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ものですが、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て、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。

RENESAS

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）

パルス幅測定編

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のパルス幅測定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のパルス幅測定機能を使用して，TI000端子から入力された信号のパルス幅を測定します。

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

資料番号　U18889JJ2V0AN00（**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，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，ソフトウエ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，ソフトウエ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。

M8E　02.11

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のパルス幅測定機能の使用例を示しています。TI000端子から入力された信号のパルス幅を8回測定します。

1. 1　初期設定の主な内容

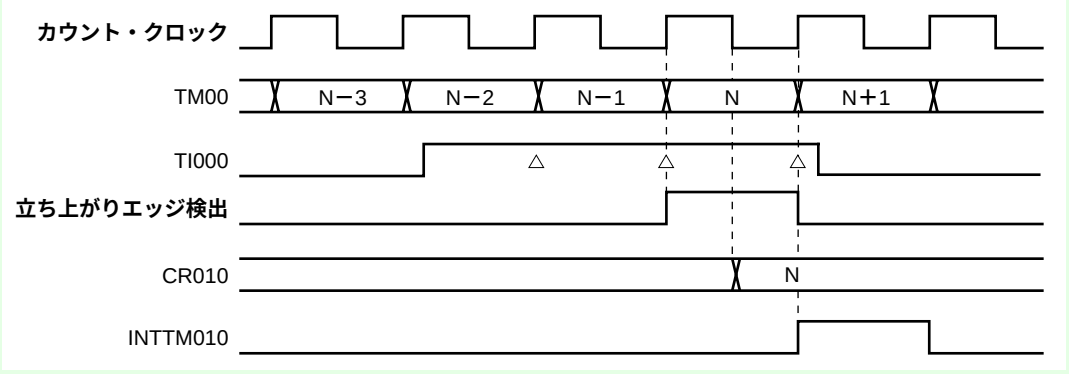
初期設定の主な内容は，次のとおりです。

●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の動作モードをコンペア・レジスタに，CR010の動作モードをキャプチャ・レジスタに設定
　・CR000に「FFFFH」を設定
　・TI000端子の有効エッジを立ち下がり，立ち上がりの両エッジに，カウント・クロックを$f_{XP}/2^2$（2 MHz）に設定
　・動作モードをTI000端子の有効エッジでクリア&スタートに設定
●INTTM000とINTTM010の割り込みを許可

注　オプション・バイトで設定します。

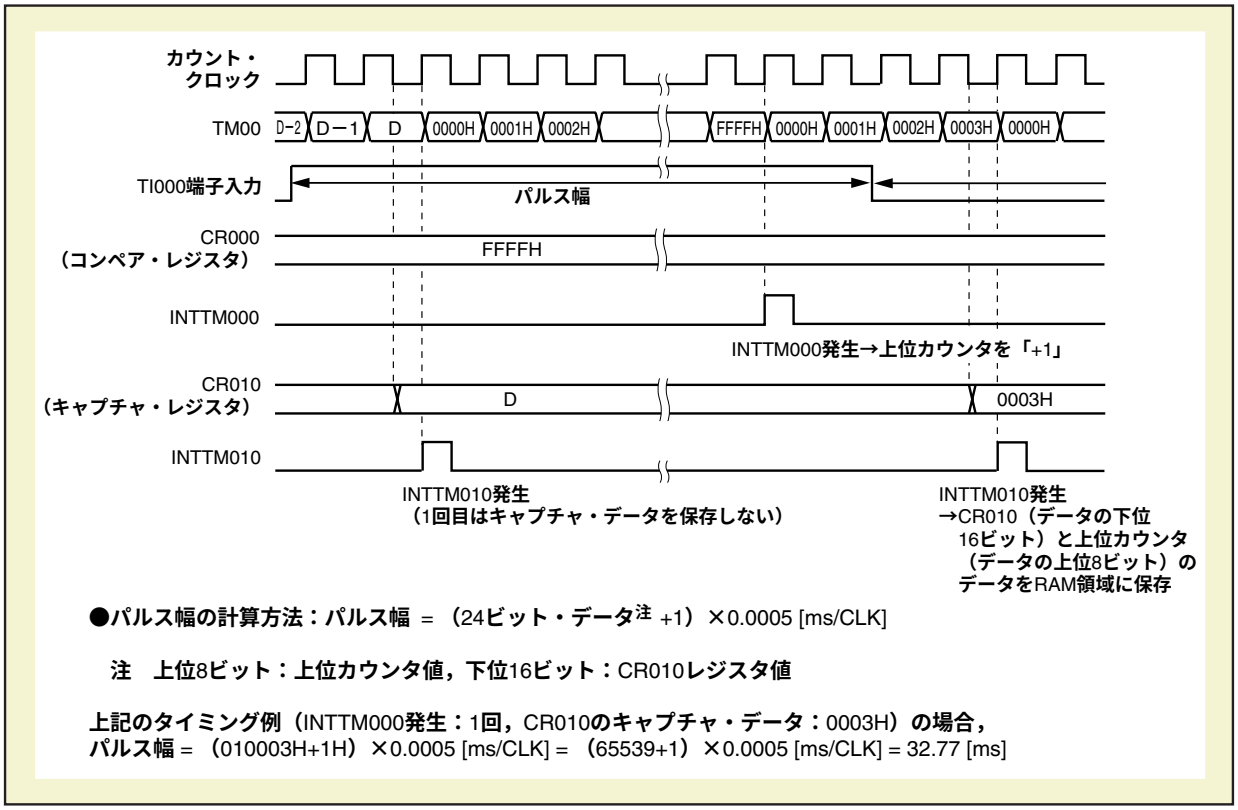
【コラム】キャプチャ動作タイミング

16ビット・タイマ／イベント・カウント00では，短いパルス幅のノイズを除去するために，TI000端子またはTI010端子の有効レベルを2回検出することではじめてキャプチャ動作を行います。したがって，カウント・クロックの2周期分より長い入力パルスが必要となります。下図に，立ち上がりエッジ指定時のCR010キャプチャ動作例を示します。



1. 2　メイン・ループ以降の内容

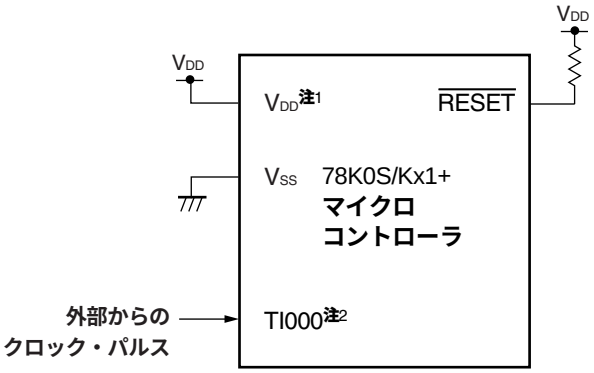
初期設定完了後は，16ビット・タイマ／イベント・カウンタ00の割り込み（INTTM000とINTTM010）発生を利用して，TI000端子から入力された信号のパルス幅を8回測定します。



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

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を説明します。



注1.　4.5 V≦VDD≦5.5 Vの電圧範囲で使用してください。

2.　TI000/INTP0/P30:　78K0S/KA1+, 78K0S/KB1+マイクロコントローラ

TI000/ANI0/TOH1/P20:　78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

注意1.　AVREF端子はVDD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.　AVSS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AVREF, AVSS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。

第3章　ソフトウエアについて

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エア初期化処理とメイン処理のソース・ファイル	●注	●注
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）	●	●
tm00cap.prw	統合開発環境 PM+用ワーク･スペース・ファイル		●
tm00cap.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。

- パルス幅測定機能　：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
- 外部からのパルス入力　：TI000[注]

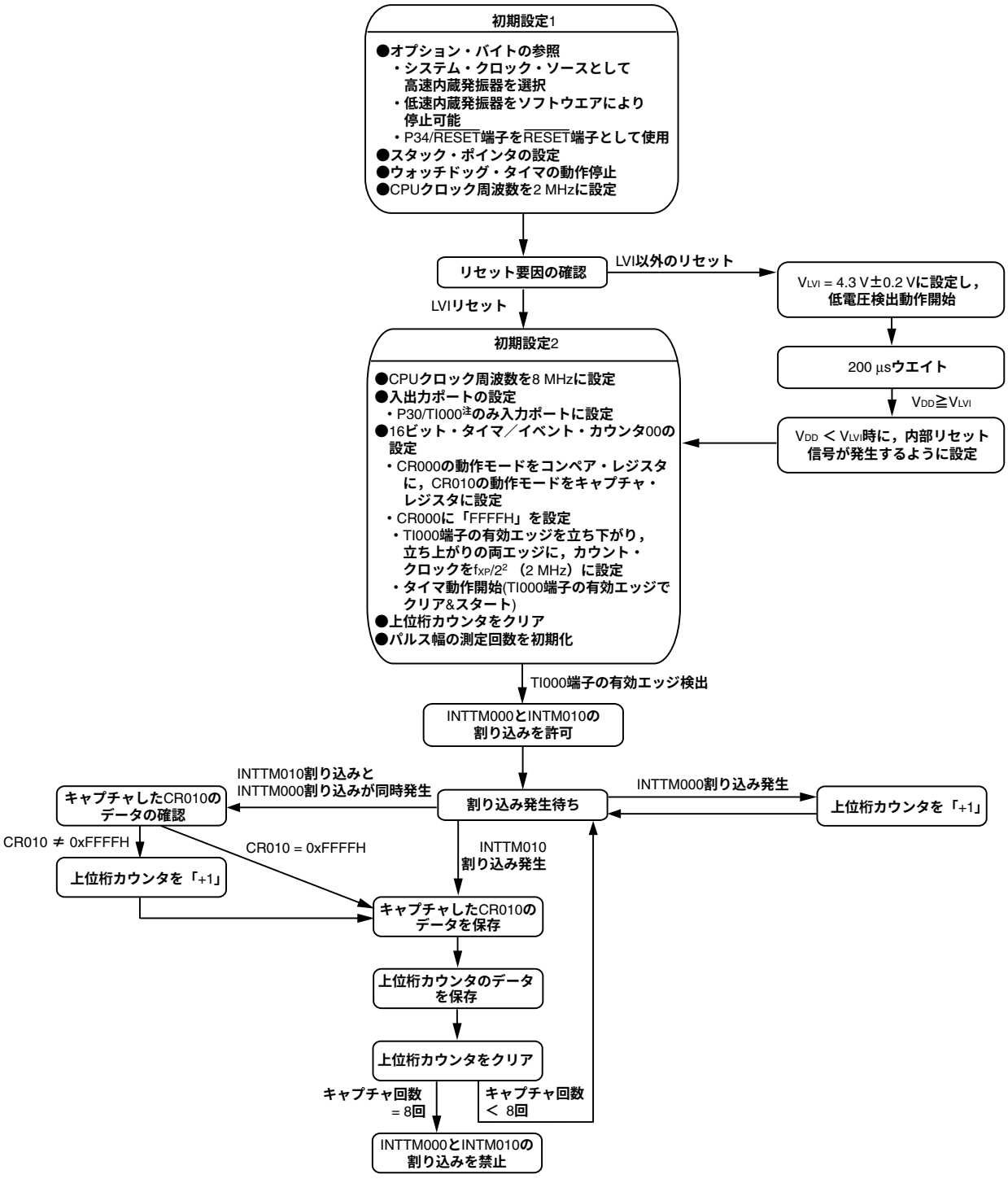
注　TI000/INTP0/P30:　78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
TI000/ANI0/TOH1/P20:　78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

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（パルス幅測定機能）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の割り込み（INTTM000, INTTM010）発生を利用して，TI000端子から入力された信号のパルス幅を8回測定します。

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。

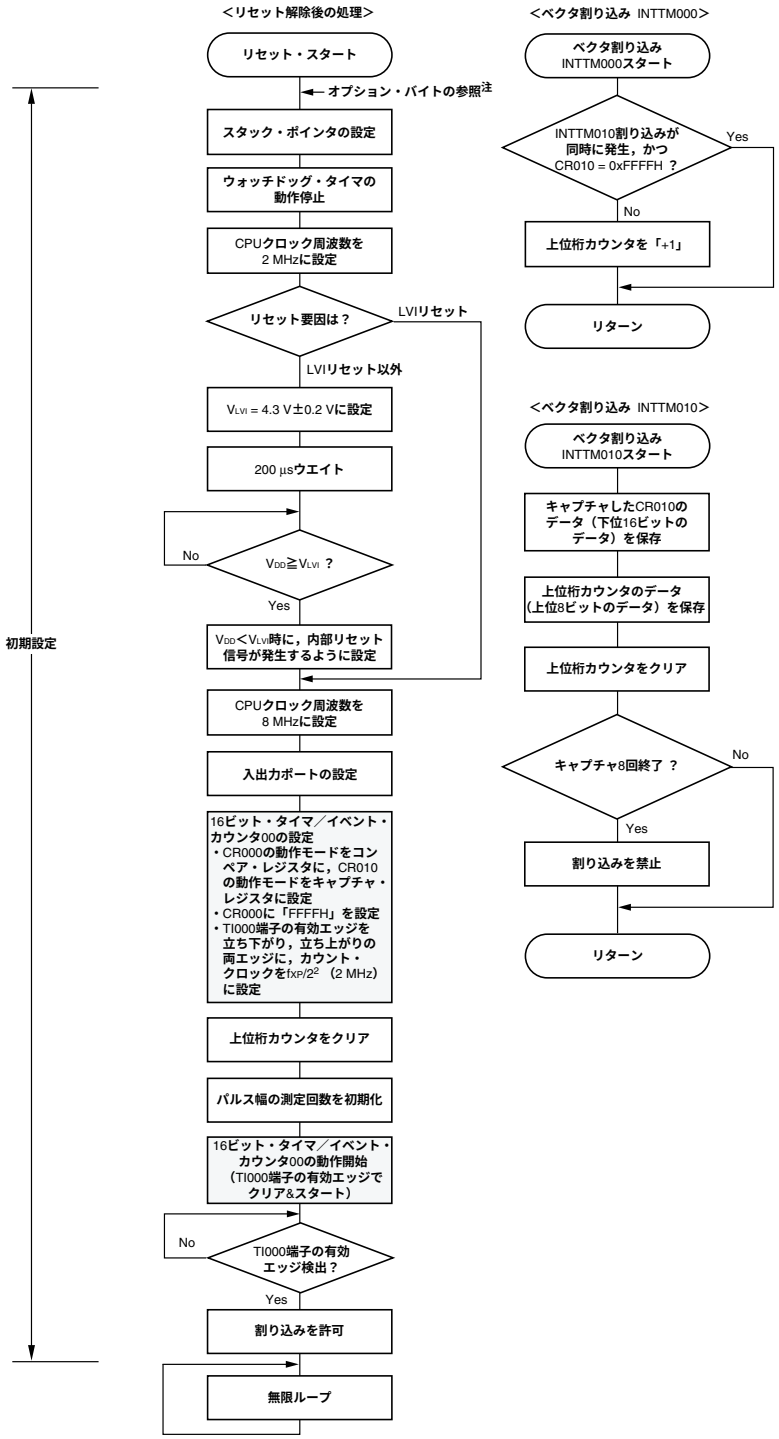


注　TI000/P30: 78K0S/KA1+, 78K0S/KB1+**マイクロコントローラ**

TI000/P20: 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エアで停止可**
- **P34/$\overline{\text{RESET}}$端子を$\overline{\text{RESET}}$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のパルス幅測定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）　2.7 V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のパルス幅測定機能の設定

16ビット・タイマ／イベント・カウンタ00のパルス幅測定機能を使用する際に使用するレジスタには，主に次の7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）
- プリスケーラ・モード・レジスタ00（PRM00）
- 16ビット・タイマ・モード・コントロール・レジスタ00（TMC00）
- 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）
- 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　パルス幅測定機能ではTI000端子のみ，またはTI000端子とTI010端子をタイマ入力として使用するので，次のように設定します。

・TI000端子

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1	PMC20 = 0

・TI010端子

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM31 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM21 = 1	PMC21 = 0

＜パルス幅測定として使用する場合の基本的な動作設定手順例＞

① CRC00レジスタの設定

② PRM00レジスタでカウント・クロックを設定

③ TMC00レジスタの設定：動作開始

注意　①，②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

CRC00

0	0	0	0	0	CRC002	CRC001	CRC000

CR000の動作モードの選択

0	コンペア・レジスタとして動作
1	キャプチャ・レジスタとして動作

CR000のキャプチャ・トリガの選択

0	TI010端子の有効エッジでキャプチャする
1	TI000端子の有効エッジの逆相でキャプチャする注

CR010の動作モードの選択

0	コンペア・レジスタとして動作
1	キャプチャ・レジスタとして動作

注　CRC000が1のとき，TI000端子の有効エッジに，立ち下がり，立ち上がりの両エッジを選択した場合には，CR000レジスタはキャプチャ動作を行えません。

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。

3. 確実にキャプチャをするためにキャプチャ・トリガは，プリスケーラ・モード・レジスタ00（PRM00）で選択したカウント・クロックの2周期分より長いパルスを必要とします。

（2）PRM00レジスタの設定

PRM00レジスタは，TM00カウンタのカウント・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4－2　プリスケーラ・モード・レジスタ00（PRM00）のフォーマット

PRM00

ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

カウント・クロック（fsam）の選択

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$
1	1	TI000端子の有効エッジ[注]

TI000端子の有効エッジの選択

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	設定禁止
1	1	立ち上がり，立ち下がりの両エッジ

TI010端子の有効エッジの選択

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	設定禁止
1	1	立ち上がり，立ち下がりの両エッジ

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を必要とします。

備考　f_{XP}：周辺ハードウエ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。

① 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

② 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

③ 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**の有効エッジをキャプチャ・トリガとして使用する場合，ノイズ除去のためにプリスケーラ・モード・レジスタ**00（PRM00）**で選択したカウント・クロック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。**

（3**）**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4－3　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**

TMC00

0	0	0	0	TMC003	TMC002	TMC001	OVF00

16**ビット・タイマ・カウンタ**00（TM00）**のオーバフロー検出**

0	**オーバフローなし**
1	**オーバフローあり**

			動作モードおよびクリア・モードの選択	TO00**の反転タイミングの選択**	**割り込み要求の発生**
0	0	0	**動作停止（**TM00**は**0**にクリア）**	**変化なし**	**発生しない**
0	0	1			
0	1	0	**フリー・ランニング・モード**	TM00 **と** CR000 **の一致または** TM00 **と** CR010 **の一致**	**＜コンペア・レジスタとして使用時＞** TM00 **と** CR000 **の一致または** TM00 **と** CR010 **の一致で発生** **＜キャプチャ・レジスタとして使用時＞** TI000**端子，**TI010**端子の有効エッジで発生**
0	1	1		TM00 **と** CR000 **の一致，**TM00**と** CR010**の一致，または** TI000 **端子の有効エッジ**	
1	0	0	TI000**端子の有効エッジでクリア&スタート**	**－**	
1	0	1			
1	1	0	TM00 **と** CR000 **の一致でクリア&スタート**	TM00 **と** CR000 **の一致または** TM00 **と** CR010 **の一致**	
1	1	1		TM00 **と** CR000 **の一致，**TM00**と** CR010**の一致，または** TI000 **端子の有効エッジ**	

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。**

（4）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4－4　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**

CR000

・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（**INTTM000）**を発生します。**

・CR000**をキャプチャ・レジスタとして使用するとき**

キャプチャ・トリガとしてTI000**端子，または**TI010**端子の有効エッジが選択できます。**TI000，TI010**端子の有効エッジは，**PRM00**レジスタで設定します。**

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**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**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，**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。**

注意5.　CR000**をキャプチャ・レジスタとして使用しているとき，レジスタ・リード期間とキャプチャ・トリガの入力が競合した場合，キャプチャ・トリガ入力が優先され，**CR000**のリード・データは不定となります。またタイマのカウント停止とキャプチャ・トリガの入力が競合した場合，キャプチャ・データは不定となります。**

6.　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。**

（5**）**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4－5　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

・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。**

・CR010**をキャプチャ・レジスタとして使用するとき**

キャプチャ・トリガとしてTI000**端子の有効エッジが選択できます。**TI000**端子の有効エッジは，**PRM00**レジスタで設定します。**

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.　CR010**をキャプチャ・レジスタとして使用しているとき，レジスタ・リード期間とキャプチャ・トリガの入力が競合した場合，キャプチャ・トリガ入力が優先され，**CR010**のリード・データは不定となります。またタイマのカウント停止とキャプチャ・トリガの入力が競合した場合，キャプチャ・データは不定となります。**

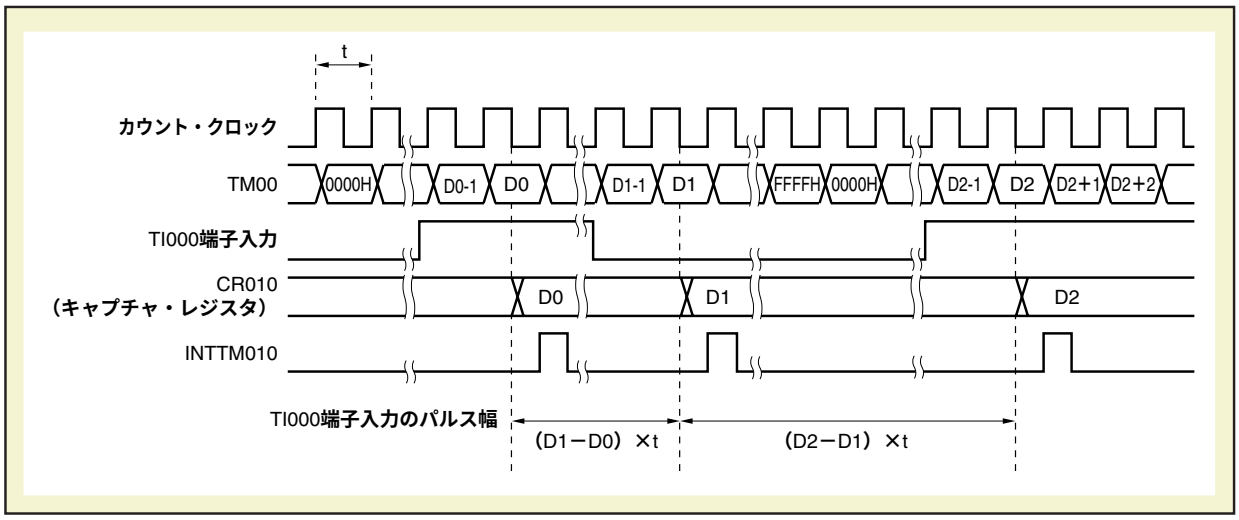
6.　TM00**カウンタ動作中に**CR010**レジスタを変更すると，誤動作する可能性があります。**

【例　1】TI000端子1本の入力信号でパルス幅を測定する場合（CR010レジスタをキャプチャ・レジスタとして使用，フリー・ランニング・モード）

TM00カウンタをフリー・ランニングで動作させているとき，TI000端子に入力される信号のパルス幅を測定します。TI000端子の有効エッジ検出により，TM00カウンタのカウント値をCR010レジスタにキャプチャします。

注意　この動作例で測定できるパルス幅は，タイマ・カウンタの1周期までです。

図4－6　TI000端子1本の入力信号でパルス幅を測定する場合のタイミング例（フリー・ランニング・モード，両エッジ指定時）



（1）レジスタの設定内容

① CRC00

0	0	0	0	0	1	x	0

CR000の動作モードの選択

0	コンペア・レジスタとして動作

CR000のキャプチャ・トリガの選択

x	設定無効

CRC010の動作モードの選択

1	キャプチャ・レジスタとして動作

② PRM00

x	x	0/1	0/1	0	0	0/1	0/1

カウント・クロック（fsam）の選択（”1, 1”は設定禁止）

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$

TI000端子の有効エッジの選択（”1, 0”は設定禁止）

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

TI010端子の有効エッジの選択

x	x	設定無効（”1, 0”は設定禁止）

③ TMC00

0	0	0	0	0	1	x	0

16ビット・タイマ・カウンタ00（TM00）のオーバフロー検出

0	オーバフローなし

動作モードおよびクリア・モードの選択

0	1	0	フリー・ランニング・モード
0	1	1	

④ PMx, PMCx

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1	PMC20 = 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。また，TI000端子の有効エッジを両エッジに，カウント・クロックをf_{XP}（システム・クロック周波数）に設定しています。

① 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）

```
SET1   PM3.0
MOV    CRC00, #00000100B
MOV    PRM00, #00110000B
MOV    TMC00, #00000100B
```

② C言語の場合（78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時）

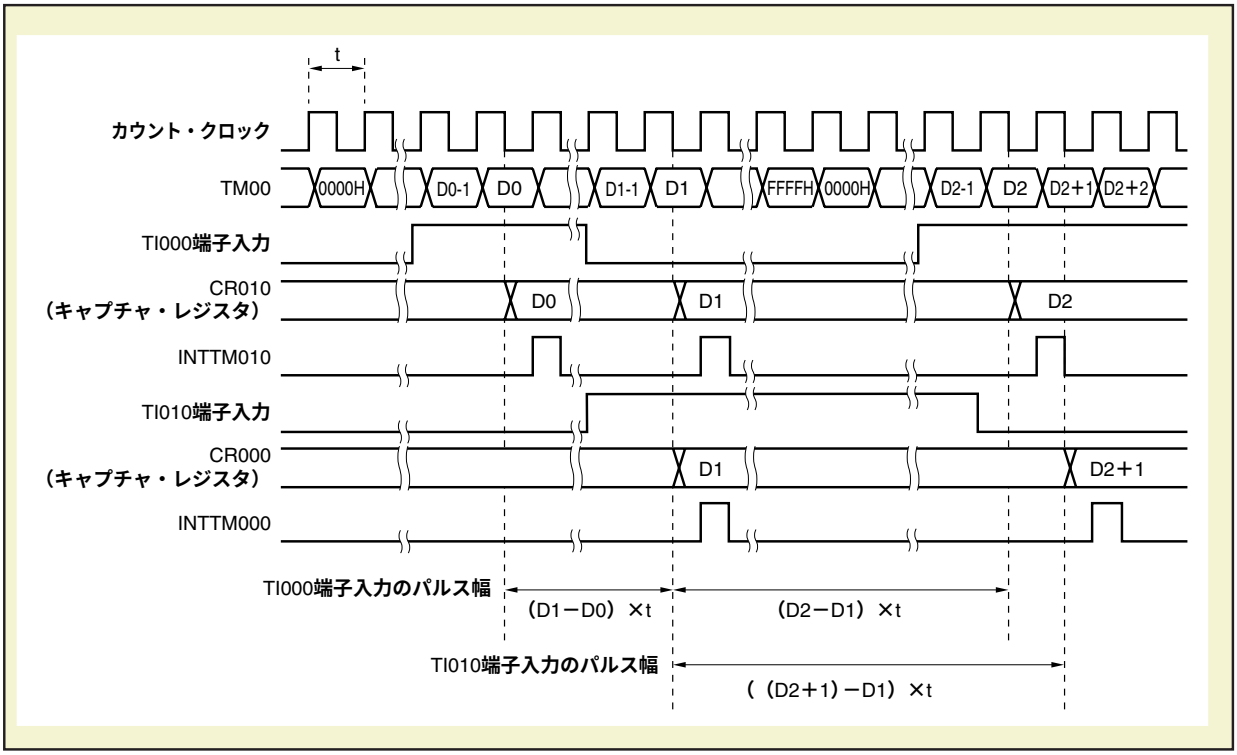
```
PM3.0 = 1;
CRC00 = 0b00000100;
PRM00 = 0b00110000;
TMC00 = 0b00000100;
```

【例　2】TI000端子およびTI010端子の2本の入力信号でパルス幅を測定する場合（CR000レジスタとCR010レジスタをキャプチャ・レジスタとして使用，フリー・ランニング・モード）

TM00カウンタをフリー・ランニングで動作させているとき，TI000端子およびTI010端子に入力される2つの信号のパルス幅を同時に測定します。TI000端子の有効エッジ検出により，TM00カウンタのカウント値をCR010レジスタにキャプチャし，TI010端子の有効エッジ検出により，TM00カウンタのカウント値をCR000レジスタにキャプチャします。

注意　この動作例で測定できるパルス幅は，タイマ・カウンタの1周期までです。

図4－7　TI000端子およびTI010端子の2本の入力信号でパルス幅を測定する場合のタイミング例（フリー・ランニング・モード，両エッジ指定時）



(1) レジスタの設定内容

① CRC00

0	0	0	0	0	1	0	1

CR000の動作モードの選択

1	キャプチャ・レジスタとして動作

CR000のキャプチャ・トリガの選択

0	TI010端子の有効エッジでキャプチャする

CRC010の動作モードの選択

1	キャプチャ・レジスタとして動作

② PRM00

0/1	0/1	0/1	0/1	0	0	0/1	0/1

カウント・クロック（fsam）の選択（”1, 1”は設定禁止）

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$

TI000端子の有効エッジの選択（”1, 0”は設定禁止）

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

TI010端子の有効エッジの選択（”1, 0”は設定禁止）

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

③ TMC00

0	0	0	0	0	1	x	0

16ビット・タイマ・カウンタ00（TM00）のオーバフロー検出

0	オーバフローなし

動作モードおよびクリア・モードの選択

0	1	0	フリー・ランニング・モード
0	1	1	

④ PMx, PMCx

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1, PM31 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1, PM21 = 1	PMC20 = 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。また，TI000端子とTI010端子の有効エッジを両エッジに，カウント・クロックをf_{XP}（システム・クロック周波数）に設定しています。

① 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）

```
SET1   PM3.0
SET1   PM3.1
MOV    CRC00, #00000101B
MOV    PRM00, #11110000B
MOV    TMC00, #00000100B
```

② C言語の場合（78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時）

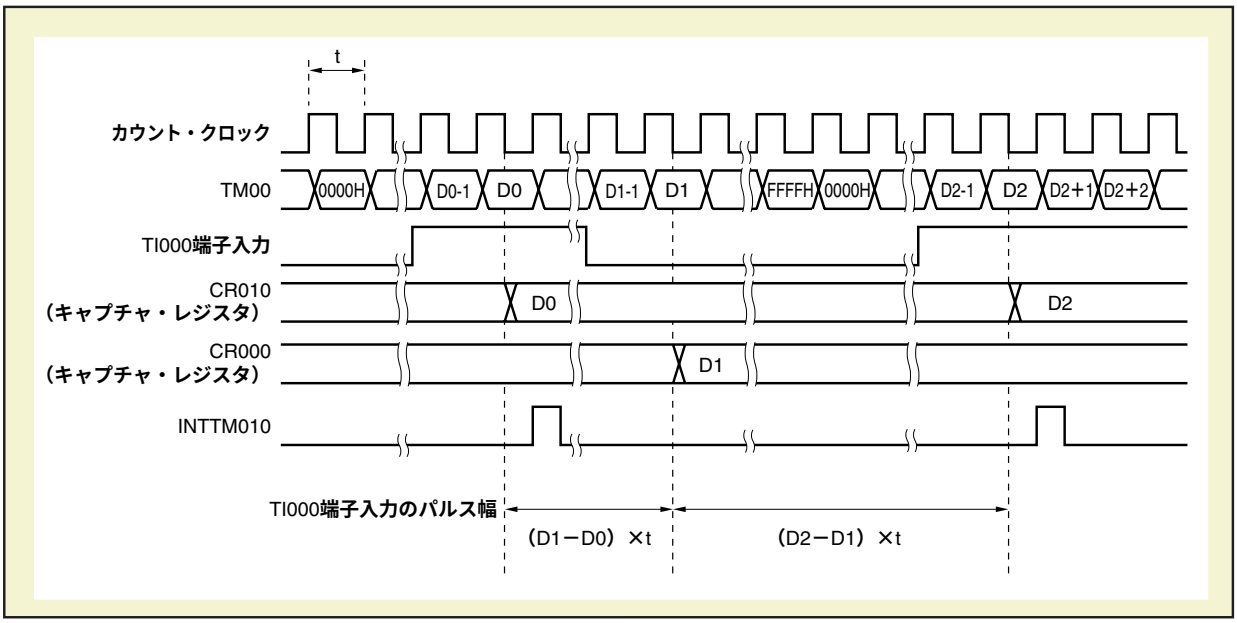
```
PM3.0 = 1;
PM3.1 = 1;
CRC00 = 0b00000101;
PRM00 = 0b11110000;
TMC00 = 0b00000100;
```

【例　3】TI000端子1本の入力信号でパルス幅を測定する場合（CR000レジスタとCR010レジスタをキャプチャ・レジスタとして使用，フリー・ランニング・モード）

TM00カウンタをフリー・ランニングで動作させているとき， TI000端子に入力される信号のパルス幅を測定します。TI000端子の有効エッジ検出により，TM00カウンタのカウント値をCR010レジスタにキャプチャし，TI000端子の有効エッジ検出の逆相により，TM00カウンタのカウント値をCR000レジスタにキャプチャします。TI000端子の有効エッジ検出の設定は，立ち上がりエッジまたは立ち下がりエッジにしてください。

注意　この動作例で測定できるパルス幅は，タイマ・カウンタの1周期までです。

図4－8　TI000端子1本の入力信号でパルス幅を測定する場合のタイミング例（フリー・ランニング・モード，立ち上がりエッジ指定時）



(1) レジスタの設定内容

① CRC00

0	0	0	0	0	1	1	1

CR000の動作モードの選択

1	キャプチャ・レジスタとして動作

CR000のキャプチャ・トリガの選択

1	TI000端子の有効エッジの逆相でキャプチャする注

CRC010の動作モードの選択

1	キャプチャ・レジスタとして動作

注　CRC000が1のとき，TI000端子の有効エッジに，立ち下がり，立ち上がりの両エッジを選択した場合には，CR000レジスタはキャプチャ動作を行えません。なお，CRC001が1のとき，TI010端子の有効エッジによるCR000レジスタへのキャプチャ動作を行えませんが，INTTM000は発生するため，TI010端子を外部割り込みとして使用することができます。

② PRM00

x	x	0	0/1	0	0	0/1	0/1

カウント・クロック（fsam）の選択（”1, 1”は設定禁止）

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$

TI000端子の有効エッジの選択

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ

TI010端子の有効エッジの選択

x	x	設定無効（”1, 0”は設定禁止）

③ TMC00

0	0	0	0	0	1	x	0

16ビット・タイマ・カウンタ00（TM00）のオーバフロー検出

0	オーバフローなし

動作モードおよびクリア・モードの選択

0	1	0	フリー・ランニング・モード
0	1	1	

④ PMx, PMCx

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1	PMC20 = 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。またTI000端子の有効エッジは立ち上がりエッジに，カウント・クロックをf_{XP}（システム・クロック周波数）に設定しています。

① 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）

```
SET1   PM3.0
MOV    CRC00, #00000111B
MOV    PRM00, #00010000B
MOV    TMC00, #00000100B
```

② C言語の場合（78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時）

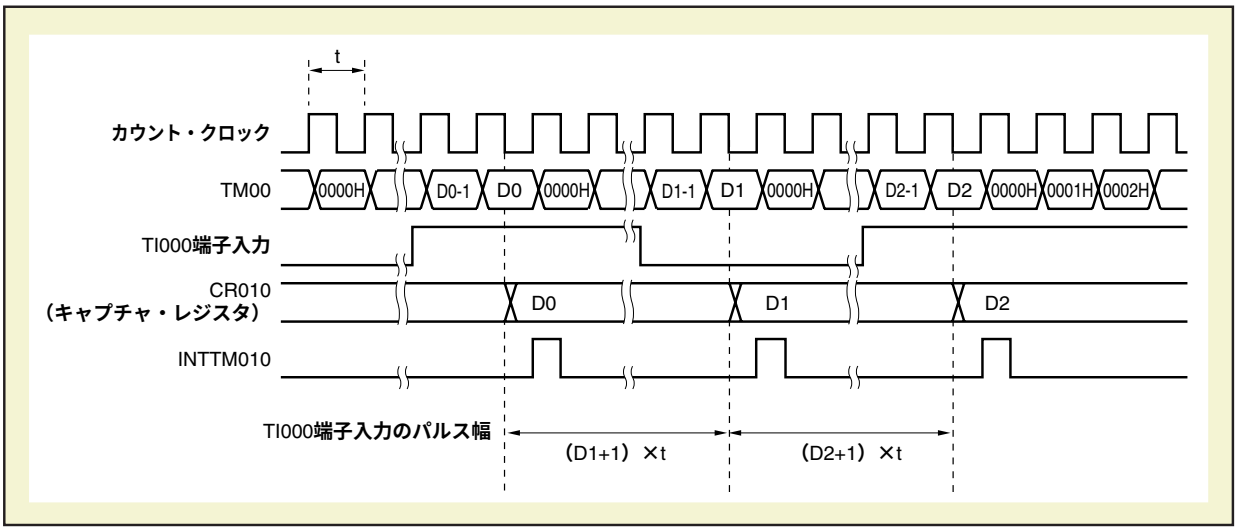
```
PM3.0 = 1;
CRC00 = 0b00000111;
PRM00 = 0b00010000;
TMC00 = 0b00000100;
```

【例　4】TI000端子1本の入力信号でパルス幅を測定する場合（CR000レジスタをキャプチャ・レジスタとして使用，TI000端子の有効エッジ入力によるクリア＆スタート・モード）

TM00カウンタをTI000端子の有効エッジ入力によるクリア＆スタート・モードで動作させているとき，TI000端子に入力される信号のパルス幅を測定します。TI000端子の有効エッジ検出により，TM00カウンタのカウント値をCR010レジスタにキャプチャしたあと，TM00カウンタをクリアしてカウントを再開し，TI000端子に入力された信号のパルス幅を測定します。

注意　この動作例で測定できるパルス幅は，タイマ・カウンタの1周期までです。

図4－9　TI000端子1本の入力信号でパルス幅を測定する場合のタイミング例（TI000端子の有効エッジ入力によるクリア＆スタート・モード，両エッジ指定時）



（1）レジスタの設定内容

① CRC00

0	0	0	0	0	1	x	0

CR000の動作モードの選択

0	コンペア・レジスタとして動作

CR000のキャプチャ・トリガの選択

x	設定無効

CRC010の動作モードの選択

1	キャプチャ・レジスタとして動作

② PRM00

x	x	0/1	0/1	0	0	0/1	0/1

カウント・クロック（fsam）の選択（”1, 1”は設定禁止）

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$

TI000端子の有効エッジの選択（”1, 0”は設定禁止）

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

TI010端子の有効エッジの選択

x	x	設定無効（”1, 0”は設定禁止）

③ TMC00

0	0	0	0	1	0	x	0

16ビット・タイマ・カウンタ00（TM00）のオーバフロー検出

0	オーバフローなし

動作モードおよびクリア・モードの選択

1	0	0	TI000端子の有効エッジでクリア&スタート
1	0	1	

④ PMx, PMCx

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1	PMC20 = 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。また，TI000端子の有効エッジを両エッジに，カウント・クロックをf_{XP}（システム・クロック周波数）に設定しています。

① 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）

```
SET1   PM3.0
MOV    CRC00, #00000100B
MOV    PRM00, #00110000B
MOV    TMC00, #00001000B
```

② C言語の場合（78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時）

```
PM3.0 = 1;
CRC00 = 0b00000100;
PRM00 = 0b00110000;
TMC00 = 0b00001000;
```

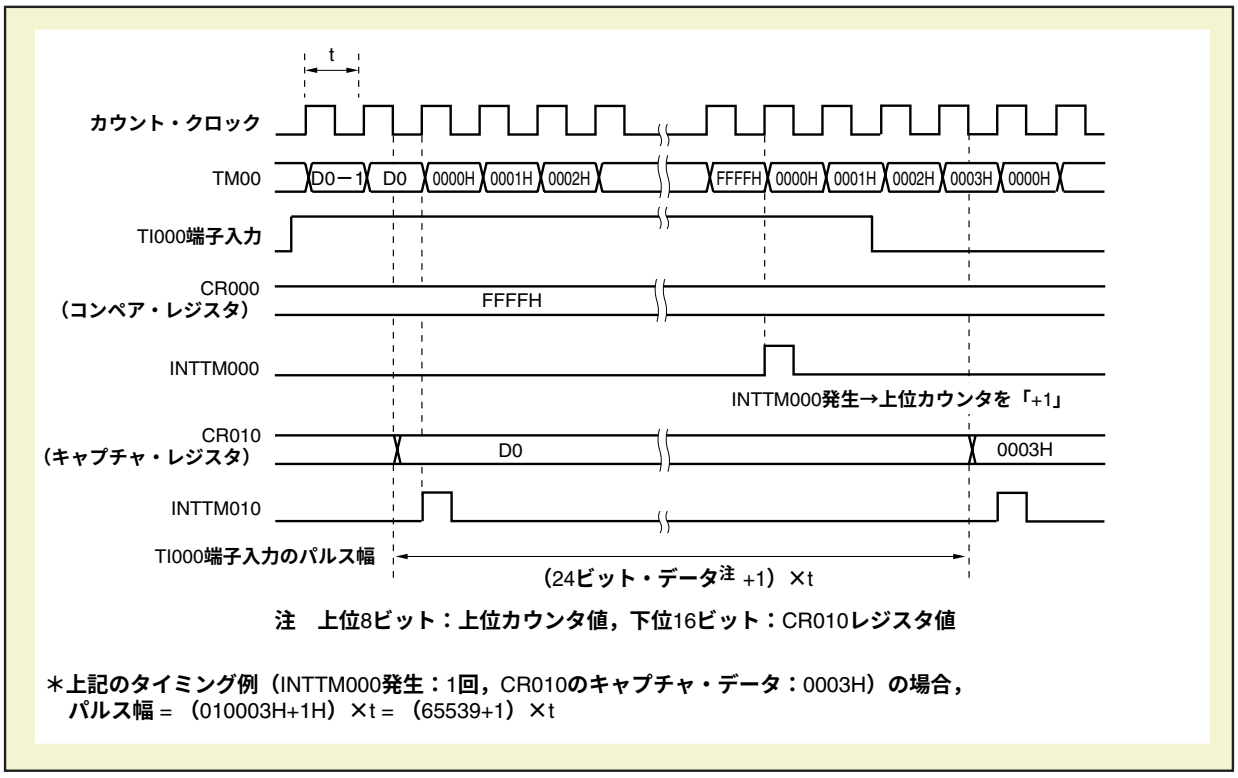
【例　5】TI000端子1本の入力信号で，TM00カウンタの1周期よりも長いパルス幅を測定する場合
（CR010レジスタをキャプチャ・レジスタ，CR000レジスタをコンペア・レジスタとして使用，TI000端子の有効エッジ入力によるクリア＆スタート・モード）
（本サンプル・プログラム・ソースと同内容）

TM00カウンタをTI000端子の有効エッジ入力によるクリア＆スタート・モードで動作させているとき，TI000端子に入力される信号のパルス幅を測定します。

CR000レジスタの動作モードをコンペア・レジスタに設定し，CR000にFFFFHを設定すると，CR000レジスタとTM00カウンタの値が一致し，TM00カウンタがFFFFHから0000Hになったときに，INTTM000割り込みが発生します。これを，オーバフロー割り込みとして使用し，INTTM000割り込みが発生したら上位桁をカウント・アップすることで，TM00カウンタの1周期よりも長いパルス幅の計測が可能となります。

TI000端子の有効エッジ検出により，TM00カウンタのカウント値をCR010レジスタにキャプチャしたあと，TM00カウンタをクリアしてカウントを再開し，TI000端子に入力された信号のパルス幅を測定します。

図4－10　TI000端子1本の入力信号で，TM00カウンタの1周期よりも長いパルス幅を測定する場合のタイミング例
（TI000端子の有効エッジ入力によるクリア＆スタート・モード，両エッジ指定時）



注　上位8ビット：上位カウンタ値，下位16ビット：CR010レジスタ値

＊上記のタイミング例（INTTM000発生：1回，CR010のキャプチャ・データ：0003H）の場合，
パルス幅 =（010003H+1H）×t =（65539+1）×t

(1) レジスタの設定内容

① CRC00

0	0	0	0	0	1	x	0

CR000の動作モードの選択

0	コンペア・レジスタとして動作

CR000のキャプチャ・トリガの選択

x	設定無効

CRC010の動作モードの選択

1	キャプチャ・レジスタとして動作

② PRM00

x	x	0/1	0/1	0	0	0/1	0/1

カウント・クロック（fsam）の選択（”1, 1”は設定禁止）

0	0	f_{XP}
0	1	$f_{XP}/2^2$
1	0	$f_{XP}/2^8$

TI000端子の有効エッジの選択（”1, 0”は設定禁止）

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

TI010端子の有効エッジの選択

x	x	設定無効（”1, 0”は設定禁止）

③ TMC00

0	0	0	0	1	0	x	0

16ビット・タイマ・カウンタ00（TM00）のオーバフロー検出

0	オーバフローなし

動作モードおよびクリア・モードの選択

1	0	0	TI000端子の有効エッジでクリア&スタート
1	0	1	

④ PMx, PMCx

	PMxレジスタ	PMCxレジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	PM30 = 1	設定不要
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	PM20 = 1	PMC20 = 0

⑤ CR000：FFFFH

(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。また，TI000端子の有効エッジを両エッジに，カウント・クロック（fsam）を $f_{XP}/2^2$ に設定しています。

① 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）

```
SET1   PM3.0
MOV    CRC00, #00000100B
MOVW   AX, #0FFFFH
MOVW   CR000, AX
MOV    PRM00, #00110001B
MOV    TMC00, #00001000B
```

② C言語の場合（78K0S/KA1+, 78K0S/KB1+マイクロコントローラ使用時）

```
PM3.0 = 1;
CRC00 = 0b00000100;
CR000 = 0xFFFF;
PRM00 = 0b00110001;
TMC00 = 0b00001000;
```

【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　5】と同内容）。

（1）アセンブリ言語

```
XMAIN  CSEG   UNIT
RESET_START:
                    •
                    •
                    •
        MOV     CRC00, #00000100B   ; CR010をキャプチャ・レジスタとして使用
        MOVW    AX,    #0FFFFH
        MOVW    CR000, AX           ; CR000はオーバフロー検出用
        MOV     PRM00, #00110001B   ; TI000端子の有効エッジを両エッジに指定、カウント・クロック = fxp/4
        MOV     TOC00, #00000000B   ; タイマ出力は行なわない

        MOV     MK0,   #0FFH        ; 最初は全ての割り込みをマスク
        MOV     IF0,   #00H         ; 無効な割り込み要求をクリアしておく
                    •
                    •
                    •
        MOV     TMC00, #00001000B   ; タイマ動作開始(TI000端子の有効エッジでクリア&スタート)
WAITCAP:
        NOP
        BF      TMIF010, $WAITCAP   ; 最初のキャプチャ完了を待つ

        MOV     IF0,   #00H         ; 割り込み要求フラグをクリア
        CLR1    TMMK000             ; INTTM000割り込みマスクを解除
        CLR1    TMMK010             ; INTTM010割り込みマスクを解除

        EI                          ; ベクタ割り込み許可

MAIN_LOOP:
        NOP
        BR      $MAIN_LOOP          ; MAIN_LOOPへ

INTERRUPT TM000:
        PUSH    AX                  ; AXレジスタのデータをスタックへ退避
                    •
                    •
                    •
        INC     HIGHCOUNT           ; 上位桁をカウント+1
                    •
                    •
                    •
INTERRUPT TM010:
        PUSH    AX                  ; AXレジスタのデータをスタックへ退避
        MOVW    AX,    CR010        ; キャプチャした値を読み出し
                    •
                    •
                    •
        MOV     A,     HIGHCOUNT    ; 上位桁を読み出し
                    •
                    •
                    •
end
```

CR010の動作モードをキャプチャ・レジスタに，CR000の動作モードをコンペア・レジスタに設定

CR000にFFFFHを設定

カウント・クロックとTI000端子の有効エッジを設定

タイマ動作開始

INTTM000とINTTM010の割り込み要求フラグをクリア

INTTM000割り込み処理を許可

INTTM010割り込み処理を許可

INTTM000割り込み発生により，割り込み処理開始

上位桁カウンタを「+1」

INTTM010割り込み発生により，割り込み処理開始

CR010レジスタのキャプチャ・データを読み出し

上位桁カウンタを読み出し

(2) C言語

```
void hdwinit(void){
        unsigned char ucCnt200us;  /* 200usウェイト用8ビット変数 */
                        •
                        •
                        •
        CRC00 = 0b00000100;        /* CR010をキャプチャ・レジスタとして使用 */
        CR000 = 0xFFFF;            /* CR000はオーバフロー検出用 */
        PRM00 = 0b00110001;        /*  TI000端子の有効エッジを両エッジに指定、カウント・クロック =fxp/4 */
        TOC00 = 0b00000000;        /* タイマ出力は行なわない */

        MK0   = 0xFF;              /* 最初は全ての割り込みをマスク */
        IF0   = 0x00;              /* 無効な割り込み要求をクリア */

        return;
}

void main(void){
        g_ucHighCount = 0;         /* 上位桁カウンタをクリア */
        g_ucTimes = 8;             /* 測定回数を初期化 */
        TMC00 = 0b00001000;        /* タイマ動作開始(TI000端子の有効エッジでクリア&スタート) */

        while(TMIF010==0){         /* 最初のキャプチャ完了を待つ */
                NOP();
        }

        IF0 = 0x00;                /* 割り込み要求フラグをクリア */
        TMMK000 = 0;               /* INTTM000割り込みマスクを解除 */
        TMMK010 = 0;               /* INTTM010割り込みマスクを解除 */

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

        while(1){                  /* 無限ループ */
                NOP();
                NOP();
        }
}

  interrupt void fn_inttm000(){
        if (!(TMIF010 && (CR010 == 0xFFFF))){    /* 同時に割り込みが発生かつCR010==0xFFFFの場合はカウントしない */
                g_ucHighCount++;   /* 上位桁をカウント+1 */
        }

        return;
}
  interrupt void fn_inttm010(){
        g_unLowLength[8 - g_ucTimes] = CR010;    /* キャプチャした値を読み出し */
        g_unHighLength[8 - g_ucTimes] = g_ucHighCount; /* 上位桁を読み出し */
                        •
                        •
                        •
        return;
}
```

CR000にFFFFHを設定

CR010の動作モードをキャプチャ・レジスタに，CR000の動作モードをコンペア・レジスタに設定

カウント・クロックとTI000端子の有効エッジを設定

タイマ動作開始

INTTM000割り込み処理を許可

INTTM000とINTTM010の割り込み要求フラグをクリア

INTTM010割り込み処理を許可

INTTM000割り込み発生により，割り込み処理開始

上位桁カウンタを「+1」

INTTM010割り込み発生により，割り込み処理開始

CR010レジスタのキャプチャ・データを読み出し

上位桁カウンタを読み出し

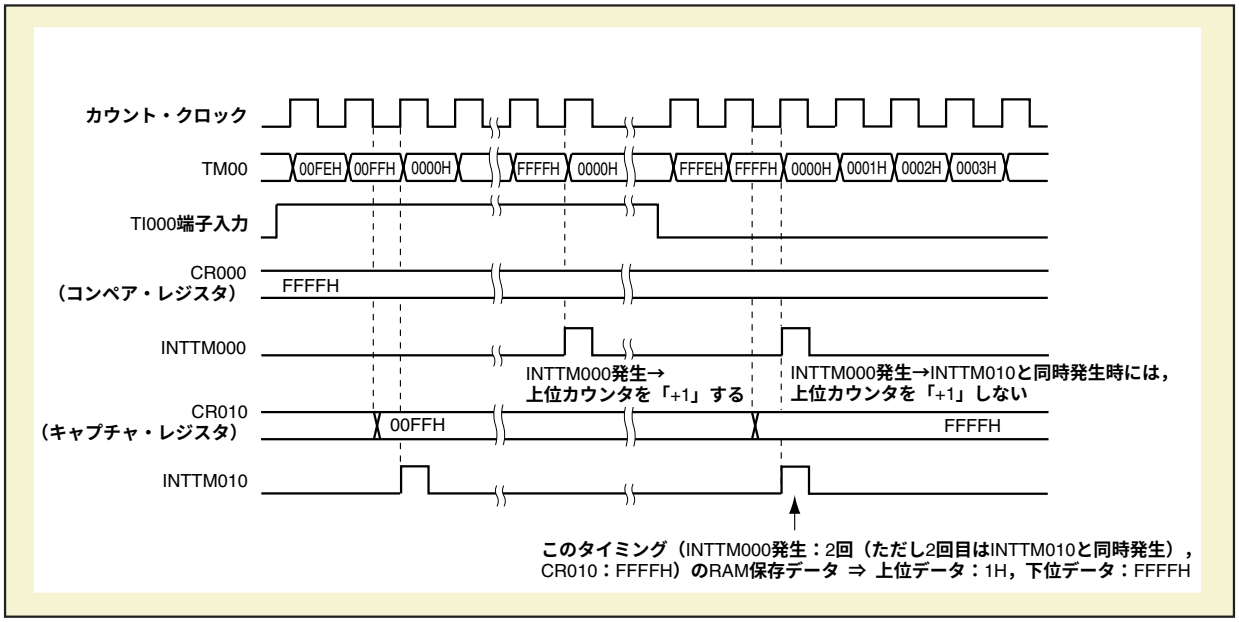
4．2　INTTM000とINTTM010の割り込み発生が競合した場合のタイミング（TM00カウンタの1周期よりも長いパルス幅を計測時）

このサンプル・プログラムでは，CR010レジスタの動作モードをキャプチャ・レジスタに，CR000レジスタの動作モードをコンペア・レジスタに設定し，CR000レジスタにFFFFHを設定しています。また，TI000端子の有効エッジでクリア&スタートに設定しています（前述の【例　5】を参照）。

この設定により，CR000レジスタとTM00カウンタの値が一致し，TM00カウンタがFFFFHから0000Hになったときに，INTTM000割り込みが発生します。これを，オーバフロー割り込みとして使用し，INTTM000割り込みが発生したら上位桁をカウント・アップすることで，TM00カウンタの1周期よりも長いパルス幅の計測が可能となります。

ここで，INTTM000割り込みがINTTM010割り込みと同じタイミングで発生した場合，INTTM000割り込みの処理が優先されます。このとき，CR010レジスタのキャプチャ・データはFFFFHとなり，上位桁のカウント・アップは不要です。したがって，INTTM000割り込みがINTTM010割り込みと同じタイミングで発生した場合は，上位桁をカウント・アップせずに，上位桁のデータとCR010レジスタのキャプチャ・データ（FFFFH）をRAM領域に保存します。

図4－11　INTTM000とINTTM010の割り込みが競合した場合のタイミング例（TI000端子の有効エッジ入力によるクリア＆スタート・モード，両エッジ指定時）



第5章　デバイスでの動作確認

この章では，ダウンロードしたサンプル・プログラムを使用し，ビルドからデバイスでの動作確認までの流れを説明します。

★ 5. 1　サンプル・プログラムのビルド

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。

また，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。

①［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。
②［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。
③「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。

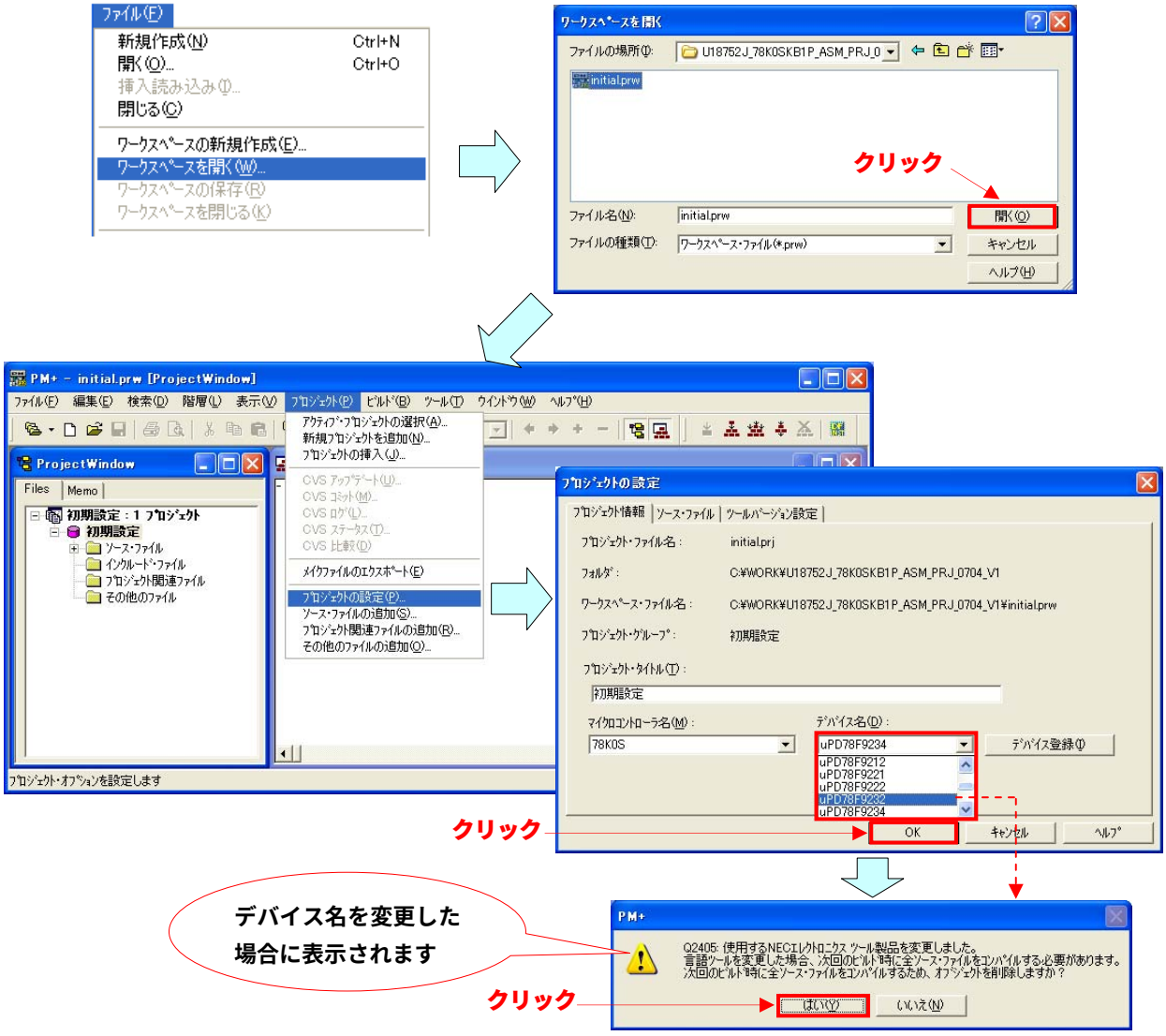
このサンプル・プログラムでは，  のアイコンを選択してダウンロードしたファイルを使用する場合，デフォルトで「標準ライブラリ固定領域を使用する」のチェックが外されています。

(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，「tm00cap.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。

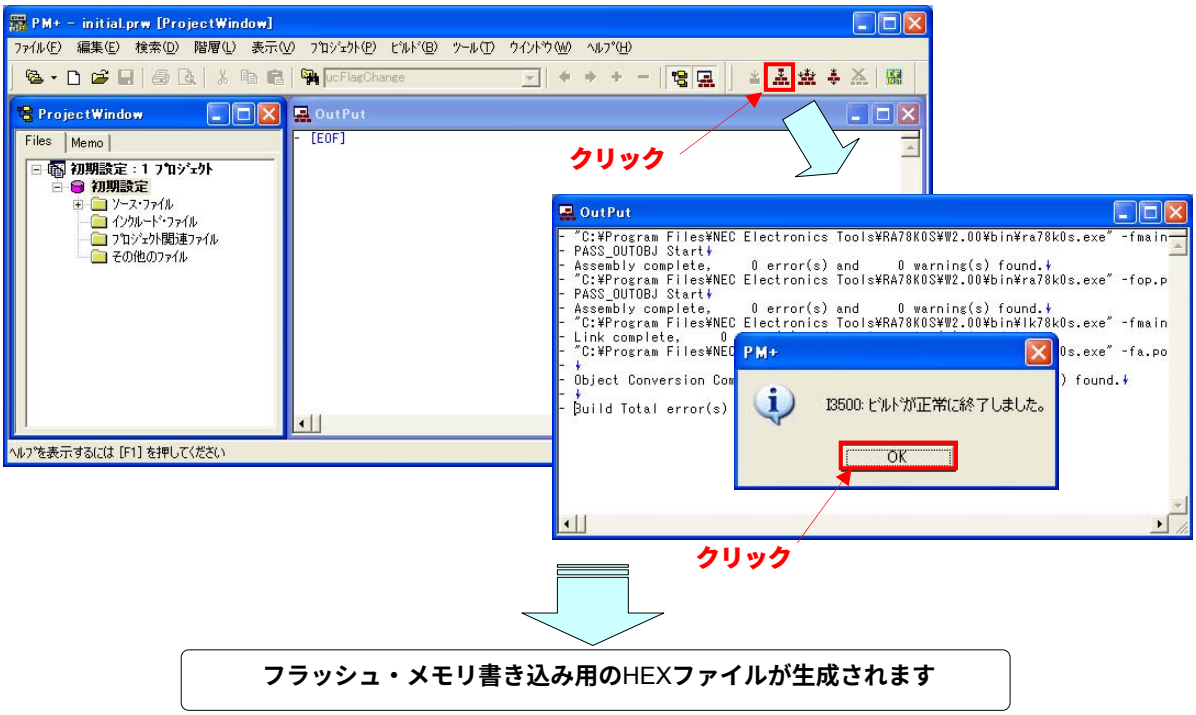
備考　下の図は，**「サンプル・プログラム（初期設定）**　LED**点灯のスイッチ制御」**の画面例です。



(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。

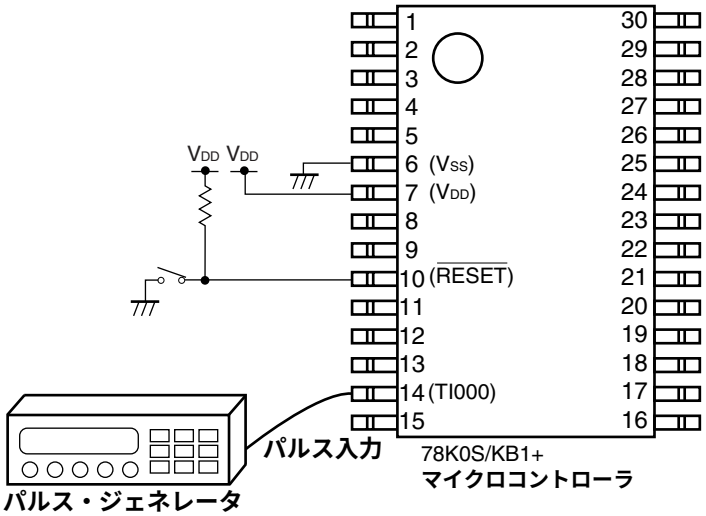


5. 2　デバイスでの動作

ここでは，デバイスでの動作確認の例を説明します。

ビルド実行により生成されたHEXファイルは，デバイスのフラッシュ・メモリに書き込みすることができます。デバイスへのフラッシュ・メモリ書き込み方法例については，各製品のフラッシュ書き込み簡単マニュアル　インフォメーション（78K0S/KU1+, 78K0S/KY1+, 78K0S/KA1+, 78K0S/KB1+）を参照してください。

デバイスと使用する周辺ハードウエア（パルス・ジェネレータ）の接続例を，次に示します。



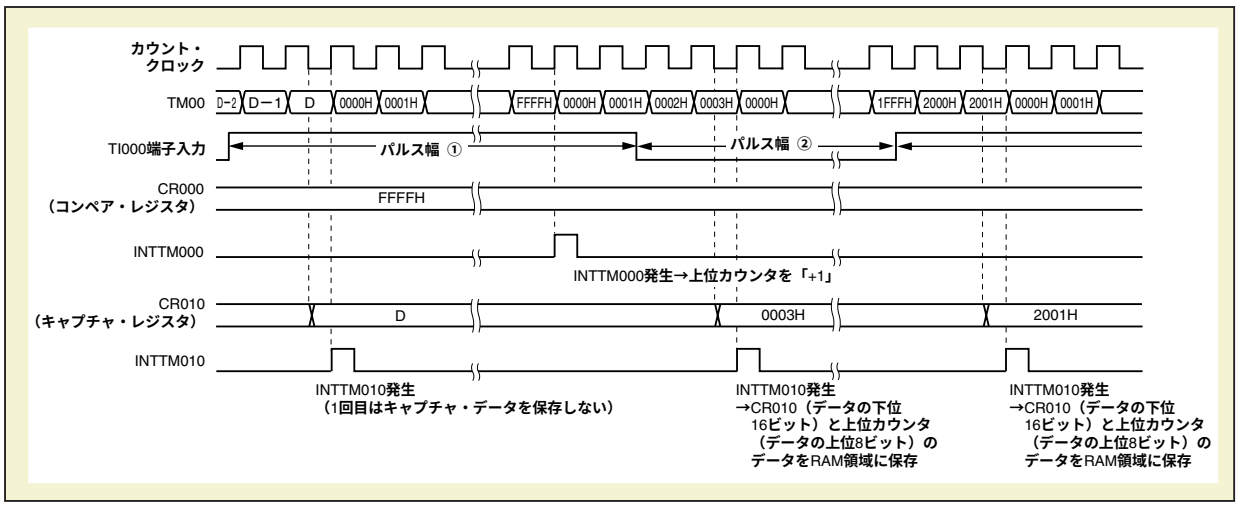
このサンプル・プログラムを書き込んだデバイスの動作例は，次のようになります。

(1) 1回目の有効エッジ検出時の動作

TI000端子にパルスが入力されると，1回目の有効エッジ検出がトリガとなり，パルス幅測定を開始します。

(2) 2回目以降の有効エッジ検出時の動作

TI000端子の有効エッジ検出は立ち下がり，立ち上がりの両エッジなので，ハイ･レベルのパルス幅とロウ・レベルのパルス幅を交互にキャプチャし，キャプチャ・データをRAMへ格納します。



＜このサンプル・プログラムのパルス幅の計算式＞

パルス幅 = （24ビット・データ[注]＋1）×0.0005 [ms/CLK]

注　上位8ビット：上位カウンタ値，下位16ビット：CR010レジスタ値

上図の動作例のパルス幅 ①とパルス幅 ②は次のようになります。

- パルス幅 ①（INTTM000発生：1回，CR010のキャプチャ・データ：0003H）
  =（010003H+1H）×0.0005 [ms/CLK] =（65539+1）×0.0005 [ms/CLK] = 32.77 [ms]）
- パルス幅 ②（INTTM000発生：0回，CR010のキャプチャ・データ：2001H）
  =（002001H+1H）×0.0005 [ms/CLK] =（8193+1）×0.0005 [ms/CLK] = 4.097 [ms]）

(3) 8回目のパルス幅測定以降の動作

合計8個のパルス幅値をRAMに格納したら，それ以降パルス幅測定は行われません。

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
78K/0S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	言語編	PDF
ユーザーズ・マニュアル	操作編	PDF
CC78K0S　Cコンパイラ	言語編	PDF
ユーザーズ・マニュアル	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）	78K0S/KU1+	PDF
インフォメーション	78K0S/KY1+	PDF
	78K0S/KA1+	PDF
	78K0S/KB1+	PDF
78K0S/Kx1+	サンプル・プログラム　スタートアップ・ガイド	PDF
アプリケーション・	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
ノート	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）　2.7 V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） PPG出力編	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（パルス幅測定）
;*******************************************************************************
;【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のパルス幅測定機能の使用例を示すもの
;である。カウントクロックをfxp/4(=2MHz)に設定して、TI000端子に入力された信号の
;立ち下がりと立ち上がりの両エッジを検出し、そのときのタイマのカウント値をCRC010
;レジスタにキャプチャする。キャプチャされた値からTI000端子に入力されたパルスの
;幅や間隔を知ることができる。CR000レジスタには0xFFFFを設定することで、オーバフロ
;ーの割り込みを発生し、タイマの周期以上の幅や間隔の計測を可能にしている。
;
;
;  ＜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
;  ・DEレジスタ、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: TI000端子の有効エッジでタイマ・カウントのクリア&スタート
;  ・CR000をオーバフローのコンペア・レジスタとして使用
;  ・CR010をキャプチャ・レジスタとして使用
;  ・TI000端子の両エッジ検出によりパルス幅をキャプチャ
;  ・カウント・クロック = fxp/4
;  ・タイマ出力は行なわない
;
;
;  <パルス幅の計算方法>
;
;  キャプチャするデータは、CR010レジスタの16ビットデータと上位桁8ビットデータを
;  合わせて24ビット長である。それぞれ下表のようにRAM領域に保存される。
;
;  パルス幅は、この24ビットデータから次のように計算される。
;
;    24ビットデータ×0.0005[ms/clk]＝パルス幅[ms]
;
;  # ここでの1[clk]はタイマ00のカウント・クロック(2MHz)
;
;  +-------------------------------------------------+
;  |    アドレス    | データ長 |        データ種別        |
;  |-------------------------------------------------|
;  | LOWLENGTH     | 16ビット | パルス幅下位データ(1回目) |
;  | LOWLENGTH + 2 | 16ビット | パルス幅下位データ(2回目) |
;  | LOWLENGTH + 4 | 16ビット | パルス幅下位データ(3回目) |
;  | LOWLENGTH + 6 | 16ビット | パルス幅下位データ(4回目) |
;  | LOWLENGTH + 8 | 16ビット | パルス幅下位データ(5回目) |
;  | LOWLENGTH +10 | 16ビット | パルス幅下位データ(6回目) |
;  | LOWLENGTH +12 | 16ビット | パルス幅下位データ(7回目) |
;  | LOWLENGTH +14 | 16ビット | パルス幅下位データ(8回目) |
;  |-------------------------------------------------|
;  | HIGHLENGTH    |  8ビット | パルス幅上位データ(1回目) |
;  | HIGHLENGTH +1 |  8ビット | パルス幅上位データ(2回目) |
;  | HIGHLENGTH +2 |  8ビット | パルス幅上位データ(3回目) |
;  | HIGHLENGTH +3 |  8ビット | パルス幅上位データ(4回目) |
;  | HIGHLENGTH +4 |  8ビット | パルス幅上位データ(5回目) |
;  | HIGHLENGTH +5 |  8ビット | パルス幅上位データ(6回目) |
;  | HIGHLENGTH +6 |  8ビット | パルス幅上位データ(7回目) |
;  | HIGHLENGTH +7 |  8ビット | パルス幅上位データ(8回目) |
;  +-------------------------------------------------+
;
;
```

```
;【ポート入出力の設定】
;
;  入力ポート：P30
;  出力ポート：P00-P03，P20-P23，P31-P33，P40-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
;
;*******************************************************************************



;===============================================================================
;
;     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)   INTLVI
      DW      RESET_START             ;(08)   INTP0
      DW      RESET_START             ;(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


;===============================================================================
;
;     RAMの定義
;
;===============================================================================
XRAM  DSEG    SADDRP
LOWLENGTH:    DS      16              ; パルス幅下位16ビット格納用テーブル
HIGHLENGTH:   DS      8               ; パルス幅上位8ビット格納用テーブル
HIGHCOUNT:    DS      1               ; パルス幅上位桁カウント用
TIMES:        DS      1               ; 測定回数カウント用
```

```
;=============================================================================
;
;     スタック領域の確保
;
;=============================================================================
XSTK  DSEG    AT      0FEE0H
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/clk]×10[clk]×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,     #00000000B      ; P30-P33の出力ラッチLow
      MOV     PM3,    #11110001B      ; P31-P33を出力ポートに、P30/TI000を入力ポートに設定

;-------------------------------------------------------------------------------
;     ポート4の設定
;-------------------------------------------------------------------------------
      MOV     P4,     #00000000B      ; P40-P47の出力ラッチLow
      MOV     PM4,    #00000000B      ; P40-P47を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;     ポート12の設定
;-------------------------------------------------------------------------------
      MOV     P12,    #00000000B      ; P120-P123の出力ラッチLow
      MOV     PM12,   #11110000B      ; 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,  #00000100B      ; CR010をキャプチャ・レジスタとして使用
      MOVW    AX,     #0FFFFH
      MOVW    CR000,  AX              ; CR000はオーバフロー検出用
      MOV     PRM00,  #00110001B      ; TI000端子の有効エッジを両エッジに指定、カウント・クロック = fxp/4
      MOV     TOC00,  #00000000B      ; タイマ出力は行なわない

;-------------------------------------------------------------------------------
;     割り込みの設定
;-------------------------------------------------------------------------------
      MOV     MK0,    #0FFH           ; 最初は全ての割り込みをマスク
      MOV     IF0,    #00H            ; 無効な割り込み要求をクリアしておく

;*******************************************************************************
;
;     メイン・ループ
;
;*******************************************************************************
      MOVW    HL,     #LOWLENGTH      ; 下位16ビット格納用テーブル・アドレスを初期化
      MOVW    DE,     #HIGHLENGTH     ; 上位8ビット格納用テーブル・アドレスを初期化
      MOV     HIGHCOUNT, #0           ; 上位カウンタをクリア
      MOV     TIMES,  #8              ; 結果の格納数を初期化

      MOV     TMC00,  #00001000B      ; タイマ動作開始(TI000端子の有効エッジでクリア&スタート)
WAITCAP:
      NOP
      BF      TMIF010, $WAITCAP       ; 最初のキャプチャ完了を待つ

      MOV     IF0,    #00H            ; 割り込み要求フラグをクリア
```

```
        CLR1    TMMK000             ; INTTM000割り込みマスクを解除
        CLR1    TMMK010             ; INTTM010割り込みマスクを解除

        EI                          ; ベクタ割り込み許可

MAIN_LOOP:
        NOP
        BR      $MAIN_LOOP          ; MAIN_LOOPへ

;*****************************************************************************
;
;       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

        BF      TMIF010, $INCHIGH   ; INTTM000割り込みと同時でなければ分岐
        MOVW    AX,     CR010       ; キャプチャした値を読み出し
        CMPW    AX,     #0FFFFH
        BZ      $ENDINTTM000        ; 同時に割り込みが発生かつCR010==0xFFFFの場合はカウントしない

INCHIGH:
        INC     HIGHCOUNT           ; 上位桁をカウント+1

ENDINTTM000:
        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_TM010:
        PUSH    AX                  ; AXレジスタのデータをスタックへ退避

        MOVW    AX,     CR010       ; キャプチャした値を読み出し
        MOV     [HL+1], A           ; 下位16ビット格納用テーブルに保存
        XCH     A,      X
        MOV     [HL],   A           ; 下位16ビット格納用テーブルに保存
        MOV     A,      HIGHCOUNT   ; 上位桁を読み出し
        MOV     [DE],   A           ; 上位8ビット格納用テーブルに保存
```

```
        INC     L                       ; 下位16ビット格納用テーブル・アドレスを+2
        INC     L
        INC     E                       ; 上位8ビット格納用テーブル・アドレスを+1
        MOV     HIGHCOUNT, #0           ; 上位桁をクリア

        DBNZ    TIMES,  $ENDINTTM000    ; 測定回数＜8なら分岐
        SET1    TMMK000                 ; INTTM000割り込みをマスク
        SET1    TMMK010                 ; INTTM010割り込みをマスク

ENDINTTM010:
        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（パルス幅測定）
*****************************************************************************
【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のパルス幅測定機能の使用例を示すもの
である。カウントクロックをfxp/4(=2MHz)に設定して、TI000端子に入力された信号の
立ち下がりと立ち上がりの両エッジを検出し、そのときのタイマのカウント値をCRC010
レジスタにキャプチャする。キャプチャされた値からTI000端子に入力されたパルスの
幅や間隔を知ることができる。CR000レジスタには0xFFFFを設定することで、オーバフロ
ーの割り込みを発生し、タイマの周期以上の幅や間隔の計測を可能にしている。


  ＜主な設定内容＞

  ・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

＜16ビット・タイマ00の設定＞

・動作モード: TI000端子の有効エッジでタイマ・カウントのクリア&スタート
・CR000をオーバフローのコンペア・レジスタとして使用
・CR010をキャプチャ・レジスタとして使用
・TI000端子の両エッジ検出によりパルス幅をキャプチャ
・カウント・クロック = fxp/4
・タイマ出力は行なわない

＜パルス幅の計算方法＞

キャプチャするデータは、CR010レジスタの16ビットデータと上位桁8ビットデータを
合わせて24ビット長である。それぞれ下表のようにRAM領域に保存される。

パルス幅は、この24ビットデータから次のように計算される。

  24ビットデータ×0.0005[ms/clk]＝パルス幅[ms]

ここでの1[clk]はタイマ00のカウント・クロック(2MHz)

変数名	データ長	データ種別
g_unLowLength[0]	16ビット	パルス幅下位データ(1回目)
g_unLowLength[1]	16ビット	パルス幅下位データ(2回目)
g_unLowLength[2]	16ビット	パルス幅下位データ(3回目)
g_unLowLength[3]	16ビット	パルス幅下位データ(4回目)
g_unLowLength[4]	16ビット	パルス幅下位データ(5回目)
g_unLowLength[5]	16ビット	パルス幅下位データ(6回目)
g_unLowLength[6]	16ビット	パルス幅下位データ(7回目)
g_unLowLength[7]	16ビット	パルス幅下位データ(8回目)
g_unHighLength[0]	8ビット	パルス幅上位データ(1回目)
g_unHighLength[1]	8ビット	パルス幅上位データ(2回目)

```
  | g_unHighLength[2] |  8ビット | パルス幅上位データ(3回目) |
  | g_unHighLength[3] |  8ビット | パルス幅上位データ(4回目) |
  | g_unHighLength[4] |  8ビット | パルス幅上位データ(5回目) |
  | g_unHighLength[5] |  8ビット | パルス幅上位データ(6回目) |
  | g_unHighLength[6] |  8ビット | パルス幅上位データ(7回目) |
  | g_unHighLength[7] |  8ビット | パルス幅上位データ(8回目) |
  +-------------------------------------------------------+


【ポート入出力の設定】

  入力ポート：P30
  出力ポート：P00-P03, P20-P23, P31-P33, P40-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       NOP                             /* NOP命令を記述可能にする */
#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 */

/*===================================================================================

      グローバル変数の定義

===================================================================================*/
sreg static unsigned int g_unLowLength[8];    /* パルス幅下位16ビット格納用16ビット変数 */
sreg static unsigned char g_unHighLength[8];  /* パルス幅上位桁格納用8ビット変数 */
sreg static unsigned char g_ucHighCount;      /* パルス幅上位桁カウント用8ビット変数 */
sreg static unsigned char g_ucTimes;          /* 測定回数カウント用8ビット変数 */
```

```
/*****************************************************************************

      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   = 0b11110001;           /* P31-P33を出力ポートに、P30/TI000を入力ポートに設定 */

/*-------------------------------------------------------------------------
      ポート4の設定
-------------------------------------------------------------------------*/
      P4    = 0b00000000;           /* P40-P47の出力ラッチLow */
      PM4   = 0b00000000;           /* P40-P47を出力ポートに設定 */

/*-------------------------------------------------------------------------
      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 = 0b00000100;           /* CR010をキャプチャ・レジスタとして使用 */
      CR000 = 0xFFFF;               /* CR000はオーバフロー検出用 */
      PRM00 = 0b00110001;           /* TI000端子の有効エッジを両エッジに指定、カウント・クロック = fxp/4 */
      TOC00 = 0b00000000;           /* タイマ出力は行なわない */
```

```
/*-------------------------------------------------------------------------
      割り込みの設定
-------------------------------------------------------------------------*/
      MK0   = 0xFF;                 /* 最初は全ての割り込みをマスク */
      IF0   = 0x00;                 /* 無効な割り込み要求をクリア */

      return;
}

/*************************************************************************

      メイン・ループ

*************************************************************************/
void main(void){
      g_ucHighCount = 0;            /* 上位桁カウンタをクリア */
      g_ucTimes = 8;                /* 測定回数を初期化 */
      TMC00 = 0b00001000;           /* タイマ動作開始(TI000端子の有効エッジでクリア&スタート) */

      while(TMIF010==0){            /* 最初のキャプチャ完了を待つ */
            NOP();
      }

      IF0 = 0x00;                   /* 割り込み要求フラグをクリア */
      TMMK000 = 0;                  /* INTTM000割り込みマスクを解除 */
      TMMK010 = 0;                  /* INTTM010割り込みマスクを解除 */

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

      while(1){                     /* 無限ループ */
            NOP();
            NOP();
      }
}

/*************************************************************************

      割り込みINTTM000

*************************************************************************/
```

```
__interrupt void fn_inttm000(){

        if (!(TMIF010 && (CR010 == 0xFFFF))){    /* 同時に割り込みが発生かつCR010==0xFFFFの場合はカウントしない */
                g_ucHighCount++;         /* 上位桁をカウント+1 */
        }

        return;
}

/************************************************************************************

        割り込みINTTM010

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

        g_unLowLength[8 - g_ucTimes] = CR010;    /* キャプチャした値を読み出し */
        g_unHighLength[8 - g_ucTimes] = g_ucHighCount;  /* 上位桁を読み出し */
        g_ucHighCount = 0;               /* 上位桁をクリア */
        g_ucTimes--;                     /* 測定回数を-1 */

        if(g_ucTimes == 0){      /* 8回目のキャプチャが終了した場合 */
                TMMK000 = 1;             /* INTTM000割り込みをマスク */
                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	pp.33-36	5. 1　**サンプル・プログラムのビルド**を変更
		p.38	**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**を変更

【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。