

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが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

78K0/Kx2

サンプル・プログラム

インターバル・タイマ編

この資料は、サンプル・プログラムの「インターバル・タイマ」の動作概要と、マイコンの基本的なインターバル・タイマの設定を説明したものです。サンプル・プログラムでは、インターバル・タイマの設定を行ったあとに、1つのLED点灯の制御を行います。

対象デバイス

78K0/KB2マイクロコントローラ
 78K0/KC2マイクロコントローラ
 78K0/KD2マイクロコントローラ
 78K0/KE2マイクロコントローラ
 78K0/KF2マイクロコントローラ

目次

- 第1章 概要 ... 3
- 第2章 回路イメージ ... 5
 - 2.1 回路イメージ ... 5
 - 2.2 マイコン以外の使用デバイス ... 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 使用するマイコン内蔵周辺機能 ... 8
 - 3.3 インターバル・タイマ設定と動作概要 ... 9
 - 3.4 フロー・チャート ... 10
- 第4章 設定方法について ... 12
 - 4.1 前処理指令 ... 12
 - 4.2 インターバル・タイマの設定 ... 13
 - 4.3 割り込みの設定 ... 26
 - 4.4 ポートの設定 ... 29
 - 4.5 メイン処理 ... 30
 - 4.6 割り込み処理 ... 31
- 第5章 システム・シミュレータ SM+での動作確認 ... 33
 - 5.1 サンプル・プログラムのビルド ... 33
 - 5.2 SM+での動作 ... 36
 - 5.3 オンチップ・デバッグ時の注意 ... 40
 - 5.4 開発環境のダウンロード、インストール ... 43
- 第6章 関連資料 ... 44
- 付録A 改版履歴 ... 45

資料番号 U19031JJ2V0AN00 (第2版)

発行年月 May 2009 NS

- 本資料に記載されている内容は2009年5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連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。また、当社製品は耐放射線設計については行っていません。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないように、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット

特別水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器

特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等

当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

M8E0710J

第1章 概 要

このサンプル・プログラムでは、インターバル・タイマの設定を行います。ここでは、16ビット・タイマ/イベント・カウンタ00 (TM00)、8ビット・タイマ/イベント・カウンタ50 (TM50)、8ビット・タイマH0 (TMH0) の3つのタイマの設定例を紹介します。

本プログラムでは、1 [ms] ごとにインターバル割り込みを発生させます。そして、この割り込み内で、P10を反転させる処理とP11に接続されたLEDをインターバル100回経過ごとに点灯/消灯 (P11の反転) をさせる処理を行います。

なお、クロック関連の設定や周辺ハードウェアの初期設定などのサンプル・プログラムの初期設定編と同内容の部分は、別ファイルとしてサブルーチン・コールにて使用しています。

(1) インターバル・タイマの設定内容

- ・タイマ・クロック設定
- ・コンペア・レジスタ設定
- ・タイマ動作許可設定
- ・割り込み設定

(2) メイン処理動作の内容

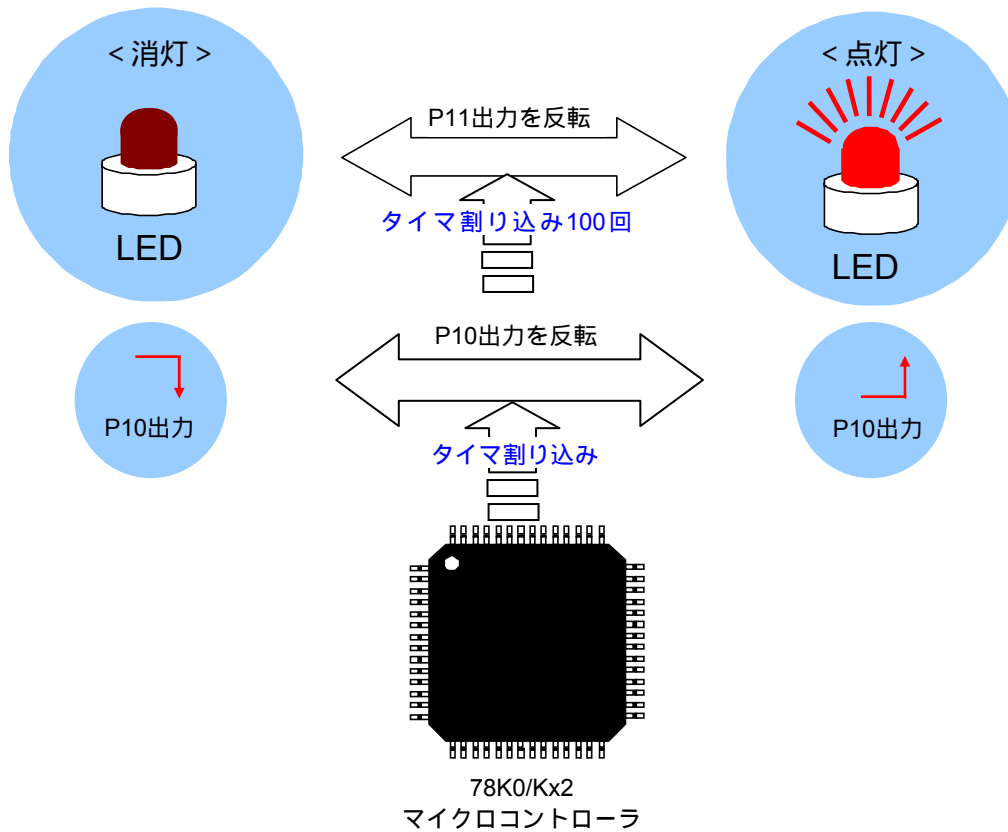
- ・インターバル・タイマを起動後、インターバル・カウンタ (割り込み回数をカウントするカウンタ) を初期化して、NOP命令のみの割り込み待ちループに入ります。

(3) 割り込み処理の内容

- ・P10の値を反転し、割り込み回数をカウントします。割り込み100回ごとにP11を反転させてLEDを点灯/消灯させます。

割り込みベクタは0020H (INTTM000 : TM00版)、001CH (INTTMH0 : TMH0版) 001EH (INTTM50 : TM50版) に設定します。

(ベクタ・テーブル0020H, 001CH, 001EH番地の割り込みハンドラ名称を“ IINIT ”から“ IINTTM000 ”, “ IINTTMH0 ”, “ INTTM50 ” に変更します。)



(4) インターバル・タイマの種類

インターバル・タイマは、デバイスによって数種類用意されています。

インターバル・タイマの種類

デバイス		78K0/KB2	78K0/KC2	78K0/KD2	78K0/KE2		78K0/KF2
		(全製品)	(全製品)	(全製品)	μPD78F0531	μPD78F0534	(全製品)
タイマ					μPD78F0532	μPD78F0535	
					μPD78F0533	μPD78F0536	
						μPD78F0537	
						μPD78F0537D	
8ビット・タイマ	H0						
	H1						
8ビット・タイマ/ イベント・カウンタ	50						
	51						
16ビット・タイマ/ イベント・カウンタ	00						
	01	-	-	-	-		

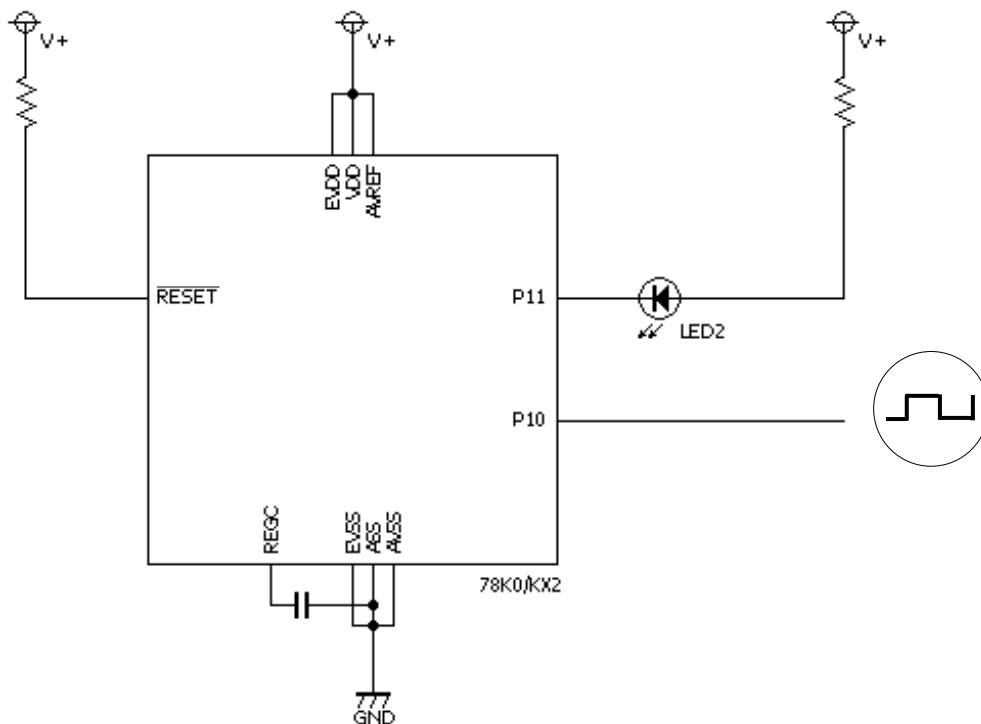
注意 デバイス使用上の注意事項については、[78K0/Kx2 ユーザーズ・マニュアル](#)を参照してください。

第2章 回路イメージ

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

2.1 回路イメージ

回路イメージを次に示します。



- 注意1. AVREF端子はVDDに直接接続してください。
2. AVSS端子はGNDに直接接続してください。
 3. REGC端子はコンデンサ (0.47 ~ 1 μ F) を介し、Vssに接続してください。
 4. EVDD端子はVDDに直接接続してください (78K0/KE2, 78K0/KF2のみ)。
 5. EVSS端子はGNDに直接接続してください (78K0/KE2, 78K0/KF2のみ)。
 6. 使用電圧と動作周波数などの詳細については、ユーザズ・マニュアルを参照してください。
 7. ポートの出力電流値には上限値があるので、そのスペックの範囲内で駆動するLEDを使用してください。

2.2 マイコン以外の使用デバイス

マイコン以外に使用するデバイスを次に示します。

(1) LED

1 [ms] ごとのインターバル・タイマの割り込みでのP10出力変化は確認しにくいので、割り込み発生確認の出力として、インターバル・タイマの割り込み100回ごとにP11に接続されたLED表示を変化(点灯/消灯)させます。




第3章 ソフトウェアについて

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

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

【C言語版】

ファイル名 ^注	説明	同封圧縮 (*.zip) ファイル		
				
Kx2_Intvl.c	インターバル・タイマの設定、メイン処理、割り込み処理のソース・ファイル			
Kx2_func.c	初期化処理を外部関数化したソース・ファイル			
Kx2_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			
Kx2_Intvl.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
Kx2_Intvl.prj	統合開発環境 PM+用プロジェクト・ファイル			
Kx2_Intvl.pri Kx2_Intvl.prs Kx2_Intvl.prm	システム・シミュレータ SM+ for 78K0/Kx2用プロジェクト・ファイル			
Kx2_Intvl0.pnl	システム・シミュレータ SM+ for 78K0/Kx2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 各ファイル名の"x"部分は、それぞれのデバイスの名前になります。

ex) 78K0/KB2の場合 "KB2_Init.c"

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 78K0/Kx2で使用するファイルを同封



: システム・シミュレータ SM+ for 78K0/Kx2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

【アセンブリ言語版】

ファイル名 ^注	説明	同封圧縮 (*.zip) ファイル		
				
Kx2_intvl.asm	インターバル・タイマの設定, メイン処理, 割り込み処理のソース・ファイル			
Kx2_subr.asm	初期化処理をサブルーチン化したソース・ファイル			
Kx2_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			
Kx2_intvl.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
Kx2_intvl.prj	統合開発環境 PM+用プロジェクト・ファイル			
Kx2_intvl.pri Kx2_intvl.prs Kx2_intvl.prm	システム・シミュレータ SM+ for 78K0/Kx2用プロジェクト・ファイル			
Kx2_intvl0.pnl	システム・シミュレータ SM+ for 78K0/Kx2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 各ファイル名の"x"部分は, それぞれのデバイスの名前になります。

ex) 78K0/KB2の場合 "KB2_init.asm"

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 78K0/Kx2で使用するファイルを同封



: システム・シミュレータ SM+ for 78K0/Kx2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 使用するマイコン内蔵周辺機能

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。

- ・ 16ビット・タイマ/イベント・カウンタ00 (TM00)
- ・ 8ビット・タイマ/イベント・カウンタ50 (TM50)
- ・ 8ビット・タイマH0 (TMH0)

また、このサンプル・プログラムでは、上記3つのタイマについての設定例を紹介しています。

- ・ 出力ポート : P10
- ・ 出力ポート (LED点灯用ポート) : P11

TM01, TM51, TMH1のサンプル・プログラムは、上記TM00, TM50, TMH0とほぼ同内容であるため、本サンプル・プログラムにおいては省略します。

なお、参考として本サンプル・プログラムにおけるTM00, TM50, TMH0との変更点をそれぞれ次に記載します。

【TM00 TM01への変更点】

- C言語 : 割り込み処理関数宣言をINTTM000からINTTM010に変更
- アセンブリ言語 : 割り込みベクタ・テーブルを0020H (INTTM000) から0022H (INTTM010) に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” でのTM00の各レジスタ名称の00を01に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” での割り込み要求フラグと割り込みマスク・フラグの名称の00を01に変更

【TM50 TM51への変更点】

- C言語 : 割り込み処理関数宣言をINTTM50からINTTM51に変更
- アセンブリ言語 : 割り込みベクタ・テーブルを001EH (INTTM50) から002AH (INTTM51) に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” でのTM50の各レジスタ名称の50を51に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” での割り込み要求フラグと割り込みマスク・フラグの名称の50を51に変更

【TMH0 TMH1への変更点】

- C言語 : 割り込み処理関数宣言をINTTMH0からINTTMH1に変更
- アセンブリ言語 : 割り込みベクタ・テーブルを001CH (INTTMH0) から001AH (INTTMH1) に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” でのTMH0の各レジスタ名称のH0をH1に変更
- C, アセンブリ言語 : “インターバル・タイマの初期化” での割り込み要求フラグと割り込みマスク・フラグの名称のH0をH1に変更

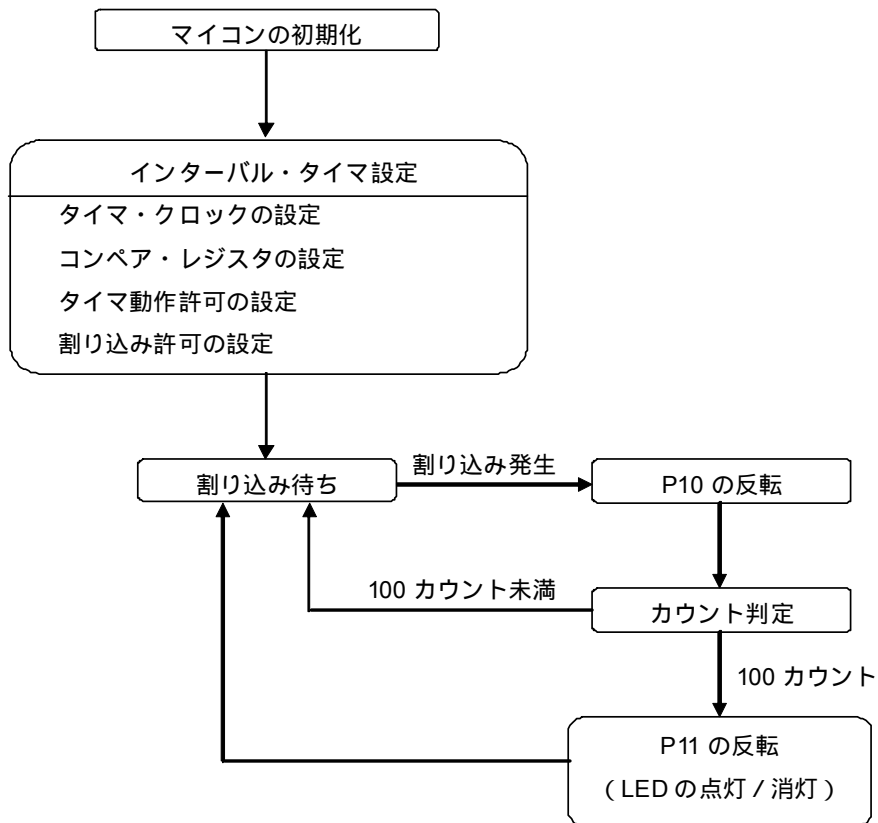
注意 本サンプル・プログラムではインターバル割り込み時間が1 [ms] と設定してありますが、上記の変更を行うとカウント・クロックを選択するレジスタの内容も変わるために割り込み時間にも変化が生じてしまうので、カウント・クロックを選択するレジスタの設定とコンペア・レジスタの設定値は使用用途に合わせて設定してください。

3.3 インターバル・タイマ設定と動作概要

このサンプル・プログラムでは、インターバル・タイマの設定を行います。

設定完了後は、設定時間に応じて割り込みが発生し、P10の反転とP11に接続されたLEDの点灯を制御します。

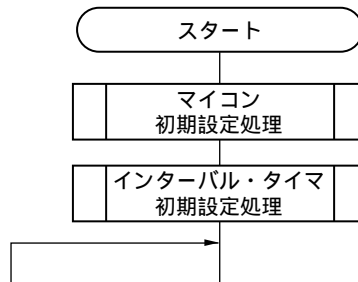
動作概要については、次の状態遷移図（ステート・チャート）に示します。



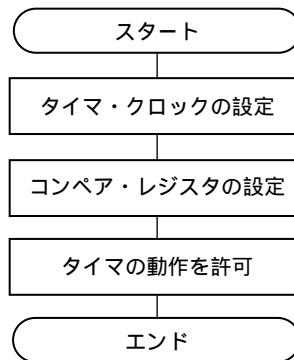
3.4 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。

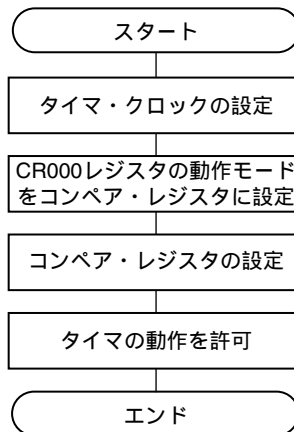
ゼネラル・フロー (C言語版 : Kx2_Intvl.c アセンブリ言語版 : Kx2_Intvl.asm)



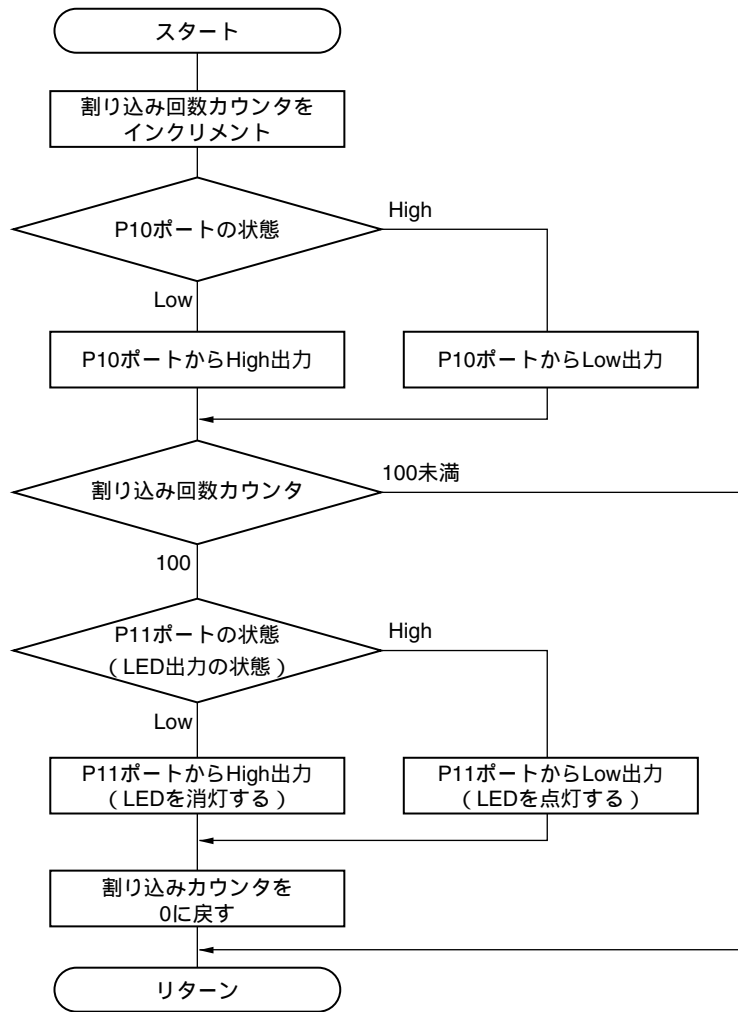
インターバル・タイマ初期設定処理 (8ビット・タイマの場合) (C言語版 : Kx2_Intvl.c アセンブリ言語版 : Kx2_Intvl.asm)



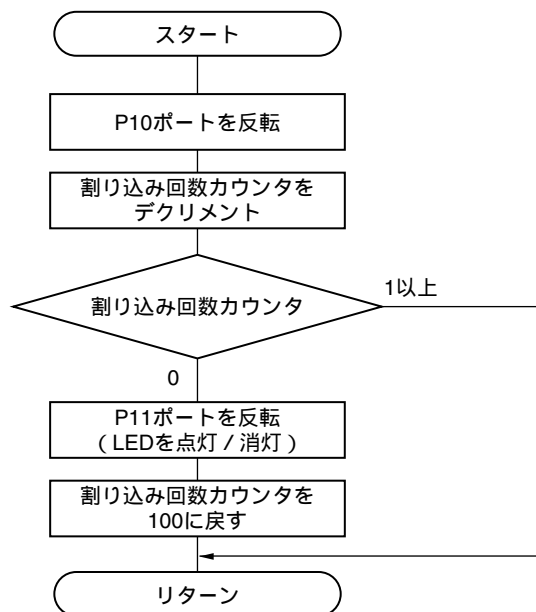
インターバル・タイマ初期設定処理 (16ビット・タイマの場合) (C言語版 : Kx2_Intvl.c アセンブリ言語版 : Kx2_Intvl.asm)



【C言語】割り込み処理 (C言語版 : Kx2_Intvl.c)



【アセンブリ言語】割り込み処理 (アセンブリ言語版 : Kx2_Intvl.asm)



第4章 設定方法について

この章では、インターバル・タイマ (TM00, TM50, TMH0) の設定, およびメイン処理について説明します。
レジスタ設定方法の詳細については、[78K0/Kx2 ユーザーズ・マニュアル](#)を参照してください。
アセンブラ命令については、[78K0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 前処理指令

C言語において、SFR領域に関する操作、CPU制御命令、割り込み関数などを使用するためには、#pragma指令にてソース・プログラムの冒頭に前処理指令を記述する必要があります。本サンプル・プログラムで使用する前処理指令は以下のとおりです。

【C言語】 (Kx2_Intvl.c)

```
/*=====
前処理指令 (#pragma指令)
=====*/
#pragma sfr <-----
#pragma di <-----
#pragma ei <-----
#pragma nop <-----
#pragma interrupt INTTM000 fn_intTimerInterval <-----
```

特殊機能レジスタ(SFR)名称を記述可能にします。

DI命令を記述可能にします。

EI命令を記述可能にします。

NOP命令を記述可能にします。

割り込み関数宣言をします。

4.2 インターバル・タイマの設定

【16ビット・タイマ/イベント・カウンタ00】

インターバル・タイマは、次の項目を設定します。

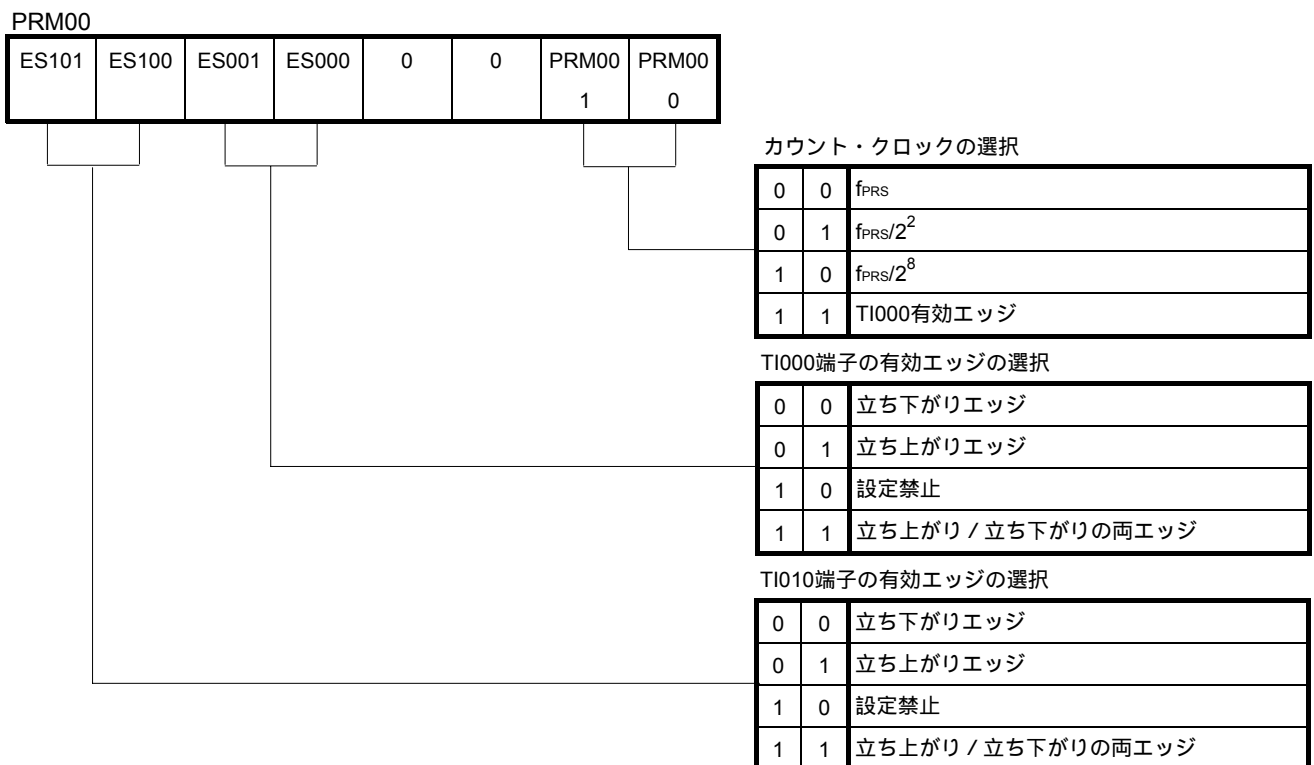
- (1) タイマ・クロック選択
- (2) タイマ・モードの設定
- (3) タイマ・コンペア・レジスタの設定
- (4) タイマ出力の設定
- (5) タイマの動作許可，タイマ出力反転条件の設定

本サンプル・プログラムでは後述の【例】の内容で設定しています。

(1) タイマ・クロック設定 (プリスケアラ・モード・レジスタ00)

16ビット・タイマ/イベント・カウンタ00のカウント・クロックおよび端子入力の有効エッジを設定します。

図4-1 プリスケアラ・モード・レジスタ00の設定

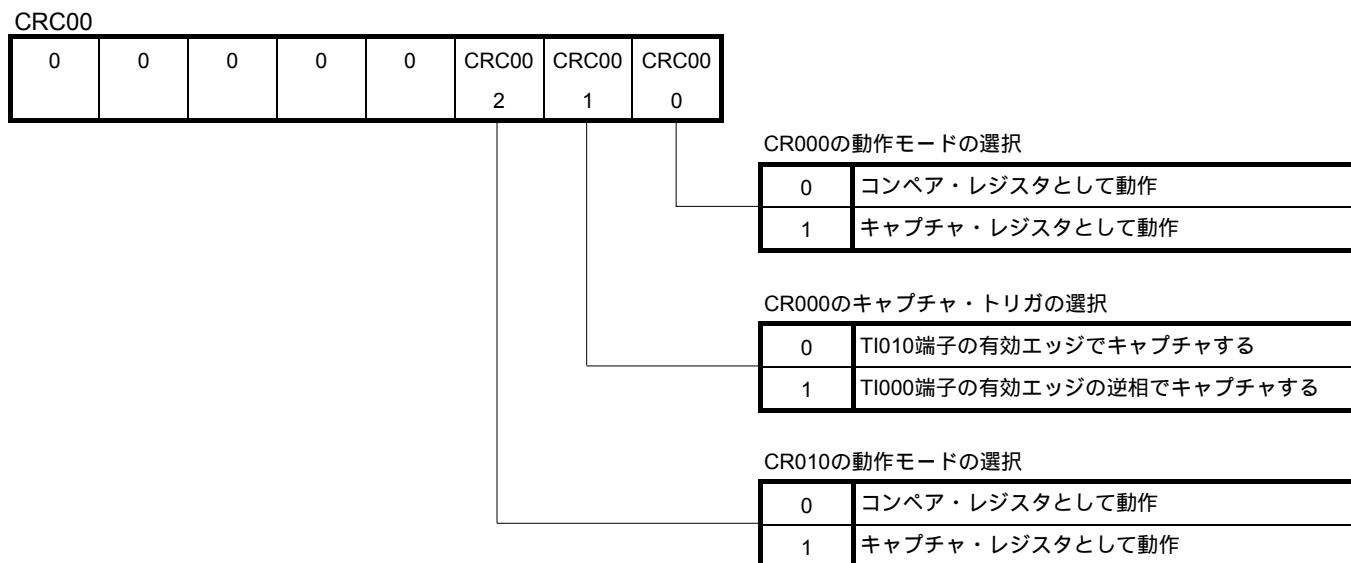


備考 f_{PRS} : 周辺ハードウェア・クロック周波数

(2) タイマ・モード設定

16ビット・タイマ/イベント・カウンタ00の動作モードの設定をします。

図4 - 2 キャプチャ/コンペア・コントロール・レジスタ00の設定



(3) タイマ・コンペア・レジスタの設定

16ビット・タイマ/イベント・カウンタ00のカウンタ値と比較し、2つの値が一致したときに割り込みを発生します。

図4 - 3 16ビット・タイマ・キャプチャ/コンペア・レジスタ000

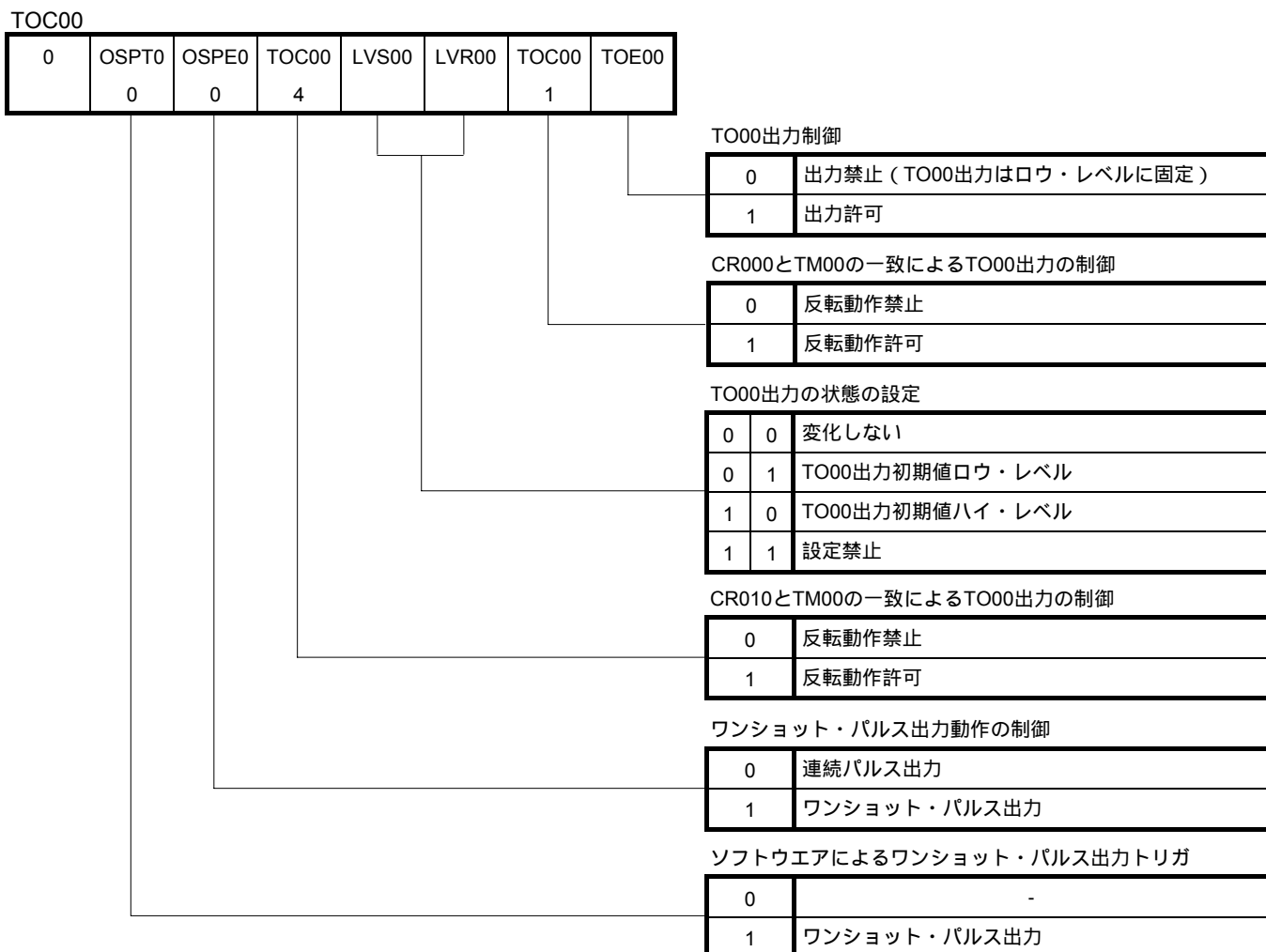


注意 CR000はタイマ停止中 (TMC003, TMC002 = 00) に書き換えを行ってください。

(4) タイマ出力の設定

16ビット・タイマ/イベント・カウンタ00のカウンタのタイマ出力の設定をします。

図4-4 16ビット・タイマ出力コントロール・レジスタ00



注意 TOC00を設定するときは、必ず次の手順で設定してください。

TOC004, TOC001のセット (1)

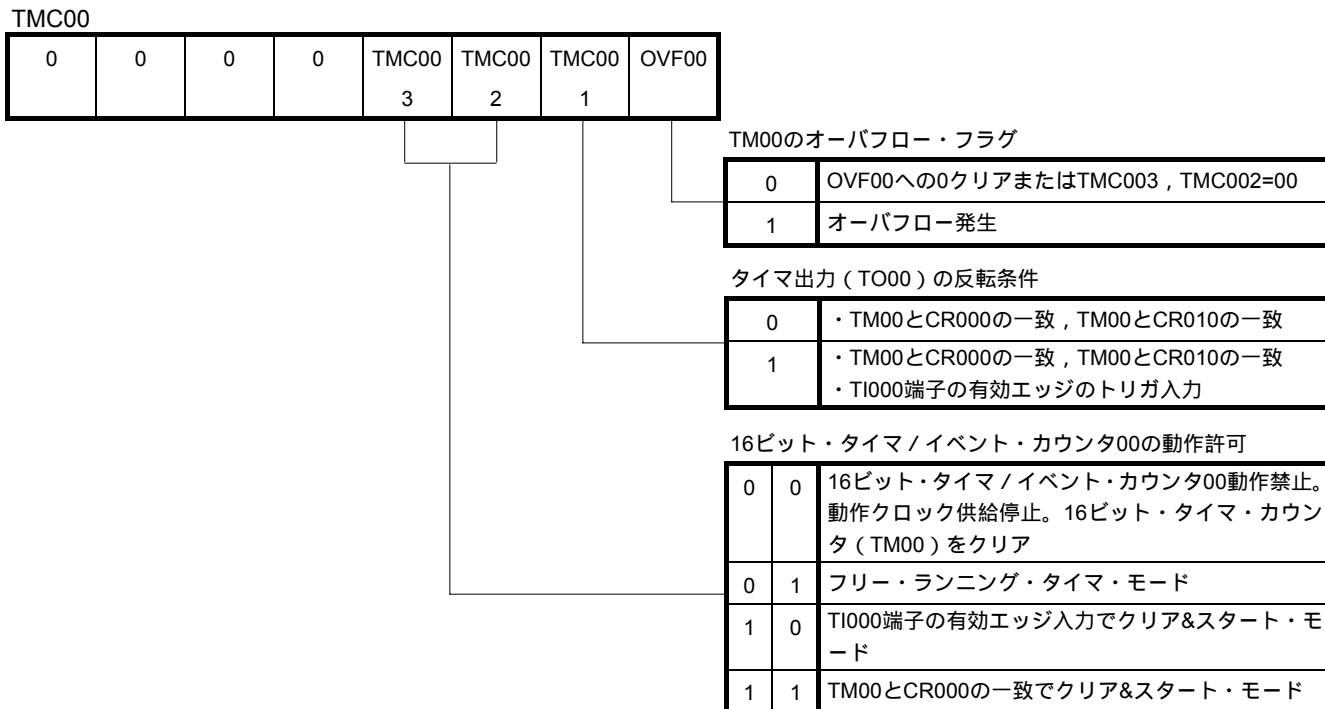
TOE00だけを単独でセット (1)

LVS00またはLVR00のどちらか片方だけをセット (1)

(5) タイマの動作許可, タイマ出力の反転条件の設定

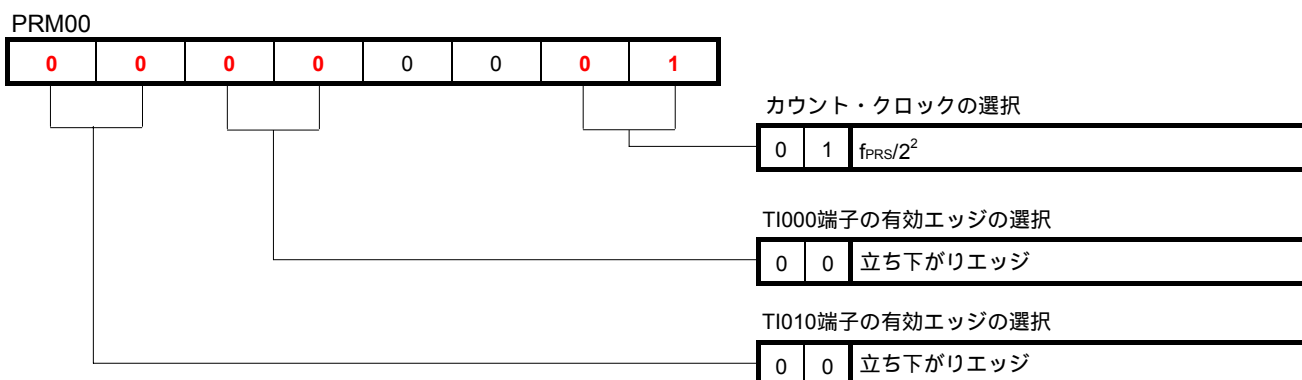
16ビット・カウンタ/イベント・カウンタ00のカウンタ・クロックの動作許可, タイマ出力反転条件の設定をします。

図4-5 16ビット・タイマ・モード・コントロール・レジスタ00



注意 16ビット・タイマ/イベント・カウンタ00は, TMC003, TMC002に00 (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, TMC003, TMC002に00を設定してください。

- 【例】
- ・ タイマ・クロックを0.5 [μ s] に設定 (8 MHz動作時)
 - ・ インターバル時間を1 [ms] に設定
- (サンプル・プログラムの設定と同内容)



CRC00



- 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
0 TI010端子の有効エッジでキャプチャする
- CR010の動作モードの選択
0 コンペア・レジスタとして動作

CR000



インターバル時間 = (1999 + 1) × カウント・クロック周期
 ここでは, 1999 (1 [ms]) に設定

TOC00



- TO00出力制御
0 出力禁止 (TO00出力は, ロウ・レベルに固定)
- CR000とTM00の一致によるTO00の出力の制御
0 反転動作禁止
- TO00の出力状態の設定
0 0 変化しない
- CR010とTM00の一致によるTO00出力の制御
0 反転動作禁止
- ワンショット・パルス出力動作の制御
0 連続パルス出力
- ソフトウェアによるワンショット・パルス出力トリガ
0 -

TMC00



- TM00のオーバーフロー・フラグ
0 OVF00への0クリアまたはTMC003, TMC002 = 00
- タイマ出力 (TO00) の反転条件
0 TM00とCR000の一致, TM00とCR010の一致
- 16ビット・タイマ/イベント・カウンタ00の動作許可
1 1 TM00とCR000の一致でクリア&スタート・モード

サンプル・プログラムでは以下ようになります。

【C言語】 (Kx2_Intvl.c)

```
PRM00 = 0b00000001; /* タイマ・クロックをfPRS/2^2に設定 */
CRC00 = 0b00000000; /* CR000をコンペア・レジスタとして動作に設定 */
CR000 = 2000-1; /* コンペア・レジスタの設定 */
TOC00 = 0b00000000; /* タイマ出力はしない */
TMC00 = 0b00001100; /* タイマの動作許可 */
      .
      .
      .
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```
MOV     PRM00, #00000001B ;タイマ・クロックをfPRS/2^2に設定
MOV     CRC00, #00000000B ;CR000をコンペア・レジスタとして動作に設定
MOVW    CR000, #2000-1    ;コンペア・レジスタの設定
MOV     TOC00, #00000000B ;タイマ出力はしない
MOV     TMC00, #00001100B ;タイマの動作許可
      .
      .
      .
```

【8ビット・タイマ/イベント・カウンタ50】

インターバル・タイマは、次の項目を設定します。

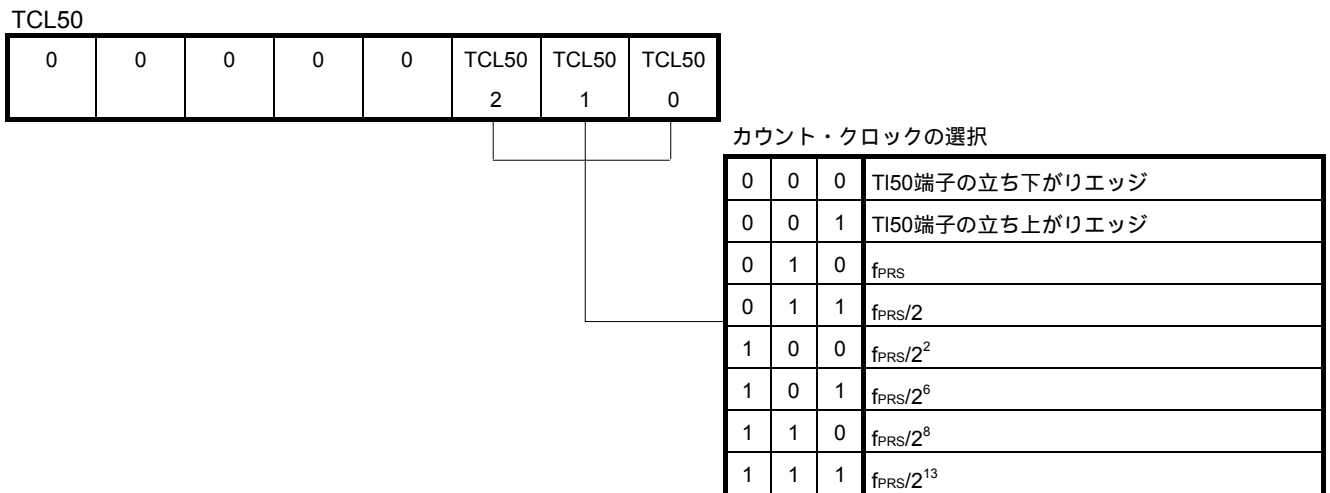
- (1) タイマ・クロック選択
- (2) タイマ・モードの設定
- (3) タイマ・コンペア・レジスタ (CR50) の設定

本サンプル・プログラムでは、後述の【例】の内容で設定しています。

(1) タイマ・クロック設定

8ビット・タイマ/イベント・カウンタ50のカウンタ・クロックおよび端子入力の有効エッジを設定します。

図4 - 6 タイマ・クロック選択レジスタ50



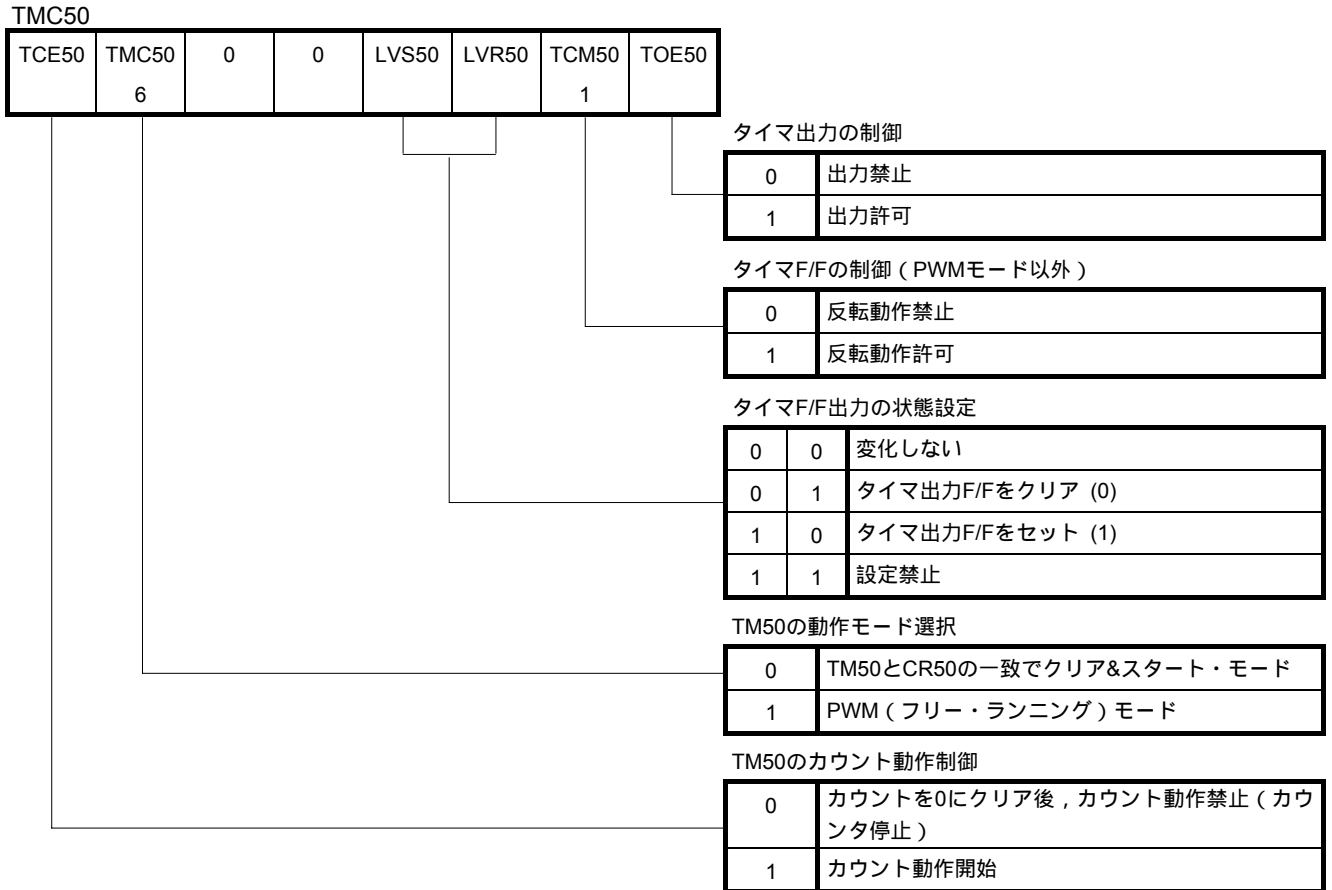
- 注意1. TCL50を同一データ以外に書き換える場合は、いったんタイマ動作を停止させてから書き換えてください。
2. ビット3-7には、必ず0を設定してください。

備考 f_{PRS} : 周辺ハードウェア・クロック周波数

(2) タイマ・モード設定

8ビット・タイマ・カウンタの動作モード，カウンタ制御，出力の状態 / 制御の設定をします。

図4 - 7 8ビット・タイマ・モード・コントロール・レジスタ50



注意1. LVS50とLVR50の設定は，PWMモード以外で有効になります。

2. 以下の設定は同時に行わないでください。また，設定は以下の順で行ってください。

TMC501，TMC506を設定

出力を許可する場合，TOE50を設定

LVS50，LVR50を設定

TCE50を設定

3. TCE50 = 1のとき，TMC50の他のビットを設定することは禁止です。

備考1. LVS50とLVR50は読み出すと，0になっています。

2. TMC506, LVS50, LVR50, TMC501, TOE50の各ビットの値は，TCE50の値に関係なくTO50端子に反映されます。

(3) 8ビット・タイマ・コンペア・レジスタ (CR50) の設定

8ビット・タイマ50のカウンタ値と比較し、2つの値が一致したときに割り込みを発生します。

図4-8 8ビット・タイマ・コンペア・レジスタ50

CR50



注意1. CR50の値は、00H-FFHの範囲で設定します。

2. TM50と値の一致でクリア&スタート・モードのときは、動作中にCR50に異なる値を書き込まないでください。

【例】 ・タイマ・クロックを8 [μs] に設定 (8 MHz動作時)

・インターバル時間を1 [ms] に設定

(サンプル・プログラムの設定と同内容)

TCL50



カウント・クロックの選択



TMC50



タイマ出力の制御



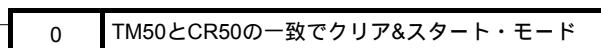
タイマF/Fの制御 (PWMモード以外)



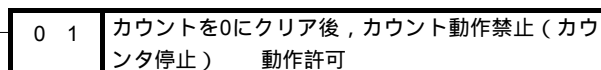
タイマF/F出力の状態設定



TM50の動作モード選択



TM50のカウント動作制御



CR50



インターバル時間 = (124 + 1) × カウント・クロック周期

ここでは、124 (1 [ms]) に設定

【C言語】 (Kx2_Intvl.c)

```
TCL50 = 0b00000101; /* タイマ・クロックをfPRS/2^6に設定 */
CR50 = 125-1; /* コンペア・レジスタの設定 */
TMC50 = 0b00000000; /* タイマ出力はしない */
TCE50 = 1; /* タイマの動作許可 */
.
.
.
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```
MOV TCL50, #00000101B ;タイマ・クロックをfPRS/2^6に設定
MOV CR50, #125-1 ;コンペア・レジスタの設定
MOV TMC50, #00000000B ;タイマ出力はしない
SET1 TCE50 ;タイマの動作許可
.
.
.
```


【8ビット・タイマH0】

インターバル・タイマは、次の項目を設定します。

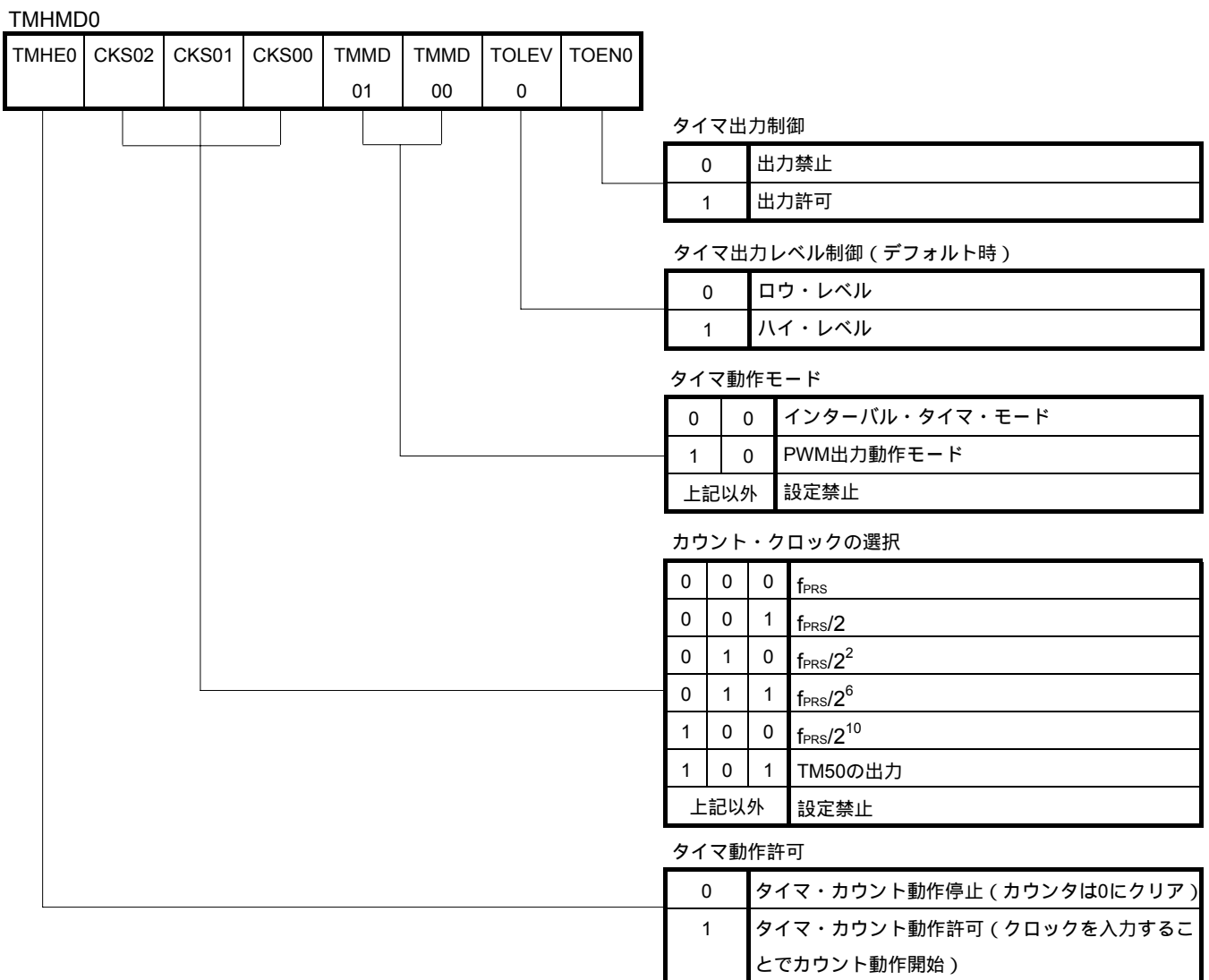
- (1) タイマ・クロック選択, タイマ・モード設定, タイマ出力制御
- (2) タイマ・コンペア・レジスタ (CMP00) の設定

本サンプル・プログラムでは、後述の【例】の内容で設定しています。

(1) タイマ・クロック設定

8ビット・タイマH0のカウンタ・クロック, タイマ・モード, タイマ出力の設定をします。

図4-9 8ビット・タイマHモード・レジスタ0

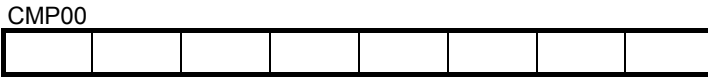


注意 TMHE1 = 1 のとき, TMHMD1の他のビットを設定することは禁止です。
ただし, リフレッシュ (同値書き込み) することは可能です。

(2) 8ビット・タイマ・コンペア・レジスタ (CMP00) の設定

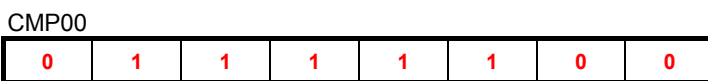
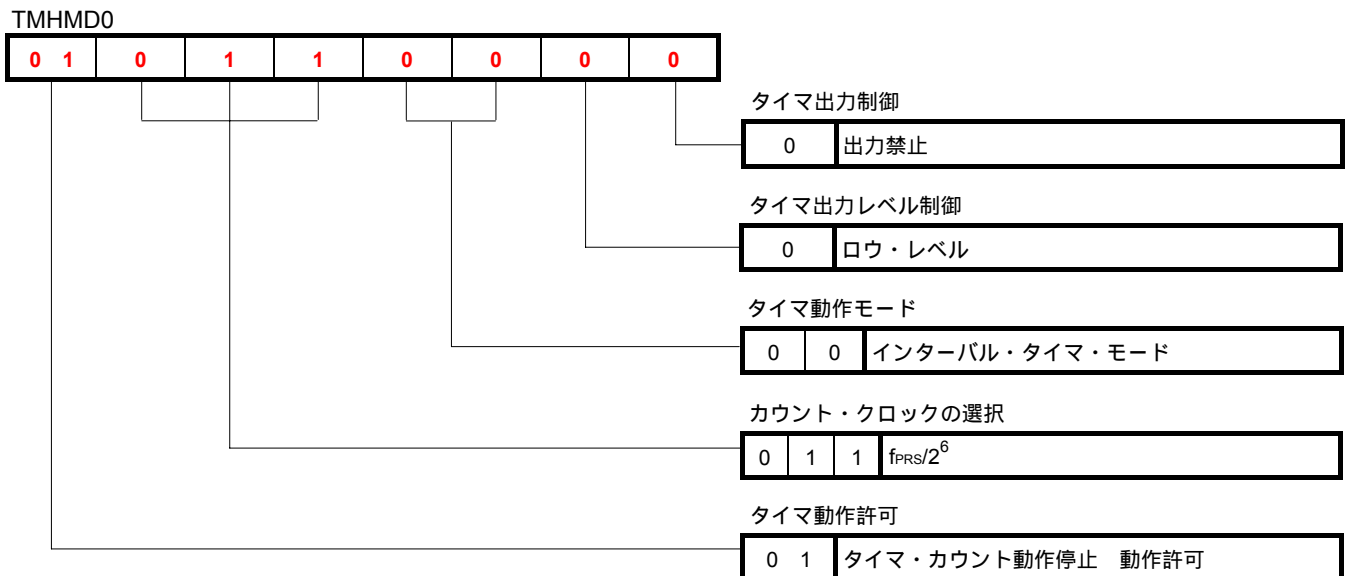
8ビット・タイマH0のカウント値と比較し、2つの値が一致したときに割り込みを発生します。

図4 - 10 8ビット・タイマHコンペア・レジスタ00



注意 CMP00は、タイマ・カウント動作中に値を書き換えないでください。
 ただし、タイマ・カウント動作中にリフレッシュ（同値書き込み）することは可能です。

- 【例】 ・タイマ・クロックを8 [μs] に設定（8 MHz動作時）
 ・インターバル時間を1 [ms] に設定
 （サンプル・プログラムの設定と同内容）



インターバル時間 = (124 + 1) × カウント・クロック周期
 ここでは、124 (1 [ms]) に設定

【C言語】 (Kx2_Intvl.c)

```

TMHMD0 = 0b00110000; /* タイマ・クロックをfPRS/2^6に設定
                       タイマ・モードをインターバル・タイマに設定
                       タイマ出力はしない */
CMP00 = 125-1; /* コンペア・レジスタの設定 */
TMHE0 = 1; /* タイマの動作許可 */
        .
        .
        .
    
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```

MOV     TMHMD0, #00110000B ;タイマ・クロックをfPRS/2^6に設定
                       ;タイマ・モードをインターバル・タイマに設定
                       ;タイマ出力はしない
MOV     CMP00, #125-1 ;コンペア・レジスタの設定
SET1    TMHE0 ;タイマの動作許可
        .
        .
        .
    
```

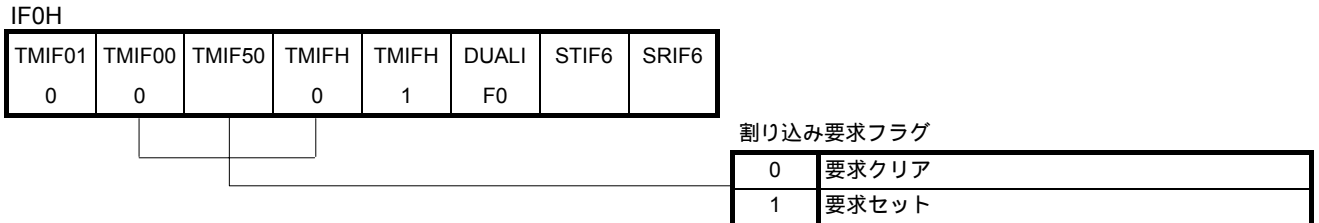
4.3 割り込みの設定

(1) 割り込み要求の設定

指定の割り込みに対して、割り込み要求フラグをクリアします。

このサンプル・プログラムでは、直接レジスタ・ビットに設定しています(【例1】、【例2】、【例3】)。

図4 - 11 割り込み要求フラグ・レジスタ (IF0H) のフォーマット

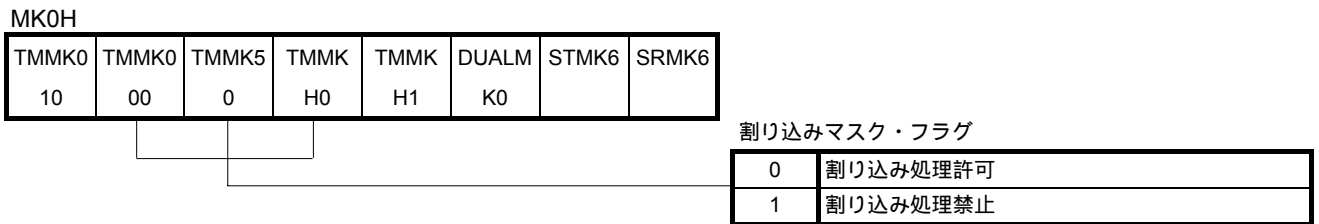


(2) 割り込みマスクの設定

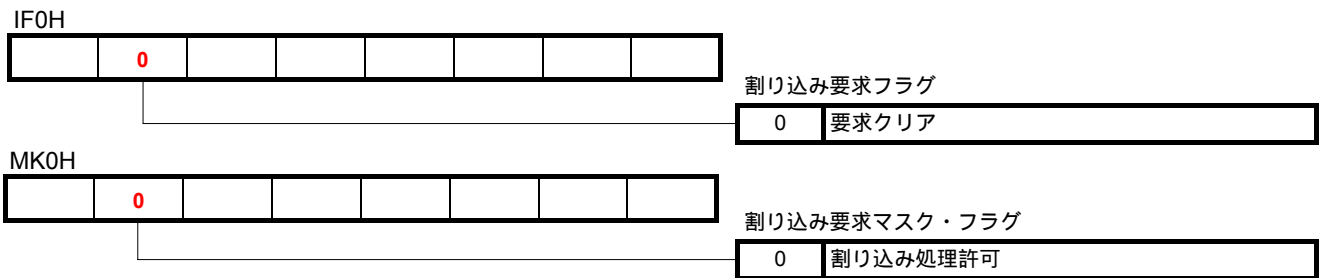
指定の割り込み処理の許可 / 禁止を設定します。

このサンプル・プログラムでは、直接レジスタ・ビットに設定しています(【例1】、【例2】、【例3】)。

図4 - 12 割り込みマスク・フラグ・レジスタ (MK0H) のフォーマット



- 【例1】
- ・ INTTM000割り込み要求 (TMIF000) をクリア
 - ・ INTTM000割り込みマスク (TMMK000) をクリア (割り込み処理許可)
- (サンプル・プログラムTM00編の設定と同内容)



サンプル・プログラムでは1ビット・アクセスにて設定しています。

【C言語】 (Kx2_Intvl.c)

```

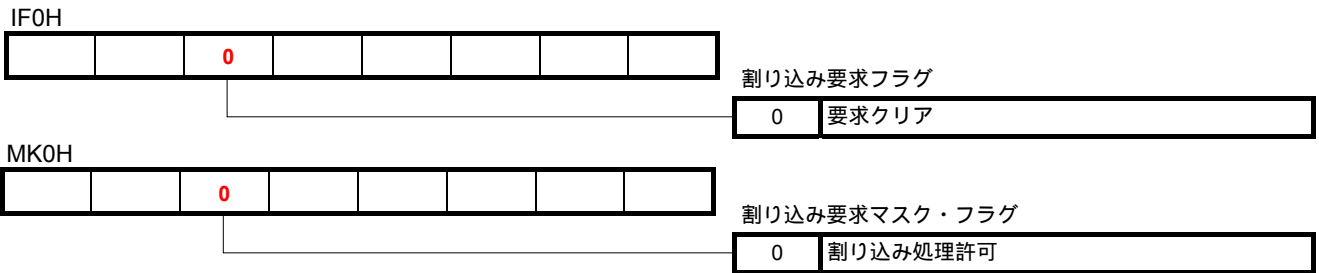
TMIF000 = 0; /* INTTM000割り込み要求クリア */
TMMK000 = 0; /* INTTM000割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```

CLR1    TMIF000    ;INTTM000割り込み要求クリア
CLR1    TMMK000    ;INTTM000割り込みマスク解除
.
.
.
    
```

- 【例2】 ・INTTM50割り込み要求 (TMIF50) をクリア
 ・INTTM50割り込みマスク (TMMK50) をクリア (割り込み処理許可)
 (サンプル・プログラムTM50編の設定と同内容)



サンプル・プログラムでは1ビット・アクセスにて設定しています。

【C言語】 (Kx2_Intvl.c)

```

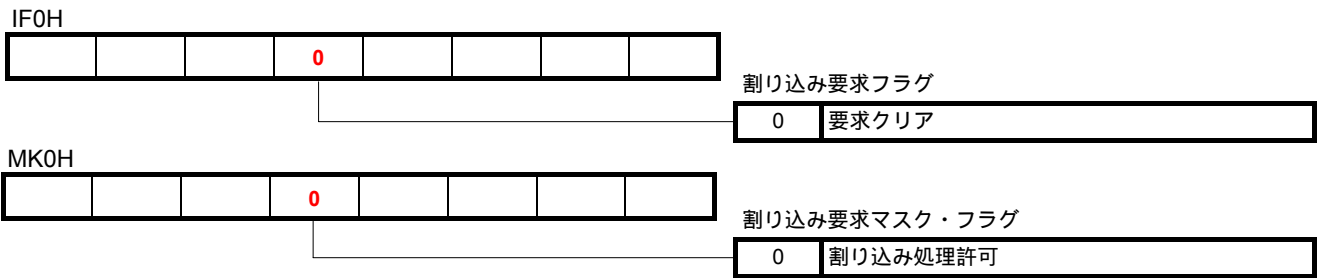
TMIFH0 = 0; /* INTTM50割り込み要求クリア */
TMMKH0 = 0; /* INTTM50割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```

CLR1    TMIF50    ;INTTM50割り込み要求クリア
CLR1    TMMK50    ;INTTM50割り込みマスク解除
.
.
.
    
```

- 【例3】
- ・INTTMH0割り込み要求 (TMIFH0) をクリア
 - ・INTTMH0割り込みマスク (TMMKH0) をクリア (割り込み処理許可)
(サンプル・プログラムTMH0編の設定と同内容)



サンプル・プログラムでは1ビット・アクセスにて設定しています。

【C言語】 (Kx2_Intvl.c)

```

TMIFH0 = 0;          /* INTTMH0割り込み要求クリア */
TMMKH0 = 0;          /* INTTMH0割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Intvl.asm)

```

CLR1    TMIFH0          ;INTTMH0割り込み要求クリア
CLR1    TMMKH0          ;INTTMH0割り込みマスク解除
.
.
.
    
```

4.4 ポートの設定

(1) ポートの入力/出力の設定

このサンプル・プログラムではインターバル・タイマの処理結果を確認するために、P10, P11を出力ポートとして使用します。後述の【例】のような内容で設定しています。

図4 - 13 ポート・モード・レジスタ1 (PM1) のフォーマット

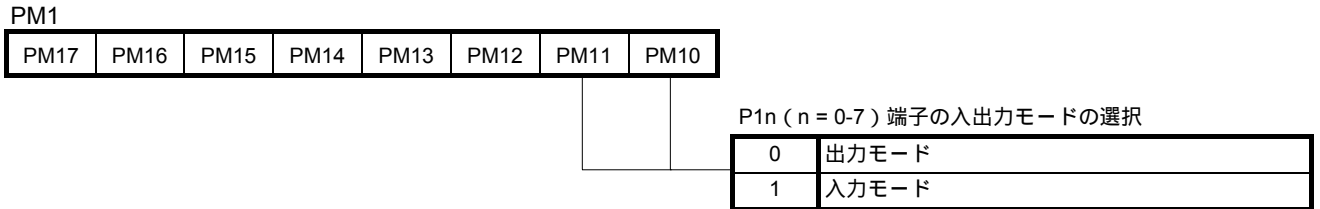
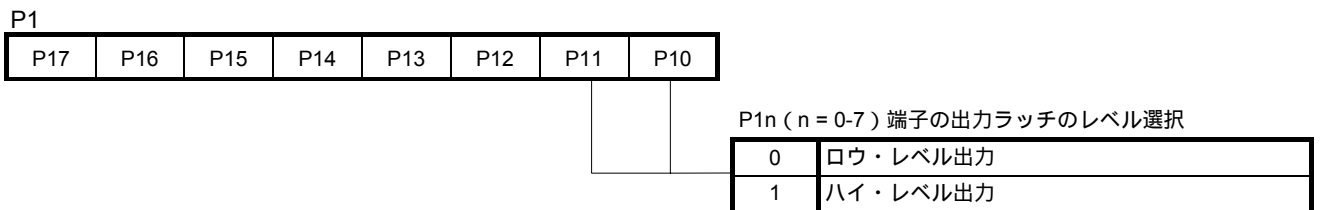
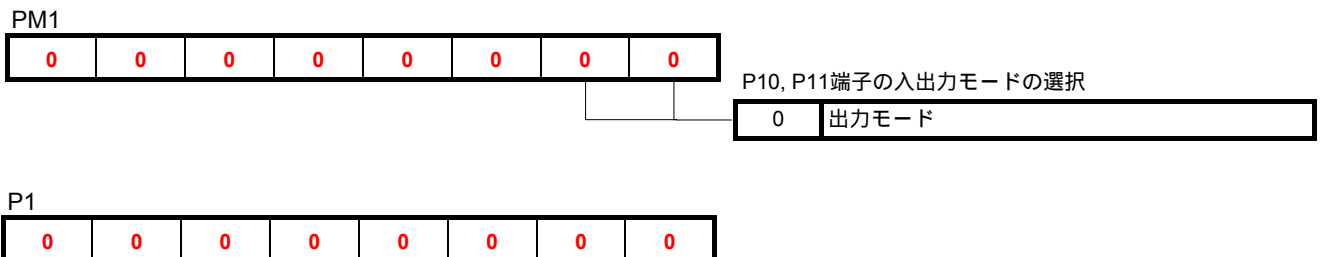


図4 - 14 ポート・レジスタ1 (P1) のフォーマット



【例】 ・ P10, P11を出力ポートに設定
(サンプル・プログラムの設定と同内容)



PM1の設定値は「00000000」、P1の設定値は「00000000」とします。

【C言語】

```
PM1 = 0b00000000;
P1 = 0b00000000;
```

【アセンブリ言語】

```
MOV PM1, #00000000B
MOV P1, #00000000B
```

4.5 メイン処理

メイン処理では、次の動作を行います。

【C言語】 (Kx2_Intvl.c)

- ・インターバル・タイマ設定関数を呼び出します（タイマ処理を開始します）。
- ・インターバル・カウンタ（割り込み回数カウンタ）を初期化（0回）します。
- ・メイン処理内では何もせず、割り込みを待つのでNOP命令のみのループ処理になります。

```

/*****
;
;   メイン・ループ
;   (インターバル・タイマを起動して、1ms毎の割り込みを待つ)
;*****/
void main(void){
    g_ucIntCnt =      0;           /* インターバル・カウンタを初期化 */
    fn_InitTimer();           /* インターバル・タイマの初期化 */
    EI();                     /* ベクタ割り込み許可 */
    while (1){
        NOP();                 /* 何もしないで、割り込み待ち */
    }
}

```

インターバル・タイマの初期化を行い、処理を開始します。

メイン処理内ではNOP命令のみで割り込みを待ちます。

【アセンブリ言語】 (Kx2_Intvl.asm)

- ・インターバル・カウンタ（割り込み回数カウンタ）を初期化（100回）します。
- ・メイン処理内では何もせず、割り込みを待つのでNOP命令のみのループ処理になります。

```

;*****/
;
;   メイン・ループ
;   (インターバル・タイマを起動して1ms毎の割り込みを待つ)
;*****/
MOV    RINTCNT, #100           ;インターバル・カウンタ初期値を100に設定
EI                                           ;ベクタ割り込み許可
MMAIN:
NOP                                           ;何もしないで、割り込み待ち
BR     MMAIN

```

インターバル・カウンタを初期化します。

メイン処理内ではNOP命令のみで割り込みを待ちます。

アセンブリ言語でのインターバル・タイマの初期化は、メイン処理に入る前に、インターバル・タイマの設定および割り込みの設定で示したアセンブリ言語版の設定例と同じ内容で初期化しています。

4.6 割り込み処理

割り込み処理では、次の動作を行います。

【C言語】 (Kx2_Intvl.c)

- ・インターバル・カウンタ（割り込み回数カウンタ）をカウントアップします。
- ・P10の値を反転します。
- ・インターバル・カウンタをチェック(==100)し、合致したらP11を反転し、インターバル・カウンタをクリア(0)します。

```

/*****
;
;   インターバル・タイマ割り込み処理
;
; *****/
__interrupt void fn_intTimerInterval(void)
{
    g_ucIntCnt++;                /* 割り込みが入る毎にインターバル
                                カウンタをインクリメント */

    /*-----
;   インターバル経過毎(割り込みが入る毎)にP10を反転させる
;-----*/
    if( P1.0 ) P1.0 = 0;        /* 割り込みが入る毎にP10反転 */
    else      P1.0 = 1;

    /*-----
;   P11に接続されたLEDはインターバル100回経過毎に点灯/消灯(P11反転)させる
;-----*/
    if(g_ucIntCnt == 100){     /* インターバル・カウンタ=100ならば処理
                                実行し、Noならば処理せずに抜ける */
        if( P1.1 ) P1.1 = 0;    /* 割り込み100回毎に反転 */
        else      P1.1 = 1;
        g_ucIntCnt = 0;        /* インターバル・カウンタを初期化 */
    }
}

```

P11の結果はLEDに反映されます

P11	LED
0	点灯
1	消灯

【アセンブリ言語】 (Kx2_Intvl.asm)

- ・ P10の値を反転。
- ・ インターバル・カウンタ (割り込み回数カウンタ) をデクリメントします。
- ・ インターバル・カウンタが0になったら, P11を反転し, インターバル・カウンタを初期化 (100) します。

```

;*****
;
;   インターバル・タイマ割り込み処理
;
;*****
IINTVL:
    XOR    P1,      #00000001B      ;P10を反転
    DBNZ  RINTCNT, $HINTEND        ;インターバル・カウンタをデクリメントし,
                                   ;0にならなければINTENDに分岐

    XOR    P1,      #00000010B      ;P11を反転
    MOV   RINTCNT, #100            ;インターバル・カウンタを初期化


HINTEND:
    RETI

```


P11の結果はLEDに反映されます

P11	LED
0	点灯
1	消灯

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C言語用のファイルを用い、サンプル・プログラムが、システム・シミュレータ SM+ for 78K0/Kx2でどのように動作するかを説明します。

5.1 サンプル・プログラムのビルド

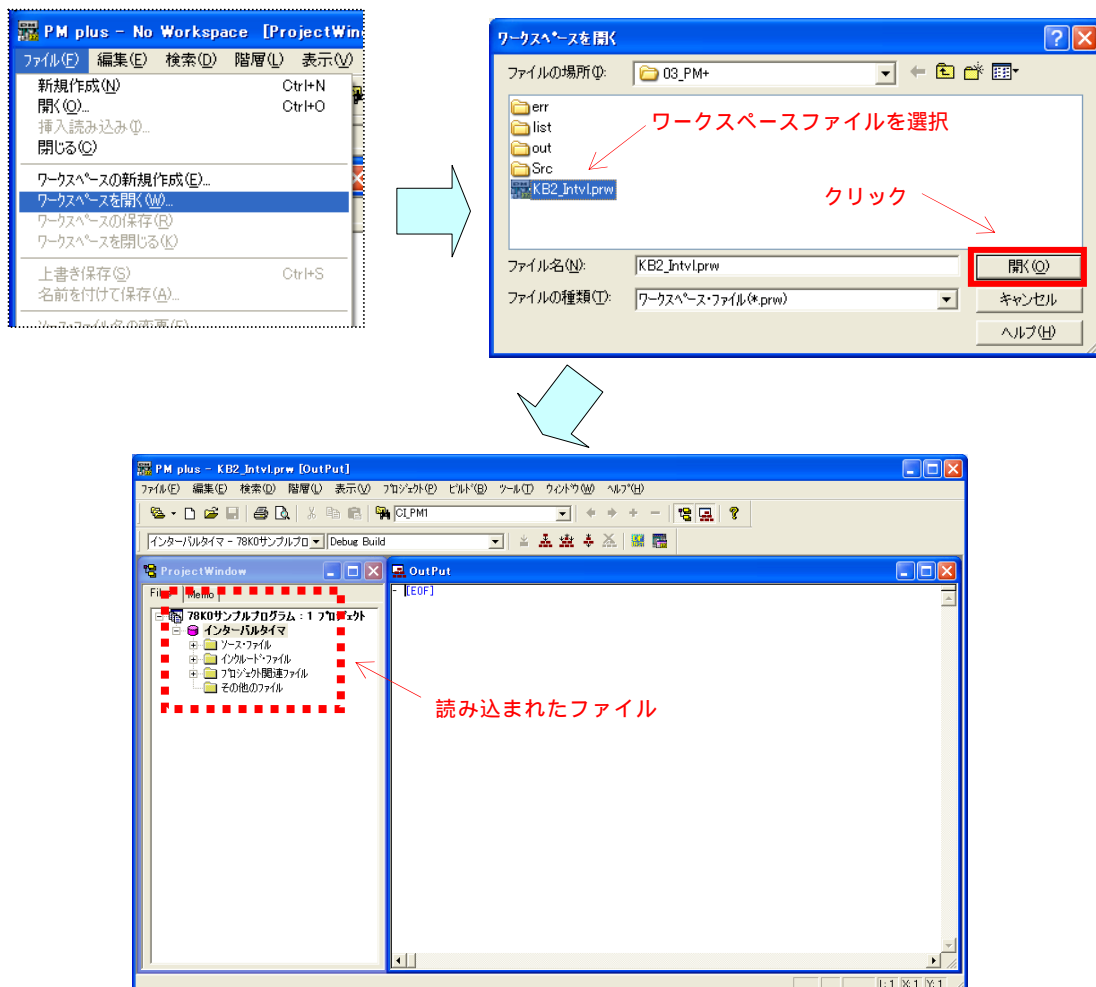
サンプル・プログラムをSM+ for 78K0/Kx2 (以降、「SM+」と表記します)で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、でダウンロードしたC言語用のファイルを用いて、統合開発環境 PM+にてビルドしてから、SM+を起動するまでの動作の一例を説明します。

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。

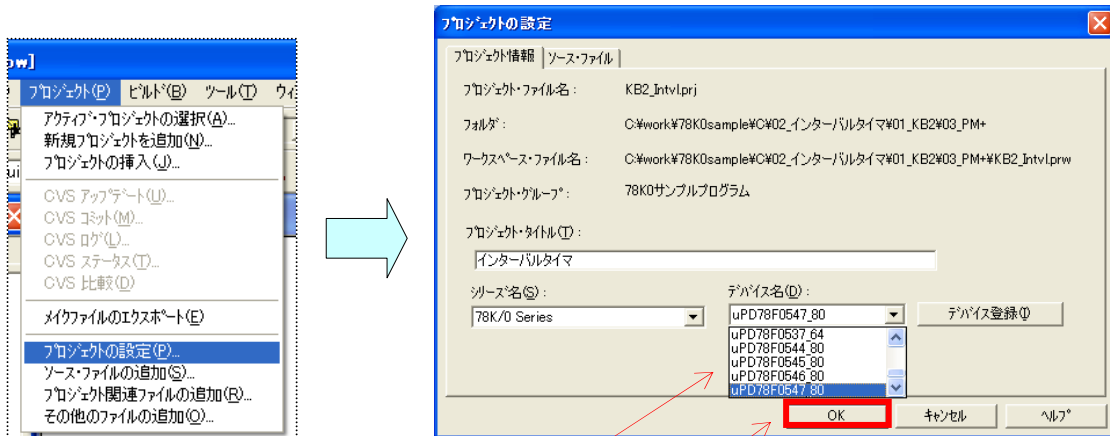
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「Kx2_Intvl.prw」^注を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。

注 ファイル名の"x"部分は対象デバイスにあわせて変更してください。

ex) 78K0/KB2の場合「KB2_Intvl.prw」



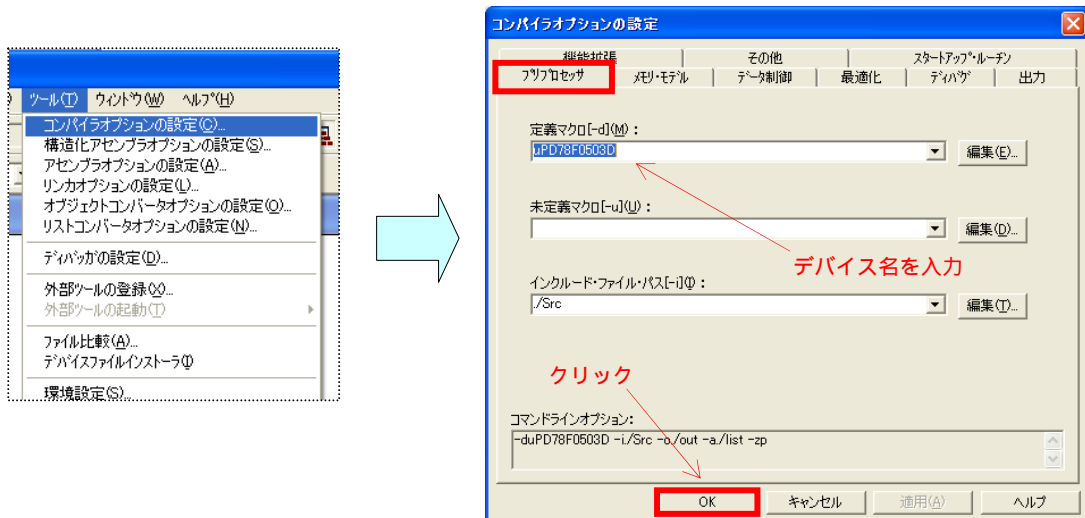
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が表示されたら、使用するデバイス名を選択（デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択）し、[OK] ボタンをクリックしてください。



デバイス名を指定 クリック


μ PD78F0500_36, μ PD78F0501_36, μ PD78F0502_36, μ PD78F0503_36は選択しないでください。

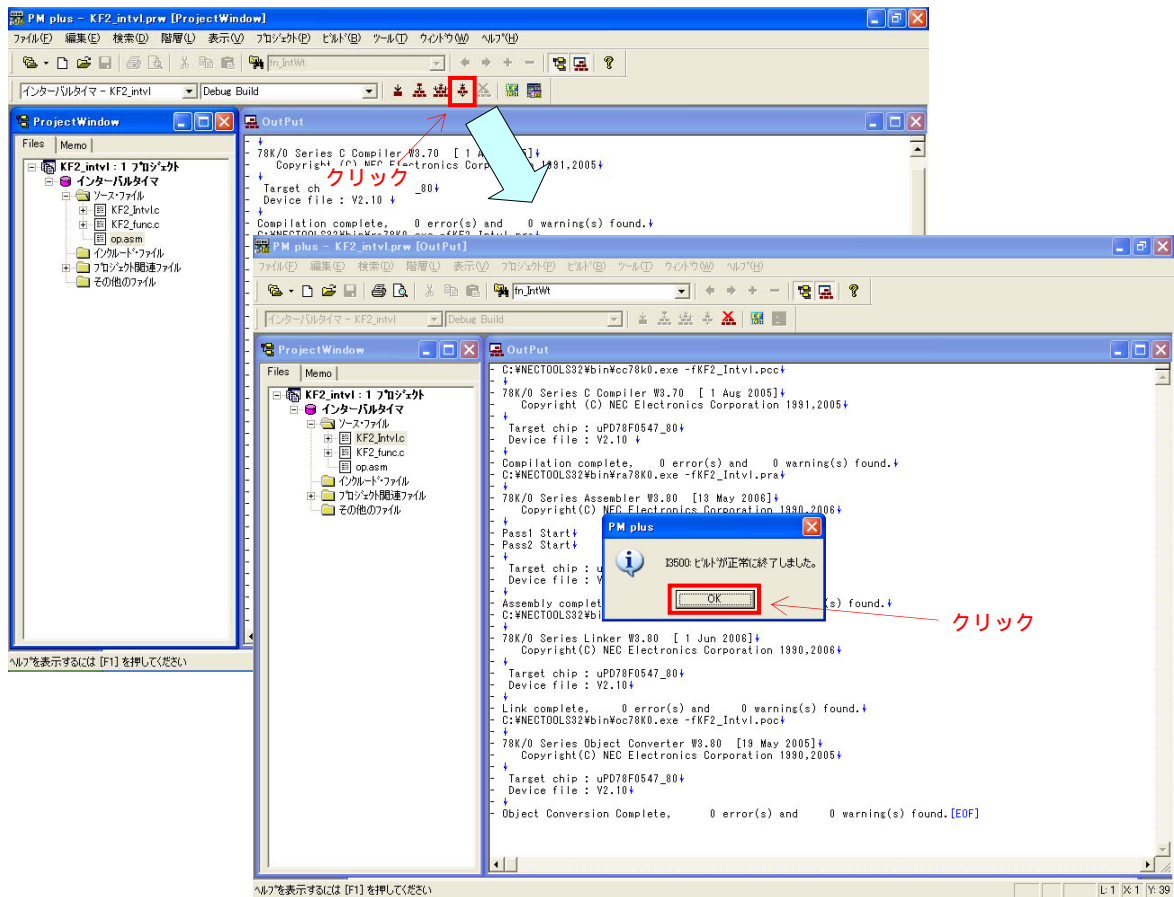
- (4) [ツール] [コンパイラオプションの設定] を選択してください。[コンパイラオプションの設定] 画面が表示されたら、[プリプロセッサ] タグページが表示されているのを確認し、その中の定義マクロ欄に使用するデバイス名を入力し、[OK] をクリックします。



デバイス名を入力

クリック

- (5)  (「ビルド ディバグ」ボタン) をクリックしてください。ソース・ファイルの「Kx2_Intvl.c」と「Kx2_func.c」と「Kx2_op.asm」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表示されます。
- (6) メッセージ画面にある [OK] ボタンをクリックすると、SM+が自動的に立ち上がります。



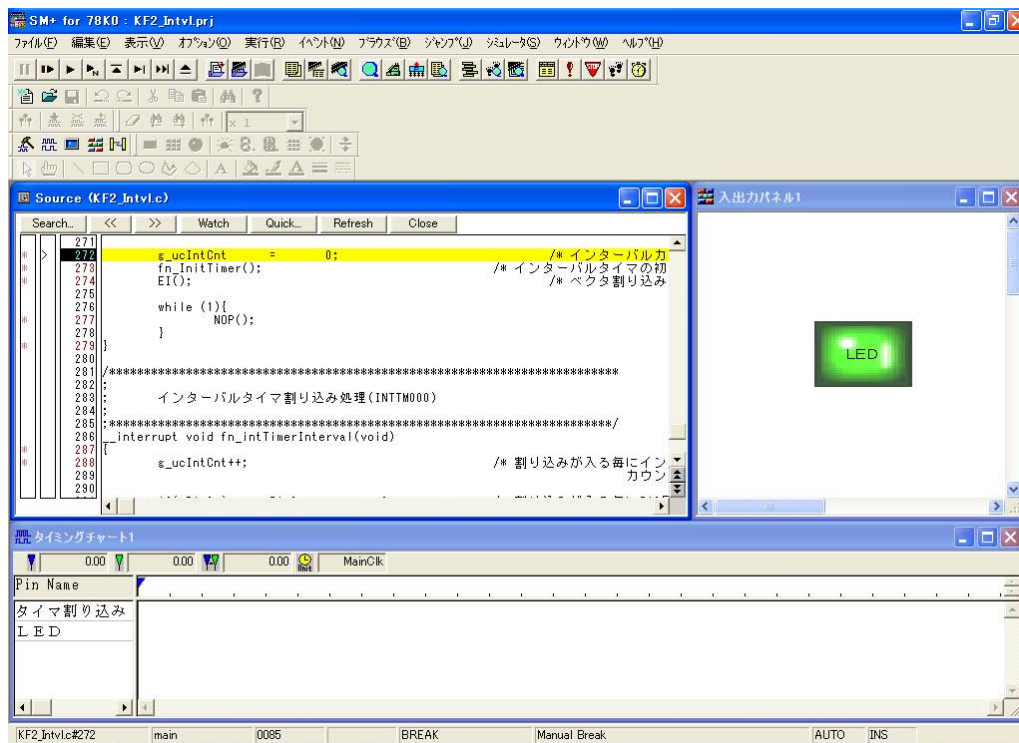
SM+が自動的に起動されます


5.2 SM+での動作

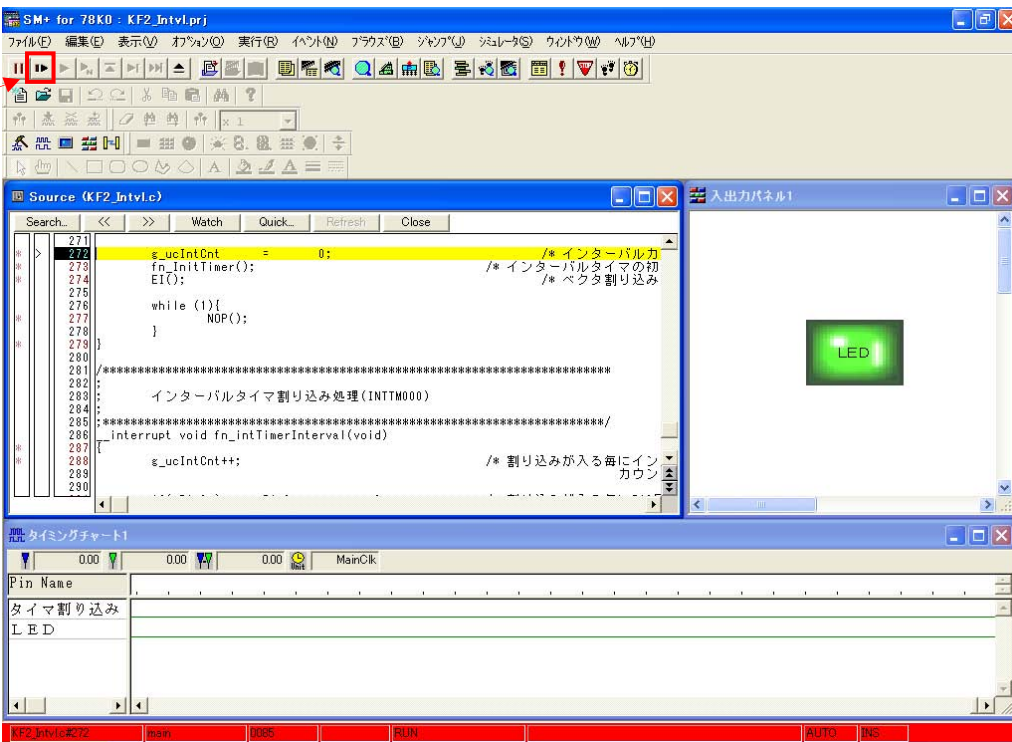
ここでは、SM+の入出力パネル・ウインドウやタイミング・チャート・ウインドウ上での動作確認の例を説明します。

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) PM+の「ビルド ディバグ」からSM+を起動(5.1を参照)すると、次のような画面になります。



- (2)  (「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。




クリック

```


271  g_ucIntCnt = 0; /* インターバルカ
272  fn_initTimer(); /* インターバルタイマの初
273  EI(); /* ベクタ割り込み
274
275  while (1){
276      NOP();
277  }
278
279
280
281  /*****
282   インターバルタイマ割り込み処理(INTTM00)
283  *****/
284
285  /*****
286   interrupt void fn_intTimerInterval(void)
287  *****/
288  {
289      g_ucIntCnt++; /* 割り込みが入る毎にイン
290                  カウン

```


プログラム実行中は、 赤になります。

(3) プログラム実行中は、入出力パネル・ウィンドウ上の [LED] が100 ms単位で点滅を繰り返します。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。

例1. LED : ONの場合

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>LED点灯</p>	<table border="1"> <tr> <td>Pin Name</td> <td></td> <td></td> </tr> <tr> <td>タイマ割り込み</td> <td></td> <td></td> </tr> <tr> <td>LED</td> <td></td> <td></td> </tr> </table> <p>または</p> <table border="1"> <tr> <td>Pin Name</td> <td></td> <td></td> </tr> <tr> <td>タイマ割り込み</td> <td></td> <td></td> </tr> <tr> <td>LED</td> <td></td> <td></td> </tr> </table>	Pin Name			タイマ割り込み			LED			Pin Name			タイマ割り込み			LED		
Pin Name																			
タイマ割り込み																			
LED																			
Pin Name																			
タイマ割り込み																			
LED																			

例2. LED : OFFの場合

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>LED消灯</p>	<table border="1"> <tr> <td>Pin Name</td> <td></td> <td></td> </tr> <tr> <td>タイマ割り込み</td> <td></td> <td></td> </tr> <tr> <td>LED</td> <td></td> <td></td> </tr> </table> <p>または</p> <table border="1"> <tr> <td>Pin Name</td> <td></td> <td></td> </tr> <tr> <td>タイマ割り込み</td> <td></td> <td></td> </tr> <tr> <td>LED</td> <td></td> <td></td> </tr> </table>	Pin Name			タイマ割り込み			LED			Pin Name			タイマ割り込み			LED		
Pin Name																			
タイマ割り込み																			
LED																			
Pin Name																			
タイマ割り込み																			
LED																			

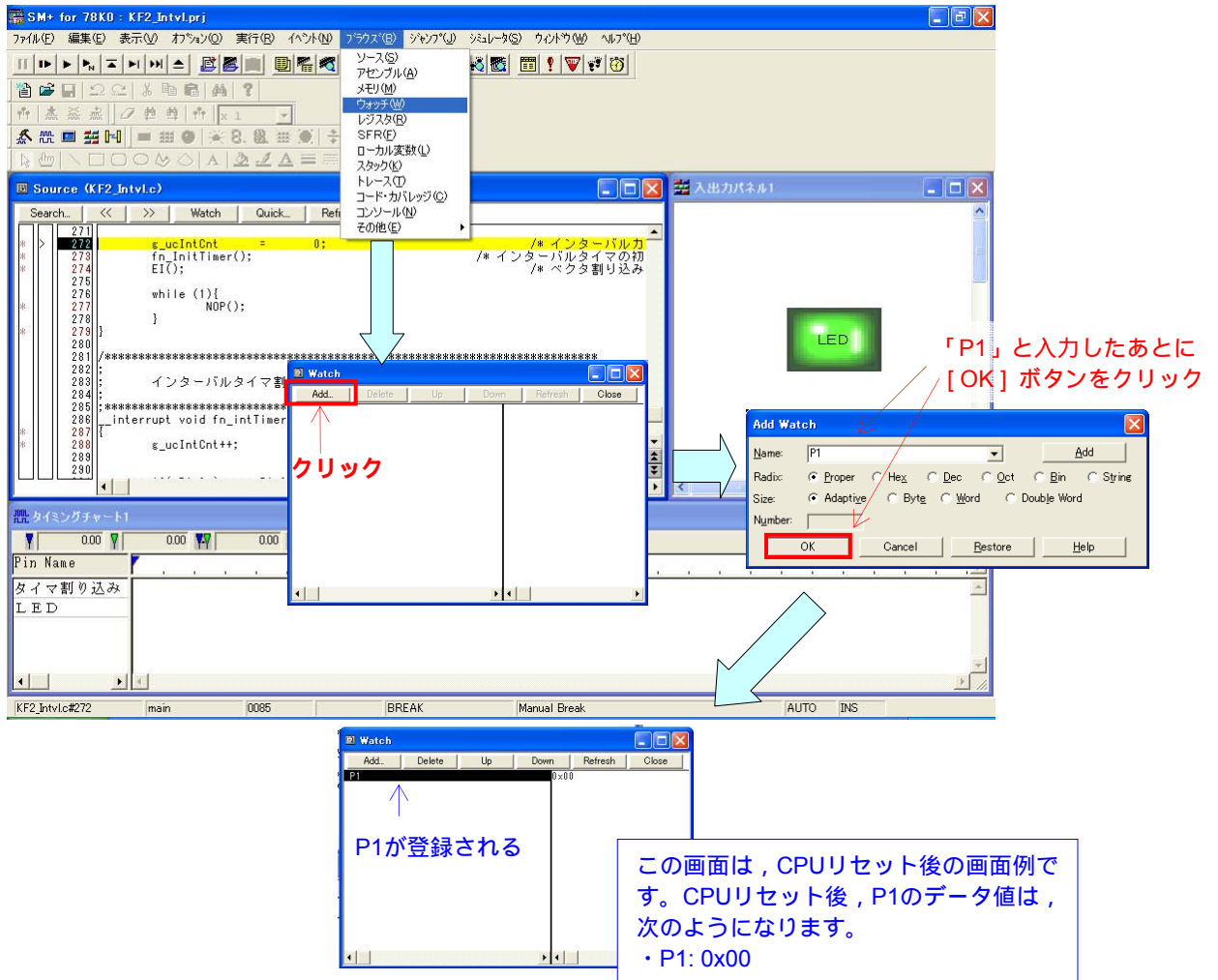
備考 H : ハイ・レベル , L : ロウ・レベル

【補 足】SM+のウォッチ機能を使用することにより，ポート1のデータ値の変化を確認することができます。

[ブラウズ] [ウォッチ] を選択してください。[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「P1」と入力し，[OK] ボタンをクリックすると，[Watch] ウィンドウに，「P1」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，ウィンドウ上のP1のデータ値が変化することを確認してください ([ストップ] (「ストップ」ボタン) をクリックすることで，確認できます)。

LED状態	[Watch] ウィンドウのデータ値
ON	P1: 0x01 または P1: 0x00
OFF	P1: 0x02 または P1: 0x03

5.3 オンチップ・デバッグ時の注意

ここでは、サンプル・プログラムを用いて、オンチップ・デバッグを行う際の手順を説明します。
 オンチップ・デバッグ機能については、ユーザズ・マニュアルを参照してください。

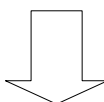
(1) オプション・バイトの設定

本サンプル・プログラムはオプション・バイトの初期設定でオンチップ・デバッグ禁止になっています。
 オプション・バイトを設定し直して、オンチップ・デバッグを許可します。
 オプション・バイト設定は Kx2_op.asm で行っています。
 次に、そのKx2_op.asmファイル内のオンチップ・デバッグ設定部分のみ抜粋して記載します。

```

;                                     印が設定値
;   DB      00000000B      ;0084H      :[オンチップデバッグ]
;           |||||++---      OCDEN1-0    :[オンチップデバッグ動作制御]
;           |||||          00:動作禁止
;           |||||          01:設定禁止
;           |||||          10:動作許可(認証失敗でフラッシュ消去せず)
;           |||||          11:動作許可(認証失敗でフラッシュ消去)
;           ++++++-----      0          必ず0に設定
    
```

動作許可(認証失敗でフラッシュ消去せず)に設定



```

;                                     印が設定値
;   DB      00000010B      ;0084H      :[オンチップデバッグ]
;           |||||++---      OCDEN1-0    :[オンチップデバッグ動作制御]
;           |||||          00:動作禁止
;           |||||          01:設定禁止
;           |||||          10:動作許可(認証失敗でフラッシュ消去せず)
;           |||||          11:動作許可(認証失敗でフラッシュ消去)
;           ++++++-----      0          必ず0に設定
    
```

(2) オンチップ・デバッグ使用領域の確保 (アセンブリ言語版のみ)

アセンブリ言語版はオンチップ・デバッグ使用領域を確保する必要があります。

本サンプル・プログラムでは、以下のようにオンチップ・デバッグ使用領域を確保しています。

```

;=====
;      ベクタテーブル
;
; このサンプル・プログラムでは割り込みは使用していない。割り込み
;ベクタ・テーブルは全て不要割り込み処理アドレスに定義する。
;=====
TVECTTBL      CSEG      AT      0000H

              DW      IRESET      ;0000H RESET入力, POC, LVI, WDT
;              DW      IINIT      ;0002Hはオンチップデバッグ用に空ける

TVECT_TBL1    CSEG      AT      0004H
              DW      IINIT      ;0004H INTLVI
    
```

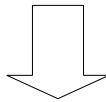
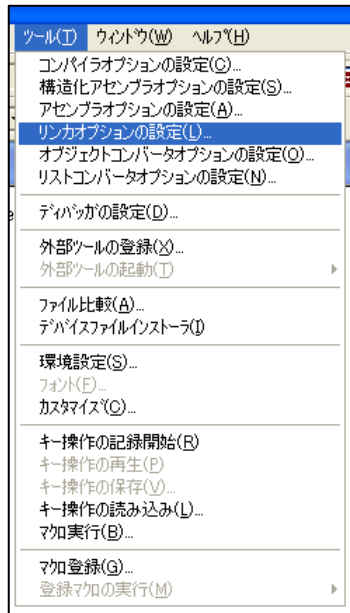
0002H番地はオンチップ・デバッグ使用領域として空けるため、0002H番地はコメント・アウトして、CSEGで再び0004H番地を定義しています。

C言語版では不要です。

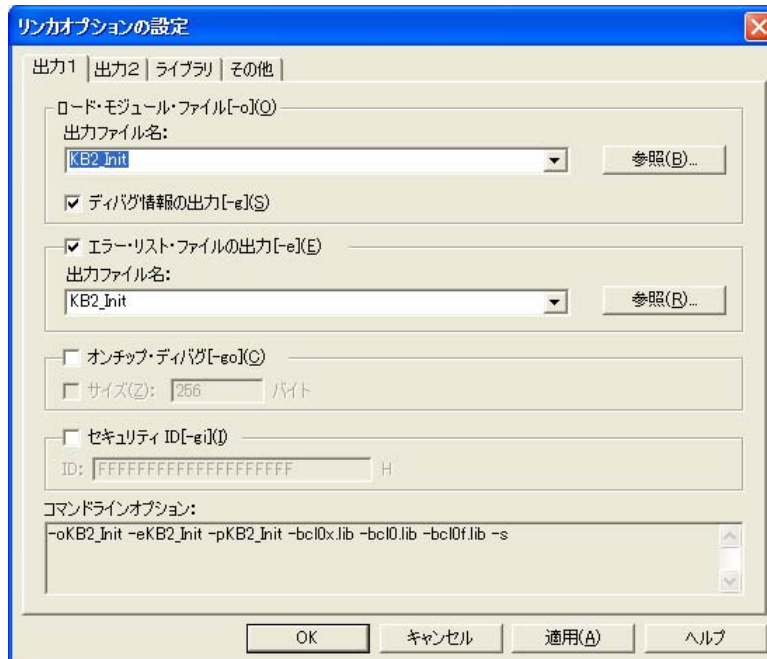
(3) リンカの設定

オンチップ・デバッグを行う場合、ビルドの際、リンカの設定を行う必要があります。

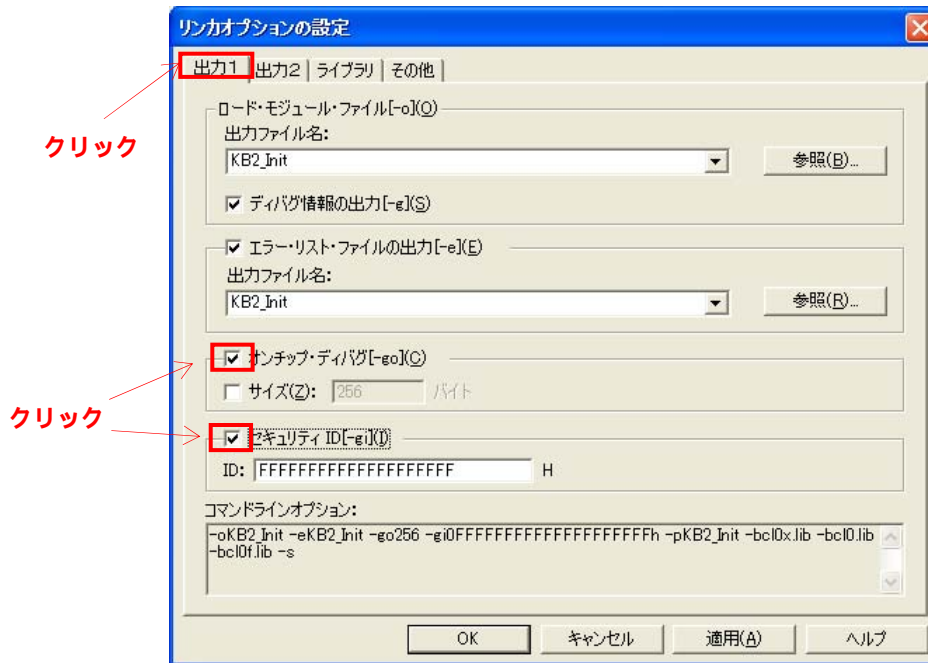
PM+の「ツール」メニューから「リンカオプションの設定」を選択してください。



「リンカオプションの設定」を選択するとリンカオプションの設定ダイアログが表示されます。



リンカオプションの設定ダイアログの「出力1」タブ上にある「オンチップ・デバッグ」と「セキュリティID」のチェックボックスをONしてください。



OKボタンを押下して設定完了です。

5.4 開発環境のダウンロード，インストール

78K0/Kx2マイクロコントローラの開発ツールのフリーツールは，次のサイトより入手可能です。

→<http://www.necel.com/micro/ja/freesoft/78k0/kx2/index.html>

「SM+ for 78K0/Kx2」「RA78K0」「CC78K0」「78K0/Kx2用デバイス・ファイル」の4ファイルをダウンロードし，インストールすることで，サンプル・プログラムの動作確認が可能となります。

ダウンロード，インストールは，上記サイトの画面および説明に従って，行ってください。

備考1. PM+は，RA78K0に同封されています。

2. ダウンロード後，登録したEメール・アドレスに，RA78K0, CC78K0, SM+ for 78K0/Kx2のプロダクトIDが送付されます。このプロダクトIDは，各ツールのインストール時に必要となります。

第6章 関連資料

資料名	和文 / 英文
78K0/Kx2 ユーザーズ・マニュアル	PDF
78K/0シリーズ 命令編 ユーザーズ・マニュアル	PDF
RA78K0 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
CC78K0 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

付録A 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	November 2007	-	-
第2版	May 2009	全般	全面見直し

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：(044)435-5111

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【資料請求先】

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

—— お問い合わせ先 ——

【営業関係、デバイスの技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00~12:00, 午後 1:00~5:00)

電 話 : (044)435-9494

E-mail : info@necel.com

【マイコン開発ツールの技術関係お問い合わせ先】

開発ツールサポートセンター

E-mail : toolsupport-micom@ml.necel.com