

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、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

アプリケーション・ノート

78K0S/Kx1+

サンプル・プログラム（16ビット・タイマ/イベント・カウンタ00）

PPG出力編

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00のPPG出力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00のPPG出力機能を使用して、任意の周期とパルス幅を持った矩形波を出力します。また、スイッチ入力回数に応じて、PPG出力のデューティを変更し、LEDの輝度を変更します。

対象デバイス

78K0S/KA1+マイクロコントローラ
 78K0S/KB1+マイクロコントローラ
 78K0S/KU1+マイクロコントローラ
 78K0S/KY1+マイクロコントローラ

目次

第1章 概要 ...	3
1.1 初期設定の主な内容 ...	3
1.2 メイン・ループ以降の内容 ...	3
第2章 回路図 ...	5
2.1 回路図 ...	5
2.2 周辺ハードウェア ...	5
第3章 ソフトウェアについて ...	6
3.1 ファイル構成 ...	6
3.2 使用する内蔵周辺機能 ...	7
3.3 初期設定と動作概要 ...	7
3.4 フロー・チャート ...	9
第4章 設定方法について ...	10
4.1 16ビット・タイマ/イベント・カウンタ00のPPG出力機能の設定 ...	10
4.2 PPG出力のデューティ設定と変更方法 ...	22
4.2.1 PPG出力のデューティ設定 ...	22
4.2.2 CR010の設定値（PPG出力のデューティ）の変更方法 ...	23
4.3 チャタリング検出時間の設定 ...	27
第5章 システム・シミュレータ SM+での動作確認 ...	29
5.1 サンプル・プログラムのビルド ...	29
5.2 SM+での動作 ...	31
第6章 関連資料 ...	36
付録A プログラム・リスト ...	37
付録B 改版履歴 ...	53

資料番号 U18890JJ2V0AN00（第2版）

発行年月 July 2008 NS

- 本資料に記載されている内容は2008年7月現在のものです、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。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等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。

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00のPPG出力機能の使用例を示しています。PPG出力のデューティを制御し、スイッチ入力回数に応じて、LEDの輝度を変更します。

1.1 初期設定の主な内容

初期設定の主な内容は、次のとおりです。

システム・クロック・ソースとして、高速内蔵発振器を選択^注

ウォッチドッグ・タイマの動作停止

V_{LVI} (低電圧検出電圧) を4.3 V ± 0.2 V に設定

V_{DD} (電源電圧) V_{LVI} になったあとに、V_{DD} < V_{LVI} を検出した場合、内部リセット (LVIリセット) 信号を発生

CPUクロック周波数を8 MHzに設定

入出力ポートの設定

16ビット・タイマ/イベント・カウンタ00の設定

- ・ CR000とCR010をコンペア・レジスタに設定
- ・ PPG出力の1周期を200 μs (0.5 μs × 400) に設定
- ・ PPG出力のパルス幅を20 μs (0.5 μs × 40) に設定
- ・ カウント・クロックをf_{XP}/2² (2 MHz) に設定
- ・ CR000とTM00の一致およびCR010とTM00の一致によるタイマ出力反転を許可に設定
- ・ タイマ出力初期値を1 (タイマ出力F/Fをセット (1)) に設定
- ・ タイマ出力 (TO00端子出力) を許可に設定
- ・ 動作モードをTM00とCR000の一致でクリア&スタートに設定

INTP1 (外部割り込み) の有効エッジを立ち下がりエッジに設定

INTP1の割り込みを許可

注 オプション・バイトで設定します。

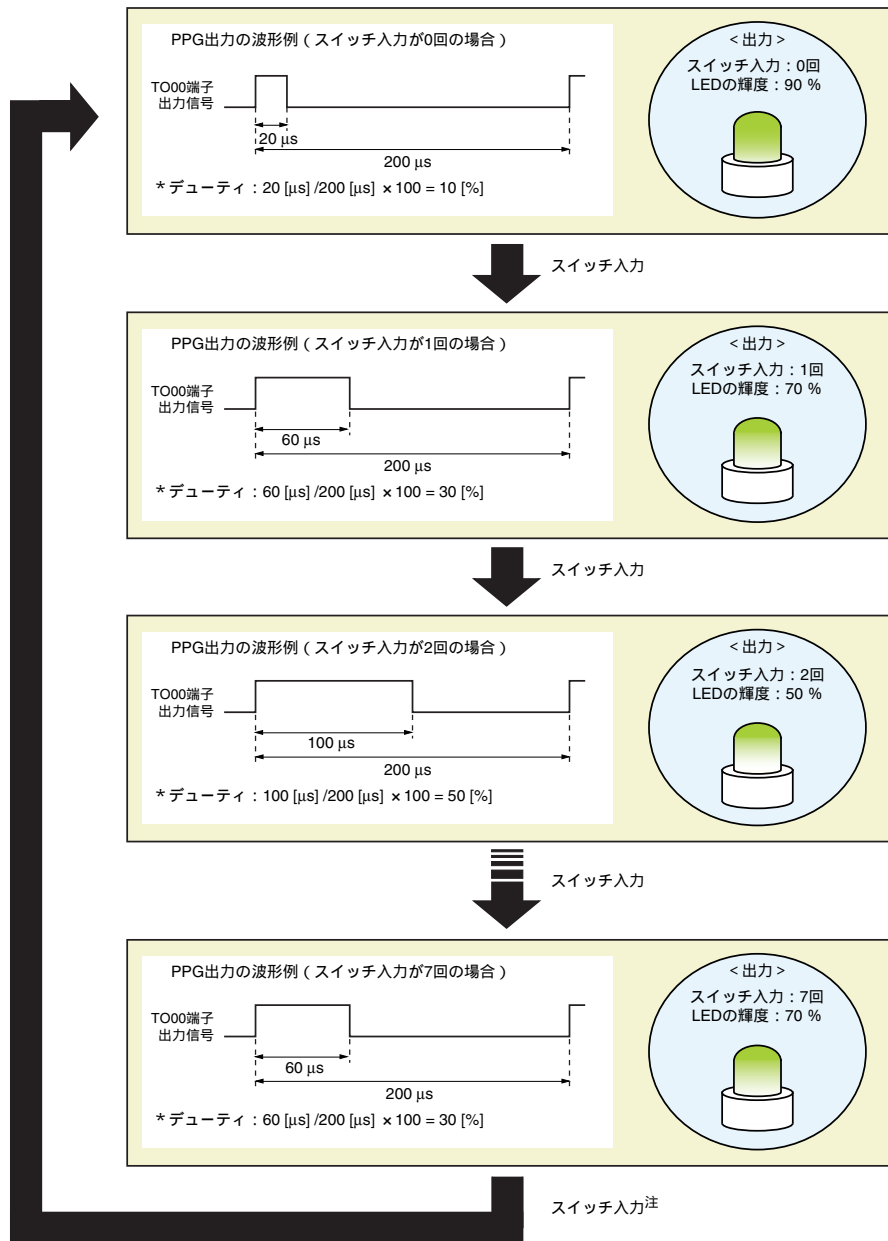
1.2 メイン・ループ以降の内容

初期設定完了後は、16ビット・タイマ/イベント・カウンタ00のPPG出力により、LEDを一定の輝度で点灯します。

スイッチ入力によるINTP1端子の立ち下がりエッジを検出したら、INTP1の割り込み処理を行います。INTP1端子の立ち下がりを検出してから、約10 ms経過後に、INTP1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、INTP1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、PPG出力のデューティを変更し、LEDの輝度を変更します。

スイッチ入力回数 ^注	0回	1回	2回	3回	4回	5回	6回	7回
PPG出力のデューティ	10 %	30 %	50 %	70 %	90 %	70 %	50 %	30 %
LEDの輝度	90 %	70 %	50 %	30 %	10 %	30 %	50 %	70 %

このサンプル・プログラムでは、PPG出力のアクティブ・レベルをハイ・レベルに設定し、LEDがロウ・レベル時に点灯するので、「LEDの輝度 = 100 - PPG出力のデューティ」となります。



注 スwitch入力が8回目以降の場合、PPG出力のデューティは0回目からの繰り返しになります。

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 ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

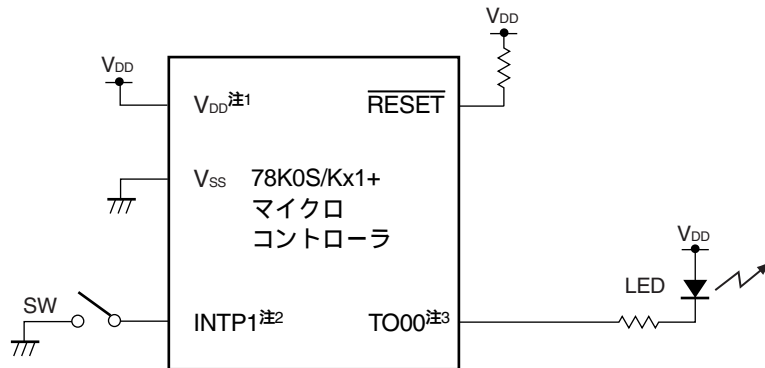
💡 【コラム】チャタリングとは
 スwitchが切り替わった直後に、接点が機械的にばたつくことにより、電気信号がONとOFFを繰り返す現象のことです。

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。



注1. 4.5 V V_{DD} 5.5 Vの電圧範囲で使用してください。

- INTP1/TxD6/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ
- TO00/TI010/INTP2/P31: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
TO00/TI010/INTP0/ANI1/P21: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

注意1. AV_{REF} 端子は V_{DD} に直接接続してください(78K0S/KA1+, 78K0S/KB1+マイクロコントローラのみ)。

2. AV_{SS} 端子はGNDに直接接続してください(78K0S/KB1+マイクロコントローラのみ)。

3. 回路図中の端子および AV_{REF} , AV_{SS} 端子以外の未使用端子はすべて出力ポートのため、オープン(未接続)にしてください。

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。

(1) スイッチ (SW)

LEDの輝度制御用の入力として、スイッチを使用します。

(2) LED




16ビット・タイマ/イベント・カウンタ00のPPG出力機能とスイッチ入力に対応した出力として、LED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。

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とメイン処理のソース・ファイル	注1	注1	
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル (システム・クロック・ソースなどを設定)			
tm00ppg.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
tm00ppg.prj	統合開発環境 PM+用プロジェクト・ファイル			
tm00ppg.pri tm00ppg.prs tm00ppg.prm	システム・シミュレータ SM+ for 78K0S/Kx1+用プロジェクト・ファイル		注2	
tm00ppg0.pnl	システム・シミュレータ SM+ for 78K0S/Kx1+用入出力パネル・ファイル (周辺ハードウェア動作を確認するために使用)		注2	
tm00ppg0.wvo	システム・シミュレータ SM+ for 78K0S/Kx1+用タイミング・チャート・ファイル (波形を確認するために使用)			

注1. 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。

2. 78K0S/KU1+マイクロコントローラには、同封されていません。

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+とシステム・シミュレータ SM+ for 78K0S/Kx1+で使用するファイルを同封



: システム・シミュレータ SM+ for 78K0S/Kx1+で使用するマイコン動作シミュレーション・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。

- ・ PPG出力機能 : 16ビット・タイマ/イベント・カウンタ00
- ・ $V_{DD} < V_{LVI}$ 検出 : 低電圧検出 (LVI) 回路
- ・ スイッチ入力 : INTP1^{注1} (外部割り込み)
- ・ PPG出力 (LED出力) : TO00^{注2} (タイマ出力)

- 注1. INTP1/TxD6/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ
2. TO00/TI010/INTP2/P31: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 TO00/TI010/INTP0/ANI1/P21: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

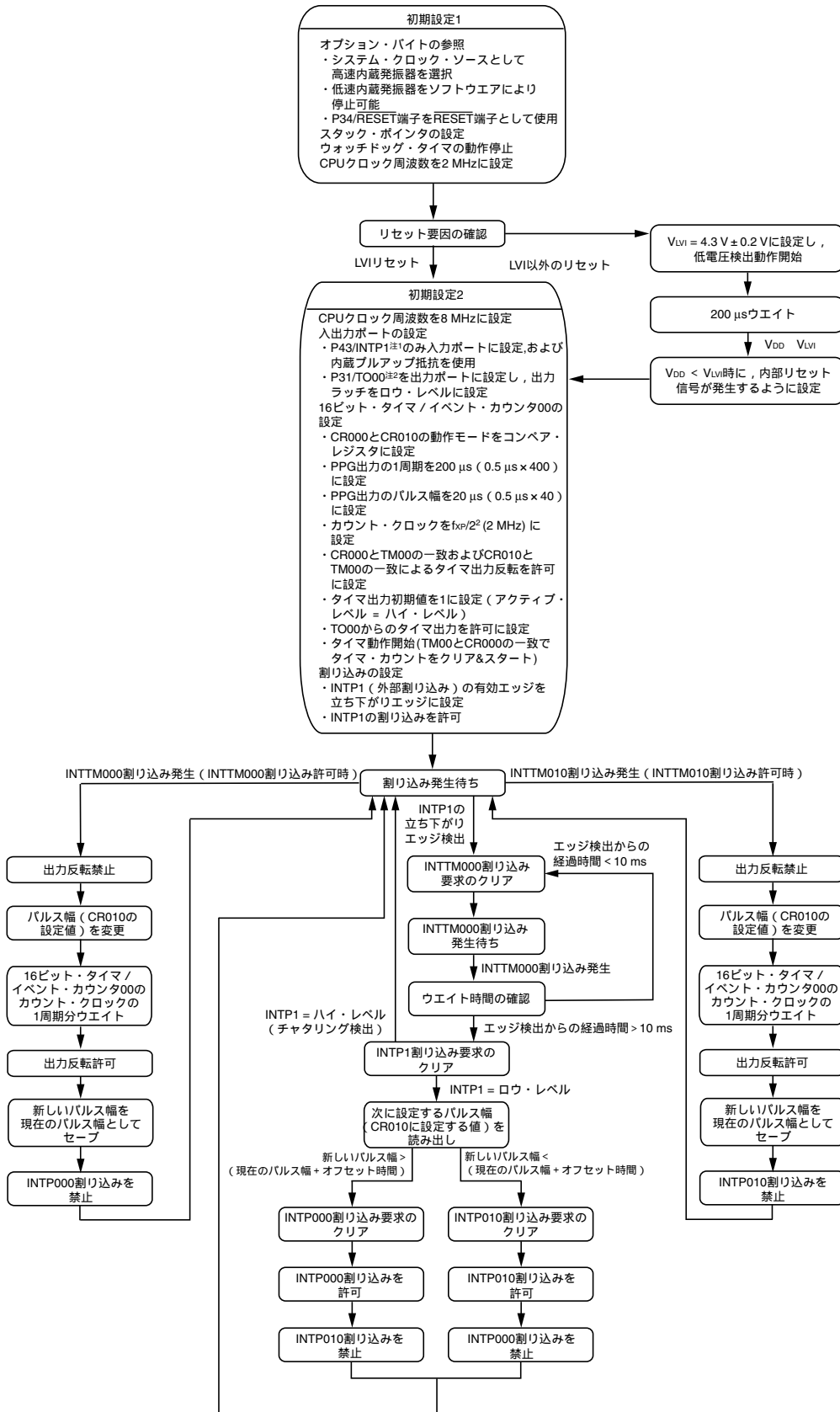
3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、低電圧検出機能の設定、クロック周波数の選択、入出力ポートの設定、16ビット・タイマ/イベント・カウンタ00 (PPG出力機能) の設定、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00のPPG出力により、LEDを一定の輝度で点灯します。

スイッチ入力によるINTP1端子の立ち下がりエッジを検出したら、INTP1の割り込み処理を行います。INTP1端子の立ち下がりを検出してから、約10 ms経過後に、INTP1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、INTP1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、PPG出力のデューティを変更し、LEDの輝度を変更します。

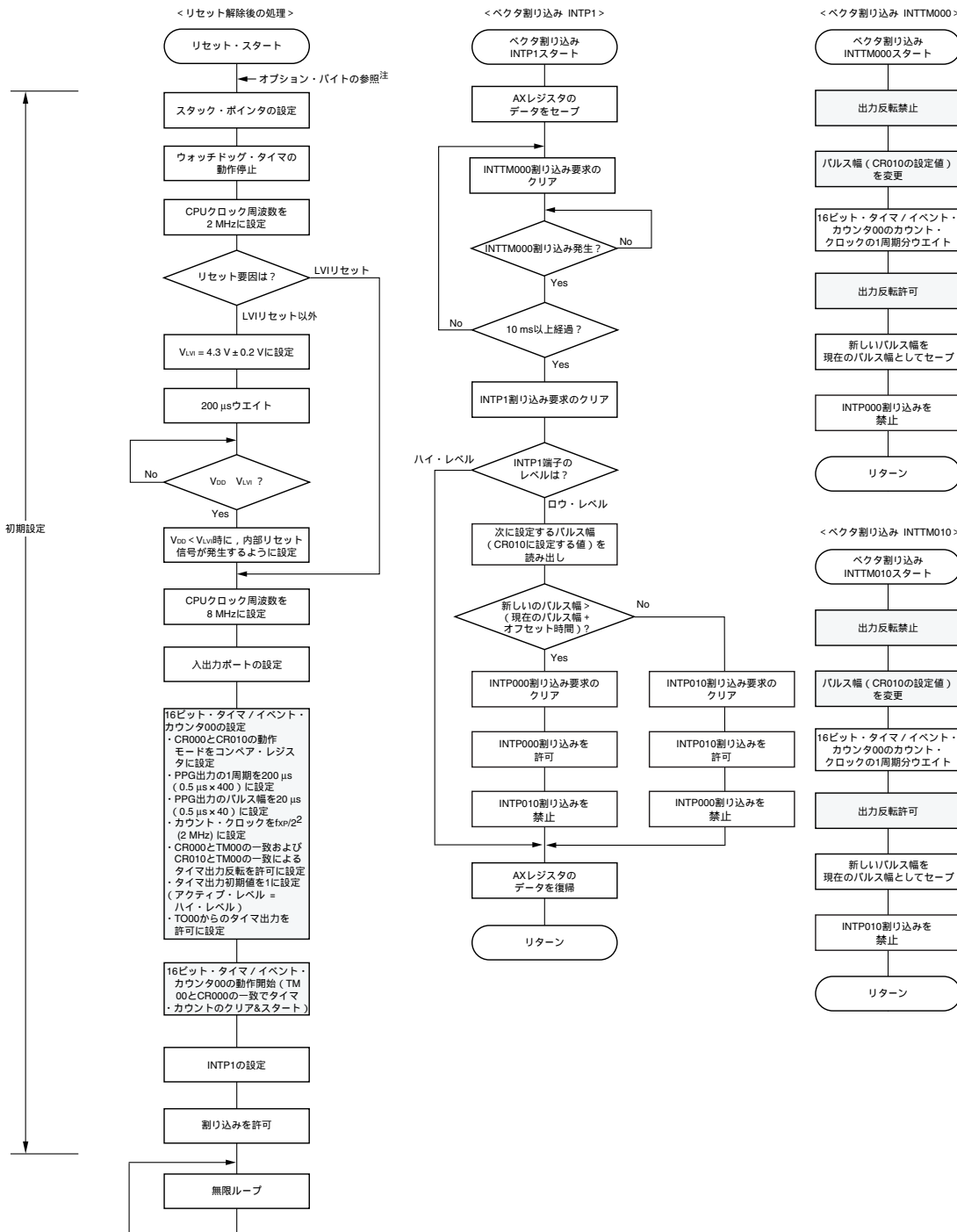
詳細については、次の状態遷移図（ステート・チャート）に示します。



- 注1. INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ
2. TO00/P31: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 TO00/P21: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

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 オプション・バイトの参照は、リセット解除後に、マイコンが自動的にを行います。このサンプル・プログラムでは、オプション・バイトの参照により、次の内容が設定されます。

- ・システム・クロック・ソースとして、高速内蔵発振クロック (8 MHz (TYP.)) を使用
- ・低速内蔵発振器をソフトウェアで停止可
- ・P34/RESET端子をRESET端子として使用

第4章 設定方法について

この章では、16ビット・タイマ/イベント・カウンタ00のPPG出力機能について説明します。

その他の初期設定については、[78K0S/Kx1+ 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、割り込みについては、[78K0S/Kx1+ 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、低電圧検出(LVI)については、[78K0S/Kx1+ サンプル・プログラム\(低電圧検出\) 2.7V未満検出時リセット発生編 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、各製品のユーザーズ・マニュアル([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#))を参照してください。

アセンブラ命令については、[78K/0Sシリーズ 命令編 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00のPPG出力機能の設定

16ビット・タイマ/イベント・カウンタ00のPPG出力機能を使用する際に使用するレジスタには、主に次の9種類があります。

- ・キャプチャ/コンペア・コントロール・レジスタ00 (CRC00)
- ・16ビット・タイマ・キャプチャ/コンペア・レジスタ000 (CR000)
- ・16ビット・タイマ・キャプチャ/コンペア・レジスタ010 (CR010)
- ・プリスケラ・モード・レジスタ00 (PRM00)
- ・16ビット・タイマ出力コントロール・レジスタ00 (TOC00)
- ・16ビット・タイマ・モード・コントロール・レジスタ00 (TMC00)
- ・ポート・レジスタx (Px) ^注
- ・ポート・モード・レジスタx (PMx) ^注
- ・ポート・モード・コントロール・レジスタx (PMCx) ^注

注 PPG出力機能では、TO00端子をタイマ出力として使用するので、Px, PMx, PMCxレジスタを次のように設定します。

	Pxレジスタ	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	P31 = 0	PM31 = 0	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	P21 = 0	PM21 = 0	PMC21 = 0

<PPG出力として使用する場合の基本的な動作設定手順例>

CRC00レジスタの設定

CR000, CR010レジスタに任意の値 (0000H < CR010 < CR000 FFFFH) を設定

PRM00レジスタでカウント・クロックを設定

TOC00レジスタの設定

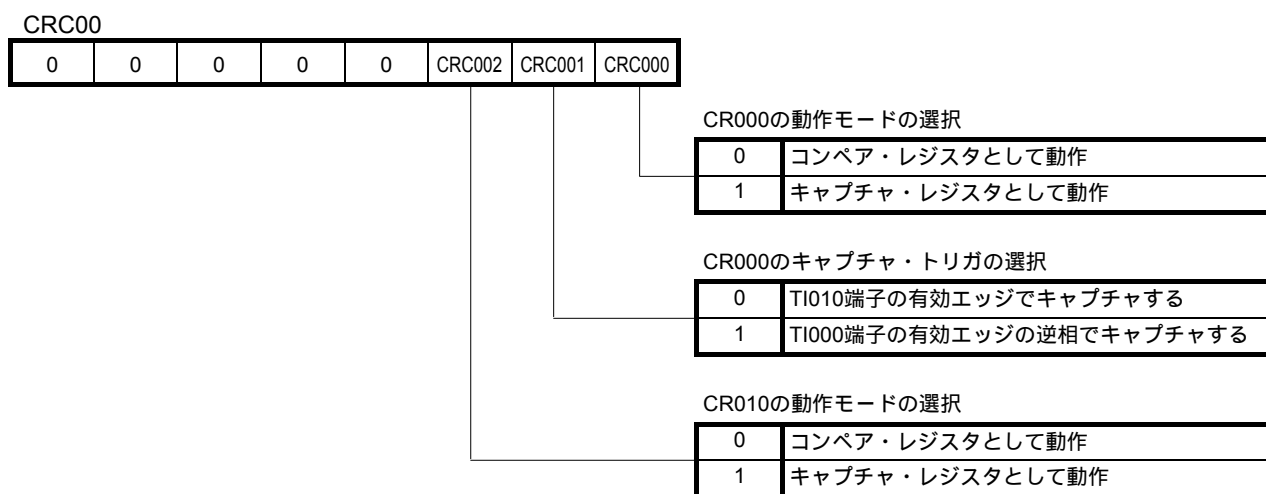
TMC00レジスタの設定：動作開始

注意 ~ は順不同です。

(1) CRC00レジスタの設定

CRC00レジスタは、CR000, CR010レジスタの動作を制御するレジスタです。

図4 - 1 キャプチャ/コンペア・コントロール・レジスタ00 (CRC00) のフォーマット



注意1. CRC00レジスタは、必ずタイマ動作を停止してから設定してください。

2. TMC00レジスタで、TM00とCR000の一致でクリア&スタート・モードを選択したとき、CR000レジスタをキャプチャ・レジスタに指定しないでください。

(2) CR000レジスタの設定

CR000レジスタは、キャプチャ・レジスタとコンペア・レジスタの機能をあわせ持ったレジスタです。

図4 - 2 16ビット・タイマ・キャプチャ/コンペア・レジスタ000 (CR000) のフォーマット



CR000をコンペア・レジスタとして使用するとき

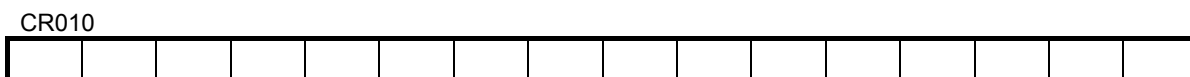
CR000に設定した値と16ビット・タイマ・カウンタ00 (TM00) のカウント値を常に比較し、一致したときに割り込み要求 (INTTM00) を発生します。

- 注意1. TM00とCR000の一致でクリア&スタート・モードの場合，CR000レジスタには0000H以外の値を設定してください。フリー・ランニング・モードおよびTI000端子の有効エッジのクリア&スタート・モードにおいて，CR000レジスタに0000Hを設定した場合は，オーバフロー（FFFFH）後，0000Hから0001Hになるときに割り込み要求（INTTM000）を発生します。
2. CR000レジスタの変更値が16ビット・タイマ・カウンタ00（TM00）の値より小さいとき，TM00レジスタはカウントを継続しオーバフローして0から再カウントします。したがって，CR000レジスタの変更後の値が変更前の値よりも小さいときは，CR000レジスタを変更後，タイマをリセットし，再スタートさせる必要があります。
 3. TM00カウンタ停止後のCR000レジスタの値は保証されません。
 4. コンペア・モードに設定したCR000レジスタは，キャプチャ・トリガが入力されてもキャプチャ動作を行いません。
 5. TM00カウンタ動作中にCR000レジスタを変更すると，誤動作する可能性があります。

(3) CR010レジスタの設定

CR010レジスタは，キャプチャ・レジスタとコンペア・レジスタの機能をあわせ持ったレジスタです。

図4 - 3 16ビット・タイマ・キャプチャ/コンペア・レジスタ010（CR010）のフォーマット



CR010をコンペア・レジスタとして使用するとき

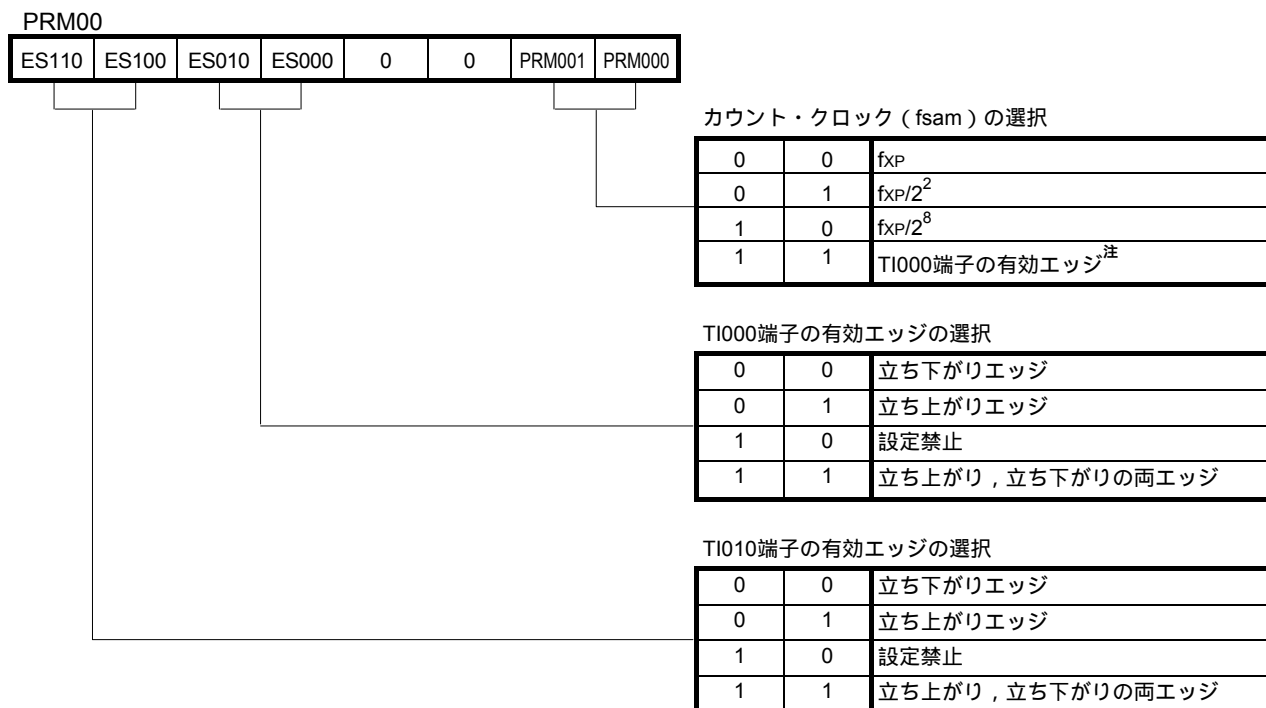
CR010に設定した値と16ビット・タイマ・カウンタ00（TM00）のカウント値を常に比較し，一致したときに割り込み要求（INTTM010）を発生します。

- 注意1. フリー・ランニング・モードおよびTI000端子の有効エッジのクリア&スタート・モードにおいて，CR010に0000Hを設定した場合は，オーバフロー（FFFFH）後，0000Hから0001Hになるときに割り込み要求（INTTM010）を発生します。
2. CR010レジスタの変更値がTM00カウンタの値より小さいとき，TM00カウンタはカウントを継続しオーバフローして0から再カウントします。したがって，CR010レジスタの変更後の値が変更前の値よりも小さいときは，CR010レジスタを変更後，タイマをリセットし，再スタートさせる必要があります。
 3. TM00カウンタ停止後のCR010レジスタの値は保証されません。
 4. コンペア・モードに設定したCR010レジスタは，キャプチャ・トリガが入力されてもキャプチャ動作を行いません。
 5. TM00カウンタ動作中にCR010レジスタを変更すると，誤動作する可能性があります。

(4) PRM00レジスタの設定

PRM00レジスタは、TM00カウンタのカウンタ・クロックおよびTI000、TI010端子入力の有効エッジを設定するレジスタです。

図4 - 4 プリスケーラ・モード・レジスタ00 (PRM00) のフォーマット



注 外部クロックは内部クロック (f_{XP}) の2周期分より長いパルスが必要とします。

備考 f_{XP} : 周辺ハードウェアへのクロックの発振周波数

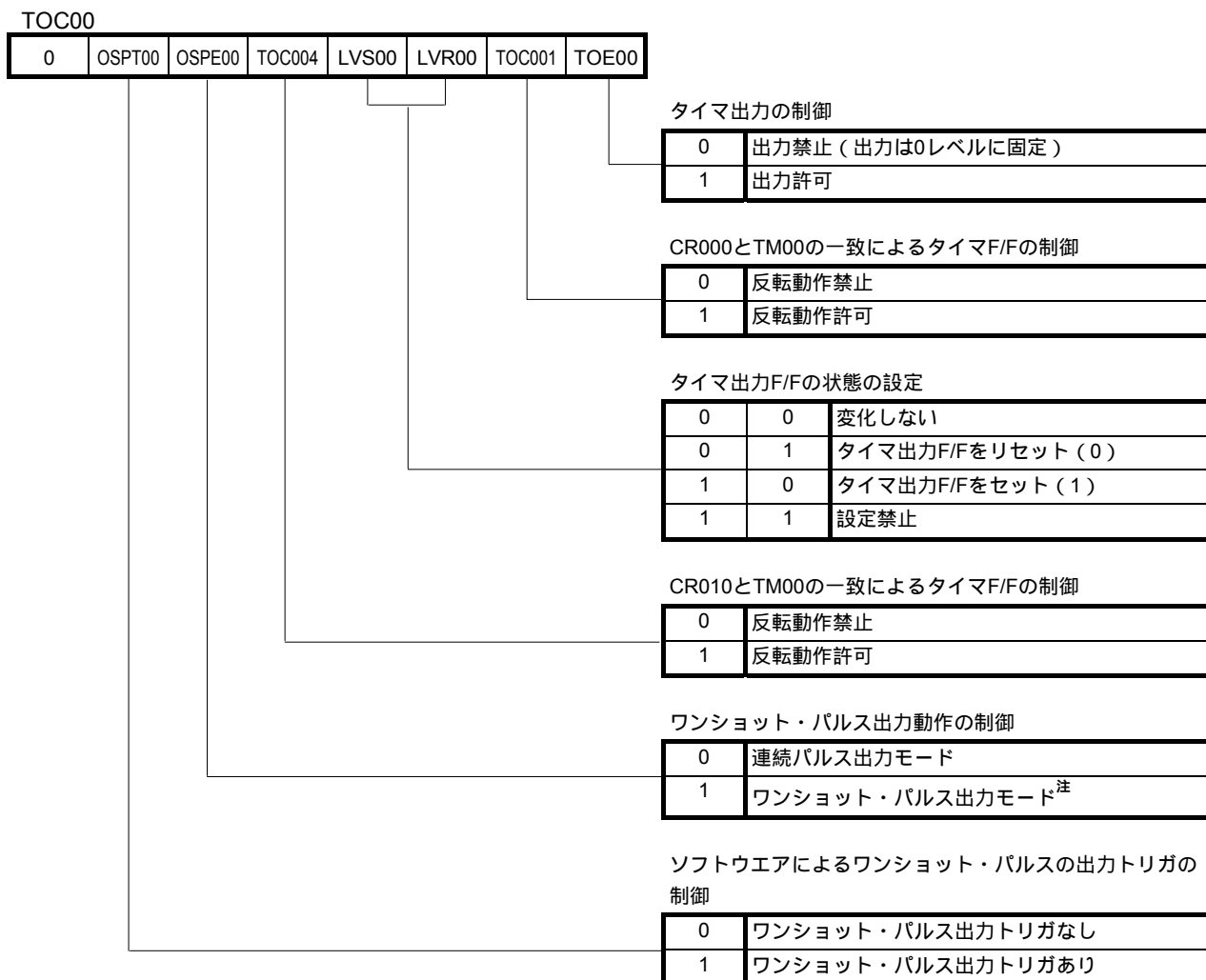
- 注意1. PRM00レジスタは、必ずタイマ動作を停止させてからデータを設定してください。
2. カウント・クロックにTI000端子の有効エッジを設定する場合、TI000端子の有効エッジでクリア & スタート・モードおよびTI000端子をキャプチャ・トリガに設定しないでください。
3. 次の場合、TI0n0端子 (n = 0, 1) の有効エッジは検出されますので、注意してください。
- システム・リセット直後、TI0n0端子にハイ・レベルを入力し、TM00動作を許可
 - TI0n0端子の有効エッジを立ち上がりまたは両エッジに指定した場合は、TM00動作の許可直後に、立ち上がりエッジを検出
 - TI0n0端子がハイ・レベルのときにTM00動作を停止し、TI0n0端子にロウ・レベルを入力したあとにTM00動作を許可
 - TI0n0端子の有効エッジを立ち下がりまたは両エッジに指定した場合は、TM00動作の許可直後に、立ち下がりエッジを検出
 - TI0n0端子がロウ・レベルのときにTM00動作を停止し、TI0n0端子にハイ・レベルを入力したあとにTM00動作を許可
 - TI0n0端子の有効エッジを立ち上がりまたは両エッジに指定した場合は、TM00動作の許可直後に、立ち上がりエッジを検出

- 注意4. TI000の有効エッジをカウント・クロックで使用する場合、ノイズ除去のために f_{XP} でサンプリングします。有効エッジをサンプリングして、有効レベルを2回検出することではじめてキャプチャ動作するため、短いパルス幅のノイズを除去できます。
5. TI010/TO00/Pxx端子を有効エッジの入力端子（TI010）として使用するときは、タイマ出力端子（TO00）として使用できません。また、タイマ出力端子（TO00）として使用するときは、有効エッジの入力端子（TI010）として使用できません。

(5) TOC00レジスタの設定

TOC00レジスタは、16ビット・タイマ/イベント・カウンタ00の出力制御回路の動作を制御するレジスタです。タイマ出力F/Fのセット/リセット、出力の反転許可/禁止、タイマ出力（TO00端子出力）の許可/禁止、ワンショット・パルス出力動作の許可/禁止およびソフトウェアによるワンショット・パルスの出力トリガを設定します。

図4 - 5 16ビット・タイマ出力コントロール・レジスタ00（TOC00）のフォーマット



注 ワンショット・パルス出力モードは、フリー・ランニング・モード、TI000端子の有効エッジでクリア&スタート・モードでのみ正常動作します。TM00とCR000の一致でクリア&スタート・モードでは、オーバフローしないためワンショット・パルス出力はできません。

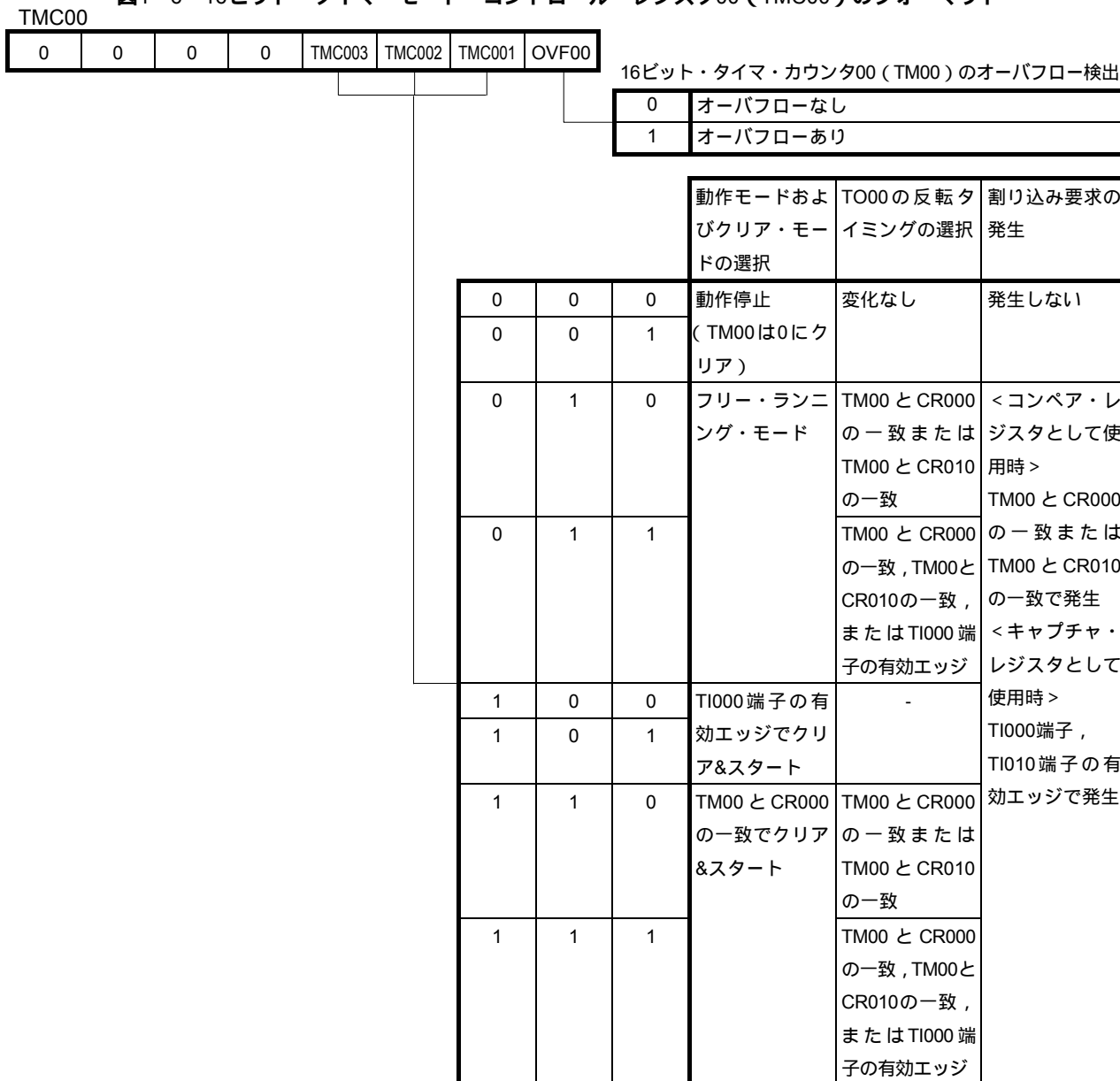
注意1. OSPT00以外は、必ずタイマ動作を停止してから設定してください。

- 注意2. LVS00, LVR00は読み出すと, 0になっています。
3. OSPT00は, データ設定後に自動的にクリアされますので, 読み出すと0になっています。
 4. OSPT00は, ワンショット・パルス出力モード以外でセット(1)しないでください。
 5. OSPT00に連続してセット(1)するとき, PRM00レジスタで選択したカウント・クロック2周期分以上のライト間隔が必要です。
 6. TOE00が0の場合, 8ビット・メモリ操作命令でTOE00, LVS00とLVR00を同時に設定してください。TOE00が1の場合, 1ビット・メモリ操作命令でLVS00とLVR00を設定することができます。
 7. TI010/TO00/Pxx端子を有効エッジの入力端子(TI010)として使用するときは, タイマ出力端子(TO00)として使用できません。また, タイマ出力端子(TO00)として使用するときは, 有効エッジの入力端子(TI010)として使用できません。

(6) TMC00レジスタの設定

TMC00レジスタは, 16ビット・タイマ/イベント・カウンタ00の動作モード, TM00カウンタのクリア・モード, 出力タイミングの設定およびオーバフローの検出するレジスタです。

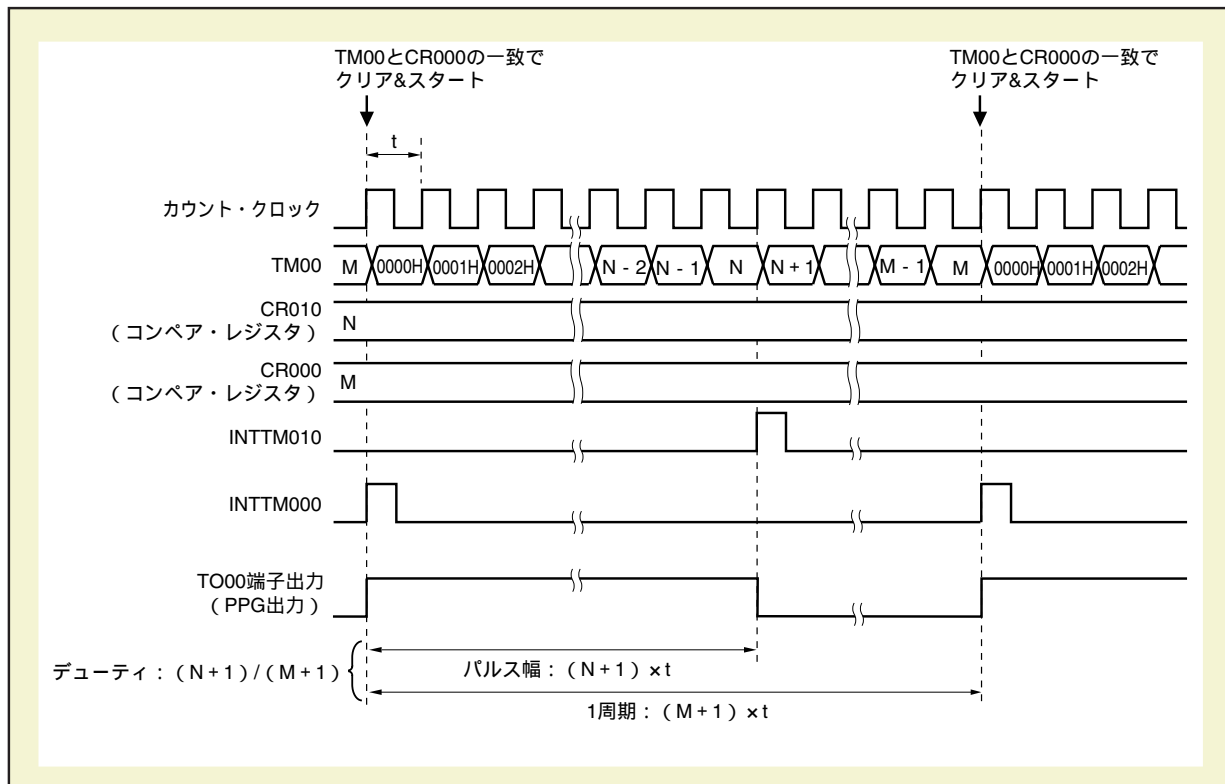
図4-6 16ビット・タイマ・モード・コントロール・レジスタ00 (TMC00) のフォーマット



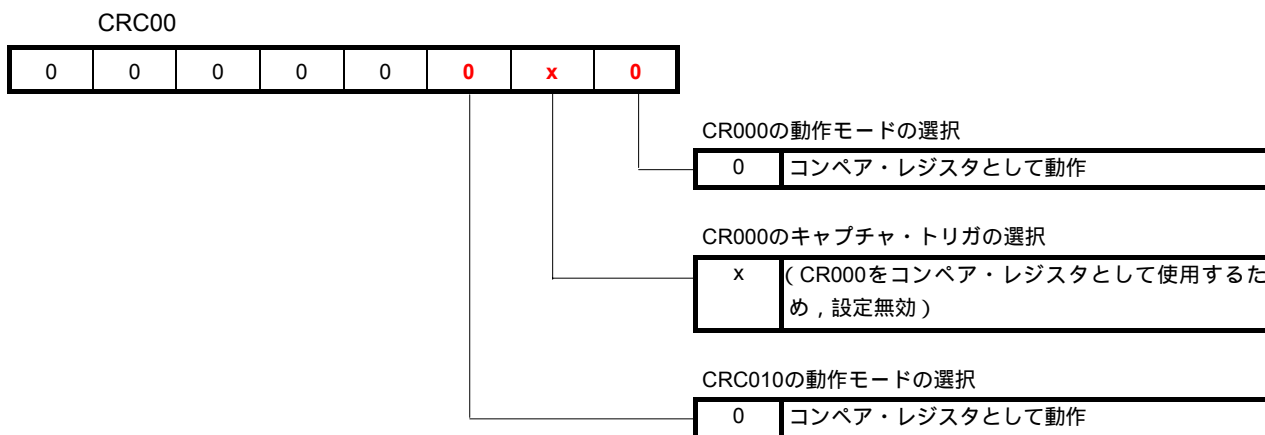
- 注意1. TM00カウンタは、TMC002, TMC003に0, 0（動作停止モード）以外の値を設定した時点で動作を開始します。動作を停止させるには、TMC002, TMC003に0, 0を設定してください。
2. OVF00フラグ以外のビットには、タイマ動作を停止してから書き込んでください。
3. タイマが停止している場合、TI000/TI010端子へ信号を入力しても、タイマ・カウントやタイマ割り込みは発生しません。
4. カウント・クロックにTI000端子の有効エッジを選択している場合を除き、STOPモードまたはシステム・クロック停止モードに設定する前に必ずタイマ動作を停止してください。システム・クロック開始時に、タイマが誤動作する可能性があります。
5. TI000端子の有効エッジは、タイマを停止してから、PRM00レジスタのビット4, 5で設定してください。
6. TM00とCR000の一致でクリア&スタート、TI000端子の有効エッジでクリア&スタート、フリー・ランニングのいずれかのモードを選択した場合、CR000レジスタの設定値がFFFFHで、TM00カウンタの値がFFFFHから0000Hに変化するとき、OVF00フラグが1に設定されます。
7. TM00カウンタがオーバフロー後、次のカウント・クロックがカウントされる（TM00カウンタが0001Hになる）前にOVF00フラグをクリアしても、再度セットされ、クリアは無効となります。
8. キャプチャ動作はカウント・クロックの立ち下がりで行われますが、割り込み要求（INTTM0n0: n = 0, 1）は次のカウント・クロックの立ち上がりで発生します。

【 例 】 PPG出力の1周期を $200 \mu\text{s}$, パルス幅を $20 \mu\text{s}$ に設定し , PPG出力を行う場合
 (カウント・クロック : $f_{XP}/2^2$ ($f_{XP} = 8 \text{ MHz}$)) (本サンプル・プログラム・ソースと同内容)

図4 - 7 PPG出力のタイミング例



(1) レジスタの設定内容



CR010

設定値 (N) : 39

・ カウント・クロック $f_{sam} = 8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$

・ パルス幅 $20 \text{ [\mu s]} = (N + 1) / 2 \text{ [MHz]}$

$$N = 20 \text{ [\mu s]} \times 2 \text{ [MHz]} - 1 = 39$$

CR000

設定値 (M) : 399

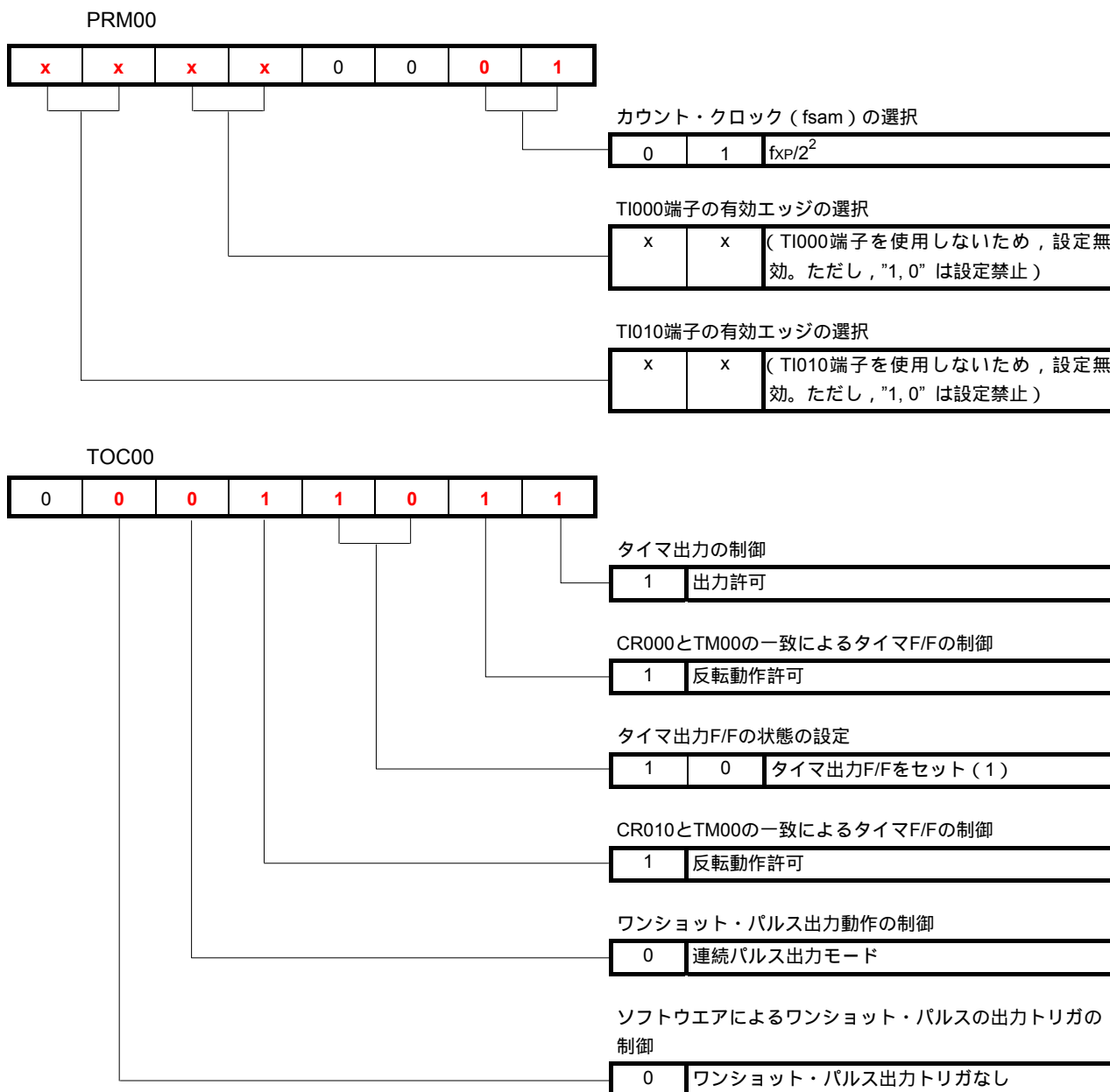
・ カウント・クロック $f_{sam} = 8 \text{ [MHz]} / 2^2 = 2 \text{ [MHz]}$

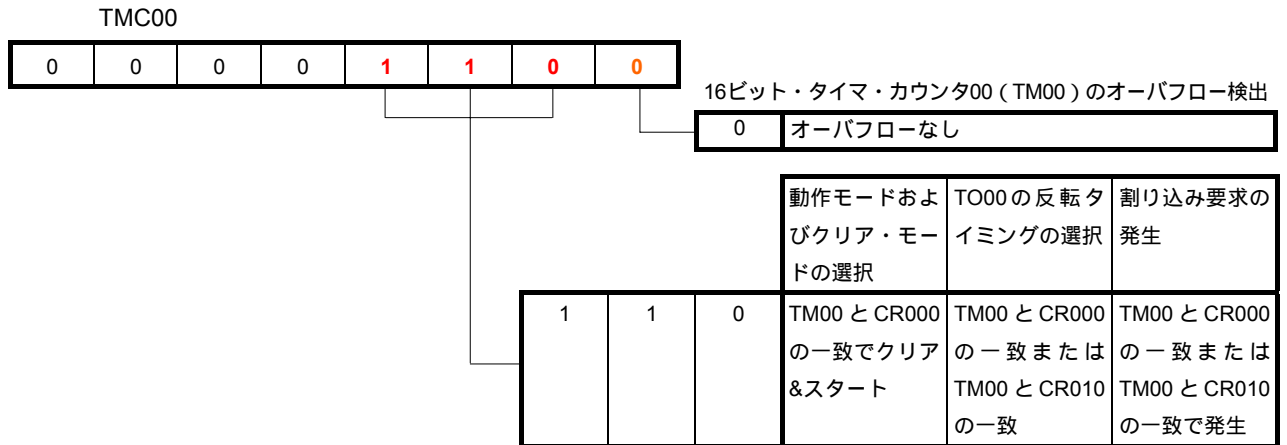
・ 1周期 $200 \text{ [\mu s]} = (M + 1) / 2 \text{ [MHz]}$

$$M = 200 \text{ [\mu s]} \times 2 \text{ [MHz]} - 1 = 399$$

注意 CR000とCR010には次の範囲の値を設定してください。

$$0000\text{H} < \text{CR010} < \text{CR000} \quad \text{FFFFH}$$





Px, PMx, PMCx

	Pxレジスタ	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	P31 = 0	PM31 = 0	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	P21 = 0	PM21 = 0	PMC21 = 0

(2) サンプル・プログラム

下記の例では、(1) レジスタの設定内容における「x」を「0」に設定しています。

アセンブリ言語の場合 (78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時)

```
CLR1 P3.1
CLR1 PM3.1
MOV CRC00, #00000000B
MOVW CR000, #399
MOVW CR010, #39
MOV PRM00, #00000001B
MOV TOC00, #00011011B
MOV TMC00, #00001100B
```

C言語の場合 (78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時)

```
P3.1 = 0;
PM3.1 = 0;
CRC00 = 0b00000000;
CR000 = 399;
CR010 = 39;
PRM00 = 0b00000001;
TOC00 = 0b00011011;
TMC00 = 0b00001100;
```

【本サンプル・プログラム・ソースからの抜粋】

付録A プログラム・リストから、16ビット・タイマ/イベント・カウンタ00の機能に関する抜粋部分を示します（前述の【例】と同内容）。

(1) アセンブリ言語

	XMAIN CSEG UNIT	
	RESET_START:	
	MOV CRC00, #00000000B	; CR000の動作モードをコンペア・レジスタに設定 ; CR000をコンペア・レジスタとして使用
PPG出力の1周期を設定	MOVW AX, #400-1	
PPG出力のパルス幅を設定	MOVW CR000, AX	; CR000は周期設定用
	MOVW AX, #40-1	
カウント・クロックの設定	MOVW CR010, AX	; CR010はパルス幅設定用
	MOV PRM00, #00000001B	; カウント・クロック = f _{xp} /4 (= 2MHz)
	MOV TOC00, #00011011B	; CR000, CR010との一致で出力反転、出力の初期値High、タイマ出力を許可
TO00端子出力の設定	MOV TMC00, #00001100B	; タイマ動作開始 (TMC00とCR000の一致でクリア&スタート)
	INTERRUPT_P1:	
	BNC \$DECDUTY	; 新しいパルス幅 < 現在のパルス幅なら分岐
INTTM000 割り込み処理を許可	CLR1 TMIF000	; 無効な割り込み要求をクリア
	CLR1 TMMK000	; INTTM000 割り込みマスクを解除
INTTM010 割り込み処理を禁止	SET1 TMMK010	; INTTM010 割り込みをマスク
	BR \$END_INTP1	; END_INTP1へ分岐
	DECDUTY:	
	CLR1 TMIF010	; 無効な割り込み要求をクリア
INTTM010 割り込み処理を許可	CLR1 TMMK010	; INTTM010 割り込みマスクを解除
	SET1 TMMK000	; INTTM000 割り込みをマスク
	END_INTP1:	
	POP AX	; AXレジスタのデータを復帰
	RETI	; 割り込み処理から復帰
	INTERRUPT TM000:	
	CLR1 TOC00.4	; 出力反転禁止
出力反転を禁止し、パルス幅変更	MOVW AX, NEXTD	
	MOVW CR010, AX	; パルス幅 (デューティ) 変更
	NOP	; タイマ00のカウント・クロック1周期分ウェイト
	NOP	
出力反転を許可	SET1 TOC00.4	; 出力反転許可
	MOVW CURRENTD, AX	; 新しいパルス幅を現在のパルス幅として保存
INTTM000 割り込み処理を禁止	SET1 TMMK000	; INTTM000 割り込みをマスク
	POP AX	; AXレジスタのデータを復帰
	RETI	; 割り込み処理から復帰
	INTERRUPT TM010:	
	PUSH AX	; AXレジスタのデータをスタックへ退避
	CLR1 TOC00.4	; 出力反転禁止
出力反転を禁止し、パルス幅変更	MOVW AX, NEXTD	
	MOVW CR010, AX	; パルス幅 (デューティ) 変更
	NOP	; タイマ00のカウント・クロック1周期分ウェイト
	NOP	
出力反転を許可	SET1 TOC00.4	; 出力反転許可
	MOVW CURRENTD, AX	; 新しいパルス幅を現在のパルス幅として保存
INTTM010 割り込み処理を禁止	SET1 TMMK010	; INTTM010 割り込みをマスク
	POP AX	; AXレジスタのデータを復帰
	RETI	; 割り込み処理から復帰

(2) C言語

```

void hdwinit(void) {
    unsigned char ucCnt200us; /* 200usウェイト用8ビット変数 */
    :
    :
    :
    CRC00 = 0b00000000; /* CR000,CR010をコンペア・レジスタとして使用 */
    CR000 = 400-1; /* CR000は周期設定用 */
    CR010 = 40-1; /* CR010はパルス幅設定用 */
    PRM00 = 0b00000001; /* カウント・クロック = fxp/4 (= 2MHz) */
    TOC00 = 0b00011011; /* CR000,CR010との一致で出力反転、出力の初期値High、タイマ出力を許可 */
    TMC00 = 0b00001100; /* タイマ動作開始 (TM00とCR000の一致でクリア&スタート) */
    :
    :
    :
    TO00端子
    出力の設定
    :
    :
    :
    __interrupt void fn_intp1(){
    :
    :
    :
    if (g_unNextD > g_unCurrentD + offsetT){ /* パルス幅を大きくするときはINTTM000割
    り込みを使用 */
        TMIF000 = 0; /* 無効な割り込み要求をクリア */
        TMMK000 = 0; /* INTTM000割り込みマスクを解除 */
        TMMK010 = 1; /* INTTM010割り込みをマスク */
    }
    else { /* パルス幅を小さくするときはINTTM010割り込みを使用 */
        TMIF010 = 0; /* 無効な割り込み要求をクリア */
        TMMK010 = 0; /* INTTM010割り込みマスクを解除 */
        TMMK000 = 1; /* INTTM000割り込みをマスク */
    }
    :
    :
    :
    interrupt void fn inttm000(){
    :
    :
    :
    TOC00.4 = 0; /* 出力反転禁止 */
    CR010 = g_unNextD; /* パルス幅(デューティ)変更 */
    NOP(); /* タイマ00のカウント・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1; /* 出力反転許可 */
    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK000 = 1; /* INTTM000割り込みをマスク */
    return;
    }
    interrupt void fn inttm010(){
    :
    :
    :
    TOC00.4 = 0; /* 出力反転禁止 */
    CR010 = g_unNextD; /* パルス幅(デューティ)変更 */
    NOP(); /* タイマ00のカウント・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1; /* 出力反転許可 */
    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK010 = 1; /* INTTM010割り込みをマスク */
    return;
    }
}
    
```

PPG出力の1周期を設定
 PPG出力のパルス幅を設定
 カウント・クロックの設定
 タイマ動作開始
 TO00端子出力の設定
 INTTM000割り込み要求フラグをクリア
 INTTM000割り込み処理を許可
 INTTM010割り込み処理を禁止
 INTTM010割り込み処理を許可
 INTTM000割り込み処理を禁止
 INTTM000割り込み発生により、割り込み処理開始
 出力反転を禁止し、パルス幅変更
 出力反転を許可
 INTTM000割り込み処理を禁止
 INTTM010割り込み発生により、割り込み処理開始
 出力反転を禁止し、パルス幅変更
 出力反転を許可
 INTTM010割り込み処理を禁止

4.2 PPG出力のデューティ設定と変更方法

4.2.1 PPG出力のデューティ設定

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00のPPG出力機能を使用し、TO00端子よりパルスを出力しています。

- ・パルス幅 = $(N + 1) / fsam$
- ・1周期 = $(M + 1) / fsam$
- ・PPG出力のデューティ = $(N + 1) / (M + 1) \times 100$

備考 N: CR010レジスタの設定値

M: CR000レジスタの設定値

fsam: 16ビット・タイマ/イベント・カウンタ00のカウント・クロック周波数

0000H < N < M FFFFH

計算例 CR010レジスタの設定値が39，CR000レジスタの設定値が399の場合（fsam = 2 MHz動作時）

- ・パルス幅 = $(N + 1) / fsam = (39 + 1) / 2 \text{ [MHz]} = 20 \text{ [}\mu\text{s]}$
- ・1周期 = $(M + 1) / fsam = (399 + 1) / 2 \text{ [MHz]} = 200 \text{ [}\mu\text{s]}$
- ・PPG出力のデューティ = $(N + 1) / (M + 1) \times 100 = (39 + 1) / (399 + 1) \times 100 = 10 \text{ [%]}$

また、スイッチ入力回数に応じて、CR010レジスタのみ設定値を変更し、PPG出力のデューティを変更しています。

スイッチ 入力回数 ^注	CR010レジスタの 設定値	CR000レジスタの 設定値	PPG出力のデューティ
0	39	399	10 % $((39 + 1) / (399 + 1)) \times 100$
1	119		30 % $((119 + 1) / (399 + 1)) \times 100$
2	199		50 % $((199 + 1) / (399 + 1)) \times 100$
3	279		70 % $((279 + 1) / (399 + 1)) \times 100$
4	359		90 % $((359 + 1) / (399 + 1)) \times 100$
5	279		70 % $((279 + 1) / (399 + 1)) \times 100$
6	199		50 % $((199 + 1) / (399 + 1)) \times 100$
7	119		30 % $((119 + 1) / (399 + 1)) \times 100$

注 スイッチ入力回数8回以降の場合、PPG出力のデューティは0回目からの繰り返しになります。

4.2.2 CR010の設定値 (PPG出力のデューティ) の変更方法

このサンプル・プログラムでは基本的に PPG出力のデューティを大きくする場合はINTTM000割り込み中に、小さくする場合はINTTM010割り込み中に、CR010レジスタ値を変更します。ただし、「INTTM010割り込み発生～CR010レジスタ値変更～タイマ出力反転許可」のオフセット時間を考慮して、実際には次のようにしています。

- ・「新しいパルス幅 > 現在のパルス幅 + オフセット時間」の場合：INTTM000割り込み処理中に変更
- ・「新しいパルス幅 < 現在のパルス幅 + オフセット時間」の場合：INTTM010割り込み処理中に変更

上記の方法でCR010レジスタ値を変更しない場合、PPG出力の1周期の期間でCR010とTM00との一致が発生しない可能性や、CR010とTM00の一致が複数回発生し、一致ごとにTO00端子の出力レベルが反転する可能性があります。

INTTM0x0 (x = 0, 1) 割り込み処理中にCR010レジスタ値を変更する場合は、次の手順で行います。

- TM00とCR010の一致によるタイマ出力反転動作を禁止する (TOC004 = 0)
- CR010を書き換える
- TM00のカウント・クロックの1周期分をウエイトする
- TM00とCR010の一致によるタイマ出力反転動作を許可する (TOC004 = 1)

次に、CR010レジスタ値を変更する際のタイミング・チャート例を示します。

図4-8 「新しいパルス幅 > 現在のパルス幅 + オフセット時間」の場合のタイミング・チャート例

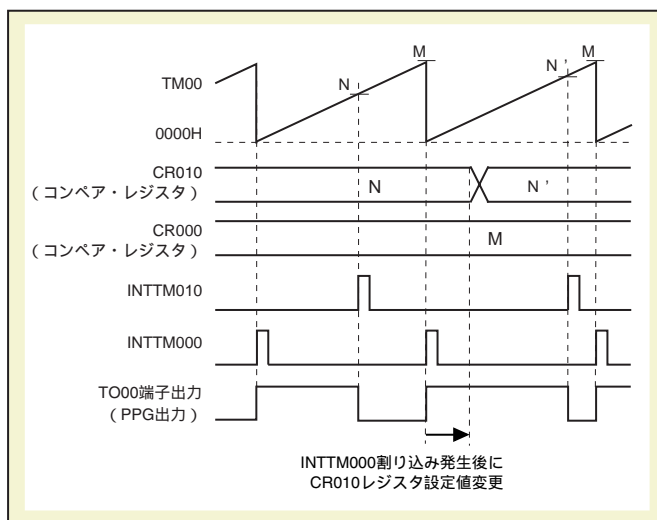
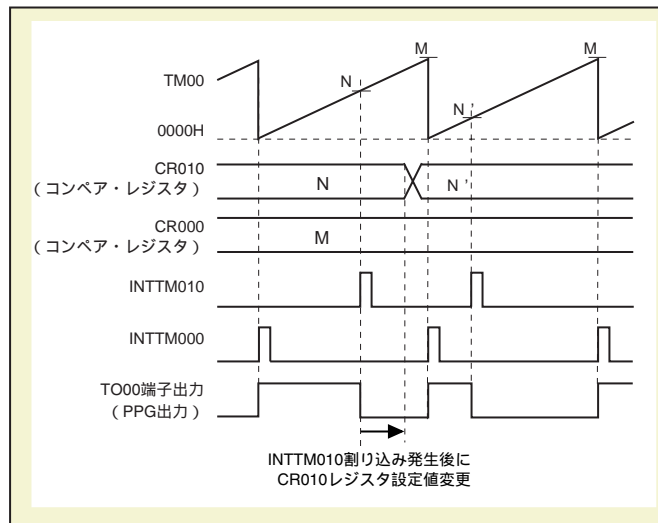


図4 - 9 「新しいパルス幅 < 現在のパルス幅 + オフセット時間」の場合のタイミング・チャート例



タイマ動作中のコンペア・レジスタの変更方法の詳細および注意事項については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) の「6.5 16ビット・タイマ/イベント・カウンタ00の注意事項」を参照してください。

【本サンプル・プログラム・ソースからの抜粋】

付録A プログラム・リストから，CR010レジスタの変更に関する抜粋部分を示します。

(1) アセンブリ言語

```

OFFSETT EQU 43/4 ; INTTM010の処理時間分の補正用・
.
.
.
INTERRUPT_P1:
.
.
.
MOVW AX, CURRENTD ; 現在のパルス幅を読み出し
ADDW AX, #OFFSETT ; オフセット時間を加算
XCH A, X
SUB A, [HL] ; 新しいパルス幅と比較
XCH A, X
SUBC A, [HL+1]

; 新しいパルス幅 < 現在のパルス幅なら分岐
BNC $DECDUTY ; 新しいパルス幅 < 現在のパルス幅なら分岐
; 無効な割り込み要求をクリア
CLR1 TMIF000 ; 無効な割り込み要求をクリア
; INTTM000割り込みマスクを解除
CLR1 TMMK000 ; INTTM000割り込みマスクを解除
; INTTM010割り込みをマスク
SET1 TMMK010 ; INTTM010割り込みをマスク
BR $END_INTP1 ; END_INTP1へ分岐

DECDUTY:
CLR1 TMIF010 ; 無効な割り込み要求をクリア
CLR1 TMMK010 ; INTTM010割り込みマスクを解除
SET1 TMMK000 ; INTTM000割り込みをマスク

.
.
.
INTERRUPT_TM000:
PUSH AX ; AXレジスタのデータをスタックへ退避
CLR1 TOC00.4 ; 出力反転禁止
MOVW AX, NEXTD
MOVW CR010, AX ; パルス幅(デューティ)変更
NOP ; タイマ00のカウント・クロック1周期分ウェイト
NOP
SET1 TOC00.4 ; 出力反転許可
MOVW CURRENTD, AX ; 新しいパルス幅を現在のパルス幅として保存
SET1 TMMK000 ; INTTM000割り込みをマスク
POP AX ; AXレジスタのデータを復帰
RETI ; 割り込み処理から復帰

INTERRUPT_TM010:
PUSH AX ; AXレジスタのデータをスタックへ退避
CLR1 TOC00.4 ; 出力反転禁止
MOVW AX, NEXTD
MOVW CR010, AX ; パルス幅(デューティ)変更
NOP ; タイマ00のカウント・クロック1周期分ウェイト
NOP
SET1 TOC00.4 ; 出力反転許可
MOVW CURRENTD, AX ; 新しいパルス幅を現在のパルス幅として保存
SET1 TMMK010 ; INTTM010割り込みをマスク
POP AX ; AXレジスタのデータを復帰
RETI ; 割り込み処理から復帰
    
```

現在のパルス
値とオフセッ
ト時間の合計
と新しいパル
ス幅を比較

「新しいパルス幅 >
現在のパルス値 +
オフセット時間」
の場合，INTTM000
割り込みを使用

「新しいパルス幅 < 現在のパルス値
+ オフセット時間」の場合，
INTTM010割り込みを使用

INTTM000
割り込み発生
タイミングで
のデューティ
(CR010)の
変更^注

INTTM010
割り込み発生
タイミングで
のデューティ
(CR010)の
変更^注

注 このサンプル・プログラムでは，割り込み処理中にCR010レジスタの設定値を変更しているのので，設定値を変更するときに，使用する割り込み処理を禁止，許可する必要はありません。

(2) C言語

```

#define offsetT 56/4 /* INTTM010の処理時間分の補正用 */
.
.
.
__interrupt void fn_intp1(){
.
.
.
    if (g_unNextD > g_unCurrentD + offsetT){ /* パルス幅を大きくするときはINTTM000割
り込みを使用 */
        TMIFF000 = 0; /* 無効な割り込み要求をクリア */
        TMMK000 = 0; /* INTTM000割り込みマスクを解除 */
        TMMK010 = 1; /* INTTM010割り込みをマスク */
    }
    else { /* パルス幅を小さくするときはINTTM010割り込みを使用 */
        TMIFF010 = 0; /* 無効な割り込み要求をクリア */
        TMMK010 = 0; /* INTTM010割り込みマスクを解除 */
        TMMK000 = 1; /* INTTM000割り込みをマスク */
    }
.
.
.
__interrupt void fn_inttm000(){
    TOC00.4 = 0; /* 出力反転禁止 */
    CR010 = g_unNextD; /* パルス幅(デューティ)変更 */
    NOP(); /* タイマ00のカウンタ・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1; /* 出力反転許可 */
    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK000 = 1; /* INTTM000割り込みをマスク */

    return;
}

__interrupt void fn_inttm010(){
    TOC00.4 = 0; /* 出力反転禁止 */
    CR010 = g_unNextD; /* パルス幅(デューティ)変更 */
    NOP(); /* タイマ00のカウンタ・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1; /* 出力反転許可 */
    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK010 = 1; /* INTTM010割り込みをマスク */

    return;
}
    
```

現在のパルス値とオフセット時間の合計と新しいパルス幅を比較

「新しいパルス幅 > 現在のパルス値 + オフセット時間」の場合、INTTM000 割り込みを使用

「新しいパルス幅 < 現在のパルス値 + オフセット時間」の場合、INTTM010 割り込みを使用

INTTM000 割り込み発生タイミングでのデューティ (CR010) の変更^注

INTTM010 割り込み発生タイミングでのデューティ (CR010) の変更^注

注 このサンプル・プログラムでは、割り込み処理中にCR010レジスタの設定値を変更しているため、設定値を変更するときに、使用する割り込み処理を禁止、許可する必要はありません。

4.3 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（INTP1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00の割り込み（INTTM000）の発生をカウントしています。

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

備考 T： INTTM000割り込み周期

T'： INTP1エッジ検出～INTP1エッジ検出後の最初のINTTM000発生までの時間（ $0 < T' < T$ ）

M： INTP1エッジ検出後のINTTM000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 ） $> 10 \text{ ms}$

計算例 INTTM000割り込み周期(T)が $200 \mu\text{s}$ ([4.2 PPG出力のデューティ設定と変更方法を参照](#))、INTP1エッジ検出後のINTTM000割り込み回数(M)が51の場合

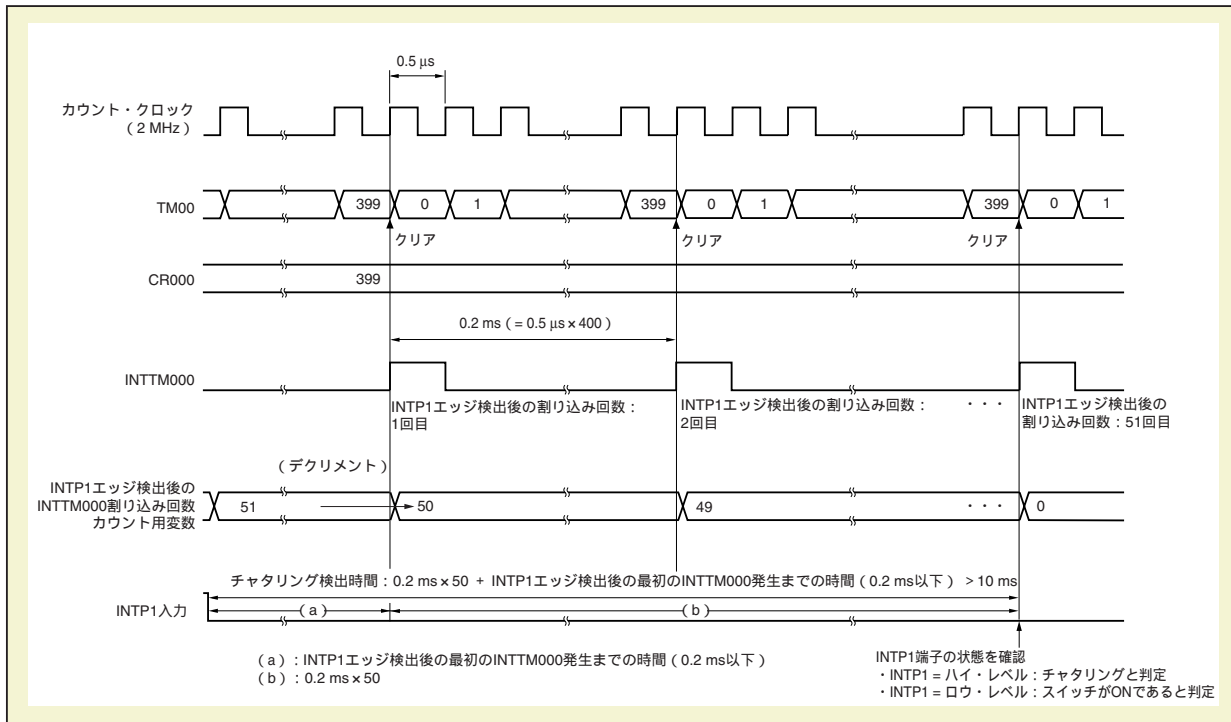
$$\begin{aligned} Tc &= T' + T \times (M - 1) \\ &= T' + 200 [\mu\text{s}] \times (51 - 1) \\ &= T' + 10000 [\mu\text{s}] \\ &= T' + 10 [\text{ms}] \end{aligned}$$

$0 < T' < 200 \mu\text{s}$ なので、


$$10 \text{ ms} < Tc < 10.2 \text{ ms}$$

チャタリング検出時間（ Tc ） $> 10 \text{ ms}$

図4 - 10 チャタリング検出のタイミング・チャート例




第5章 システム・シミュレータ SM+での動作確認

この章では、のアイコンを選択してダウンロードしたアセンブリ言語用のファイル（ソース・ファイル+プロジェクト・ファイル）を用い、サンプル・プログラムが、システム・シミュレータ SM+ for 78K0S/Kx1+でどのように動作するかを説明します。

注意 SM+ for 78K0S/Kx1+は、78K0S/KU1+マイクロコントローラには対応していません（2008年7月現在）。
したがって、78K0S/KU1+マイクロコントローラはSM+ for 78K0S/Kx1+で動作確認することはできません。

5.1 サンプル・プログラムのビルド

サンプル・プログラムをSM+ for 78K0S/Kx1+（以降、「SM+」と表記します）で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、サンプル・プログラムのビルド方法について、のアイコンからダウンロードしたアセンブリ言語用サンプル・プログラム（ソース・プログラム+プロジェクト・ファイル）を使用し、説明します。その他のダウンロードしたプログラムのビルド方法については、[78K0S/Kx1+ サンプル・プログラム スタートアップ・ガイド アプリケーション・ノート](#)を参照してください。

PM+操作方法の詳細については、[PM+ プロジェクト・マネージャ ユーザーズ・マニュアル](#)を参照してください。

【コラム】ビルドのエラー


PM+でビルドしているときに「A006 File not found 'C:\%NECTOOLS32\LIB78K0S\%s0sl.rel'」または、「*** ERROR F206 Segment '@@DATA' can't allocate to memory - ignored.」というエラー・メッセージが出た場合、次の手順にてコンパイラオプションの設定を変更してください。

[ツール] [コンパイラオプションの設定]を選択してください。

[コンパイラオプションの設定]ダイアログが開いたら、「スタートアップ・ルーチン」タグを選択してください。

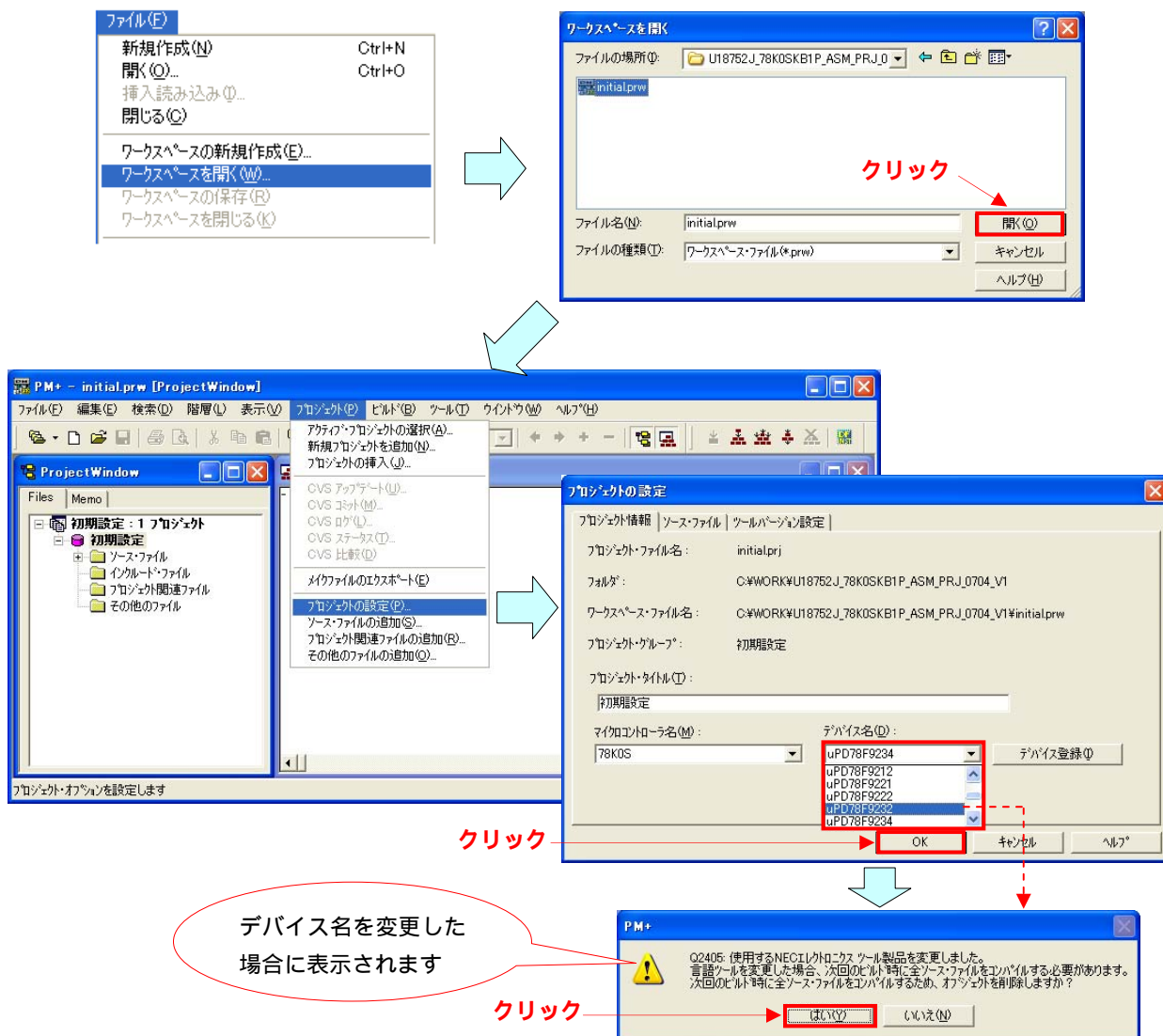
「標準ライブラリ固定領域を使用する」のチェックを外してください（それ以外のチェックは、そのまま）。


「標準ライブラリ固定領域を使用する」のチェックを外すと、標準ライブラリ固定領域として確保されていた118バイトのRAM領域が使用可能になりますが、標準ライブラリ（getchar関数やmalloc関数など）を使用できなくなります。

このサンプル・プログラムでは、のアイコンを選択してダウンロードしたファイルを使用する場合、デフォルトで「標準ライブラリ固定領域を使用する」のチェックが外されています。

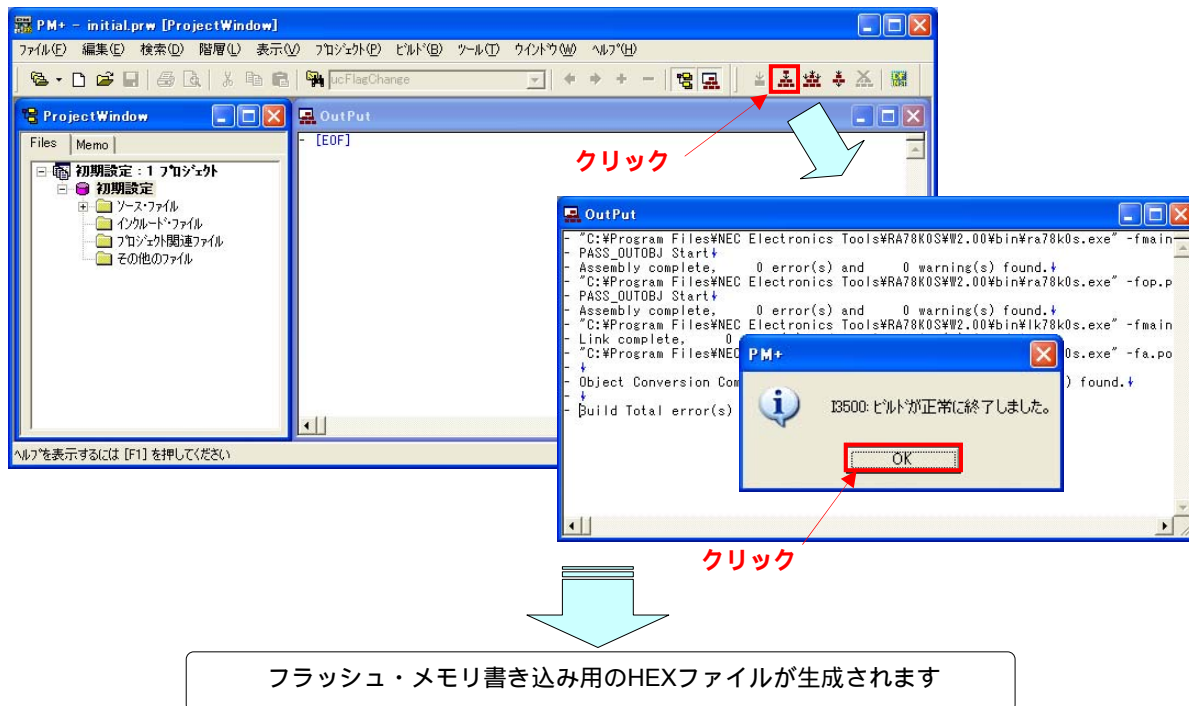
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「tm00ppg.prw」を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が立ち上がったら、使用するデバイス名を選択(デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択)し、[OK] ボタンをクリックしてください。

備考 下の図は、「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。



- (4)  (「ビルド」ボタン)をクリックしてください。ソース・ファイルが正常にビルドされると、「I3500: ビルドが正常に終了しました」というメッセージ画面が立ち上がります。
- (5) メッセージ画面にある [OK] ボタンをクリックしてください。フラッシュ・メモリ書き込み用のHEXファイルが作成されます。

備考 下の図は、「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。




5.2 SM+での動作

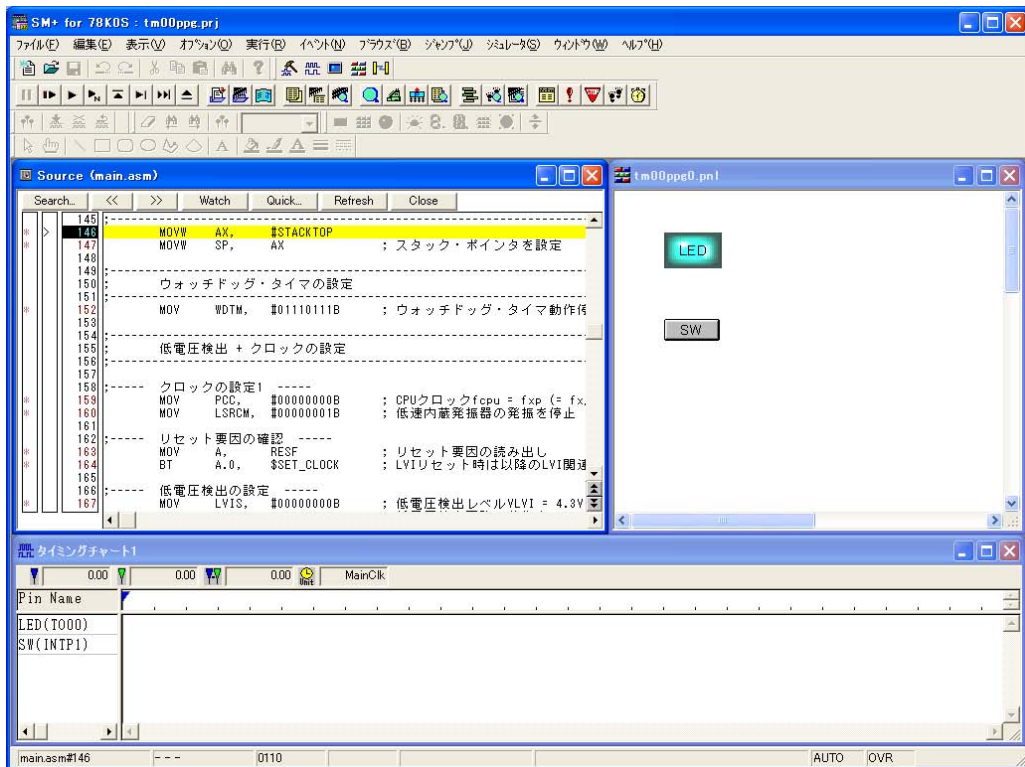
ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。


SM+操作方法の詳細については、[SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル](#)を参照してください。

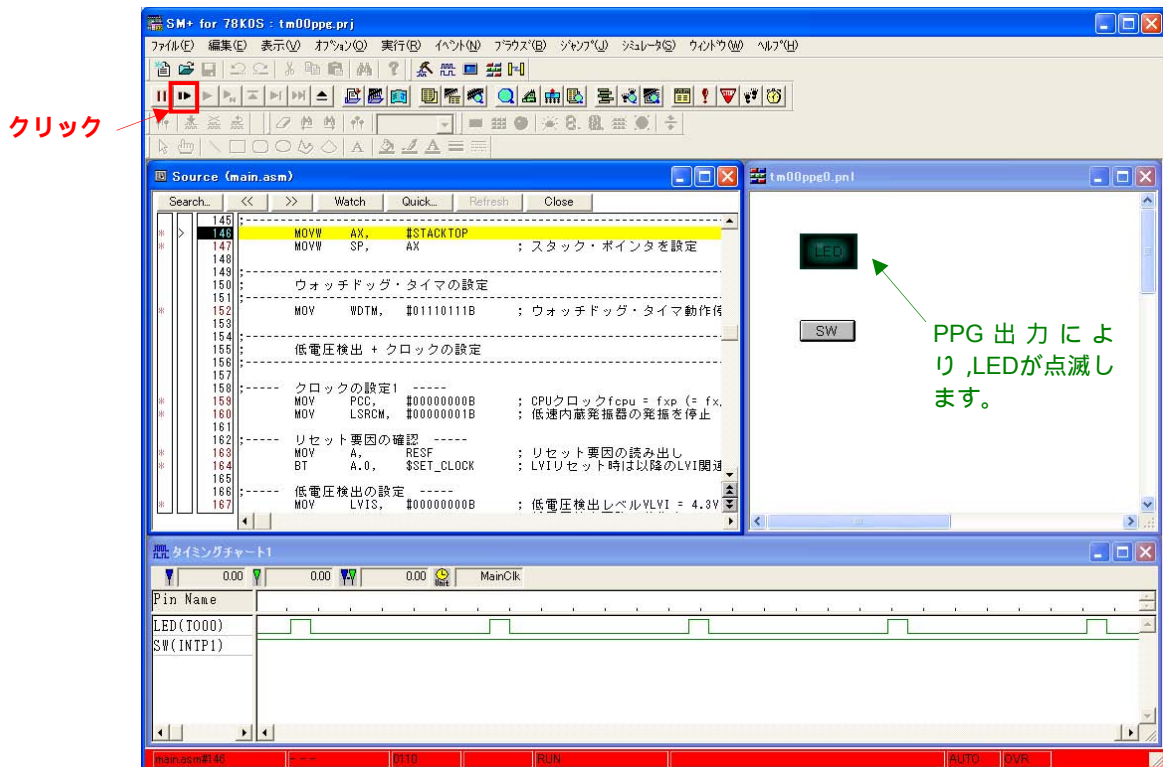
- (1) SM+ for 78K0S/Kx1+ W1.02をPM+ Ver6.30の環境で使用するために、次のサイトにあるPDFファイルを参照して、「外部ツールの登録」を行い、SM+を起動してください。

・ <http://www.necel.com/micro/ja/freesoft/pdf/ZUD-CD-07-0189.pdf>

- (2) SM+を起動すると、次のような画面になります（これは、のアイコンを選択してダウンロードしたアセンブリ言語のソース・ファイルを使用した場合の画面例です）。



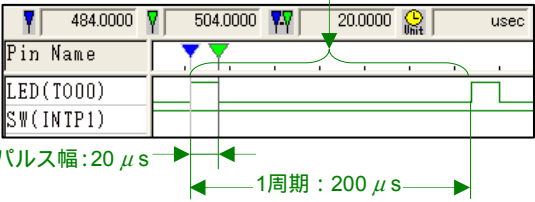


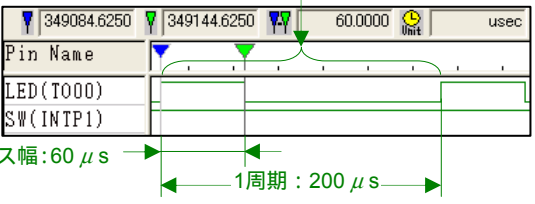


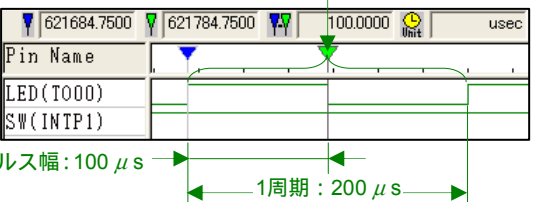


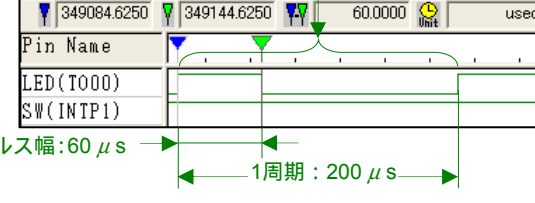


- (3) （「リスタート」ボタン）をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。



プログラム
実行中は、
赤になります。

- (4) プログラム実行中に、入出力パネル・ウィンドウ上の [SW] ボタンをクリックしてください。[SW] ボタンの入力回数により、PPG出力のデューティを変更します。
 入出力パネル・ウィンドウ上の [LED] の輝度およびタイミング・チャート・ウィンドウ上の波形が変化することを確認してください。

入出力パネル・ウィンドウ	タイミング・チャート・ウィンドウ
<p>輝度：90% </p> <p>クリックしない </p>	<p>デューティ10%のPPG出力</p>  <p>Pin Name LED (T000) SW (INTP1)</p> <p>パルス幅：20 μs 1周期：200 μs</p>
<p>輝度：70% </p> <p>1回クリック → </p>	<p>デューティ30%のPPG出力</p>  <p>Pin Name LED (T000) SW (INTP1)</p> <p>パルス幅：60 μs 1周期：200 μs</p>
<p>輝度：50% </p> <p>2回クリック → </p>	<p>デューティ50%のPPG出力</p>  <p>Pin Name LED (T000) SW (INTP1)</p> <p>パルス幅：100 μs 1周期：200 μs</p>
<p>⋮</p>	<p>⋮</p>
<p>輝度：70% </p> <p>7回クリック → </p>	<p>デューティ30%のPPG出力</p>  <p>Pin Name LED (T000) SW (INTP1)</p> <p>パルス幅：60 μs 1周期：200 μs</p>

注

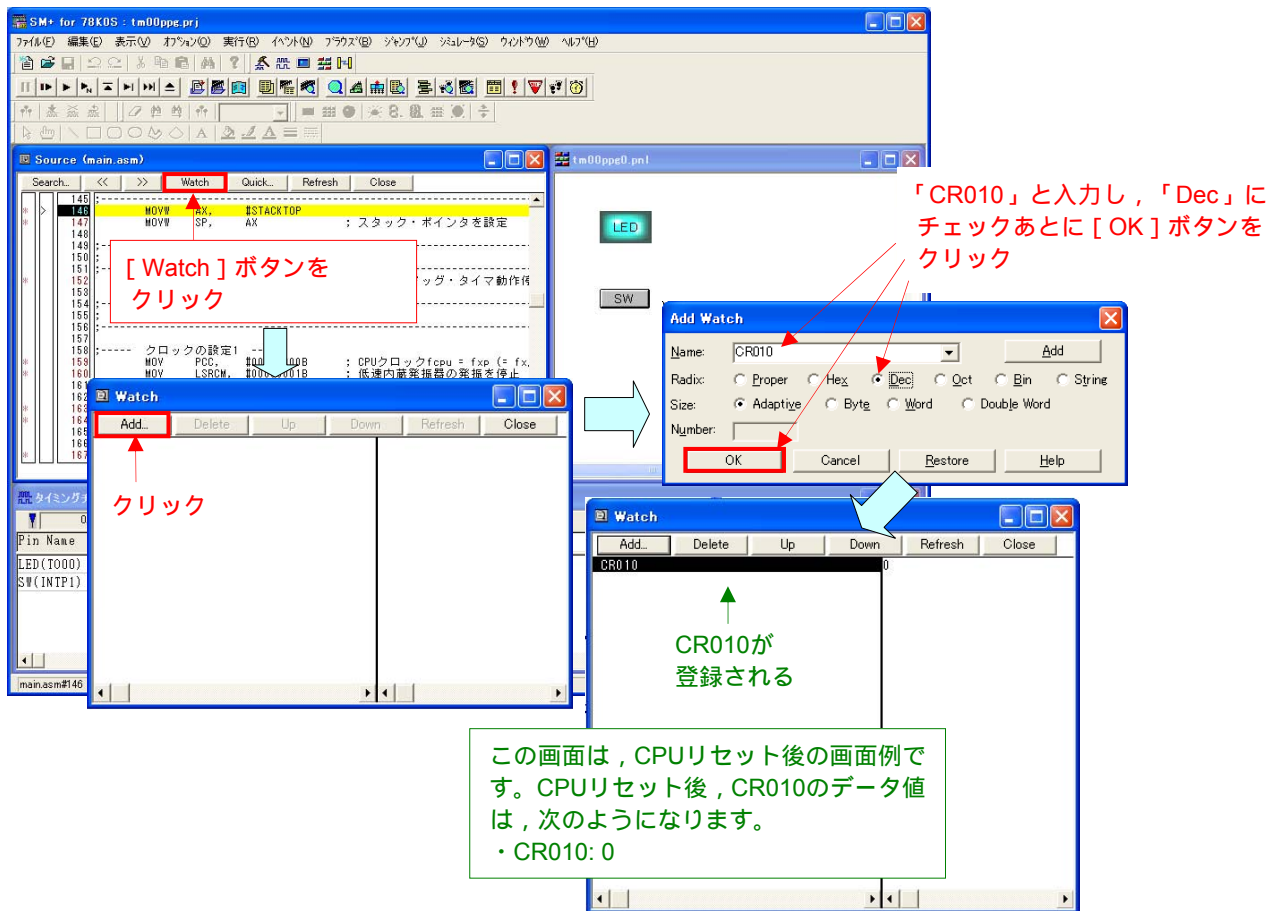
注 スイッチ入力が8回目以降の場合、PPG出力のデューティは0回目からの繰り返しになります。

【補足1】 SM+のウォッチ機能を使用することにより，CR010レジスタのデータ値の変化を確認することができます。

ソース・ウインドウの [Watch] ボタンをクリックしてください。 [Watch] ウィンドウが立ち上がります。

[Add] ボタンをクリックすると， [Add Watch] ウィンドウが立ち上がります（このとき， [Watch] ウィンドウは開いたままです）。

Nameに「CR010」と入力し，Radixの「Dec」にチェックしたあと， [OK] ボタンをクリックすると， [Watch] ウィンドウに，「CR010」が登録され， [Add Watch] ウィンドウが閉じられます。




プログラムを実行し，入出力パネル・ウインドウ上の [SW] ボタンをクリックしてください。 [SW] ボタンの入力回数により， [Watch] ウィンドウ上のCR010のデータ値が変化することを確認してください。

[SW] ボタンの入力回数 ^注	[Watch] ウィンドウのデータ値
0	CR010: 39
1	CR010: 119
2	CR010: 199
3	CR010: 279
4	CR010: 359
5	CR010: 279
6	CR010: 199
7	CR010: 119

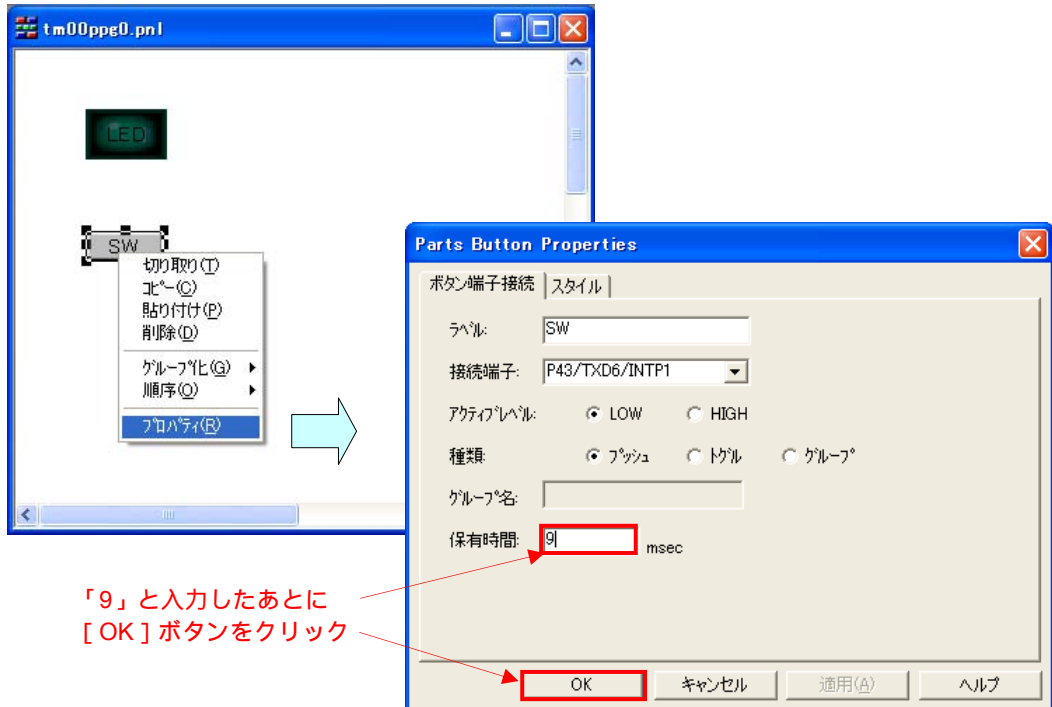
注 8回目以降は，0回目からの繰り返しになります。


【補足2】 [SW] ボタンの保有時間を10 ms以下に設定することにより，チャタリング検出が行われているかを確認することができます。

ツール・バーの  を選択してください。

入出力パネル・ウィンドウ上の [SW] ボタンを右クリックし， [プロパティ] を選択してください。

保有時間に「9」を入力し， [OK] ボタンをクリックしてください。



ツール・バーの  を選択してください。

プログラムを実行し， [SW] ボタンをクリックしてください。ボタンの保有時間が9 msのため， [SW] ボタンをクリックしても，チャタリングであると判定され，PPG出力のデューティは変化しません。

第6章 関連資料

資料名		和文 / 英文
78K0S/KU1+ ユーザーズ・マニュアル		PDF
78K0S/KY1+ ユーザーズ・マニュアル		PDF
78K0S/KA1+ ユーザーズ・マニュアル		PDF
78K0S/KB1+ ユーザーズ・マニュアル		PDF
78K0Sシリーズ 命令編 ユーザーズ・マニュアル		PDF
RA78K0S 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
CC78K0S 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
SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル		PDF
フラッシュ書き込み簡単マニュアル (MINICUBE2編) インフォメーション	78K0S/KU1+	PDF
	78K0S/KY1+	PDF
	78K0S/KA1+	PDF
	78K0S/KB1+	PDF
78K0S/Kx1+ アプリケーション・ ノート	サンプル・プログラム スタートアップ・ガイド	PDF
	サンプル・プログラム (初期設定) LED点灯のスイッチ制御編	PDF
	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	PDF
	サンプル・プログラム (低電圧検出) 2.7V未満検出時リセット発生編	PDF
	サンプル・プログラム (16ビット・タイマ/イベント・カウンタ00) インターバル・タイマ編	PDF
	サンプル・プログラム (16ビット・タイマ/イベント・カウンタ00) 外部イベント・カウンタ編	PDF
	サンプル・プログラム (16ビット・タイマ/イベント・カウンタ00) パルス幅測定編	PDF
	サンプル・プログラム (16ビット・タイマ/イベント・カウンタ00) ワンショット・パルス出力編	PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0S/KB1+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      78K0S/KB1+シリーズ
;
;*****
; 78K0S/KB1+シリーズ      サンプル・プログラム
;*****
; 16ビット・タイマ00 (PPG出力)
;*****
; 【履歴】
; 2007.7.--      新規作成
;*****
;
; 【概要】
;
; 本サンプルプログラムは、16ビット・タイマ00のPPG出力機能の使用例を示すものである。
; 周期200usのPPG出力を行い、スイッチ入力の度にデューティ幅を20%ずつ変更する。
;
;
; <主な設定内容>
;
; ・ウォッチドッグ・タイマの動作停止
; ・低電圧検出電圧VLVIを4.3V ± 0.2Vに設定
; ・VDD VLVIとなった後にVDD < VLVIとなった場合、内部リセット信号発生(低電圧検出回路)
; ・CPUクロックを8MHzに設定
; ・周辺ハードウェアへの供給クロックを8MHzに設定
; ・外部割り込みINTP1の有効エッジ：立ち下りエッジに設定
; ・スイッチ入力時のチャタリング検出時間 = 10ms
; ・HLレジスタを割り込み処理で使用する(=グローバル変数のように使用する)
;
;
; <16ビット・タイマ00の設定>
; ・動作モード：TM00とCR000の一致でタイマ・カウントのクリア&スタート
; ・カウント・クロック = fxp/4 (2MHz)
; ・CR000,CR010をコンペア・レジスタとして使用
```

```

;   ・ CR000は周期200usに初期化
;   ・ CR010はパルス幅20usに初期化
;   ・ CR000,CR010との一致で出力反転(PPG出力)
;   ・ 出力の初期値(=アクティブレベル)をHighに設定
;   ・ P31の出力ラッチをLowに設定(T000出力を使用するため)
;   ・ P31を出力モードに設定(T000出力を使用するため)
;
;
;   <スイッチ入力回数とPPG出力のデューティ>
;
;   +-----+
;   | SW入力回数 | PPG出力の | LEDの輝度 |
;   |             | デューティ |           |
;   |-----|
;   |   0回   |   10 %   |   90 %   |
;   |   1回   |   30 %   |   70 %   |
;   |   2回   |   50 %   |   50 %   |
;   |   3回   |   70 %   |   30 %   |
;   |   4回   |   90 %   |   10 %   |
;   |   5回   |   70 %   |   30 %   |
;   |   6回   |   50 %   |   50 %   |
;   |   7回   |   30 %   |   70 %   |
;   +-----+
;   # 8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 : P43
;   出力ポート : P00-P03, P20-P23, P30-P33, P40-P42, P44-P47, P120-P123, P130
;   未使用のポートは全て出力ポートに設定しておく
;
; *****
;
; =====
;
;   シンボル定義
;
; =====
OFFSETT      EQU      43/4                ; INTTM010の処理時間分の補正用

```



```

;=====
;
;   ベクタ・テーブルの設定
;
;=====

```

```

XVCT  CSEG  AT      0000H
      DW   RESET_START      ;(00)  RESET
      DW   RESET_START      ;(02)  --
      DW   RESET_START      ;(04)  --
      DW   RESET_START      ;(06)  INTLV1
      DW   RESET_START      ;(08)  INTP0
      DW   INTERRUPT_P1     ;(0A)  INTP1
      DW   RESET_START      ;(0C)  INTTMH1
      DW   INTERRUPT_TM000   ;(0E)  INTTM000
      DW   INTERRUPT_TM010   ;(10)  INTTM010
      DW   RESET_START      ;(12)  INTAD
      DW   RESET_START      ;(14)  --
      DW   RESET_START      ;(16)  INTP2
      DW   RESET_START      ;(18)  INTP3
      DW   RESET_START      ;(1A)  INTTM80
      DW   RESET_START      ;(1C)  INTSRE6
      DW   RESET_START      ;(1E)  INTSR6
      DW   RESET_START      ;(20)  INTST6

```

```

;=====
;
;   ROMの定義
;
;=====

```

```

XROM  CSEG  AT      0100H
PWIDTH:  DW   40-1      ; パルス幅 20us(デューティ10%)
      DW   40*3-1      ; パルス幅 60us(デューティ30%)
      DW   40*5-1      ; パルス幅100us(デューティ50%)
      DW   40*7-1      ; パルス幅140us(デューティ70%)
      DW   40*9-1      ; パルス幅180us(デューティ90%)
      DW   40*7-1      ; パルス幅140us(デューティ70%)
      DW   40*5-1      ; パルス幅100us(デューティ50%)
      DW   40*3-1      ; パルス幅 60us(デューティ30%)

```

```

;=====
;
;   RAMの定義
;
;=====
XRAM DSEG  SADDRP
NEXTD:          DS    2          ; 新しいパルス幅(デューティ)保存用
CURRENTD:      DS    2          ; 現在のパルス幅(デューティ)保存用

;=====
;
;   スタック領域の確保
;
;=====
XSTK DSEG  AT    OFEE0H
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 = FF00H

;*****
;
;   リセット解除後の初期化処理
;
;*****
XMAIN CSEG  UNIT
RESET_START:
;-----
;   スタック・ポインタの設定
;-----
        MOVW  AX,    #STACKTOP
        MOVW  SP,    AX          ; スタック・ポインタを設定

;-----
;   ウォッチドッグ・タイマの設定
;-----
        MOV   WDTM,  #01110111B ; ウォッチドッグ・タイマ動作停止

;-----
;   低電圧検出 + クロックの設定
;-----

;----- クロックの設定1 -----
        MOV   PCC,   #00000000B ; CPUクロックfcpu = fxp (= fx/4 = 2MHz)

```

```

MOV    LSRCM, #00000001B    ; 低速内蔵発振器の発振を停止

;----- リセット要因の確認 -----
MOV    A,      RESF          ; リセット要因の読み出し
BT     A.0,    $SET_CLOCK    ; LVIリセット時は以降のLVI関連処理を省略し、SET_CLOCKへ

;----- 低電圧検出の設定 -----
MOV    LVIS,   #00000000B    ; 低電圧検出レベルVLVI = 4.3V ± 0.2Vに設定
SET1   LVION   ; 低電圧検出回路の動作許可

MOV    A,      #40           ; 200usウェイト用のカウント値を代入
;----- 200usウェイト -----
WAIT_200US:
DEC    A
BNZ    $WAIT_200US          ; 0.5[us/cIk]×10[cIk]×40[count] = 200[us]

;----- VDD VLVI待ち処理 -----
WAIT_LVI:
NOP
BT     LVIF,   $WAIT_LVI    ; VDD < VLVIなら分岐

SET1   LVIMD   ; VDD < VLVI時に内部リセット信号が発生するように設定

;----- クロックの設定2 -----
SET_CLOCK:
MOV    PPCC,   #00000000B    ; 周辺ハードウェアへの供給クロックfxp = fx (= 8MHz)
; -> CPUクロックfcpu = fxp = 8MHz

;-----
;   ポート0の設定
;-----
MOV    P0,     #00000000B    ; P00-P03の出力ラッチLow
MOV    PM0,    #11110000B    ; P00-P03を出力ポートに設定

;-----
;   ポート2の設定
;-----
MOV    P2,     #00000000B    ; P20-P23の出力ラッチLow
MOV    PM2,    #11110000B    ; P20-P23を出力ポートに設定

;-----
;   ポート3の設定
;-----

```

```
MOV P3, #0000000B ; P30-P33の出力ラッチLow
MOV PM3, #1111000B ; P30-P33を出力ポートに設定
```

; ポート4の設定

```
MOV P4, #0000000B ; P40-P47の出力ラッチLow
MOV PU4, #00001000B ; P43に内蔵プルアップ抵抗を使用
MOV PM4, #00001000B ; P40-P42,P44-P47を出力ポートに、P43を入力ポートに設定
```

; ポート12の設定

```
MOV P12, #0000000B ; P120-P123の出力ラッチLow
MOV PM12, #1111000B ; P120-P123を出力ポートに設定
```

; ポート13の設定

```
MOV P13, #00000001B ; P130の出力High
```

; 16ビット・タイマ00の設定

```
MOV CRC00, #00000000B ; CR000,CR010をコンペア・レジスタとして使用
MOVW AX, #400-1
MOVW CR000, AX ; CR000は周期設定用
MOVW AX, #40-1
MOVW CR010, AX ; CR010はパルス幅設定用
MOV PRM00, #00000001B ; カウント・クロック = fxp/4 (= 2MHz)
MOV TOC00, #00011011B ; CR000,CR010との一致で出力反転、出力の初期値High、タイマ出力を
```

許可

```
MOV TMC00, #00001100B ; タイマ動作開始(TM00とCR000の一致でクリア&スタート)
```

; RAM、汎用レジスタの初期化

```
MOVW CURRENTD, AX ; パルス幅の初期値を現在のパルス幅として保存
MOVW HL, #PWIDTH ; パルス幅のテーブル・アドレスを指定
```

; 割り込みの設定

```
MOV    INTM0, #0000000B    ; INTP1の有効エッジ = 立下りエッジ
MOV    IFO,    #00H        ; 無効割り込み要求をクリアしておく
CLR1   PMK1              ; INTP1割り込みマスク解除

EI                                           ; ベクタ割り込み許可

;*****
;
;   メイン・ループ
;
;*****
MAIN_LOOP:
    NOP
    BR    $MAIN_LOOP      ; MAIN_LOOPへ

;*****
;
;   外部割り込みINTP1
;
;*****
INTERRUPT_P1:
    PUSH  AX              ; AXレジスタのデータをスタックへ退避

;----- チャタリング対策の10msウェイト -----
    MOV  A,    #50+1      ; 200us x 50 = 10ms
WAIT_CHAT:
    CLR1  TMIF00         ; INTTM000割り込み要求フラグをクリア
WAIT_INT:
    BF    TMIF00, $WAIT_INT ; INTTM000割り込み待ち
    DEC  A              ; Aレジスタを-1
    BNZ  $WAIT_CHAT     ; A = 0でなければ分岐

    CLR1  PIF1          ; INTP1割り込み要求をクリア

;----- チャタリング検出の判定 -----
    BT   P4.3, $END_INTP1 ; スイッチ入力がないと分岐する

;----- 新しいパルス幅の読み出し -----
    MOV  A,    L          ; テーブル・アドレスを読み出し
    ADD  A,    #2         ; テーブル・アドレスを+2
    AND  A,    #00001111B ; ビット0-3以外をマスク
    MOV  L,    A
    MOV  A,    [HL]      ; パルス幅下位8ビットを読み出し
```

```

XCH  A,    X
MOV  A,    [HL+1]      ; パルス幅上位8ビットを読み出し
MOVW NEXTD, AX        ; 新しいパルス幅を保存

;----- 現在のパルス幅と比較 -----
MOVW  AX,    CURRENTD  ; 現在のパルス幅を読み出し
ADDW  AX,    #OFFSETT  ; オフセット時間を加算
XCH  A,    X
SUB   A,    [HL]       ; 新しいパルス幅と比較
XCH  A,    X
SUBC  A,    [HL+1]

;----- パルス幅を変更する割り込み処理を指定 -----
BNC   $DECDUTY        ; 新しいパルス幅 < 現在のパルス幅なら分岐
CLR1  TMIF000        ; 無効な割り込み要求をクリア
CLR1  TMMK000        ; INTTM000割り込みマスクを解除
SET1  TMMK010        ; INTTM010割り込みをマスク
BR    $END_INTP1     ; END_INTP1へ分岐
DECDUTY:
CLR1  TMIF010        ; 無効な割り込み要求をクリア
CLR1  TMMK010        ; INTTM010割り込みマスクを解除
SET1  TMMK000        ; INTTM000割り込みをマスク

END_INTP1:
POP   AX             ; AXレジスタのデータを復帰
RETI                          ; 割り込み処理から復帰

;*****
;
;
;   割り込みINTTM000
;
;*****
INTERRUPT_TM000:
PUSH  AX             ; AXレジスタのデータをスタックへ退避

CLR1  TOC00.4       ; 出力反転禁止
MOVW  AX,    NEXTD
MOVW  CR010, AX     ; パルス幅(デューティ)変更
NOP                          ; タイマ00のカウント・クロック1周期分ウェイト
NOP
SET1  TOC00.4       ; 出力反転許可

MOVW  CURRENTD, AX   ; 新しいパルス幅を現在のパルス幅として保存

```

```

SET1  TMMK000          ; INTTM000割り込みをマスク

POP   AX              ; AXレジスタのデータを復帰
RETI                      ; 割り込み処理から復帰
    
```

```

;*****
;
; 割り込みINTTM010
;
;*****
;
    
```

INTERRUPT_TMO10:

```

PUSH  AX              ; AXレジスタのデータをスタックへ退避

CLR1  TOC00.4        ; 出力反転禁止
MOVW  AX, NEXTD
MOVW  CR010, AX      ; パルス幅(デューティ)変更
NOP                    ; タイマ00のカウンタ・クロック1周期分ウェイト
NOP
SET1  TOC00.4        ; 出力反転許可

MOVW  CURRENTD, AX   ; 新しいパルス幅を現在のパルス幅として保存
SET1  TMMK010        ; INTTM010割り込みをマスク

POP   AX              ; AXレジスタのデータを復帰
RETI                      ; 割り込み処理から復帰
    
```

end

main.c (C言語版)

```

/*****
    
```

NEC Electronics 78K0S/KB1+シリーズ

```

*****
    
```

78K0S/KB1+シリーズ サンプル・プログラム

```

*****
    
```

16ビット・タイマ00 (PPG出力)

```

*****
    
```

【履歴】

2007.7.-- 新規作成

```

*****
    
```

【概要】

本サンプルプログラムは、16ビット・タイマ00のPPG出力機能の使用例を示すものである。
 周期200usのPPG出力を行い、スイッチ入力の度にデューティ幅を20%ずつ変更する。

< 主な設定内容 >

- ・割り込みで起動される関数の宣言: INTP1 -> fn_intp1()
- ・割り込みで起動される関数の宣言: INTTM000 -> fn_inttm000()
- ・割り込みで起動される関数の宣言: INTTM010 -> fn_inttm010()
- ・ウォッチドッグ・タイマの動作停止
- ・低電圧検出電圧VLVIを4.3V ± 0.2Vに設定
- ・VDD VLVIとなった後にVDD < VLVIとなった場合、内部リセット信号発生(低電圧検出回路)
- ・CPUクロックを8MHzに設定
- ・周辺ハードウェアへの供給クロックを8MHzに設定
- ・外部割り込みINTP1の有効エッジ: 立ち下りエッジに設定
- ・スイッチ入力時のチャタリング検出時間 = 10ms

< 16ビット・タイマ00の設定 >

- ・動作モード: TM00とCR000の一致でタイマ・カウンタのクリア&スタート
- ・カウンタ・クロック = f_{xp}/4 (2MHz)
- ・CR000, CR010をコンペア・レジスタとして使用
- ・CR000は周期200usに初期化
- ・CR010はパルス幅20usに初期化
- ・CR000, CR010との一致で出力反転(PPG出力)
- ・出力の初期値(=アクティブレベル)をHighに設定
- ・P31の出力ラッチをLowに設定(T000出力を使用するため)
- ・P31を出力モードに設定(T000出力を使用するため)

< スイッチ入力回数とPPG出力のデューティ >

SW入力回数	PPG出力の デューティ	LEDの輝度
0回	10 %	90 %
1回	30 %	70 %
2回	50 %	50 %
3回	70 %	30 %
4回	90 %	10 %
5回	70 %	30 %

	6回		50 %		50 %	
	7回		30 %		70 %	

+-----+

8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：P43

出力ポート：P00-P03, P20-P23, P30-P33, P40-P42, P44-P47, P120-P123, P130

未使用のポートは全て出力ポートに設定しておく

*****/

/*=====

前処理指令

=====*/

```
#pragma SFR /* 特殊機能レジスタ(SFR)名を記述可能にする */
#pragma EI /* EI命令を記述可能にする */
#pragma DI /* DI命令を記述可能にする */
#pragma NOP /* NOP命令を記述可能にする */
#pragma interrupt INTP1 fn_intp1 /* 割り込み関数宣言:INTP1 */
#pragma interrupt INTTM000 fn_inttm000 /* 割り込み関数宣言:INTTM000 */
#pragma interrupt INTTM010 fn_inttm010 /* 割り込み関数宣言:INTTM000 */
```

```
#define offsetT 56/4 /* INTTM010の処理時間分の補正用 */
```

/*=====

グローバル変数、定数テーブルの定義

=====*/

```
sreg unsigned char g_ucSWcnt = 0; /* スイッチ入力カウンタ用8ビット変数 */
sreg unsigned int g_unNextD = 40-1; /* 新しいパルス幅(デューティ)保存用16ビット変数 */
sreg unsigned int g_unCurrentD = 40-1; /* 現在のパルス幅(デューティ)保存用16ビット変数 */
const unsigned int g_unOutData[8] = {40-1,40*3-1,40*5-1,40*7-1,40*9-1,40*7-1,40*5-1,40*3-1}; /* 出力デューティ・テーブル */
```

```

/*****
リセット解除後の初期化処理

*****/
void hdwinit(void){
    unsigned char ucCnt200us;    /* 200usウェイト用8ビット変数 */

/*-----
    ウォッチドッグ・タイマの設定 + 低電圧検出 + クロックの設定
-----*/

/* ウォッチドッグ・タイマの設定 */
    WDTM = 0b01110111;    /* ウォッチドッグ・タイマ動作停止 */

/* クロックの設定1 */
    PCC = 0b00000000;    /* CPUクロックfcpu = fxp (= fx/4 = 2MHz) */
    LSRCM = 0b00000001;    /* 低速内蔵発振器の発振を停止 */

/* リセット要因の確認 */
    if (!(RESF & 0b00000001)){    /* LVIリセット時は以降のLVI関連処理を省略 */

        /* 低電圧検出の設定 */
        LVIS = 0b00000000;    /* 低電圧検出レベルVLVI = 4.3V±0.2Vに設定 */
        LVION = 1;    /* 低電圧検出回路の動作許可 */

        for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){    /* 約200usウェイト */
            NOP();
        }

        while (LVIF){    /* VDD VLVI待ち */
            NOP();
        }

        LVIMD = 1;    /* VDD < VLVI時に内部リセット信号が発生するように設定 */
    }

/* クロックの設定2 */
    PPCC = 0b00000000;    /* 周辺ハードウェアへの供給クロックfxp = fx (= 8MHz)
                           -> CPUクロックfcpu = fxp = 8MHz */

/*-----
    ポート0の設定
-----*/

```

```

P0    = 0b00000000;          /* P00-P03の出力ラッチLow */
PM0   = 0b11110000;          /* P00-P03を出力ポートに設定 */

/*-----*/
      ポート2の設定
/*-----*/

P2    = 0b00000000;          /* P20-P23の出力ラッチLow*/
PM2   = 0b11110000;          /* P20-P23を出力ポートに設定 */

/*-----*/
      ポート3の設定
/*-----*/

P3    = 0b00000000;          /* P30-P33の出力ラッチLow */
PM3   = 0b11110000;          /* P30-P33を出力ポートに設定 */

/*-----*/
      ポート4の設定
/*-----*/

P4    = 0b00000000;          /* P40-P47の出力ラッチLow */
PU4   = 0b00001000;          /* P43に内蔵プルアップ抵抗を使用 */
PM4   = 0b00001000;          /* P40-P42,P44-P47を出力ポートに、P43を入力ポートに設定 */

/*-----*/
      ポート12の設定
/*-----*/

P12   = 0b00000000;          /* P120-P123の出力ラッチLow */
PM12  = 0b11110000;          /* P120-P123を出力ポートに設定 */

/*-----*/
      ポート13の設定
/*-----*/

P13   = 0b00000001;          /* P130の出力High */

/*-----*/
      16ビット・タイマ00の設定
/*-----*/

CRC00 = 0b00000000;          /* CR000,CR010をコンペア・レジスタとして使用 */
CR000 = 400-1;                /* CR000は周期設定用 */
CR010 = 40-1;                 /* CR010はパルス幅設定用 */
PRM00 = 0b00000001;          /* カウント・クロック = fxp/4 (= 2MHz) */
TOC00 = 0b00011011;          /* CR000,CR010との一致で出力反転、出力の初期値High、タイマ出力
を許可 */
TMC00 = 0b00001100;          /* タイマ動作開始(TM00とCR000の一致でクリア&スタート) */

```

```

/*-----
   割り込みの設定
-----*/

INTM0 = 0b00000000;      /* INTP1の有効エッジ = 立下りエッジ */
IF0 = 0x00;              /* 無効な割り込み要求をクリア */
PMK1 = 0;                /* INTP1割り込みマスクを解除 */

return;
}

/*****

   メイン・ループ

*****/

void main(void){

    EI();                  /* ベクタ割り込み許可 */

    while (1){
        NOP();
    }
}

/*****

   割り込みINTP1

*****/

__interrupt void fn_intp1(){
    unsigned char ucChat;      /* チャタリング除去用8ビット変数 */

    for (ucChat = 0; ucChat <50+1; ucChat++){      /* 10msウェイト用ループ */
        TMIF00 = 0;          /* INTTM000割り込みフラグをクリア */
        while (!TMIF00){    /* INTTM000割り込み待ち */
            NOP();
        }
    }

    PIF1 = 0;                /* INTP1割り込み要求をクリア */

    if (!P4.3){              /* 10ms以上SWオンの場合の処理 */
        g_ucSWcnt += 1;      /* スイッチ入力回数を+1 */
    }
}

```

```
g_ucSWcnt &= 0b00000111;          /* ビット0-2以外をマスク */

g_unNextD = g_unOutData[g_ucSWcnt]; /* 新しいパルス幅を保存 */

if (g_unNextD > g_unCurrentD + offsetT){ /* パルス幅を大きくするときはINTTM000割
り込みを使用 */
    TMIF000 = 0;          /* 無効な割り込み要求をクリア */
    TMMK000 = 0;          /* INTTM000割り込みマスクを解除 */
    TMMK010 = 1;          /* INTTM010割り込みをマスク */
}
else {                    /* パルス幅を小さくするときはINTTM010割り込みを使用 */
    TMIF010 = 0;          /* 無効な割り込み要求をクリア */
    TMMK010 = 0;          /* INTTM010割り込みマスクを解除 */
    TMMK000 = 1;          /* INTTM000割り込みをマスク */
}
}

return;
}

/*****

割り込みINTTM000

*****/

__interrupt void fn_inttm000(){

    TOC00.4 = 0;          /* 出力反転禁止 */
    CR010 = g_unNextD;    /* パルス幅(デューティ)変更 */
    NOP();                /* タイマ00のカウンタ・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1;          /* 出力反転許可 */

    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK000 = 1;          /* INTTM000割り込みをマスク */

    return;
}
```

```

/*****
割り込みINTTM010

*****/
__interrupt void fn_inttm010(){

    TOC00.4 = 0;          /* 出力反転禁止 */
    CRO10 = g_unNextD;   /* パルス幅(デューティ)変更 */
    NOP();               /* タイマ00のカウンタ・クロック1周期分ウェイト */
    NOP();
    TOC00.4 = 1;        /* 出力反転許可 */

    g_unCurrentD = g_unNextD; /* 新しいパルス幅を現在のパルス幅として保存 */
    TMMK010 = 1;         /* INTTM010割り込みをマスク */

    return;
}

```

op.asm (アセンブリ言語版とC言語版共通)

```

=====
;
;
; オプション・バイトの設定
;
=====
OPBT      CSEG  AT      0080H
          DB      10011100B      ; オプション・バイトの設定
;
;          |||
;          |||+----- 低速内蔵発振器はソフトウェアで停止可能
;          |++----- 高速内蔵発振クロック(8MHz)を使用
;          +-----  P34/RESET端子をリセット端子として使用

          DB      11111111B      ; プロテクト・バイトの設定(セルフプログラミング用)
;
;          ||| |||
;          +----- 全てのブロックへの書き込み許可

end

```

付録B 改版履歴

本文欄外の 印は、本版で改訂された主な箇所を示しています。この" "をPDF上でコピーして「検索する文字列」に指定することによって、改版箇所を容易に検索できます。

版 数	発行年月	改版箇所	改版内容
第1版	September 2007	-	-
第2版	July 2008	p.29	第5章 システム・シミュレータ SM+での動作確認 ・注意文中の（2007年9月現在）を（2008年7月現在）に変更
		pp.29-31	5.1 サンプル・プログラムのビルドを変更
		p.31	5.2 SM+での動作 ・（1）を追加
		p.36	第6章 関連資料 ・フラッシュ書き込み簡単マニュアル（MINICUBE2編） インフォメーションを追加

【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。
