

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

アプリケーション・ノート

78K0/Kx2

サンプル・プログラム

外部イベント・カウンタ編

この資料は、サンプル・プログラムの「外部イベント・カウンタ」の動作概要と、外部イベント・カウンタ機能の設定を説明したものです。サンプル・プログラムでは、外部イベント・カウンタ機能の設定を行ったあとに、入力イベントに応じ、LED出力制御を行います。

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 外部イベント・カウンタ機能の設定と動作概要 ... 9
 - 3.4 フロー・チャート ... 10
- 第4章 設定方法について ... 11
 - 4.1 前処理指令 ... 11
 - 4.2 外部イベント・カウンタの設定 ... 12
 - 4.3 割り込みの設定 ... 23
 - 4.4 ポートの設定 ... 25
 - 4.5 メイン処理 ... 26
 - 4.6 割り込み処理 ... 27
- 第5章 システム・シミュレータ SM+での動作確認 ... 28
 - 5.1 サンプル・プログラムのビルド ... 28
 - 5.2 SM+での動作 ... 31
 - 5.3 オンチップ・デバッグ時の注意 ... 35
 - 5.4 開発環境のダウンロード、インストール ... 38
- 第6章 関連資料 ... 39
- 付録A 改版履歴 ... 40

資料番号 U19034JJ1V0AN00 (第1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、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、一切その責を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。また、当社製品は耐放射線設計については行っておりません。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないよう、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。

第1章 概 要

このサンプル・プログラムでは、外部イベント・カウンタの設定を行います。

78K0/Kx2では8ビット・タイマまたは16ビット・タイマの外部イベント・カウンタ機能を使用して実現します。

本プログラムでは、TI000端子から入力される外部からのクロック・パルス立ち下がりエッジをカウントし、10回カウントごとに割り込み（INTTM000）を発生させ、その割り込み内でP1をインクリメントします。そして、そのカウント・アップしていくP1のバイナリ値に合わせ、P1に接続された8個のLEDを点灯させます。

(1) 外部イベント・カウンタ設定内容

- ・ タイマ関連の設定
- ・ 出力ポート（LED接続）の設定
- ・ 入力ポート（クロック入力）の設定
- ・ 割り込み許可設定

(2) メイン処理動作の内容

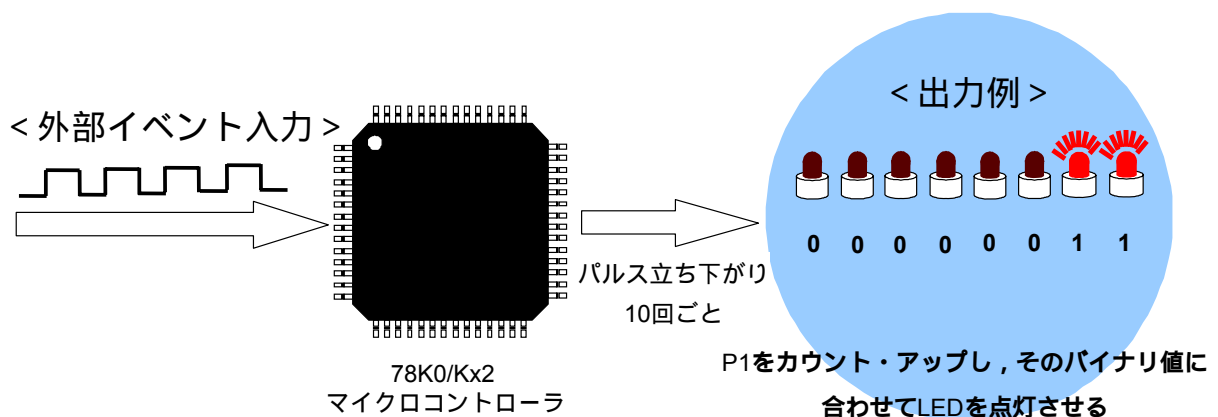
- ・ 外部イベント・カウンタを起動し、NOP命令のみの割り込み待ちループに入ります。

(3) 割り込み処理動作の内容

- ・ LEDの点灯制御（LED接続ポートをインクリメント）

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

（ベクタ・テーブル0020H、001EH番地の割り込みハンドラ名称を“ IINIT ” から “ IINTTM000 ” ， “ IINTTM50 ” に変更します。）



(4) 外部イベント・カウンタの種類

外部イベント・カウンタは、8ビット・タイマ、16ビット・タイマの外部イベント・カウンタ機能を使用して実現します。

使用できるのは、16ビット・タイマ/イベント・カウンタ00, 01^註、8ビット・タイマ/イベント・カウンタ50, 51の4つです。

注 製品によっては00のみ使用可能です（次表参照）。

16ビット・タイマの使用対応デバイス

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

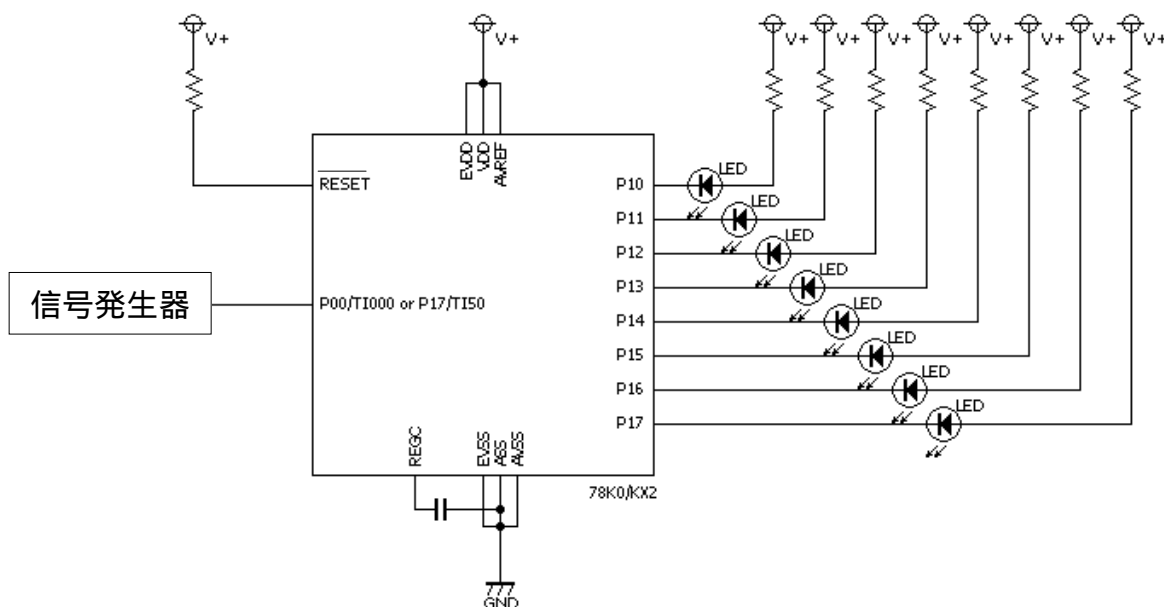
注意 デバイス使用上の注意事項については、[78K0/Kx2 ユーザーズ・マニュアル](#)を参照してください。

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. AV_{REF}端子はV_{DD}に直接接続してください。
2. AV_{SS}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 μF) を介し、V_{SS}に接続してください。
 4. EV_{DD}端子はV_{DD}に直接接続してください (78K0/KE2, 78K0/KF2のみ)。
 5. EV_{SS}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) 信号発生器

タイマ入力端子P00/TI000またはP17/TI50に外部イベントを入力するために使用します。

(2) LED

外部イベント入力に対応した出力として、LEDを8個使用します。




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PPG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_Event.c | 外部イベント・カウンタ機能の設定、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_Event.prw | 統合開発環境 PM+用ワーク・スペース・ファイル | | | |
| Kx2_Event.prj | 統合開発環境 PM+用プロジェクト・ファイル | | | |
| Kx2_Event.pri Kx2_Event.prs Kx2_Event.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_Event0.pnl | システム・シミュレータ SM+ for 78K0/Kx2用入力パネル・ファイル (周辺ハードウェア動作を確認するために使用) | | | |
| Kx2_Event0.wvi | システム・シミュレータ SM+ for 78K0/Kx2用信号データ・ファイル (信号発生器をシミュレートし、信号を入力するために使用) | | | |

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_Event.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_Event.asm | 外部イベント・カウンタ機能の設定，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_Event.prw | 統合開発環境 PM+用ワーク・スペース・ファイル | | | |
| Kx2_Event.prj | 統合開発環境 PM+用プロジェクト・ファイル | | | |
| Kx2_Event.pri Kx2_Event.prs Kx2_Event.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_Event0.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） | | | |
| Kx2_Event0.wvi | システム・シミュレータ SM+ for 78K0/Kx2用信号データ・ファイル（信号発生器をシミュレートし，信号を入力するために使用） | | | |

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_Event.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
- ・ 8ビット・タイマ50

また、このサンプル・プログラムでは、上記2つのタイマについての設定例を紹介しています。

- ・ 入力ポート : P00/TI000 or P17/TI50
- ・ 出力ポート : P1

TM01, TM51のサンプル・プログラムは、上記TM00, TM50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INTTM001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0038H (INTTM001) に変更
- C, アセンブリ言語 : “ 外部イベント・カウンタの初期化 ” でのTM00の各レジスタ名称の00を01に変更, TM00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
- C, アセンブリ言語 : “ 外部イベント・カウンタの初期化 ” でのP00とPM00(TI000) をP05とPM05(TI001) に変更

【TM50 TM51への変更点】

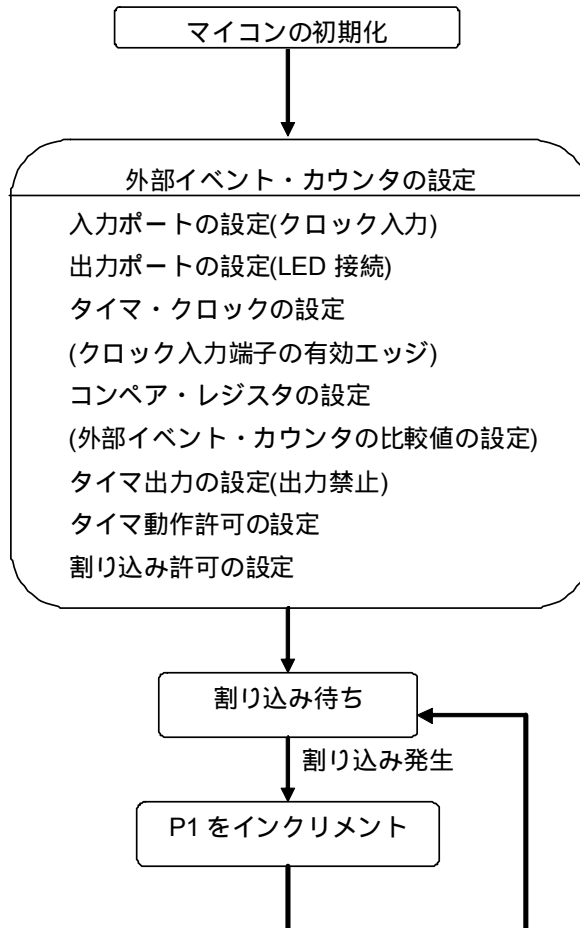
- C言語 : 割り込み処理関数宣言をINTTM50からINTTM51に変更
- アセンブリ言語 : 割り込みベクタ・テーブルを001EH (INTTM50) から002AH (INTTM51) に変更
- C, アセンブリ言語 : “ 外部イベント・カウンタの初期化 ” でのTM50の各レジスタ名称の50を51に変更, TM50の割り込み要求フラグと割り込みマスク・フラグの名称の50を51に変更
- C, アセンブリ言語 : “ 外部イベント・カウンタの初期化 ” でP17とPM17 (TI50) をP30とPM30 (TI51) に変更

3.3 外部イベント・カウンタ機能の設定と動作概要

このサンプル・プログラムでは、外部イベント・カウンタ機能の設定を行います。

設定完了後は、入力ポートP00/TI000またはP17/TI50に入力されるパルスを検出し、パルスの立ち下がりを10回検出することにP1をカウントアップし、P1に接続された8個のLEDを点灯させます。

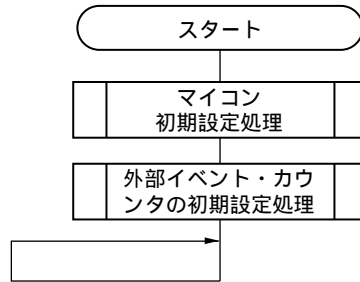
処理概要については、状態遷移図（ステート・チャート）に示します。



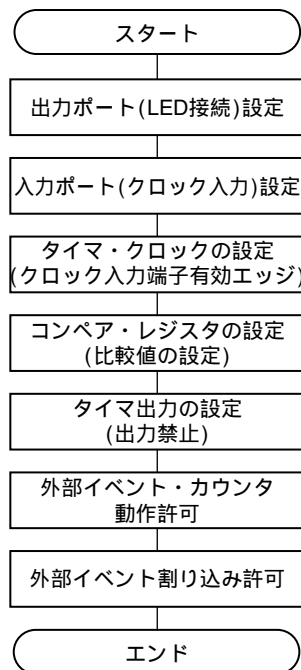
3.4 フロー・チャート

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。

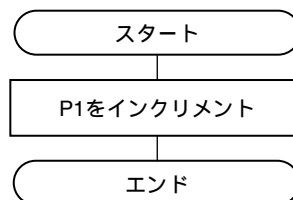
ゼネラル・フロー (C言語版 : Kx2_Event.c アセンブリ言語版 : Kx2_Event.asm)



外部イベント・カウンタ初期設定処理 (C言語版 : Kx2_Event.c アセンブリ言語版 : Kx2_Event.asm)



外部イベント割り込み処理 (C言語版 : Kx2_Event.c アセンブリ言語版 : Kx2_Event.asm)



第4章 設定方法について

この章では、外部イベント・カウンタ（16ビット・カウンタ/イベント・カウンタ00）の設定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。ここでは、TM00版のみを例として説明します。

【C言語】 (Kx2_Event.c)

```
/*=====
   前処理指令 (#pragma指令)
   =====*/
#pragma sfr ←
#pragma di ←
#pragma ei ←
#pragma nop ←
#pragma interrupt INTTM000 fn_inttm000 ←
```

| |
|---------------------------|
| 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 外部イベント・カウンタの設定

【16ビット・タイマ/イベント・カウンタ00】

外部イベント・カウンタは、次の項目を設定します。

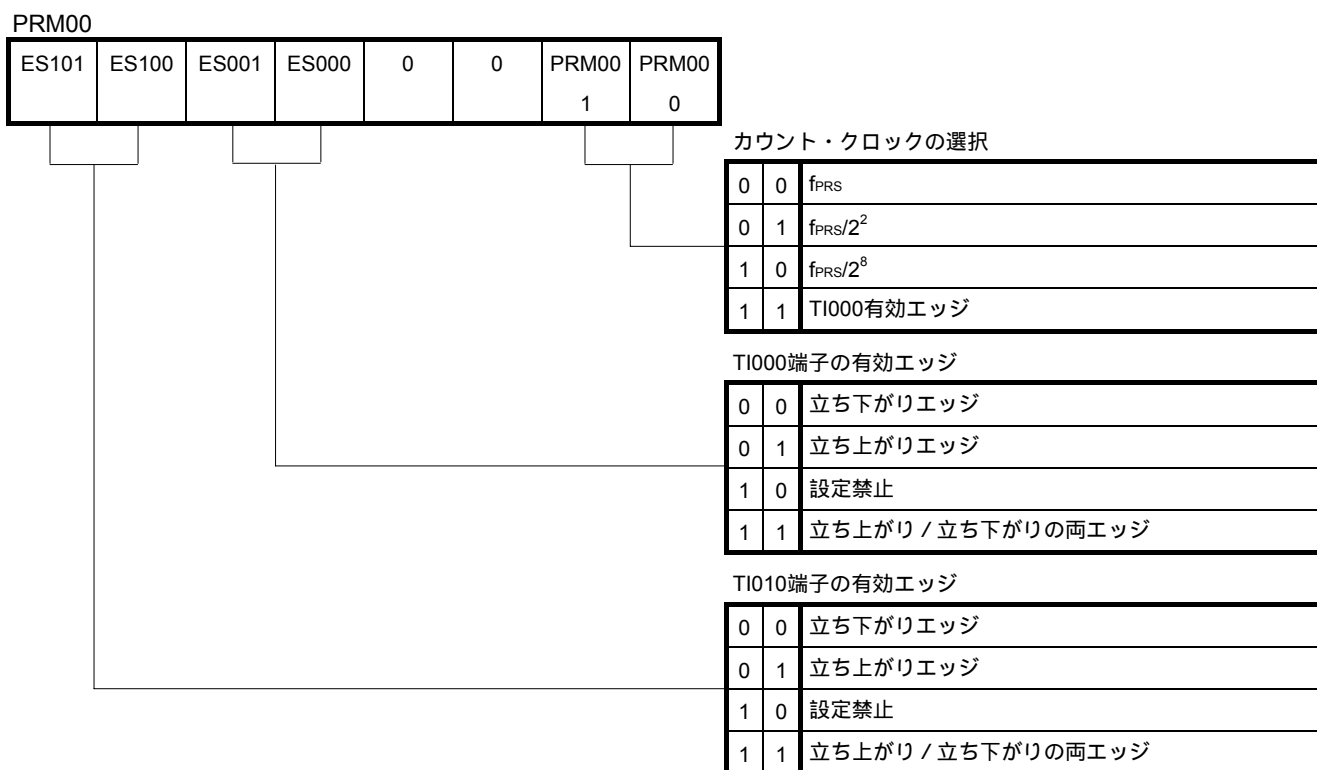
- (1) タイマ・クロックの設定 (TI000の有効エッジ)
- (2) CR000を“コンペア・レジスタとして動作”に設定
- (3) コンペア・レジスタの設定
- (4) タイマ出力の設定 (出力禁止)
- (5) タイマ動作許可

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) カウント・クロックの設定

TM00のカウント・クロック、およびTI000, TI001端子入力の有効エッジを設定します。

図4 - 1 プリスケーラ・モード・レジスタ00



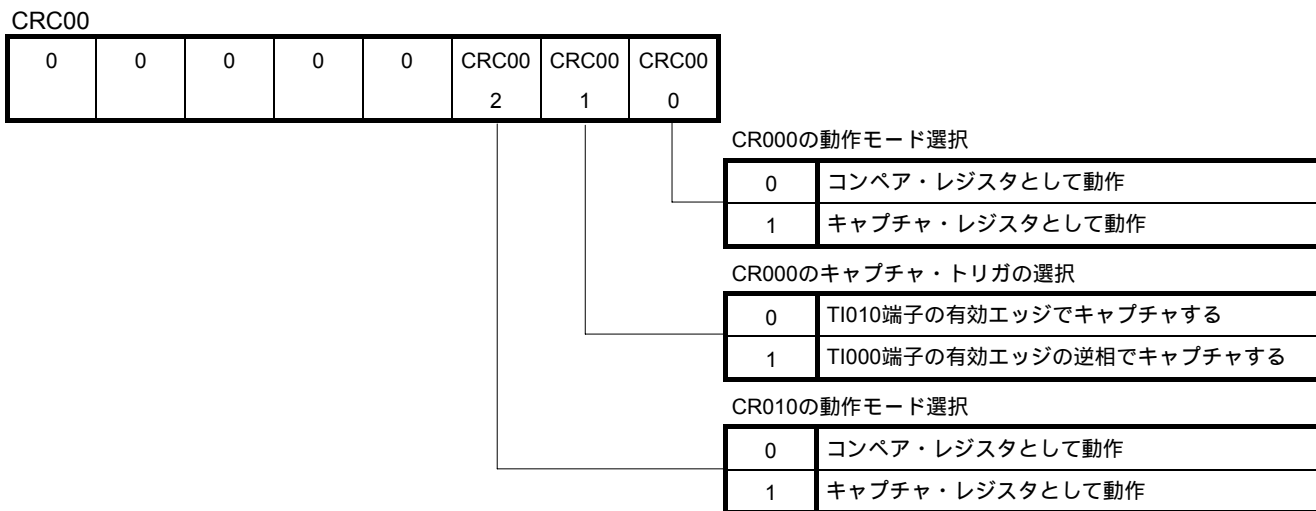
注意 外部クロックには、内部クロック (f_{PRS}) の2周期分より長いパルスが必要です。

備考 f_{PRS} : 周辺ハードウェア・クロック周波数

(2) コンペア・レジスタの制御の設定

CR000, CR001の動作を制御します。

図4 - 2 キャプチャ/コンペア・コントロール・レジスタ00



- 注意1. TI010端子から有効エッジが検出された場合、キャプチャ動作は行われませんが、外部割り込み信号としてINTTM000信号が発生します。
2. TMC003, TMC002 = 11を設定した場合は、CRC000には必ず0を設定してください。

(3) 16ビット・タイマ・キャプチャ/コンペア・レジスタ (CR000/CR010) の設定

コンペア・レジスタとして使用するときにはCR000 (CR010) に設定した値とTM00を常に比較し、一致したときに割り込み信号を発生します。

図4 - 3 16ビット・タイマ・キャプチャ/コンペア・レジスタ000 (010)

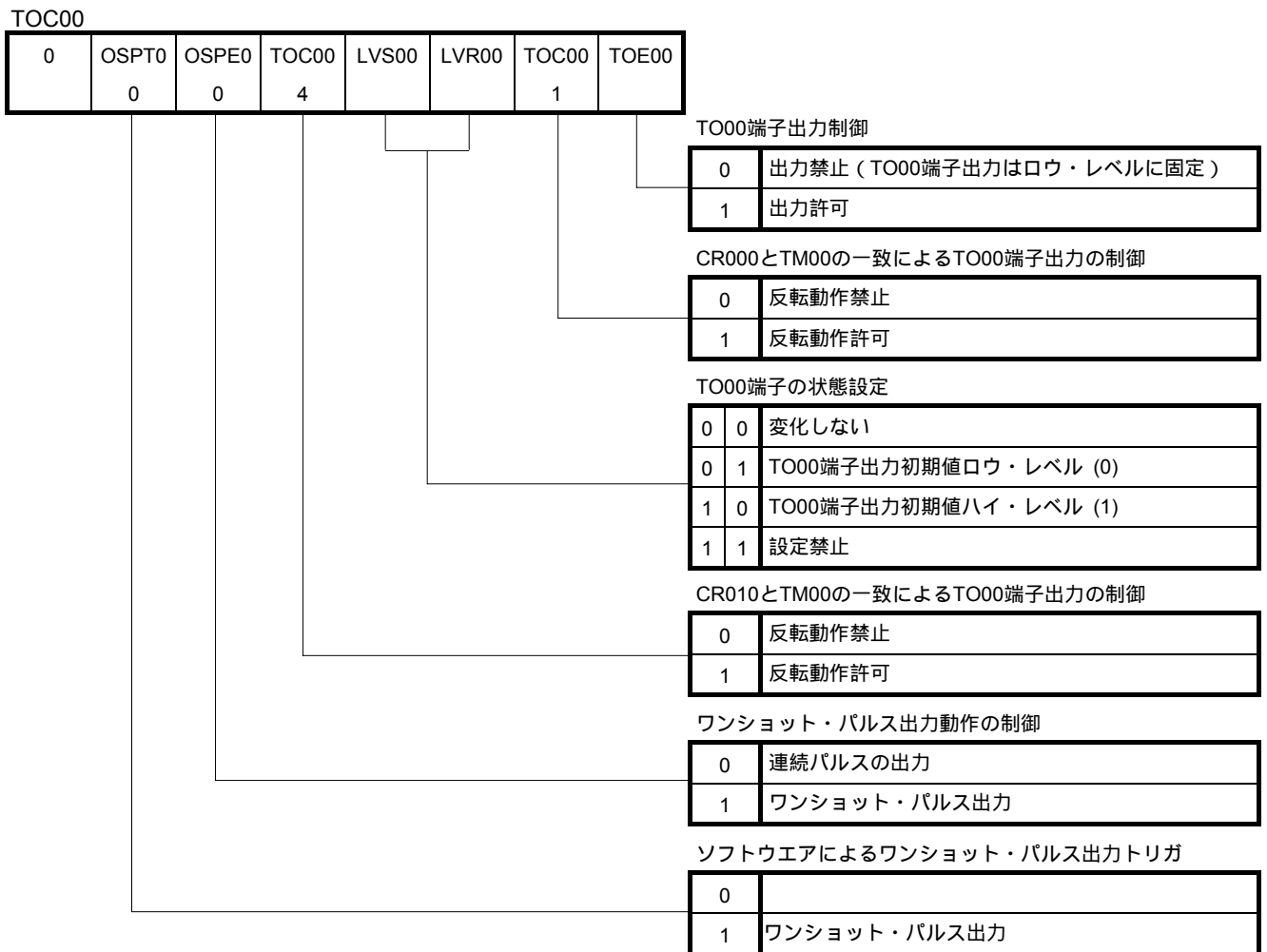


- 注意1. 外部イベント・カウンタとして動作する場合は、CR010は使用しません。
2. 外部イベント・カウンタとして動作する場合は、0000H < CR000 FFFFHの範囲で設定します。

(4) タイマ出力の設定

TOC00端子出力を制御します。

図4 - 4 16ビット・タイマ出力コントロール・レジスタ00

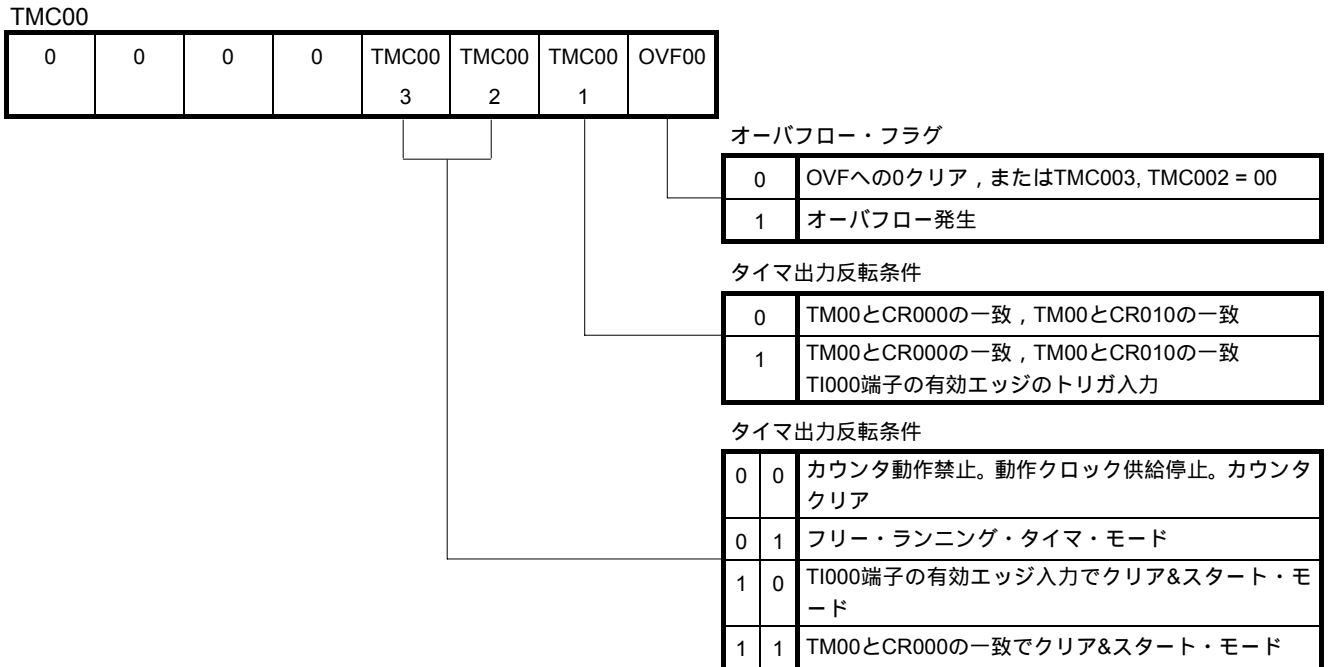


- 注意1. TOC001 = 0でも割り込み信号 (INTTM000) は発生します。
2. TOC004 = 0でも割り込み信号 (INTTM010) は発生します。
3. ワンショット・パルス出力は、フリーランニング・タイマ・モード, TI000端子の有効エッジ入力でクリア & スタート・モードのときに正常に動作します。
4. OSPT00は、リード値は常に0です。ワンショット・パルス出力以外ではセット (1) しないでください。

(5) タイマ・モードの設定

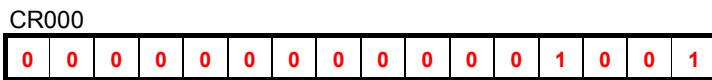
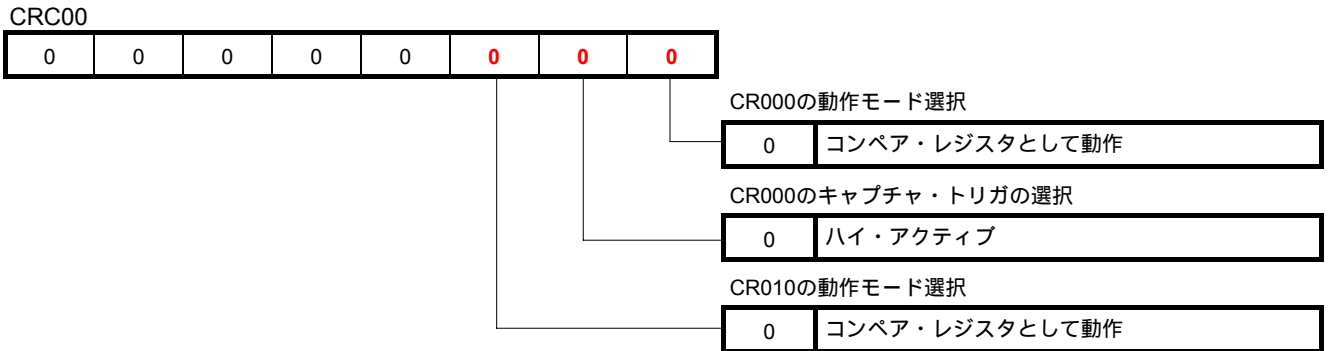
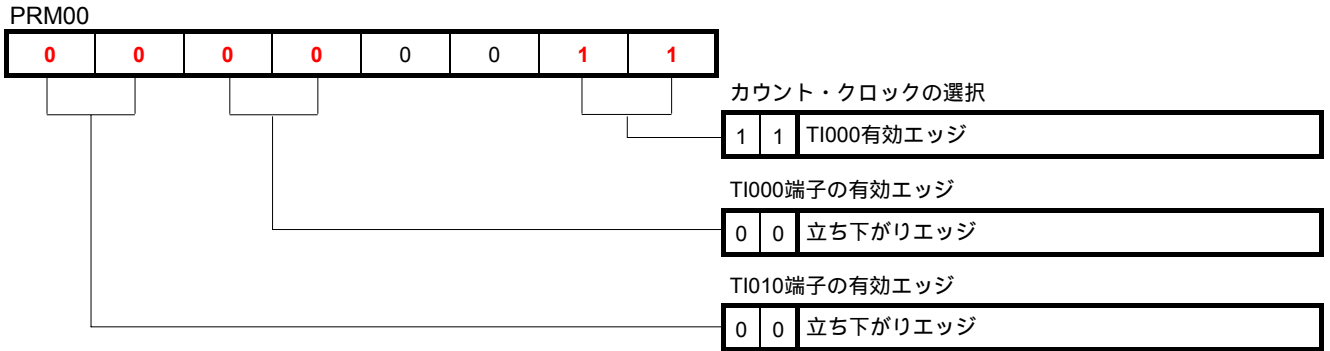
16ビット・カウンタ/イベント・カウンタ00の動作モード，カウンタ制御，出力の状態/制御の設定をします。

図4 - 5 16ビット・タイマ・モード・コントロール・レジスタ00



注意 OVF00はすべての動作モードで，TM00の値がFFFFHから0000Hになるときセット(1)されます。1を書き込むことでもセットできます。

- 【例】
- ・ タイマ・クロックをTI000有効エッジに設定
 - ・ CR000をコンペア・レジスタとして動作に設定
 - ・ コンペア・レジスタ (CR000) の初期設定
 - ・ タイマ出力禁止に設定
 - ・ タイマ動作許可
- (サンプル・プログラムの設定と同内容)



$$\text{INTTM000発生タイミング} = \text{有効エッジ検出回数} \times (9 + 1)$$

↑
CR000設定値

ここでは、9回に設定 10回の外部イベント検出で割り込み発生

TOC00



TO00出力制御

0 出力禁止 (TO00出力は, ロウ・レベルに固定) 許可

CR000とTM00の一致によるTO00の出力の制御

0 反転動作許可

TO00の出力状態の設定

0 0 変化しない TO00出力初期値ハイ・レベル

CR010とTM00の一致によるTO00出力の制御

0 反転動作許可

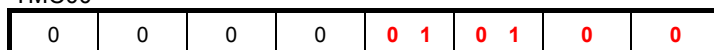
ワンショット・パルス出力動作の制御

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



カウント・クロックの選択

0 クリア

タイマ出力反転条件

0 TM00とCR000の一致、TM00とCR010の一致

タイマ出力反転条件

0 0 カウンタ動作禁止。動作クロック供給停止。カウンタ・クリア
TM00とCR000の一致でクリア&スタート・モード



【コラム】外部イベント・カウンタ機能でのINTTM000発生タイミング

16ビット・タイマ/イベント・カウンタにおいて, 外部イベント・カウンタとして使用する場合, INTTM000発生タイミングが以下のように初回と2回目以降で異なります。

- ・ 初回のINTTM000発生タイミング
= 外部イベント入力の有効エッジ検出回数 × (CR000設定値 + 2)
- ・ 2回目以降のINTTM000発生タイミング
= 外部イベント入力の有効エッジ検出回数 × (CR000設定値 + 1)

INTTM001発生タイミングも上記と同様のタイミングで発生します。

サンプル・プログラムでは以下ようになります。

【C言語】 (Kx2_Event.c)

```

/*=====
;      外部イベント・カウンタ定義
;      外部イベント・カウンタの比較値（割り込み発生タイミング）を定義します。
;      割り込み発生タイミングを変更したい場合には、ここでの定義値を変更
;      してください。
;=====*/
#define C_CntCR      10      /* ここでは10回とする */

PRM00 = 0b00000011; /* [プリースケーラ・モード・レジスタ] */
CRC00 = 0b00000000; /* [キャプチャ/コンペア制御] */
CR000 = C_CntCR-1; /* [キャプチャ/コンペア・レジスタ] */
TOC00 = 0b00000000; /* [タイマ出力制御] */
TMC00 = 0b00001100; /* [モード制御]

```

【アセンブリ言語】 (Kx2_Event.asm)

```

/*=====
;      外部イベント・カウンタ定義
;      アセンブリ言語版でも外部イベント・カウンタの比較値（割り込み発生タイミング）
;      を定義します。
;      割り込み発生タイミングを変更したい場合には、ここでの定義値を変更してください。
;=====*/
CCNTCR EQU 10 ; ここでは10回とする

MOV PRM00, #00000011B ; [プリースケーラモードレジスタ]
MOV CRC00, #00000000B ; [キャプチャ/コンペア制御]
MOVW CR000, #CCNTCR-1 ; [キャプチャ/コンペアレジスタ]
MOV TOC00, #00000000B ; [タイマ出力制御]
MOV TMC00, #00001100B ; [モード制御]

```

【8ビット・タイマ50】

外部イベント・カウンタは、次の項目を設定します。

- (1) タイマ・クロックの設定 (TI50の有効エッジ)
- (2) コンペア・レジスタの設定
- (3) タイマ出力の設定 (出力禁止)
- (4) タイマ動作許可

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・カウンタ/イベント・カウンタ50のカウンタ・クロックおよびTI50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

TCL50

| | | | | | | | |
|---|---|---|---|---|------------|------------|------------|
| 0 | 0 | 0 | 0 | 0 | TCL 502 | TCL 501 | TCL 500 |
|---|---|---|---|---|------------|------------|------------|

カウンタ・クロックの選択

| | | | |
|---|---|---|------------------|
| 0 | 0 | 0 | TI50端子の立ち下がりエッジ |
| 0 | 0 | 1 | TI50端子の立ち上がりエッジ |
| 0 | 1 | 0 | f_{PRS} |
| 0 | 1 | 1 | $f_{PRS}/2$ |
| 1 | 0 | 0 | $f_{PRS}/2^2$ |
| 1 | 0 | 1 | $f_{PRS}/2^6$ |
| 1 | 1 | 0 | $f_{PRS}/2^8$ |
| 1 | 1 | 1 | $f_{PRS}/2^{13}$ |

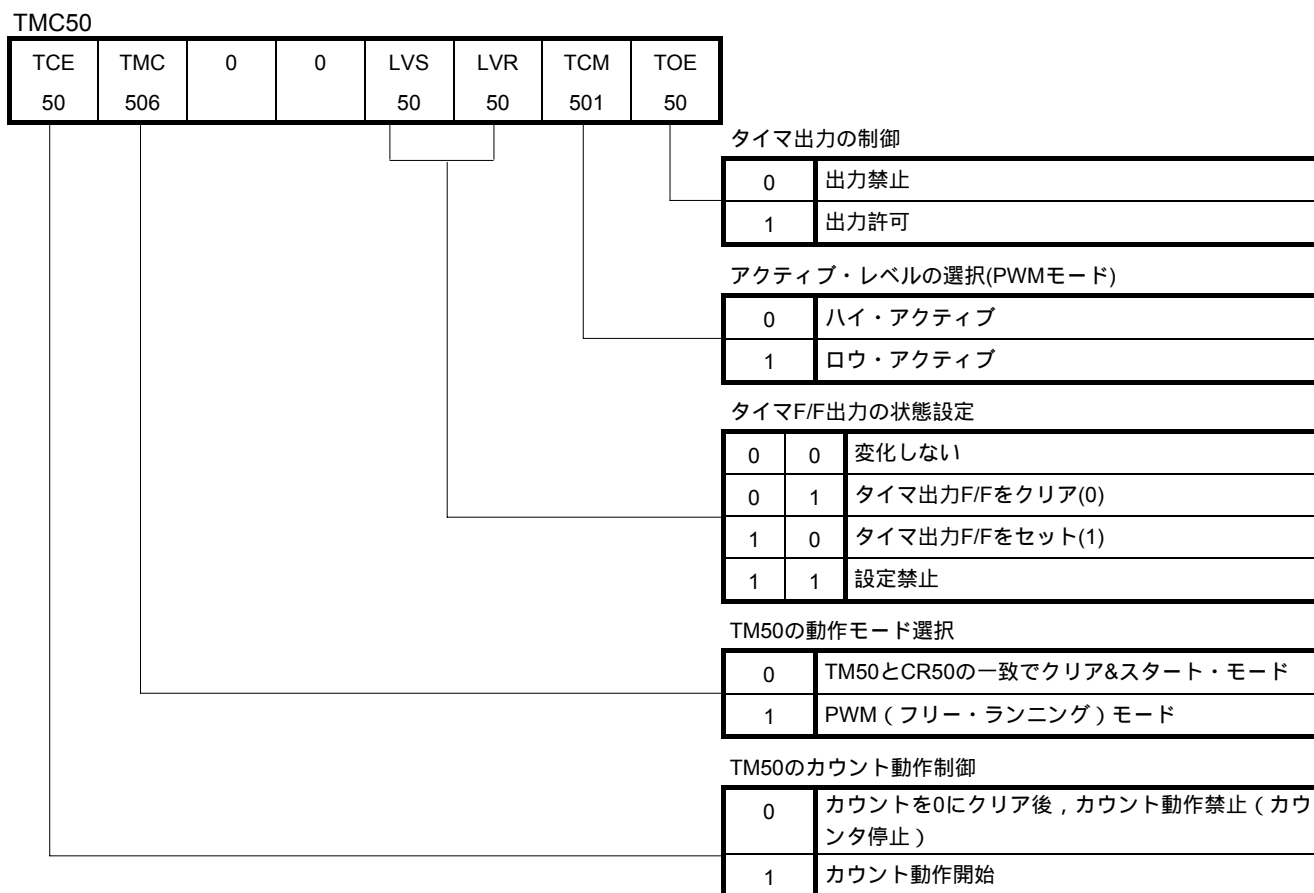
- 注意1. TCL50を同一データ以外に書き換える場合は、いったんタイマ動作を停止させてから書き換えてください。
 2. ビット3-7には、必ず0を設定してください。

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。

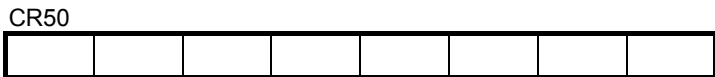
備考1. LVS50とLVR50は読み出すと，0になっています。

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

コンペア・レジスタとして使用するときにはCR50 (CR51) に設定した値とTM50を常に比較し、一致したときに割り込み要求 (INTTM50) を発生します。

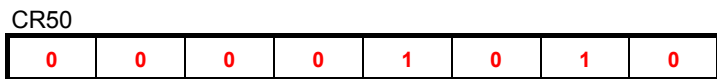
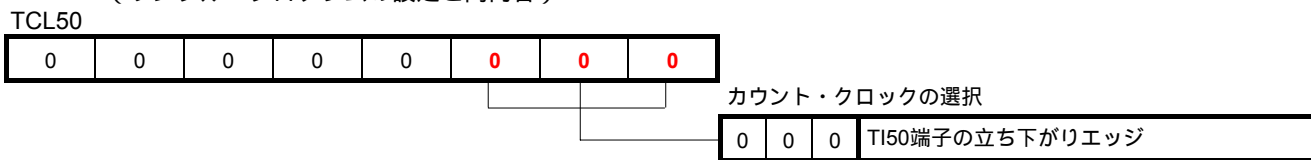
図4-8 8ビット・タイマ・コンペア・レジスタ50



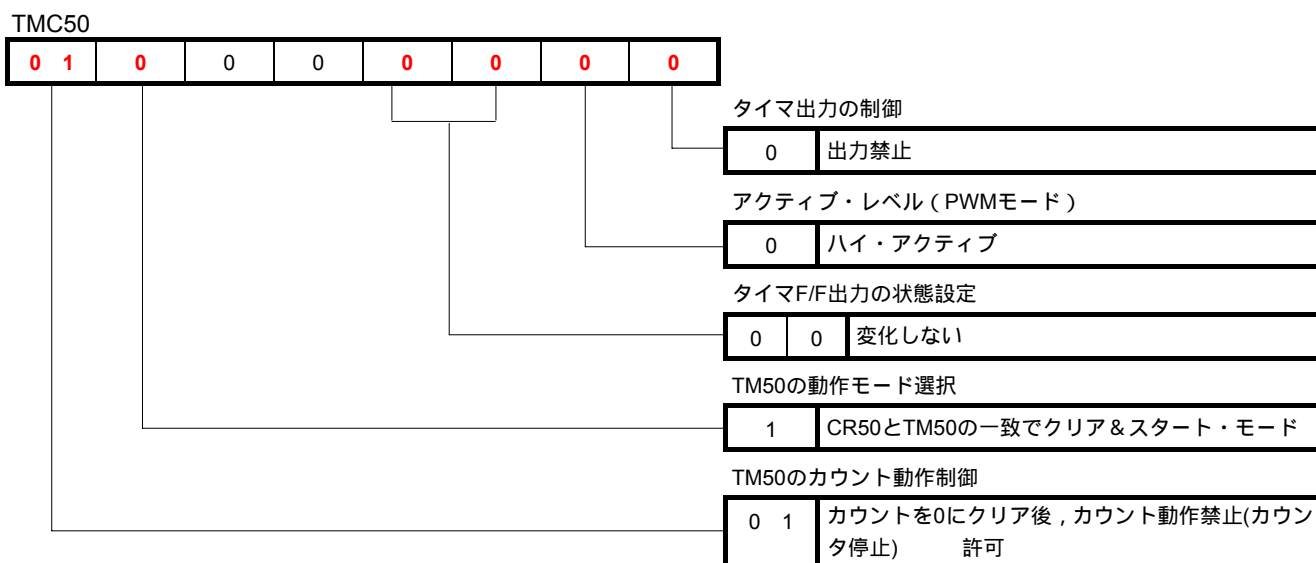
注意1. CR50の値は、00H-FFHの範囲で設定します。

2. TM5nとCR5nの一致でクリア&スタート・モード時は、動作中にCR5n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TI50の立ち下がりエッジに設定
 - ・コンペア・レジスタ (CR50) の初期設定
 - ・タイマ出力禁止に設定
 - ・タイマ動作許可
- (サンプル・プログラムの設定と同内容)



ここでは、10に設定 10回の外部イベント検出で割り込み発生



サンプル・プログラムでは以下のようになります。

【C言語】 (Kx2_Event.c)

```

/*=====
;      外部イベント・カウンタ定義
;
;      外部イベント・カウンタの比較値(割り込み発生タイミング)を定義します。
;      割り込み発生タイミングを変更したい場合には、ここでの定義値を変更
;      してください。
;=====*/

#define C_CntCR      10      /* ここでは10回とする */
.
.
.
.

TCL50   =      0b00000000;   /* [タイマ・クロック選択] */
CR50    =      C_CntCR;     /* [コンペア・レジスタ] */
TMC50   =      0b00000000;   /* [モード制御] */
TCE50   =      1;          /* TMC50レジスタのTCE50をセット
                           カウント動作開始 */
.
.
.

```

【アセンブリ言語】 (Kx2_Event.asm)

```

/*=====
;      外部イベント・カウンタ定義
;
;      アセンブリ言語版でも外部イベント・カウンタの比較値(割り込み発生タイミング)を
;      定義します。
;      割り込み発生タイミングを変更したい場合には、ここでの定義値を変更してください。
;=====*/

CCNTCR      EQU      10      ; ここでは10回とする
.
.
.
.

MOV     TCL50, #00000000B    ; [タイマ・クロック選択]
MOV     CR50,  #CCNTCR      ; [コンペア・レジスタ]
MOV     TMC50, #00000000B    ; [モード制御]
SET1    TCE50                ; TMC50レジスタのTCE50をセット
;      カウント動作開始
.
.
.

```

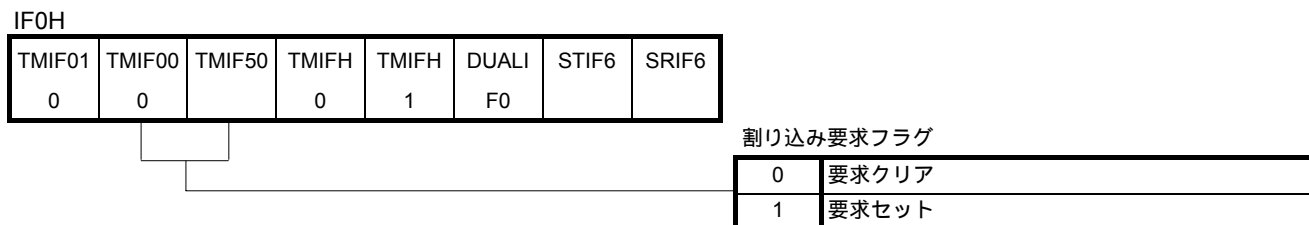

4.3 割り込みの設定

(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】)。

図4 - 9 割り込み要求フラグ・レジスタ (IF0H) のフォーマット

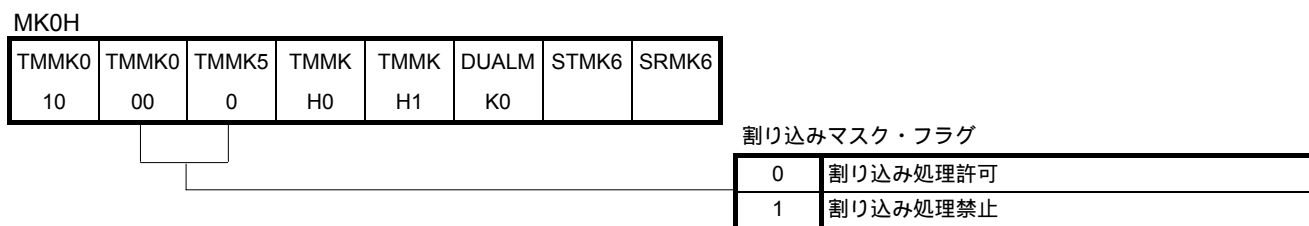


(2) 割り込みマスクの設定

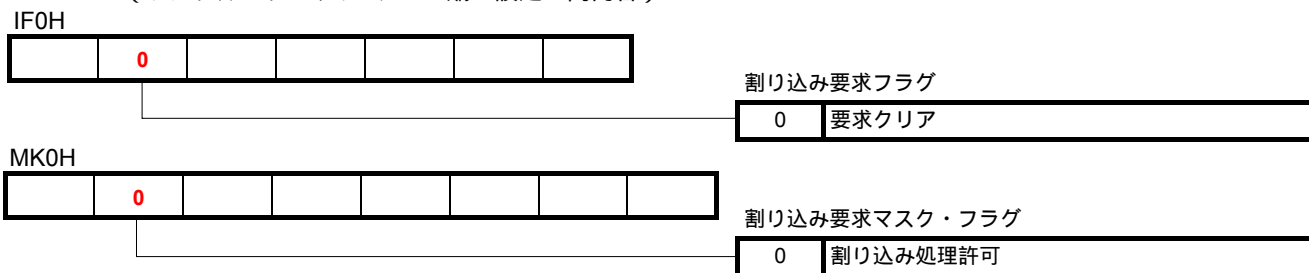
指定の割り込み処理の許可 / 禁止を設定します。

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】)。

図4 - 10 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_Event.c)

```

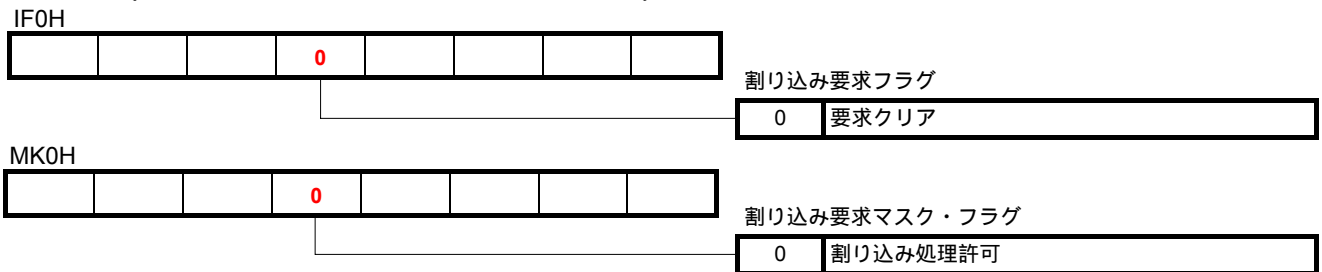
TMIF000 = 0; /* INTTM000割り込み要求クリア */
TMMK000 = 0; /* INTTM000割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Event.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_Event.c)

```

TMIF50 = 0; /* INTTM50割り込み要求クリア */
TMMK50 = 0; /* INTTM50割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Event.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
.
.
.
    
```

4.4 ポートの設定

(1) ポートの出力の設定

このサンプル・プログラムでは外部イベント・カウンタの処理結果を確認するために、P10を出力ポートとして使用し、クロック入力用のTI000およびTI50を入力ポートとして使用します。後述の【例】でTM00版の設定例を示します。TM50版はクロック入力ポートを変更するだけなので、省略します。

図4 - 11 ポート・モード・レジスタ1 (PM1) のフォーマット

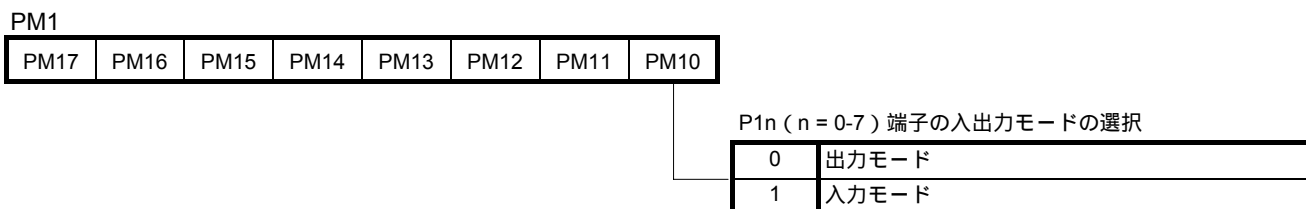
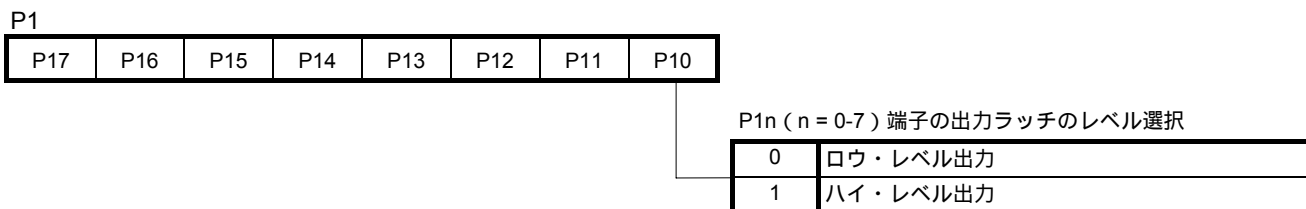
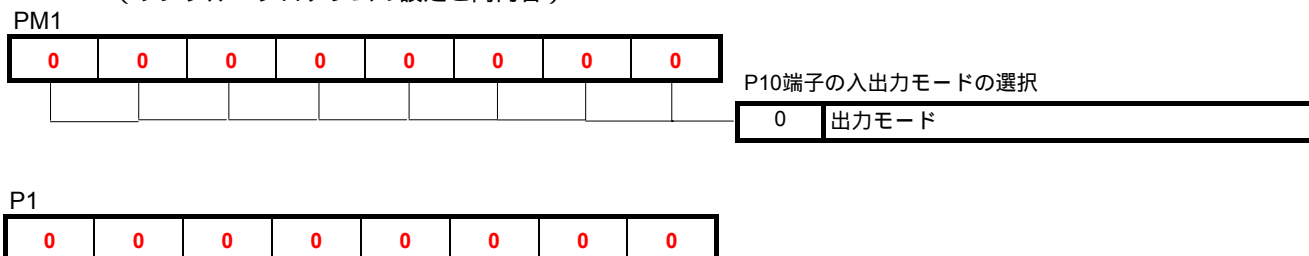


図4 - 12 ポート・レジスタ1 (P1) のフォーマット



- 【例】
- ・ P1を出力ポートに設定
 - ・ P00/TI000を入力ポートに設定
- (サンプル・プログラムの設定と同内容)



【C言語】

```
P1 = 0b00000000;
PM1 = 0b00000000;
P0 = 0b00000001;
```

【アセンブリ言語】

```
MOV P1, #00000000B
MOV PM1, #00000000B
MOV P0, #00000001B
```

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_Event.c)

- ・外部イベント・カウンタ設定関数を呼び出します(タイマ処理を開始します)。
- ・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
;*****/
void main(void){

    fn_InitTimer();          /* 外部イベント・カウンタの初期化 */
    EI();                    /* ベクタ割り込み許可          */

    while (1){
        NOP();
    }
}

```

← 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_Event.asm)

- ・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
;*****/
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

```

← メイン処理内ではNOP命令のみで割り込みを待ちます。

アセンブリ言語版での外部イベント・カウンタの初期化は、メイン処理に入る前に、外部イベント・カウンタの設定、割り込みの設定およびポート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_Event.c)

- ・割り込み発生ごとにP1をインクリメントします。

```

/*****
;      割り込み処理
;*****/
__interrupt void      fn_inttm000(void){

    P1    +=      0b00000001;          /* P1をインクリメント      */
}

```

P1をインクリメントし、P1に接続されたLEDを点灯していきます。

【アセンブリ言語】 (Kx2_Event.asm)

- ・割り込み発生ごとにP1をインクリメントします。


```

;*****/
;      割り込み処理
;*****/
IINTTM000:
    ADD    P1,      #00000001B          ;P1をインクリメント
    RETI


```

P1をインクリメントし、P1に接続されたLEDを点灯していきます。

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

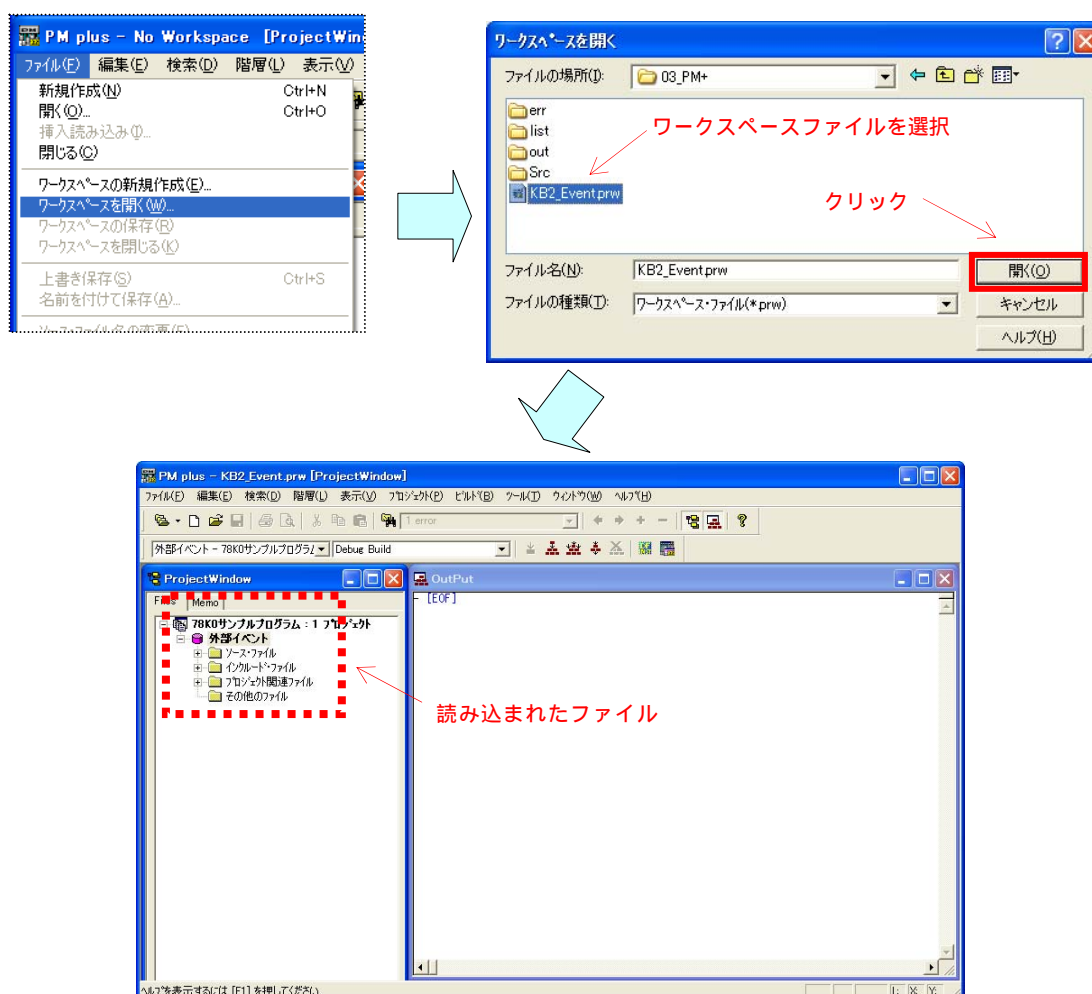
サンプル・プログラムをSM+ for 78K0/Kx2 (以降、「SM+」と表記します)で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、でダウンロードしたC言語用のファイルを用いて、統合開発環境 PM+にてビルドしてから、SM+を起動するまでの動作の一例を説明します。

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。

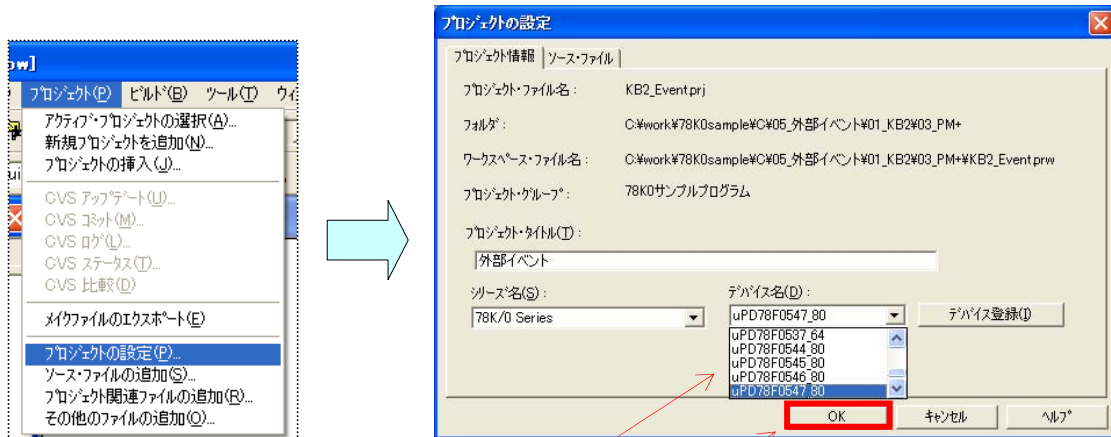
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「Kx2_Event.prw」^注を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。

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_Event.prw」



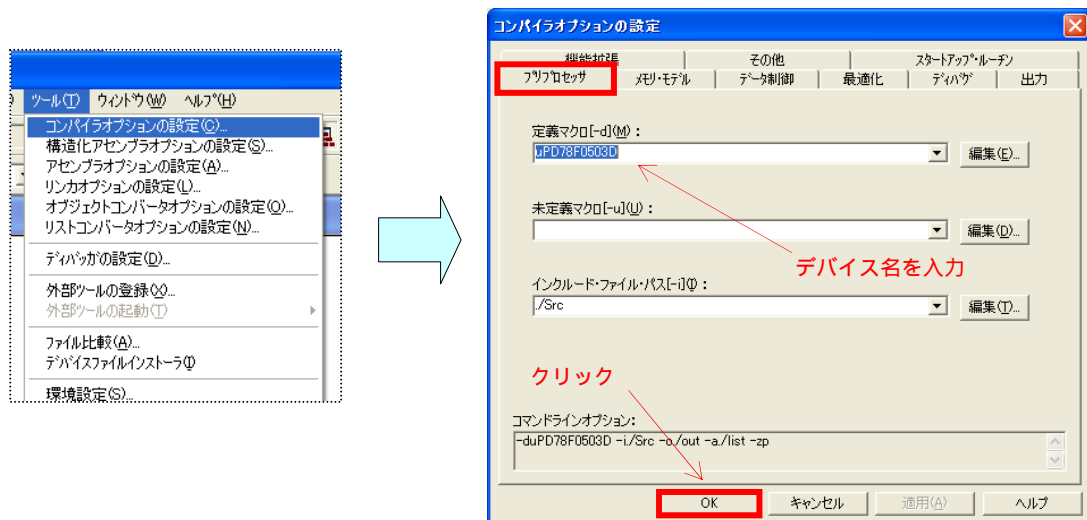
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が表示されたら、使用するデバイス名を選択（デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択）し、[OK] ボタンをクリックしてください。




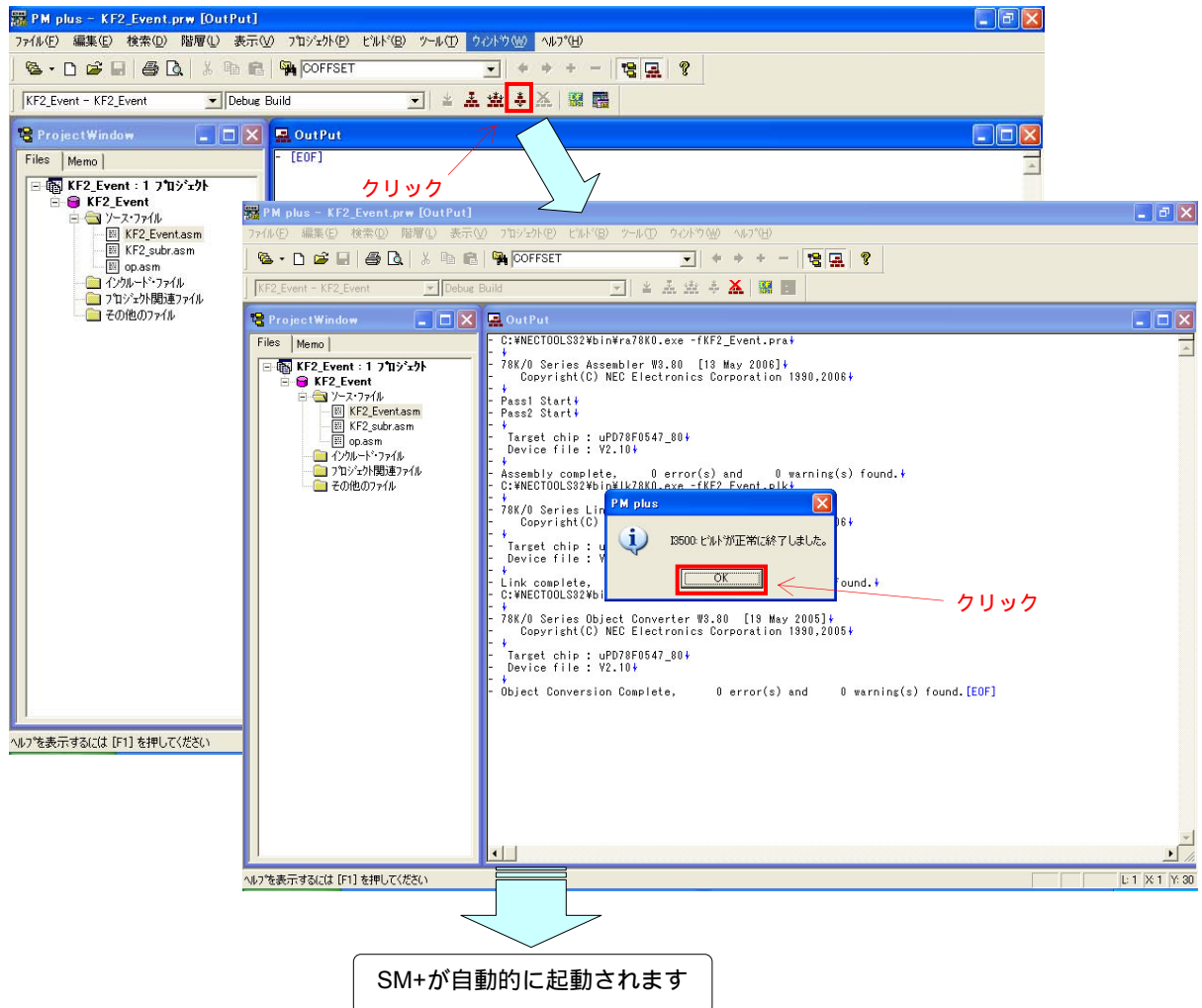
デバイス名を指定 クリック

μPD78F0500_36, μPD78F0501_36, μPD78F0502_36, μPD78F0503_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。入力するデバイス名は、「Kx2_Res.h」ヘッダファイルの先頭部のコメントを確認してください（以下は78K0/KB2の場合）。



- (5)  (「ビルド デバッグ」ボタン)をクリックしてください。ソース・ファイルの「Kx2_Event.c」と「Kx2_Init.c」と「KB2_Mem.sam」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。

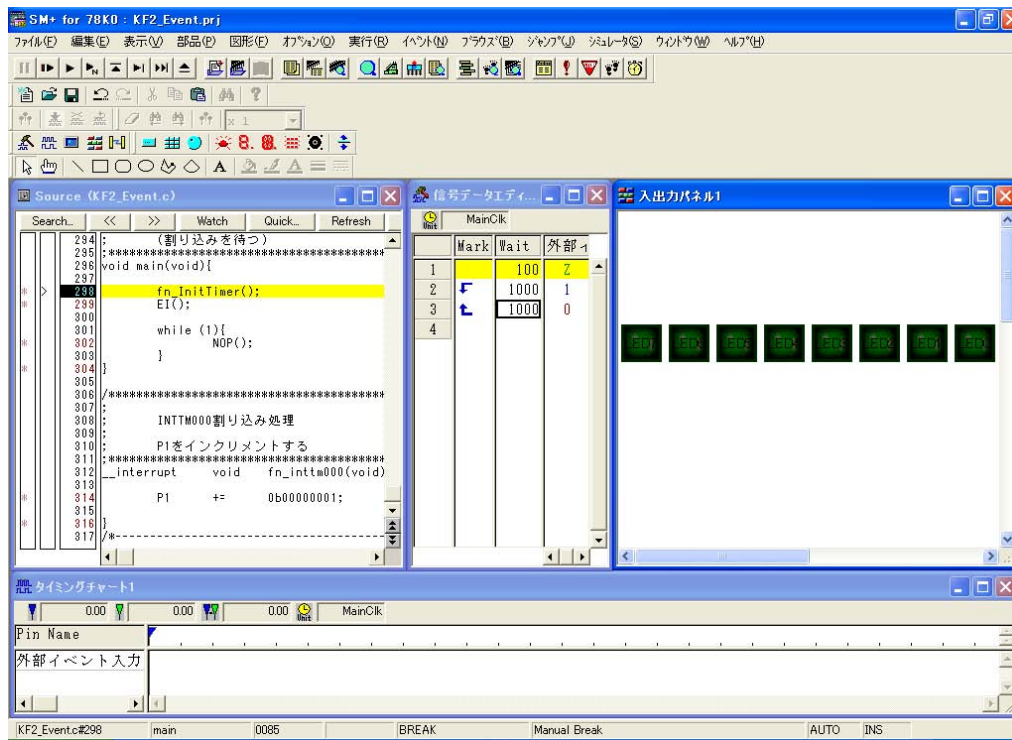



5.2 SM+での動作

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。

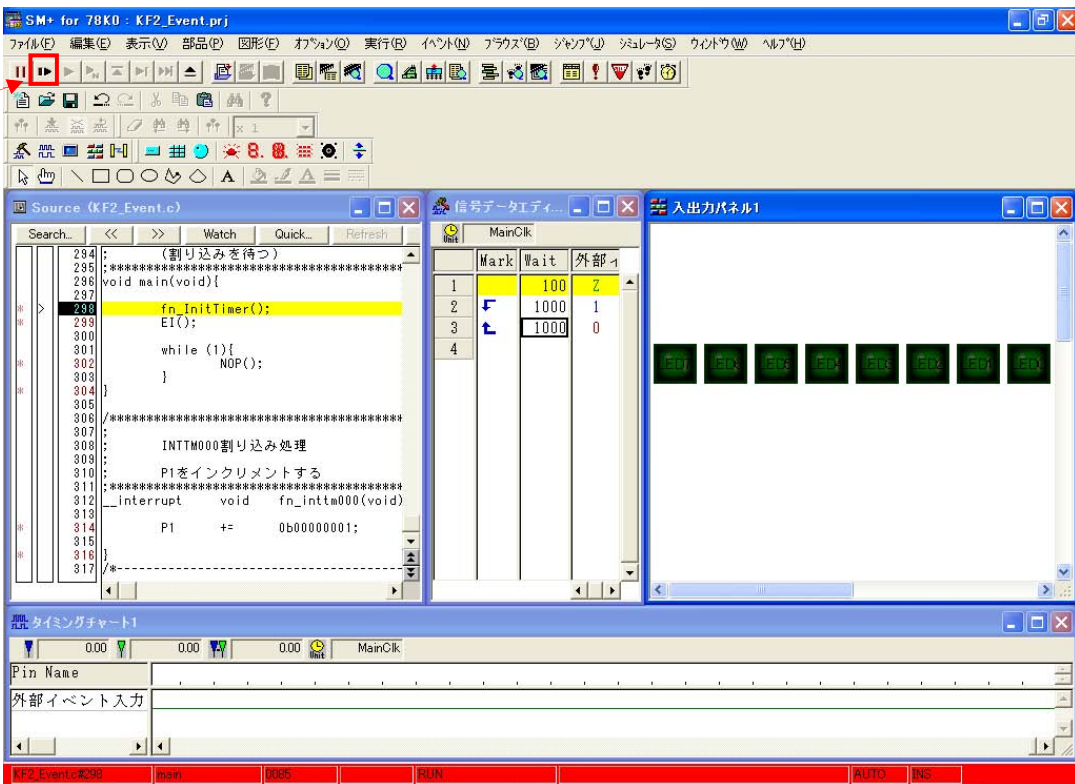
SM+操作方法の詳細については、[SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル](#)を参照してください。

(1) 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



Source (KF2_Event.c)

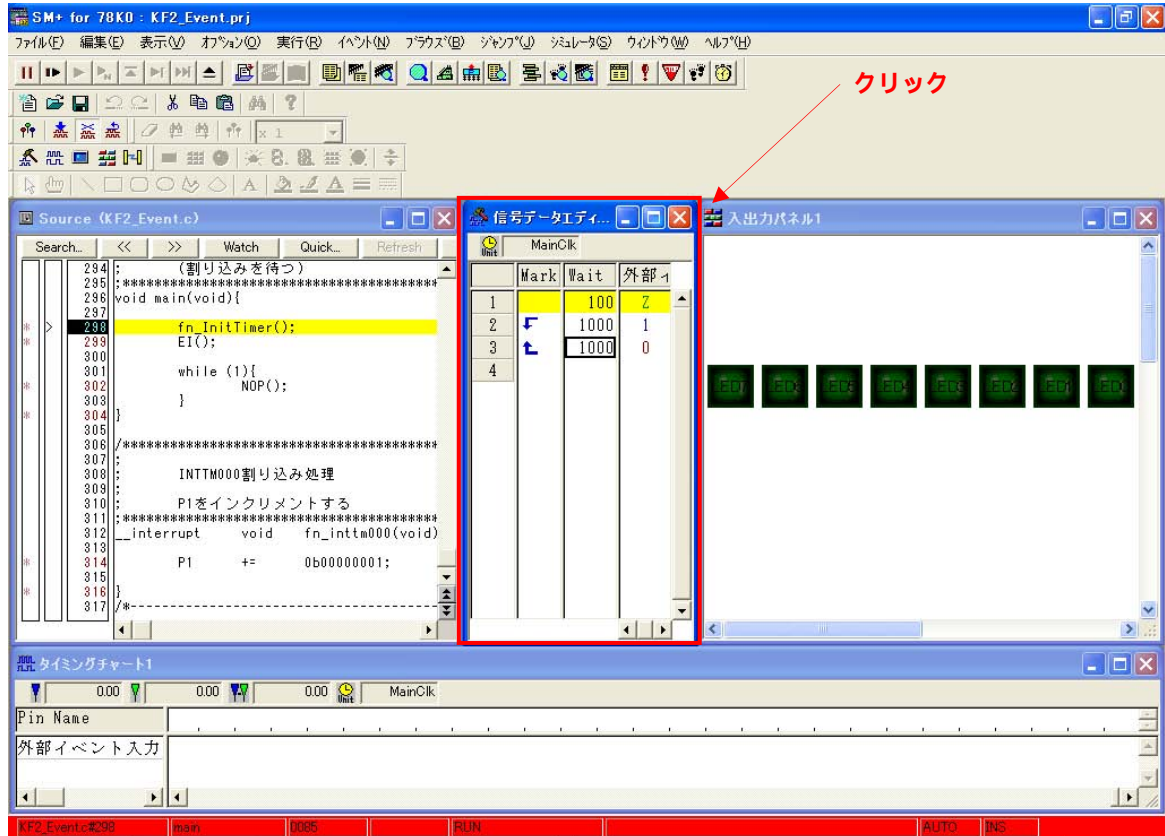
```

294: (割り込みを待つ)
295: *****
296: void main(void){
297:
298:     fn_initTimer();
299:     EI();
300:
301:     while (1){
302:         NOP();
303:     }
304: }
305:
306: /*****
307:
308:     INTTM000割り込み処理
309:
310:     P1をインクリメントする
311:     *****
312:     _interrupt void fn_intm000(void)
313:
314:     P1 += 0b00000001;
315:
316:
317: *****/
    
```

| Mark | Wait | 外部イ |
|------|------|-----|
| 1 | 100 | 2 |
| 2 | 1000 | 1 |
| 3 | 1000 | 0 |
| 4 | | |

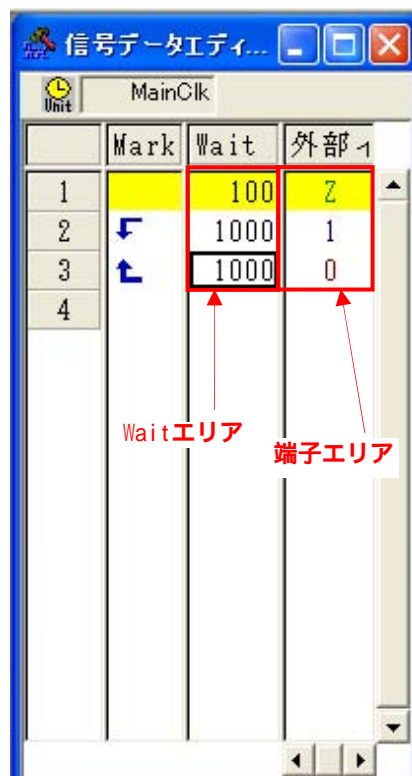
プログラム実行中は、→赤になります。


(3) 信号データ・エディタ・ウィンドウをクリックしてアクティブにしてください。



信号データ・エディタは端子入力をシミュレーションで行うための機能で、「Wait」エリアに設定された時間経過後、「端子」エリアに登録されている端子に「High(1)」、「Low(0)」、「Hi Z(Z)」のいずれかを入力します。

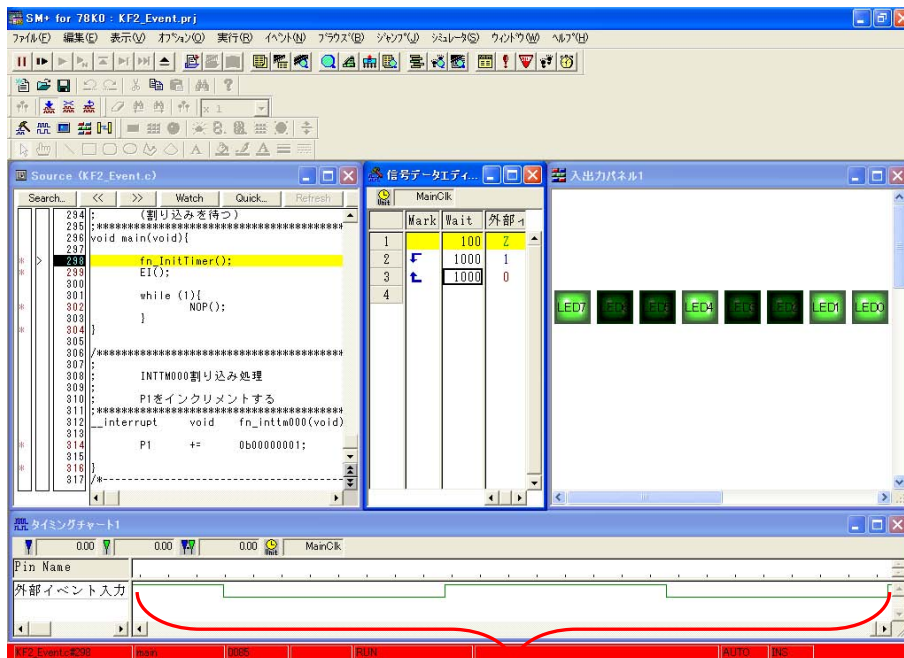
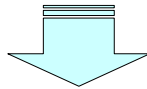
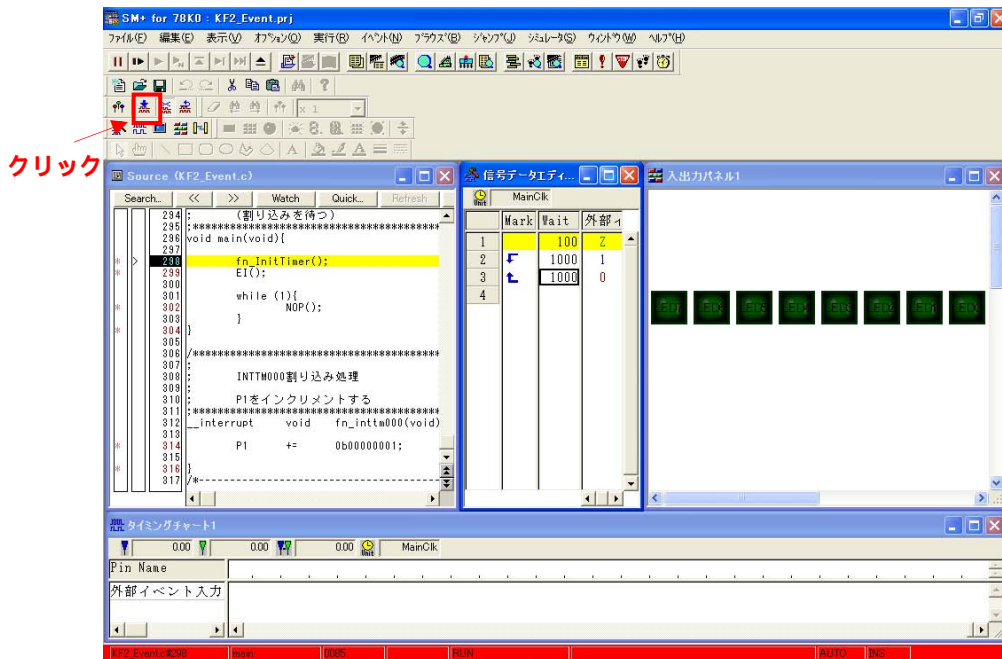
詳細は[SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル](#)を参照してください。



- (4)  (「信号入力の開始」ボタン)をクリックしてください。

信号の入力が開始され、次のような画面になります。

タイミング・チャートの外部イベント入力端子の波形が変化し、信号データ・エディタから信号入力が行われていることがわかります



信号入力が開始されます

割り込み発生ごとにインクリメントされるP1の状態(バイナリ値)に合わせ、LEDが点灯していきます。

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。

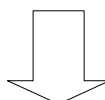
(1) オプション・バイトの設定

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
    
```

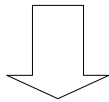
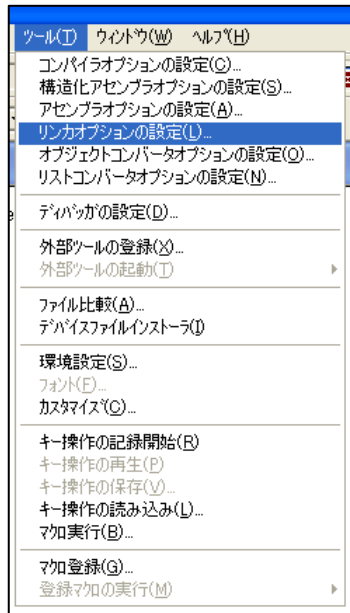
0002H番地はオンチップ・デバッグ使用領域として空けるため、0002H番地はコメント・アウトしてCSEGにより再び0004H番地を定義しています。

C言語版では不要です。

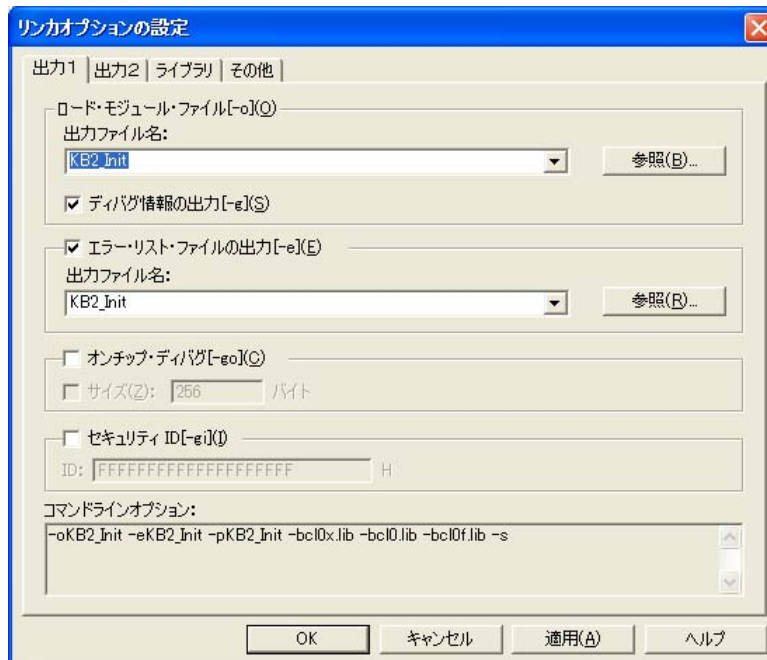
(3) リンカの設定

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。

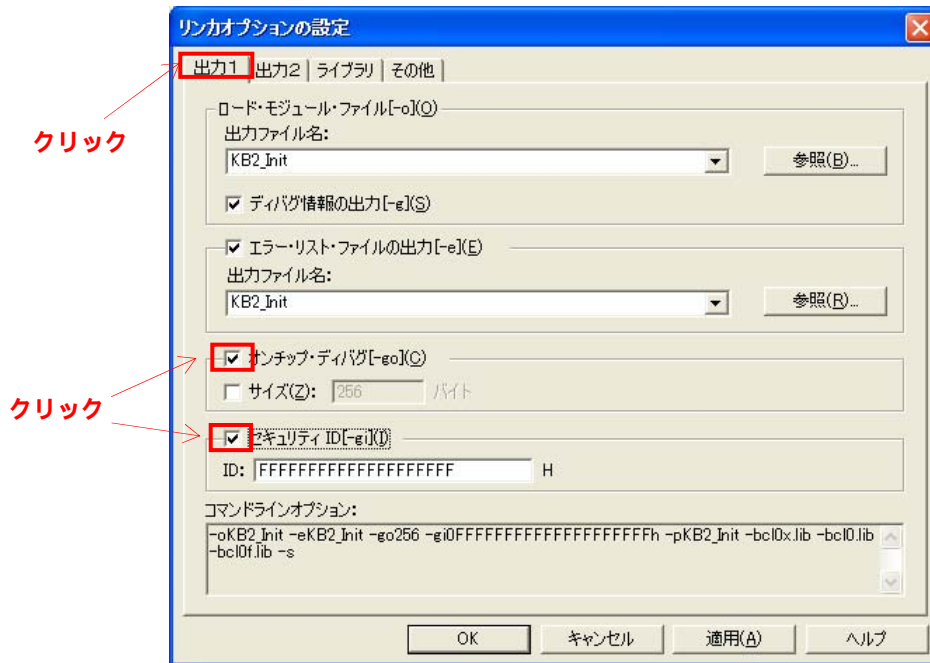
PM+の「ツール」メニューから「リンカオプションの設定」を選択してください。



「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 ユーザーズ・マニュアル | 言語編 | PDF |
| | 操作編 | PDF |
| CC78K0 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 |

付録A 改版履歴

| 版 数 | 発行年月 | 改版箇所 | 改版内容 |
|-----|----------|------|------|
| 第1版 | May 2009 | - | - |

【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
