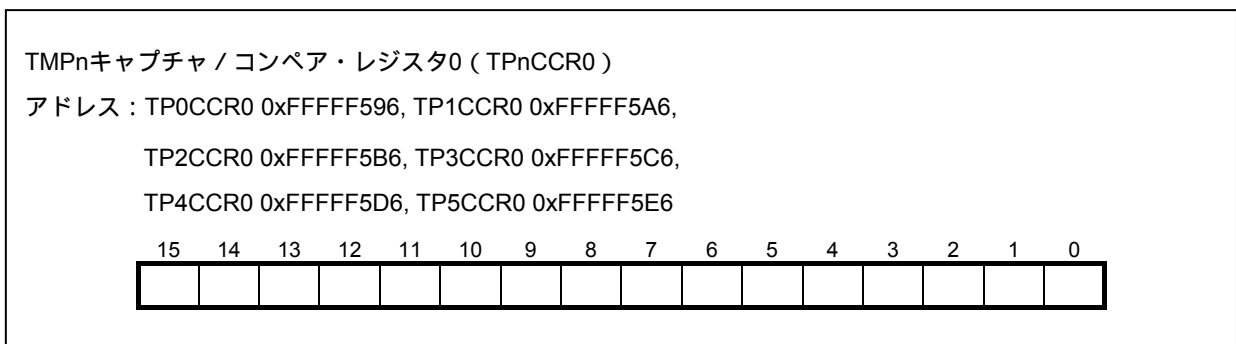
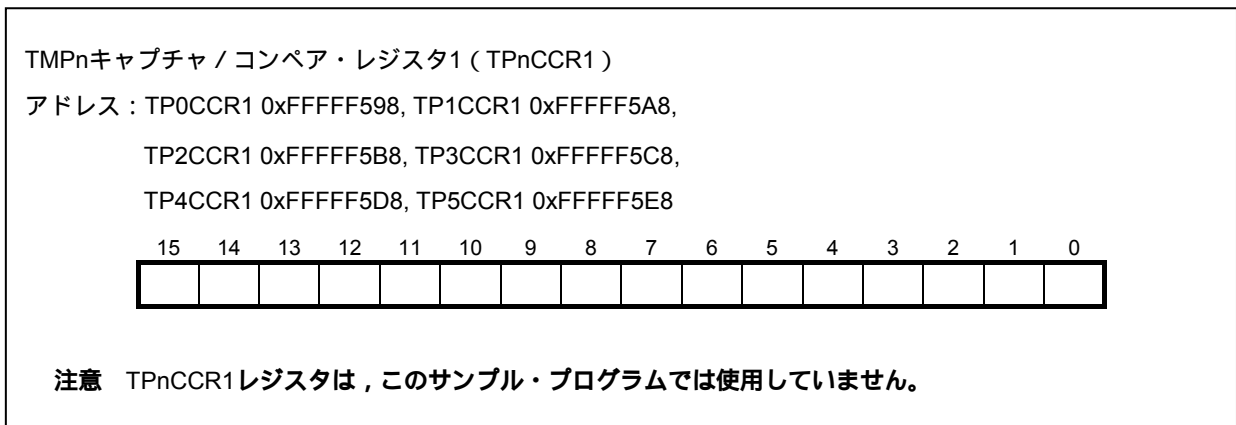


図4 - 8 TPnCCR1レジスタのフォーマット



16ビット・タイマ/イベント・カウンタQ (TMQ) では、TQ0CCR0-TQ0CCR3まで4つのキャプチャ/コンペア・レジスタが存在します。使用方法については、TPnCCR0、TPnCCR1レジスタと同様です。

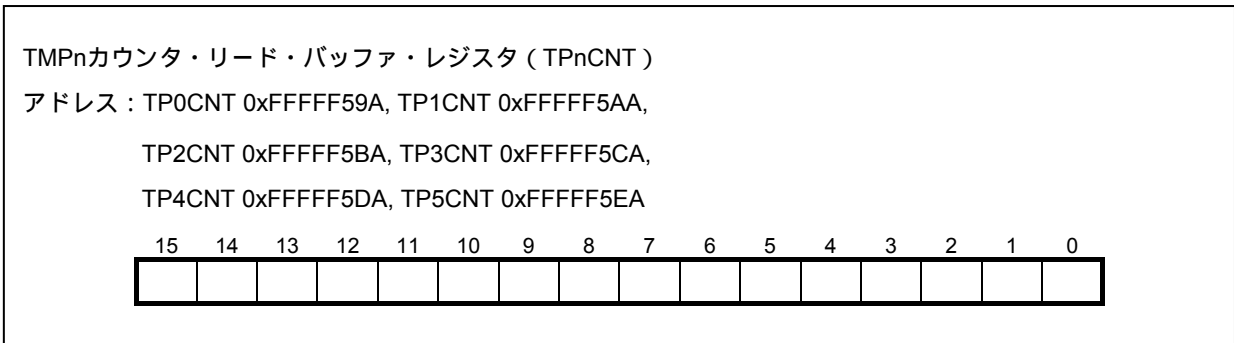
4.1.8 タイマ・カウント値の参照

TPnCNTレジスタは、16ビットのカウント値をリードできるリード・バッファ・レジスタです。

タイマ動作中 (TPnCTL0.TPnCEビット = 1のとき) に、本レジスタの値をリードすることにより、現在のカウント値がリード可能です。

タイマ停止中 (TPnCTL0.TPnCEビット = 0のとき) に、本レジスタの値をリードすると0x0000となります。このサンプル・プログラムでは、カウント値を参照する必要がないため使用していません。

図4 - 9 TPnCNTレジスタのフォーマット



【例 1】16ビット・タイマ/イベント・カウンタP (TMP) をインターバル・タイマ・モードとして使用し、周波数をfxx (fxx = 20 MHz)、インターバル周期を2 msに設定し、タイマ動作を開始する場合 (サンプル・プログラムと同内容)

・設定手順

- カウント・クロックをfxxに設定します。
- インターバル・タイマ・モードに設定します。
- コンペア・レジスタに値を設定します。
- タイマの動作を開始します。
- 割り込みのマスクを解除します。

・プログラム例 (サンプル・プログラムと同内容)

```
#pragma interrupt INTTP0CC0 f_int_inttmp0 /* タイマ(INTTP0CC0)割り込みハンドラ指定*/
                                f_int_inttmp0関数を割り込みハンドラとして登録
const unsigned short COMP_TBL [] ={(40000-1),/* 2 msインターバル用コンペア値 */
                                    (20000-1), /* 1 msインターバル用コンペア値 */
                                    (10000-1), /* 0.5 msインターバル用コンペア値 */
                                    (5000-1)}; /* 0.25 msインターバル用コンペア値 */

static void f_init_timer( void )
{
    /* 変数の初期化 */
    g_blk_ptn_cnt = 0; /* コンペア・レジスタ設定パターン格納変数初期化 */
    g_led_int_cnt = 0; /* タイマ(INTTP0CC0)割り込み回数カウント変数初期化*/

    /* タイマ機能設定 */
    TP0CTL0 = 0x00; /* カウント・クロック = fxx(20 MHz) */
    TP0CTL1 = 0x00; /* インターバル・タイマ・モード指定 */
    TP0CCR0 = COMP_TBL[g_blk_ptn_cnt]; /* コンペア・レジスタに値を設定 */
    TP0CE = 1; /* タイマ(TMP)動作開始 */
    /* CMP_TBL[0]の値(40000-1)を設定することにより、
    インターバル時間を2 msに設定 */
    /* 割り込み機能設定 */
    TP0CCIC0 = 0x07; /* INTTP0CC0 優先度7, マスク解除 */

    return;
}

__interrupt
void f_int_inttmp0( void )
{
    f_int_subtmp0();
    return; /* __interrupt修飾子によりretiに */
}
```

【例 2】16ビット・タイマ/イベント・カウンタQ (TMQ) をインターバル・タイマ・モードとして使用し、周波数をfxx (fxx = 20 MHz)、インターバル周期を2 msに設定し、タイマ動作を開始する場合

・設定手順

- カウント・クロックをfxxに設定します。
- インターバル・タイマ・モードに設定します。
- コンペア・レジスタに値を設定します。
- タイマの動作を開始します。
- 割り込みのマスクを解除します。

・プログラム例

```
#pragma interrupt INTTQ0CC0 f_int_inttmq0 /* タイマ(INTTQ0CC0)割り込みハンドラ指定 */
const unsigned short COMP_TBL[] ={(40000-1), /* 2 msインターバル用コンペア値 */
                                   (20000-1), /* 1 msインターバル用コンペア値 */
                                   (10000-1), /* 0.5 msインターバル用コンペア値 */
                                   (5000-1)}; /* 0.25 msインターバル用コンペア値 */

static void f_init_timer( void )
{
    /* 変数の初期化 */
    g_blk_ptn_cnt = 0; /* コンペア・レジスタ設定パターン格納変数初期化 */
    g_led_int_cnt = 0; /* タイマ(INTTQ0CC0)割り込み回数カウント変数初期化 */

    /* タイマ機能設定 */
    TQ0CTL0 = 0x00; /* カウント・クロック = fxx(20 MHz) */
    TQ0CTL1 = 0x00; /* インターバル・タイマ・モード指定 */
    TQ0CCR0 = COMP_TBL[g_blk_ptn_cnt]; /* コンペア・レジスタに値を設定 */
    TQ0CE = 1; /* タイマ(TMQ)動作開始 */

    /* 割り込み機能設定 */
    TQ0CCIC0 = 0x07; /* INTTQ0CC0 優先度7, マスク解除 */

    return;
}

__interrupt
void f_int_inttmq0( void )
{
    f_int_subtmp0();

    return; /* __interrupt修飾子によりretiに */
}
```

割り込み要因としてINTTQ0CC0を指定

f_int_inttmq0関数を割り込みハンドラとして登録

CMP_TBL[0]の値(40000-1)を設定することにより、インターバル時間を2msに設定

INTTQ0CC0割り込み発生により、割り込み処理開始

4.2 LED1点滅周期とチャタリング検出時間の設定

このサンプル・プログラムでは、LED1点滅周期とチャタリング検出時間を次のように設定しています。

(1) LED1点滅周期の設定

このサンプル・プログラムでは、16ビット・タイマ/イベント・カウンタP (TMP) の割り込み (INTTP0CC0) が250回発生するごとに、LED1出力が反転します。

- ・ 割り込み周期 (インターバル時間) = $(N + 1) / f_{CNT}$
- ・ LED1出力反転周期 = 割り込み周期 × 割り込み回数
- ・ LED1点滅周期 = LED1出力反転周期 × 2

備考 N : TP0CCR0レジスタの設定値

f_{CNT} : 16ビット・タイマ/イベント・カウンタP (TMP) のカウント・クロック周波数

計算例 TP0CCR0レジスタの設定値が39999の場合 (カウント・クロック周波数 = 20 MHz動作時)

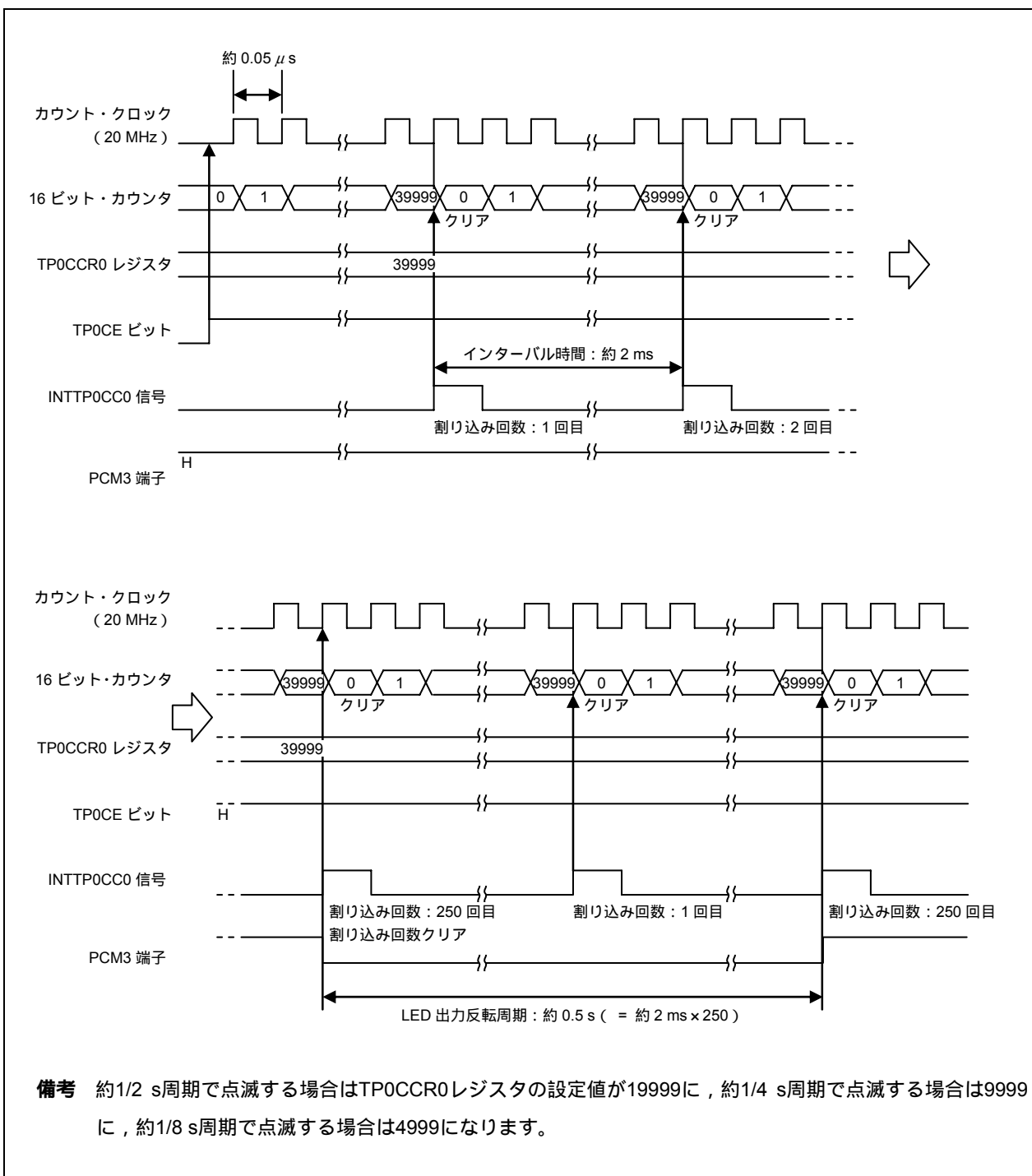
- ・ 割り込み周期 (インターバル時間) = $(N + 1) / f_{CNT} = (39999 + 1) / 20 \text{ MHz} = 2 \text{ ms}$
- ・ LED1出力反転周期 = 割り込み周期 × 割り込み回数 = $2 \text{ ms} \times 250 \text{ 回} = 500 \text{ ms}$
- ・ LED1点滅周期 = LED1出力反転周期 × 2 = $500 \text{ ms} \times 2 = 1 \text{ s}$

また、スイッチ入力の回数に応じて、TP0CCR0レジスタの設置値を変更し、LED1点滅周期を変更しています。

スイッチ入力回数 ^注	TP0CCR0レジスタの設定値	割り込み周期	LED1点滅周期
0	39999	約2 ms ((39999 + 1) / 20 MHz)	約1 s (約2 ms × 250回 × 2)
1	19999	約1 ms ((19999 + 1) / 20 MHz)	約0.5 s (約1 ms × 250回 × 2)
2	9999	約0.5 ms ((9999 + 1) / 20 MHz)	約0.25 s (約0.5 ms × 250回 × 2)
3	4999	約0.25 ms ((4999 + 1) / 20 MHz)	約0.125 s (約0.25 ms × 250回 × 2)

注 4回目以降は、0回目からの点滅周期の繰り返しになります。

図4 - 10 LED1点滅周期のタイミング・チャート例 (LED1が1 s周期で点滅する場合)



備考 約1/2 s周期で点滅する場合はTP0CCR0レジスタの設定値が19999に、約1/4 s周期で点滅する場合は9999に、約1/8 s周期で点滅する場合は4999になります。

(2) チャタリング検出時間の設定

このサンプル・プログラムでは、スイッチ入力（INTP0割り込み発生）時のチャタリング対策として、10 ms以下のチャタリングを除去するように、16ビット・タイマ/イベント・カウンタP（TMP）の割り込み（INTTP0CC0）の発生をカウントしています。

チャタリング除去にINTTP0CC0割り込みを利用することで、チャタリング検出中でも、INTTP0CC0割り込みのカウントを継続することができます。これにより、チャタリングが原因でLED1点滅周期がずれることを抑えることができます。

$$\cdot \text{チャタリング検出時間 (Tc)} = T' + T \times (M - 1)$$

備考 T：INTTP0CC0割り込み周期

T'：INTP0エッジ検出～INTP0エッジ検出後の最初のINTTP0CC0発生までの時間

$$(0 < T' < T)$$

M：INTP0エッジ検出後のINTTP0CC0割り込み回数

$T \times (M - 1) = 10 \text{ ms}$ となるように設定すると、

$$Tc = T' + 10 \text{ ms}$$

$0 < T' < T$ なので、

$$10 \text{ ms} < Tc < T + 10 \text{ ms}$$

チャタリング検出時間 (Tc) > 10 ms

計算例 割り込み周期 (T) が2 ms (4.2 (1) LED1点滅周期の設定の計算例を参照)、INTP0エッジ検出後のINTTP0CC0割り込み回数 (M) が6の場合

$$\begin{aligned} Tc &= T' + T \times (M - 1) \\ &= T' + 2 \text{ ms} \times (6 - 1) \\ &= T' + 10 \text{ ms} \end{aligned}$$

$0 < T' < 2 \text{ ms}$ なので、

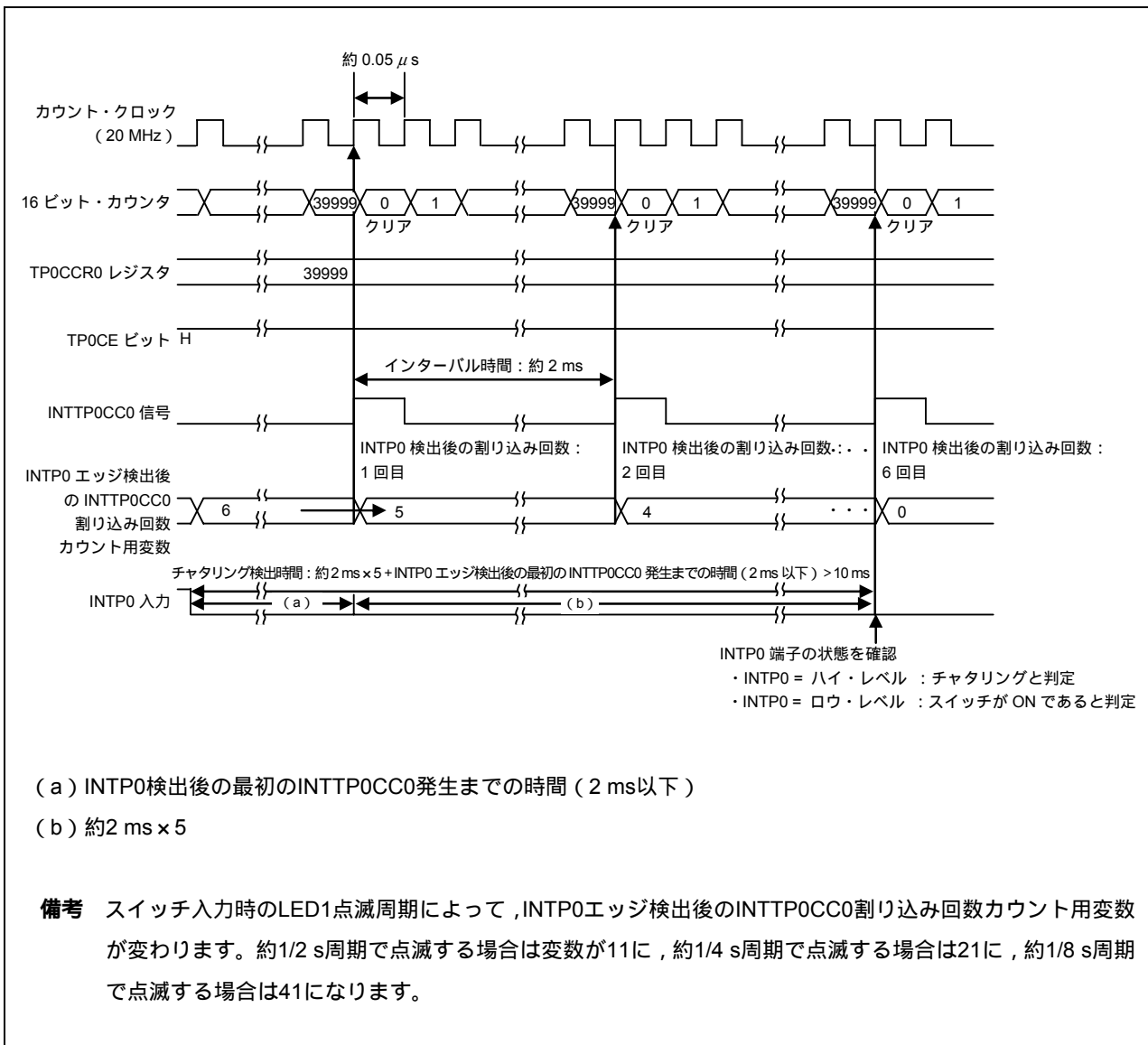
$$10 \text{ ms} < Tc < 12 \text{ ms}$$

チャタリング検出時間 (Tc) > 10 ms

このサンプル・プログラムでは、スイッチ入力時の割り込み周期とINTP0エッジ検出後のINTTP0CC0割り込み回数を次のように対応しています。

LED1点滅周期	割り込み周期	INTP0エッジ検出後のINTTP0CC0割り込み回数	チャタリング検出時間
約1 s	約2 ms	6	10 ms < Tc < 12 ms
約0.5 s	約1 ms	11	10 ms < Tc < 11 ms
約0.25 s	約0.50 ms	21	10 ms < Tc < 10.50 ms
約0.125 s	約0.25 ms	41	10 ms < Tc < 10.25 ms

図4 - 11 チャタリング検出のタイミング・チャート例（スイッチ入力時にLED1が約1 s周期で点滅している場合）



第5章 関連資料

資料名	和文 / 英文
V850ES/JF3-L ハードウェア編 ユーザーズ・マニュアル	PDF
V850ES/JG3-L ハードウェア編 ユーザーズ・マニュアル	PDF
PM+ Ver.6.30 ユーザーズ・マニュアル	PDF
CA850 Ver.3.20 Cコンパイラ・パッケージ 操作編 ユーザーズ・マニュアル	PDF
CA850 Ver.3.20 Cコンパイラ・パッケージ C言語編 ユーザーズ・マニュアル	PDF
CA850 Ver.3.20 Cコンパイラ・パッケージ アセンブリ言語編 ユーザーズ・マニュアル	PDF
CA850 Ver.3.20 Cコンパイラ・パッケージ リンク・ディレクティブ編 ユーザーズ・マニュアル	PDF
V850ES アーキテクチャ編 ユーザーズ・マニュアル	PDF
QB-MINI2 プログラミング機能付きオンチップ・デバッグ・エミュレータ ユーザーズ・マニュアル	PDF
ID850QB Ver.3.40 統合デバッガ 操作編 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、V850ES/Jx3-L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。

```
opt_b.s
#-----
#
#   NEC Electronics   V850ES/Jx3-L マイクロコントローラ
#
#-----
#   V850ES/JG3-L JF3-L サンプル・プログラム
#-----
#   インターバル・タイマ・モード編
#-----
# 【履歴】
#   2009.1.--   新規作成
#-----
# 【概要】
#   本サンプル・プログラムは、オプション・バイトの設定を行う
#-----

.section "OPTION_BYTES"
.byte 0b00000101 -- 0x7a (5MHz: 発振安定時間 6.554ms に設定)
.byte 0b00000000 -- 0x7b
.byte 0b00000000 -- 0x7c
.byte 0b00000000 -- 0x7d 0x7b-0x7f番地には0x00を設定する必要がある
.byte 0b00000000 -- 0x7e
.byte 0b00000000 -- 0x7f
```

```
minicube2.s
#-----
#
#   NEC Electronics   V850ES/Jx3-L マイクロコントローラ
#
#-----
#   V850ES/JG3-L JF3-L サンプル・プログラム
#-----
#   インターバル・タイマ・モード編
#-----
# 【履歴】
#   2009.1.--   新規作成
#-----
# 【概要】
#   本サンプル・プログラムは、MINICUBE2使用時に必要なリソースの確保を行っている
#   (CSIB0を用いてMINICUBE2を使用する場合の例)
#-----

--monitorROMセクションとして2Kバイトの空間を確保
.section "MonitorROM", const
.space 0x800, 0xff

--デバッグ用割り込みベクタの確保
.section "DBG0"
.space 4, 0xff

--シリアル通信用受信割り込みベクタの確保
.section "INTCB0R"
.space 4, 0xff

--MonitorRAMセクションとして16バイトの空間を確保
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4
```

```

AppNote_LVI.dir

# Sample link directive file (not use RTOS/use internal memory only)
#
# Copyright (C) NEC Electronics Corporation 2002
# All rights reserved by NEC Electronics Corporation.
#
# This is a sample file.
# NEC Electronics assumes no responsibility for any losses incurred by customers or
# third parties arising from the use of this file.
#
# Generated      : PM+ V6.31 [ 9 Jul 2007]
# Sample Version : E1.00b [12 Jun 2002]
# Device        : uPD70F3738 (C:¥Program Files¥NEC Electronics Tools¥DEV¥DF3738.800)
# Internal RAM   : 0x3ffb000 - 0x3ffefff
#
# NOTICE:
# Allocation of SCONST, CONST and TEXT depends on the user program.
#
# If interrupt handler(s) are specified in the user program then
# the interrupt handler(s) are allocated from address 0 and
# SCONST, CONST and TEXT are allocated after the interrupt handler(s).

SCONST : !LOAD ?R {
    .sconst      = $PROGBITS ?A .sconst;
};

CONST : !LOAD ?R {
    .const       = $PROGBITS ?A .const;
};

TEXT : !LOAD ?RX {
    .pro_epi_runtime = $PROGBITS ?AX .pro_epi_runtime;
    .text           = $PROGBITS ?AX .text;
};

### MINICUBE2用###
MROMSEG : !LOAD ?R 0x03F800{
    MonitorROM = $PROGBITS ?A MonitorROM;
};

```

内蔵ROMサイズが128 Kバイト
製品の場合は 0x01F800

デフォルトのリンク・ディレティブ
ファイルとの差分（追加コード）

MINICUBE2用の予約領域を確保。

```

SIDATA : !LOAD ?RW V0x3ffb000 {
    .tidata.byte    = $PROGBITS ?AW .tidata.byte;
    .tibss.byte     = $NOBITS ?AW .tibss.byte;
    .tidata.word    = $PROGBITS ?AW .tidata.word;
    .tibss.word     = $NOBITS ?AW .tibss.word;
    .tidata         = $PROGBITS ?AW .tidata;
    .tibss          = $NOBITS ?AW .tibss;
    .sidata         = $PROGBITS ?AW .sidata;
    .sibss          = $NOBITS ?AW .sibss;
};

```

```

DATA : !LOAD ?RW V0x3ffb100 {
    .data           = $PROGBITS ?AW .data;
    .sdata          = $PROGBITS ?AWG .sdata;
    .sbss           = $NOBITS ?AWG .sbss;
    .bss            = $NOBITS ?AW .bss;
};

```

```

### MINICUBE2用###
MRAMSEG : !LOAD ?RW V0x03FFEFF0{
MonitorRAM = $NOBITS ?AW MonitorRAM;
};

```

デフォルトのリンク・ディレクティブ
ファイルとの差分（追加コード）

MINICUBE2用の予約領域を確保。

```

__tp_TEXT @ %TP_SYMBOL;
__gp_DATA @ %GP_SYMBOL & __tp_TEXT{DATA};
__ep_DATA @ %EP_SYMBOL;

```

```
main.c
/*-----*/
/*
/*   NEC Electronics   V850ES/Jx3-L マイクロコントローラ
/*
/*-----*/
/*   V850ES/JG3-L サンプル・プログラム
/*-----*/
/*   インターバル・タイマ・モード編
/*-----*/
/* 【履歴】
/*   2009.1.--   新規作成
/*-----*/
/* 【概要】
/*   本サンプル・プログラムは、16ビット・タイマ/イベント・カウンタ (TMP)
/*   インターバル・タイマ・モード機能の使用例を示すものです。
/*   16ビット・タイマ/イベント・カウンタ (TMP) 割り込みを利用してPCM3端子出力を反転し、
/*   LED1を点滅させます。
/*   LED1の点滅周期は、スイッチ入力割り込み時にタイマのコンペア・レジスタを書き換える
/*   ことで変更します。
/*
/*
/*   <主な設定内容>
/*   ・pragma指令にて、割り込みハンドラの設定、周辺IOレジスタ名を記述可能にする
/*   ・プロトタイプ宣言を実施
/*   ・コンペア・レジスタ設定テーブル、10msウエイト用カウント値テーブルを定義
/*   ・コンペア・レジスタ設定パターン格納変数、タイマ (INTTP0CC0) 割り込み回数カウント変数を定義
/*   ・内蔵周辺I/Oへのバス・ウエイト、ウォッチドッグ・タイマの動作停止、クロック設定処理
/*   ・未使用ポートの初期化
/*   ・スイッチ (INTP0) 割り込みポート (立ち下りエッジ)、LED1出力ポートの初期化
/*   ・タイマ (TMP) の初期化
/*   <スイッチ (INTP0) 割り込み処理>
/*   ・コンペア・レジスタ設定パターン更新
/*     (スイッチ入力時のチャタリング除去時間10ms)
/*   <タイマ (INTTP0CC0) 割り込み処理>
/*   ・LED1の点滅処理
```

```

/*
/* <スイッチ入力回数とLED1点灯パターン>
/*
/* +-----+
/* | SW入力回数 | LED1点滅周期 |
/* | (P03/INTP0) | (PCM3) |
/* |-----|-----|
/* | 0回 | 1 秒 |
/* | 1回 | 1/2 秒 |
/* | 2回 | 1/4 秒 |
/* | 3回 | 1/8 秒 |
/* +-----+
/* # 4回目以降は0回からの繰り返し
/*
/*
/* 【ポート入出力の設定】
/*
/* 入力ポート：P03 (INTP0)
/* 出力ポート：PCM3
/* 空きポート：P02,P04-P06, P10-P11, P30-P39, P50-P55, P70-P711, P90-P915,
/* PCM0-PCM2, PCT0, PCT1, PCT4, PCT6, PDH0-PDH5, PDL0-PDL15
/* 空きのポートは全て出力ポート (Low出力) に設定しておく
/*
/*-----*/

/*-----*/
/* pragma指令 */
/*-----*/
#pragma ioreg /* 周辺IOレジスタを記述可にする */
#pragma interrupt INTP0 f_int_intp0 /* スイッチ (INTP0) 割り込みハンドラ指定*/
#pragma interrupt INTP0CC0 f_int_inttmp0 /* タイマ (INTTP0CC0) 割り込みハンドラ指定*/

/*-----*/
/* プロトタイプ宣言 */
/*-----*/

void main( void ); /* メイン関数 */
static void f_init( void ); /* 初期化関数 */
static void f_init_clk_bus_wdt2( void ); /* クロック・バス・WDT初期化関数 */
static void f_init_port_func( void ); /* ポート/兼用機能初期化関数 */
static void f_init_timer( void ); /* タイマ初期化関数 */
static void f_int_subtmp0( void ); /* タイマ (INTTP0CC0) 割り込みサブルーチン*/

```

TMQを使用する場合は、INTTQ0CC0を指定

```
/*-----*/
/*      テーブルの設定      */
/*-----*/
const unsigned short COMP_TBL[] ={(40000-1), /* 2msインターバル用コンペア値 */
                                   (20000-1), /* 1msインターバル用コンペア値 */
                                   (10000-1), /* 0.5msインターバル用コンペア値 */
                                   (5000-1)}; /* 0.25msインターバル用コンペア値 */

const unsigned char WAIT_TBL[] = {(5+1), /* 10msウエイトの調整用定数定義(2msインターバル用) */
                                   (10+1), /* 10msウエイトの調整用定数定義(1msインターバル用) */
                                   (20+1), /* 10msウエイトの調整用定数定義(0.5msインターバル用) */
                                   (40+1)}; /* 10msウエイトの調整用定数定義(0.25msインターバル用) */

/*-----*/
/*      グローバル変数の設定      */
/*-----*/
static unsigned char g_blk_ptn_cnt; /* コンペア・レジスタ設定パターン格納変数 */
static unsigned char g_led_int_cnt; /* タイマ(INTTP0CC0) 割り込み回数カウンタ変数 */

/*****
/*      メインモジュール      */
*****/
void main(void)
{
    f_init(); /* 初期化実施 */

    __EI(); /* 割り込み許可 */

    while(1); /* メイン・ループ(無限ループ) */

    return;
}
```

```

/*-----*/
/*      初期化モジュール      */
/*-----*/
static void f_init( void )
{
    f_init_clk_bus_wdt2();          /* 内蔵周辺I/Oへのバス・ウエイト ,WDT2の停止 ,クロック設定処理 */

    f_init_port_func();           /* ポート/兼用機能の設定 */

    f_init_timer();               /* タイマ (TMP)の初期設定*/

    return;
}

/*-----*/
/* クロック・バス・WDT2初期化      */
/*-----*/
static void f_init_clk_bus_wdt2( void )
{
    VSWC = 0x01;                  /* 内蔵周辺へのバス・ウエイト設定 */
                                /* OCDMを「通常動作モード」に指定 */

    #pragma asm
        st.b   r0, PRCMD
        st.b   r0, OCDM
    #pragma endasm

    RSTOP = 1;                   /* 内蔵発振器を停止          */
    WDTM2 = 0x00;                /* ウォッチドッグ・タイマを停止 */

                                /* クロック分周を「なし」に設定 */

    #pragma asm
        push   r10
        mov    0x80, r10
        st.b   r10, PRCMD
        st.b   r10, PCC
        pop    r10
    #pragma endasm

    PLLCTL = 0x03;               /* PLLモードに設定 */

    return;
}

```

```

#pragma asm
    st.b   r0, PRCMD
    st.b   r0, OCDM
#pragma endasm

```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```

#pragma asm
    push   r10
    mov    0x80, r10
    st.b   r10, PRCMD
    st.b   r10, PCC
    pop    r10
#pragma endasm

```



```

/*-----*/
/*   ポート/兼用機能の設定   */
/*-----*/
static void f_init_port_func( void )
{
    P0   = 0x00;          /* P02-P06を出力LOWに設定 */
    PM0  = 0x8B;          /* P03を入力ラッチに接続 */
    PMC0 = 0x08;          /* P03をINTP0入力に設定 */

    P1   = 0x00;          /* P10,P11を出力LOWに設定 */
    PM1  = 0xFC;          /* JF3-L版での設定値は0xFE   JF3-LではP10のみ */
    P3   = 0x0000;        /* P30-P39を出力LOWに設定 */
    PM3  = 0xFC00;        /* JF3-L版での設定値は0xFCC0 JF3-LではP30-P35,P38,P39 */
    PMC3 = 0x0000;

#ifdef MINICUBE2
    /* MINICUBE2使用時はP4をCSIB0として使用する為, */
    /* P4は空き端子として初期化しない(QB-V850ESJG3L-TB) */
    P4   = 0x00;          /* P40-P42を出力LOWに設定 */
    PM4  = 0xF8;
    PMC4 = 0x00;
#endif

    P5   = 0x00;          /* P50-P55を出力LOWに設定 */
    PM5  = 0xC0;
    PMC5 = 0x00;

    P7H  = 0x00;          /* P70-P71を出力LOWに設定 */
    P7L  = 0x00;          /* JF3-L版では,レジスタが JF3-L版ではP70-P77
    PM7H = 0xF0;          存在しない為,設定しない。
    PM7L = 0x00;

    P9   = 0x0000;        /* P90-P915を出力LOWに設定 */
    PM9  = 0x0000;        /* JF3-L版での設定値は0x1C3C JF3-L版ではP90,P91,P96-P99,P913-P915
    PMC9 = 0x0000;

    PCM  = 0x08;          /* PCM0-PCM2を出力LOWに,PCM3に消灯パターンを設定*/
    PMCM = 0xF0;
    PMCCM = 0x00;

    PCT  = 0x00;          /* PCT0,1,4,6を出力LOWに設定 */
    PMCT = 0xAC;

```

```

PMCCCT = 0x00;

PDH = 0x00; /* PDH0-PDH5を出力LOWに設定 */
PMDH = 0xC0; /* JF3-L版での設定値は0xFC JF3-L版ではPDH0,PDH1 */
PMCDH = 0x00;

PDL = 0x0000; /* PDL0-PDL15を出力LOWに設定 */
PMDL = 0x0000;
PMCDL = 0x0000;

/* 割り込み機能設定 */
INTF0 = 0x08; /* INTPO立ち下りエッジ指定 */
INTRO = 0x00; /*
PIC0 = 0x07; /* INTPO 優先度7, マスク解除

return;
}

/*-----*/
/* タイマの初期設定 */
/*-----*/
static void f_init_timer( void )
{
    /* タイマ機能設定 */
    g_blk_ptn_cnt = 0; /* コンペア・レジスタ設定パターン格納変数初期化*/
    g_led_int_cnt = 0; /* タイマ(INTTP0CC0) 割り込み回数カウンタ変数初期化*/

    /* タイマ機能設定 */
    TP0CTL0 = 0x00; /* カウント・クロック = fxx(20MHz) */
    TP0CTL1 = 0x00; /* インターバル・タイマ・モード指定 */
    TP0CCR0 = COMP_TBL[g_blk_ptn_cnt]; /* コンペア・レジスタに値を設定 */
    TP0CE = 1; /* タイマ(TMP) 動作開始

    /* 注意：インターバル・タイマ・モードでは、以下のレジスタは設定しません */
    /* <未使用レジスタ> */
    /* - TP0ICC0レジスタ */
    /* - TP0ICC1レジスタ */
    /* - TP0ICC2レジスタ */
    /* - TP0OPT0レジスタ */
    /* - TP0CNTレジスタ

    /* 割り込み機能設定 */
    TP0CCIC0 = 0x07; /* INTTP0CC0 優先度7, マスク解除

return;
}

```

```

/*****
/* スイッチ (INTP0) モジュール */
/*****

__interrupt
void f_int_intp0( void )
{
    unsigned int loop_wait;

    /* チャタリング用 10msウエイト */
    for( loop_wait = WAIT_TBL[g_blk_ptn_cnt]; loop_wait > 0; loop_wait-- )
    {
        while( !TPOCCIF0 ) /* タイマ (INTTPOCC0) 割り込み要求フラグ判定 */
        {
            __nop();
            /* TMQを使用する場合は、
            TQOCCIF0ビットの値を監視 */
        }
        TPOCCIF0 = 0; /* タイマ (INTTPOCC0) 割り込み要求フラグをクリア */
        f_int_subtmp0(); /* タイマ (INTTPOCC0) 割り込み処理 */
    }
    /* TMQを使用する場合は、
    TQOCCIF0ビットに値を設定 */
    if( ( P0 & 0x08 ) == 0x00 ) /* ウエイト後のSW1押下状態認識 */
    {
        g_blk_ptn_cnt++; /* コンペア・レジスタ設定パターンを変更(4種類) */
        g_blk_ptn_cnt &= 3;
        /* TMQを使用する場合は、
        TQOCCR0レジスタに値を設定 */
        TPOCCR0 = COMP_TBL[g_blk_ptn_cnt]; /* コンペア・レジスタ値変更*/

        g_led_int_cnt = 0; /* タイマ (INTTPOCC0) 割り込み回数をクリア*/
    }

    PIC0 &= (unsigned char)~0x80; /* FailSafe：複数要求をクリア */

    return; /* __interrupt修飾子によりretiに */
}

```

```
/* **** */
/* タイマ (INTTP0CC0) 割り込みモジュール */
/* **** */

__interrupt
void f_int_inttmp0( void )
{
    f_int_subtmp0();

    return; /* __interrupt修飾子によりretiに */
}

/* **** */
/* タイマ (INTTP0CC0) 割り込みサブルーチン */
/* **** */

static void f_int_subtmp0( void )
{
    g_led_int_cnt++; /* タイマ (INTTP0CC0) 割り込み発生回数を更新 */

    if( g_led_int_cnt >= 250 ) /* タイマ (INTTP0CC0) 割り込み発生回数の判定 */
    {
        g_led_int_cnt = 0; /* タイマ (INTTP0CC0) 割り込み回数カウンタ変数を初期化 */
        PCM.3 ^= 1; /* LED1を反転 */
    }

    return;
}
```

〔メモ〕

【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

【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/>

【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。

お問い合わせ先

【営業関係、デバイスの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