

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

方形波出力編

この資料は、サンプル・プログラムの「方形波出力」の動作概要と、マイコンの基本的な方形波出力の設定を説明したものです。サンプル・プログラムでは、方形波出力の設定を行ったあとに、その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
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	方形波出力設定と動作概要	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	方形波出力の設定	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	変数・定数の定義	31
4.7	割り込み処理	32
第5章	システム・シミュレータ SM+での動作確認	35
5.1	サンプル・プログラムのビルド	35
5.2	SM+での動作	38
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 U19035JJ1V0AN00 (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いません。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

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

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

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

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

(注)

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

第1章 概 要

このサンプル・プログラムでは、方形波出力の設定を行います。

ここでは、16ビット・タイマ/イベント・カウンタ01 (TM01)、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。

方形波出力は、インターバル・タイマ編のサンプル・プログラム内“インターバル・タイマの初期化”でタイマ出力の設定値を変更し、タイマ出力端子 (TO01 or TO50 or TOH0) の兼用ポートを出力モードに設定するだけで容易に実現できます。本プログラムでは、その設定例として方形波出力の周波数を1 [kHz] ~ 2 [kHz] の範囲で変化させています。

このために、タイマ出力を反転するタイマ・カウンタとコンペア・レジスタとの一致タイミングでの割り込み要求を使用します。割り込み要求は500 [μs] ~ 250 [μs] 周期で発生します。割り込み要求発生から割り込み受付までが10クロック程度で、割り込み処理中で**タイマを停止し、コンペア・レジスタを書き換え、タイマを再起動**するまでの時間が30クロック弱となるのでタイマが再開するまでの時間は40クロック (5 [μs]) となります。これは割り込み要求の発生周期の最大2%となります。これは、高速内蔵クロックの周波数制度よりも小さいので、無視しても構わないレベルと考えることが可能です。また、タイマが再起動するまでの時間のプログラム処理は毎回同じで、割り込み要求の受付タイミングだけが変動するので、この分での変動はさらに小さくなります。

なお、TMH0はタイマ動作を停止させるとタイマ出力もクリアされロウ・レベルになってしまうので、タイマ出力がハイ・レベルの時にタイマ動作を停止させると、正しい方形波が出力されません。そのため、タイマ出力がロウ・レベルの時にタイマ動作を停止させてコンペア・レジスタの設定値を更新させる方法を使用します。その方法として、TMH0版だけ割り込みが2回発生ごとにコンペア・レジスタの設定値を更新します。

TM00では、 $\text{タイマ・クロック} = f_{\text{PRS}}/2^6$ が選択できません。そのため、本サンプル・プログラムでは、TM50編、TMH0編とタイマ・クロックを同様にするため、TM01の設定例を紹介しています。よって、TM01編では78K0/KF2と78K0/KE2のみのサンプル・プログラムを紹介しています (78K0/KD2, 78K0/KC2, 78K0/KB2はTM01非搭載であるため)。

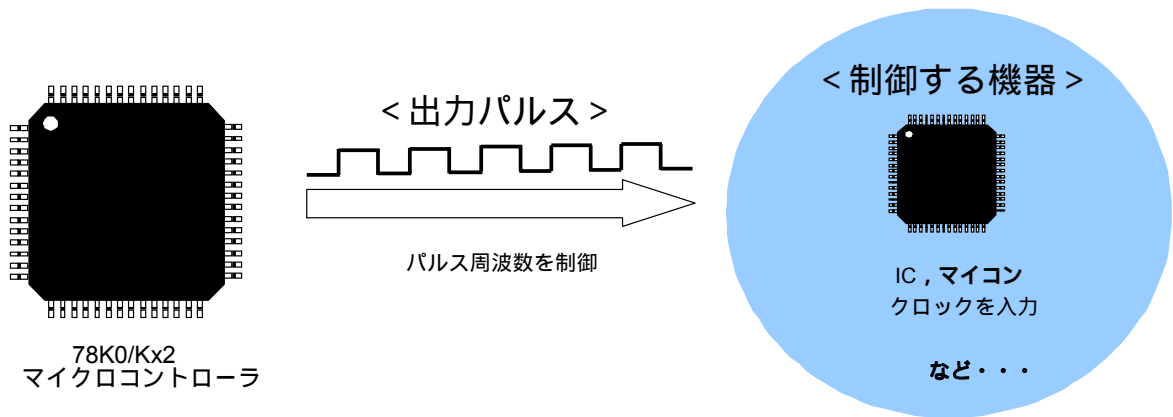
TM00を使用する場合には、使用するタイマ・クロックと出力したい方形波周波数からコンペア・レジスタの設定値を算出し、変更する必要があります。

(1) 方形波出力設定内容

- ・出力ポート (タイマ出力端子) の設定
- ・タイマ・クロックの設定
- ・コンペア・レジスタの設定
- ・タイマ動作許可設定
- ・割り込みの設定

(2) メイン処理動作の内容

- ・ 方形波出力タイマを起動し，NOP命令のみの割り込み待ちループに入ります。



(3) 方形波出力の種類

方形波出力は，タイマを使用して実現します。

方形波出力使用可能デバイス

デバイス		78K0/KB2	78K0/KC2	78K0/KD2	78K0/KE2		78K0/KF2
		(全製品)	(全製品)	(全製品)	μ PD78F0531	μ PD78F0534	(全製品)
タイマ					μ PD78F0532	μ PD78F0535	
					μ PD78F0533	μ PD78F0536	
8ビット・タイマ	H0					μ PD78F0537	
	H1					μ PD78F0537D	
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	-	-	-	-		

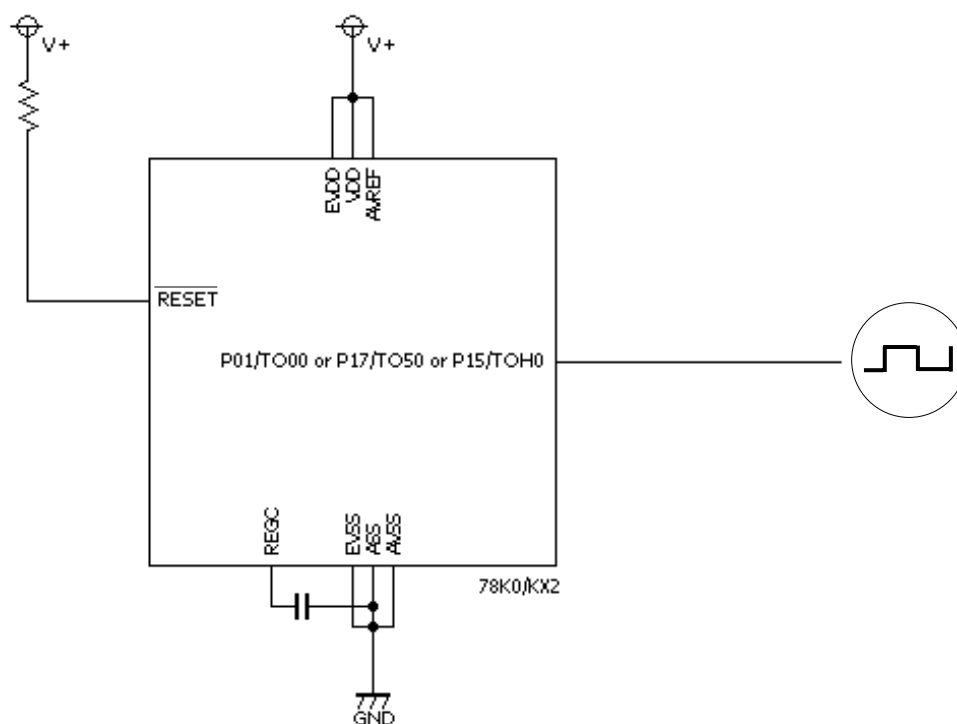
注意 デバイス使用上の注意事項については，[78K0/Kx2 ユーザーズ・マニュアル](#)を参照してください。

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. 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 ($0.47 \sim 1 \mu F$) 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、ユーザーズ・マニュアルを参照してください。




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・プログラムの方形波出力設定と動作概要、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_Square.c	方形波出力の設定、メイン処理のソース・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_Square.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
Kx2_Square.prj	統合開発環境 PM+用プロジェクト・ファイル			
Kx2_Square.pri	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_Square.prs				
Kx2_Square.prm				

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_Square.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_Square.asm	方形波出力の設定，メイン処理のソース・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_Square.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
Kx2_Square.prj	統合開発環境 PM+用プロジェクト・ファイル			
Kx2_Square.pri	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_Square.prs				
Kx2_Square.prm				

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_Square.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01 (TM01)
- ・ 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 (タイマ出力) : P06/TO01 or P17/TO50 or P15/TOH0
- TM00, TM51, TMH1のサンプル・プログラムは、上記TM01, 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TM01, TM50, TMH0との変更点をそれぞれ以下に記載します。

【TM01 TM00への変更点】

- C言語 : 割り込み処理関数宣言をINTTM001からINTTM000に変更
- アセンブリ言語 : 割り込みベクタテーブルを0038H (INTTM001) から0020H (INTTM000) に変更
- C, アセンブリ言語 : “方形波出力の初期化” でのTM01の各レジスタ名称の01を00に変更
- C, アセンブリ言語 : “方形波出力の初期化” でP06とPM06 (TO01) の設定をP01とPM01 (TO00) に変更
- C, アセンブリ言語 : “方形波出力の初期化” 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01を00に変更

注意 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方形波周波数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。

【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 : “方形波出力の初期化” 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 : “方形波出力の初期化” でP17とPM17 (TO50) の設定をP33とPM33 (TO51) に変更
- C, アセンブリ言語 : “方形波出力の初期化” 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】

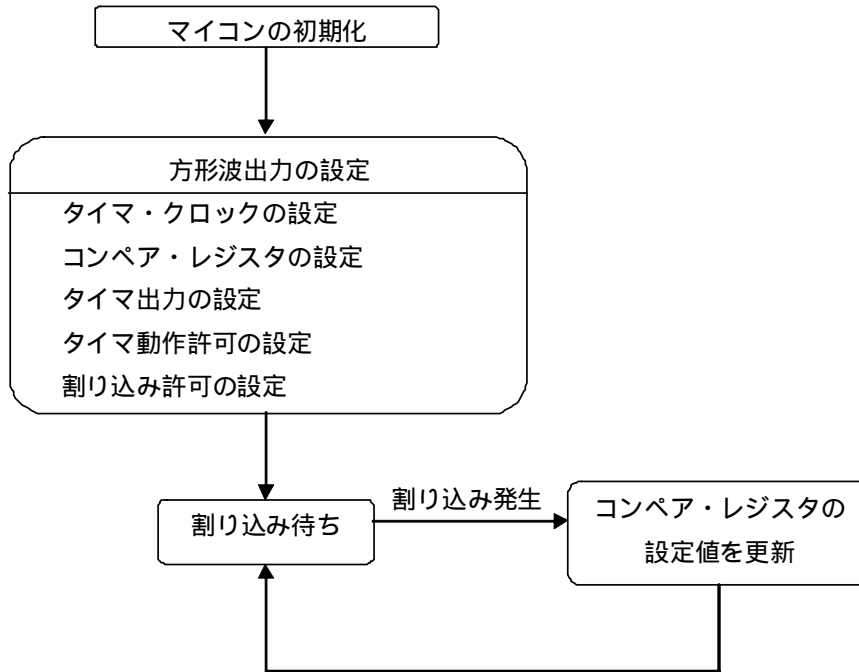
- C言語 : 割り込み処理関数宣言をINTTMH0からINTTMH1に変更
- アセンブリ言語 : 割り込みベクタテーブルを001CH (INTTMH0) から001AH (INTTMH1) に変更
- C, アセンブリ言語 : “方形波出力の初期化” でのTMH0の各レジスタ名称のH0をH1に変更
- C, アセンブリ言語 : “方形波出力の初期化” でP15とPM15 (TOH0) の設定をP16とPM16 (TOH1) に変更
- C, アセンブリ言語 : “方形波出力の初期化” での割り込み要求フラグと割り込みマスク・フラグの名称のH0をH1に変更

3.3 方形波出力設定と動作概要

このサンプル・プログラムでは、方形波出力の設定を行います。

設定完了後は、設定時間に応じてポートへ出力を繰り返します。

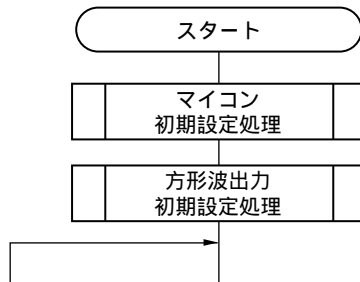
動作概要については、次の状態遷移図（ステート・チャート）に示します。



3.4 フロー・チャート

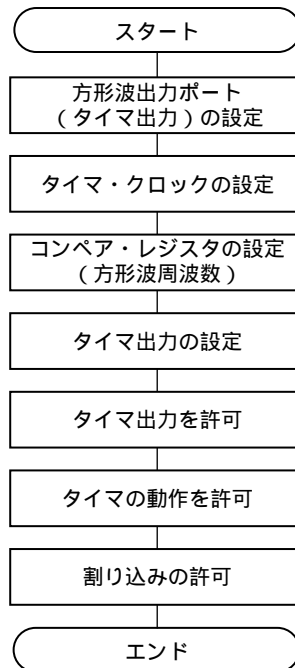
このサンプル・プログラムのフロー・チャートを次に示します。

TM01版, TM50版, TMH0版ゼネラル・フロー (C言語版: Kx2_Square.c アセンブリ言語版: Kx2_Square.asm)

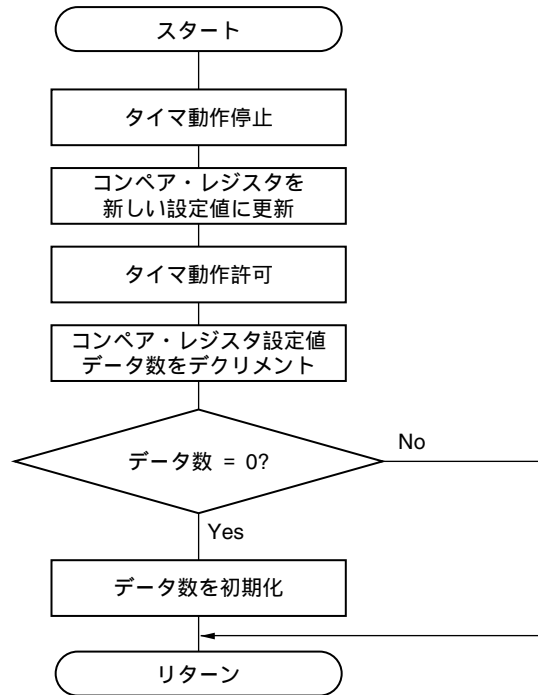


TM01版, TM50版, TMH0版方形波出力初期設定処理

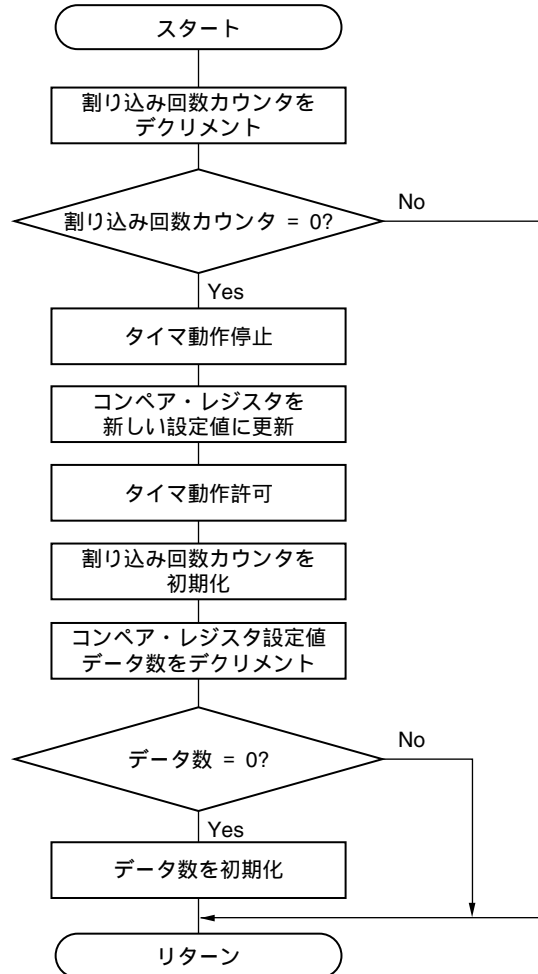
(C言語版: Kx2_Square.c アセンブリ言語版: Kx2_Square.asm)



TM01版, TM50版割り込み処理 (C言語版: Kx2_Square.c アセンブリ言語版: Kx2_Square.asm)



TMH0版割り込み処理 (C言語版: Kx2_Square.c アセンブリ言語版: Kx2_Square.asm)



第4章 設定方法について

この章では、方形波出力 (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。ここでは, TM01版のみを例として紹介します。

【C言語】 (Kx2_Square.c)

```
/*=====
   前処理指令 (#pragma指令)
   =====*/
#pragma sfr <-----
#pragma di <-----
#pragma ei <-----
#pragma nop <-----
#pragma interrupt INTTM001 fn_intTimerSquare <-----
```

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 方形波出力の設定

【16ビット・タイマ/イベント・カウンタ01】

方形波出力は、次の項目を設定します。

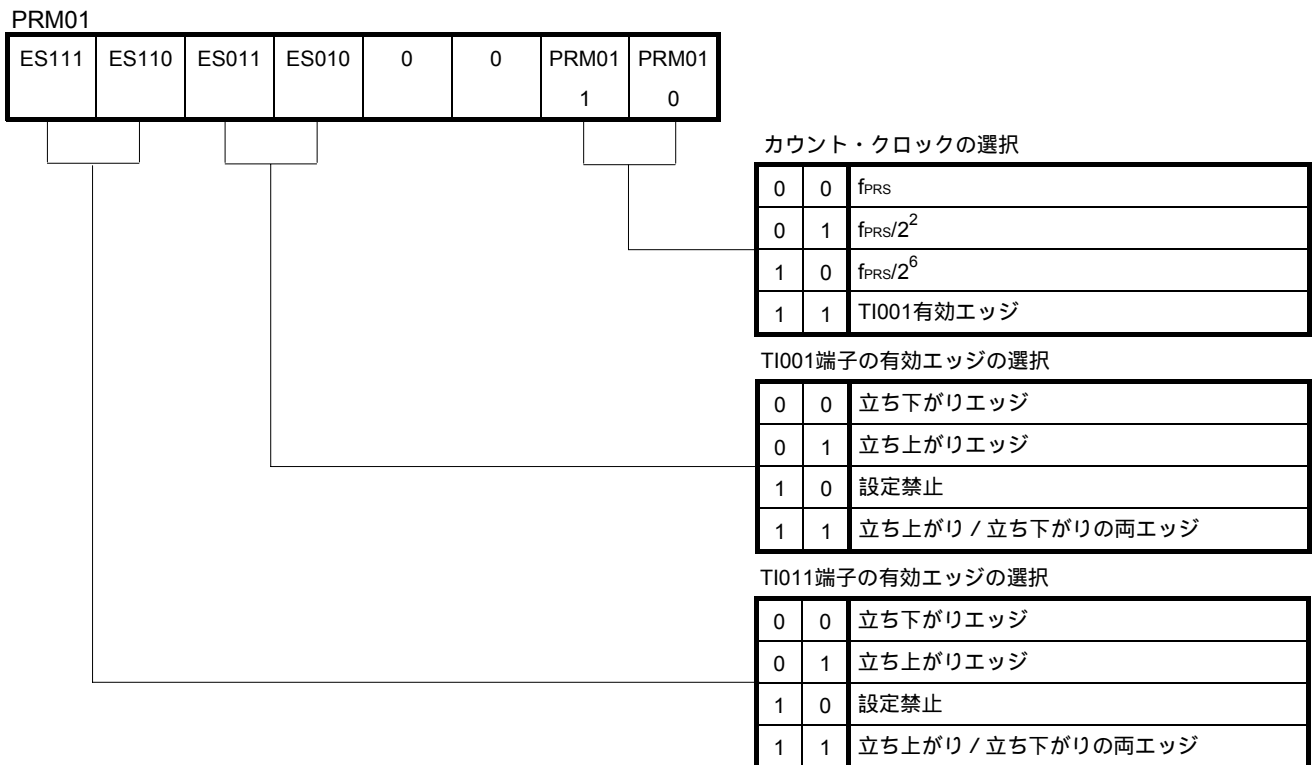
- (1) タイマ・クロック選択
- (2) タイマ・モードの設定
- (3) タイマ・コンペア・レジスタの設定
- (4) タイマ出力の設定
- (5) タイマ動作許可、タイマ出力反転条件の設定

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01)

16ビット・タイマ/イベント・カウンタ01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01の設定

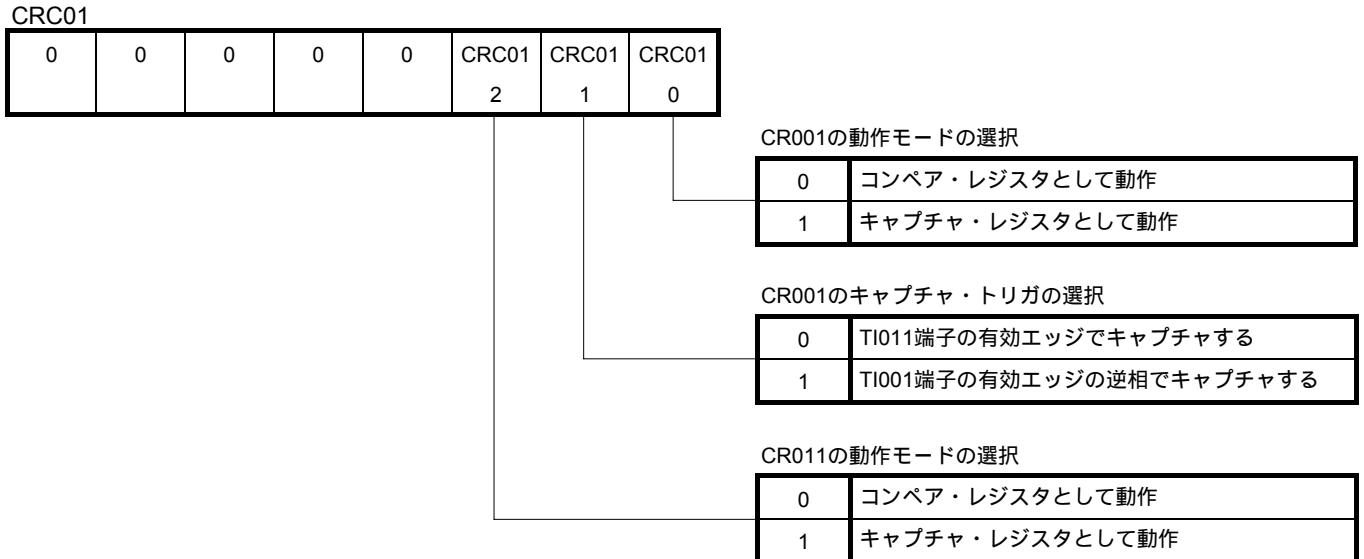


備考 f_{PRS} : 周辺ハードウェア・クロック周波数

(2) タイマ・モード設定

16ビット・タイマ/イベント・カウンタ01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01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01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001

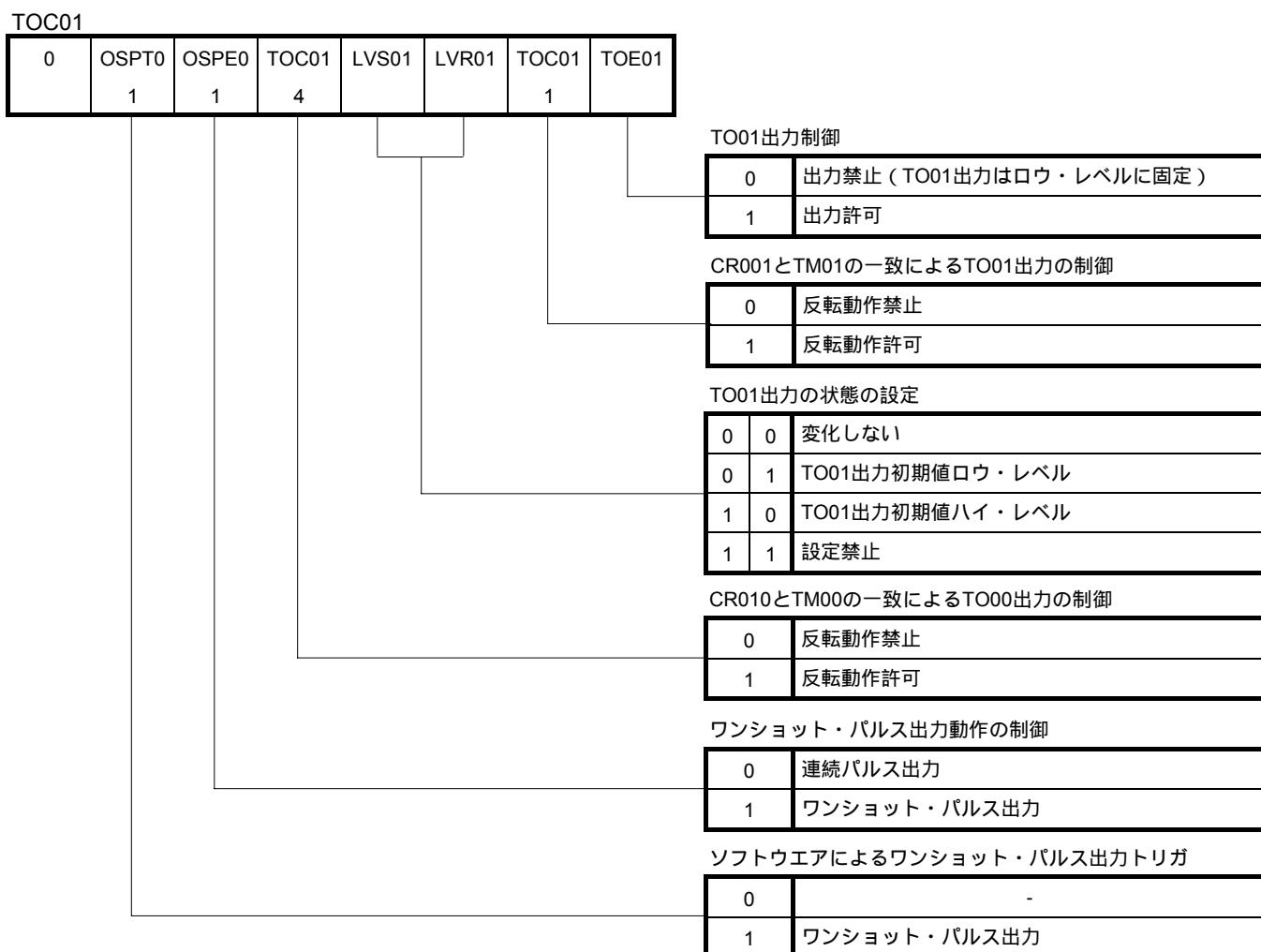


注意 CR001はタイマ停止中 (TMC013, TMC012 = 00) に書き換えを行ってください。

(4) タイマ出力の設定

16ビット・タイマ/イベント・カウンタ01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01



注意 TOC01を設定するときは、必ず次の手順で設定してください。

TOC014, TOC011のセット (1)

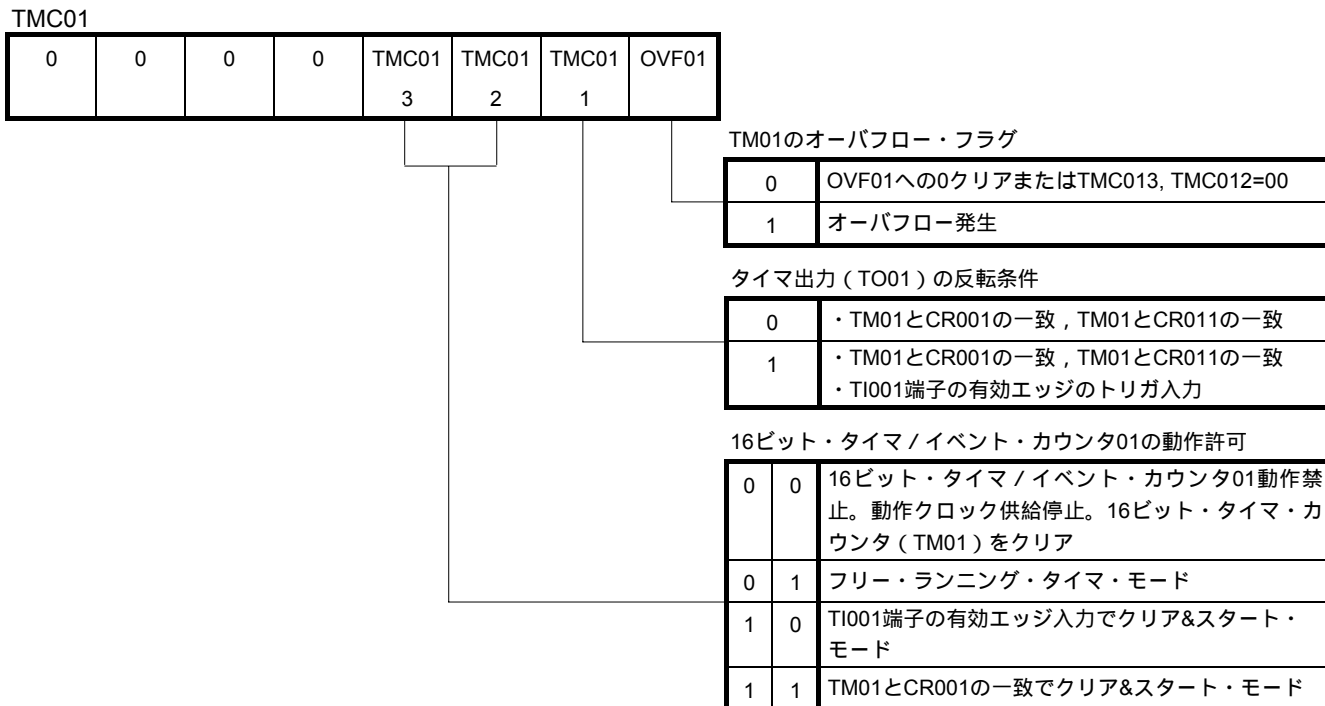
TOE01だけを単独でセット (1)

LVS01またはLVR01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

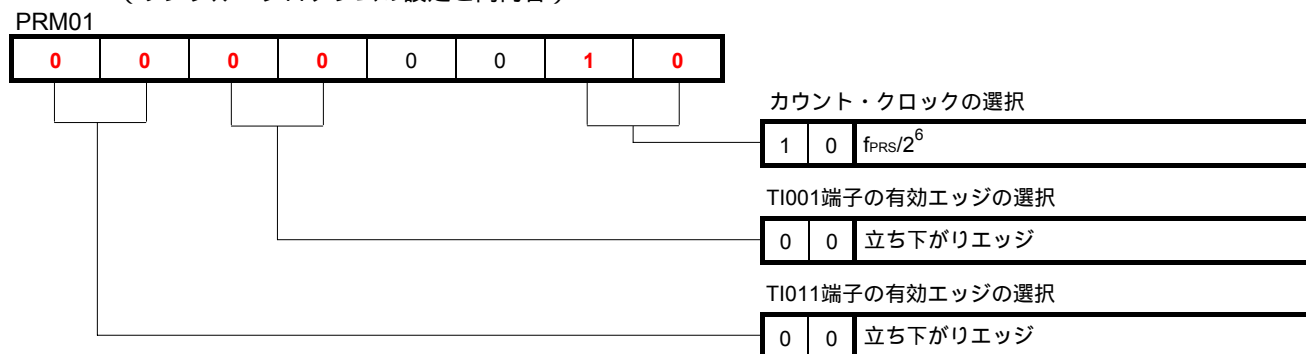
16ビット・カウンタ/イベント・カウンタ01のカウンタ・クロックの動作許可, タイマ出力反転条件の設定をします。

図4-5 16ビット・タイマ・モード・コントロール・レジスタ01



注意 16ビット・タイマ/イベント・カウンタ01は, TMC013, TMC012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, TMC013, TMC012に00を設定してください。

- 【例】
- ・タイマ・クロックを $f_{PRS}/2^6$ に設定
 - ・コンペア・レジスタに初期値0000Hを設定 (サンプル・プログラムの設定と同内容)



CRC01

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

CR001の動作モードの選択

0 コンペア・レジスタとして動作

CR001のキャプチャ・トリガの選択

0 TI011端子の有効エッジでキャプチャする

CR011の動作モードの選択

0 コンペア・レジスタとして動作

CR001

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

下記の計算式よりコンペア値を計算し、方形波周波数を初期値1 [kHz] に設定します。

$$\text{コンペア・レジスタ設定値} = (\text{タイマ・クロック周波数} / \text{方形波周波数}) / 2$$

TOC01

0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---

TO01出力制御

0 1 出力禁止 出力許可

CR001とTM01の一致によるTO01の出力の制御

1 反転動作許可

TO01の出力状態の設定

0 0 変化しない

CR011とTM01の一致によるTO01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 -

TMC01

0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---

TM01のオーバーフロー・フラグ

0 OVF01への0クリアまたはTMC013, TMC012=00

タイマ出力 (TO01) の反転条件

0 TM01とCR001の一致, TM01とCR011の一致

16ビット・タイマ/イベント・カウンタ01の動作許可

1 1 TM01とCR001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_Square.c)

```
PRM01 = 0b00000010; /* タイマ・クロックをfPRS/2^6に設定 */
CRC01 = 0b00000000; /* CR001をコンペア・レジスタとして動作に設定 */
CR001 = 62; /* コンペア・レジスタに初期値を設定 */
TOC01 = 0b00000010; /* タイマ出力反転許可 */
TOE01 = 1; /* タイマ出力動作許可 */
TMC01 = 0b00001100; /* タイマの動作許可 */
.
.
.
```

【アセンブリ言語】 (Kx2_Square.asm)

```
MOV PRM01, #00000010B ;タイマ・クロックをfPRS/2^6に設定
MOV CRC01, #00000000B ;CR001をコンペア・レジスタとして動作に設定
MOVW CR001, #62 ;コンペア・レジスタに初期値を設定
MOV TOC01, #00000010B ;タイマ出力反転許可
SET1 TOE01 ;タイマ出力動作許可
MOV TMC01, #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】

方形波出力は、次の項目を設定します。

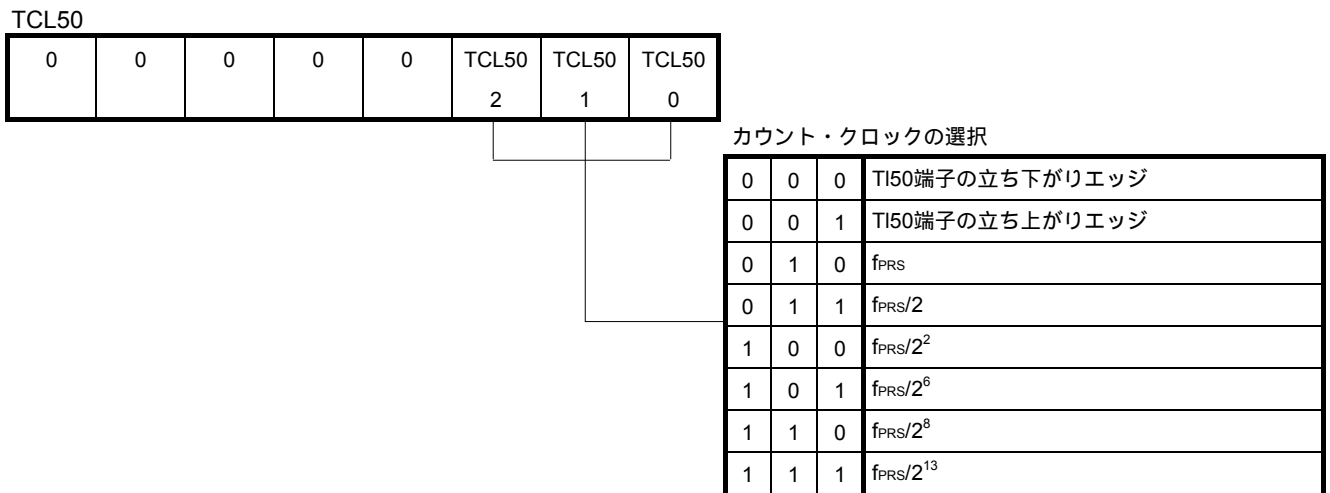
- (1) タイマ・クロック選択
- (2) タイマ・コンペア・レジスタ (CR50) の設定
- (3) タイマ出力の設定 (出力許可)
- (4) タイマ動作許可

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・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



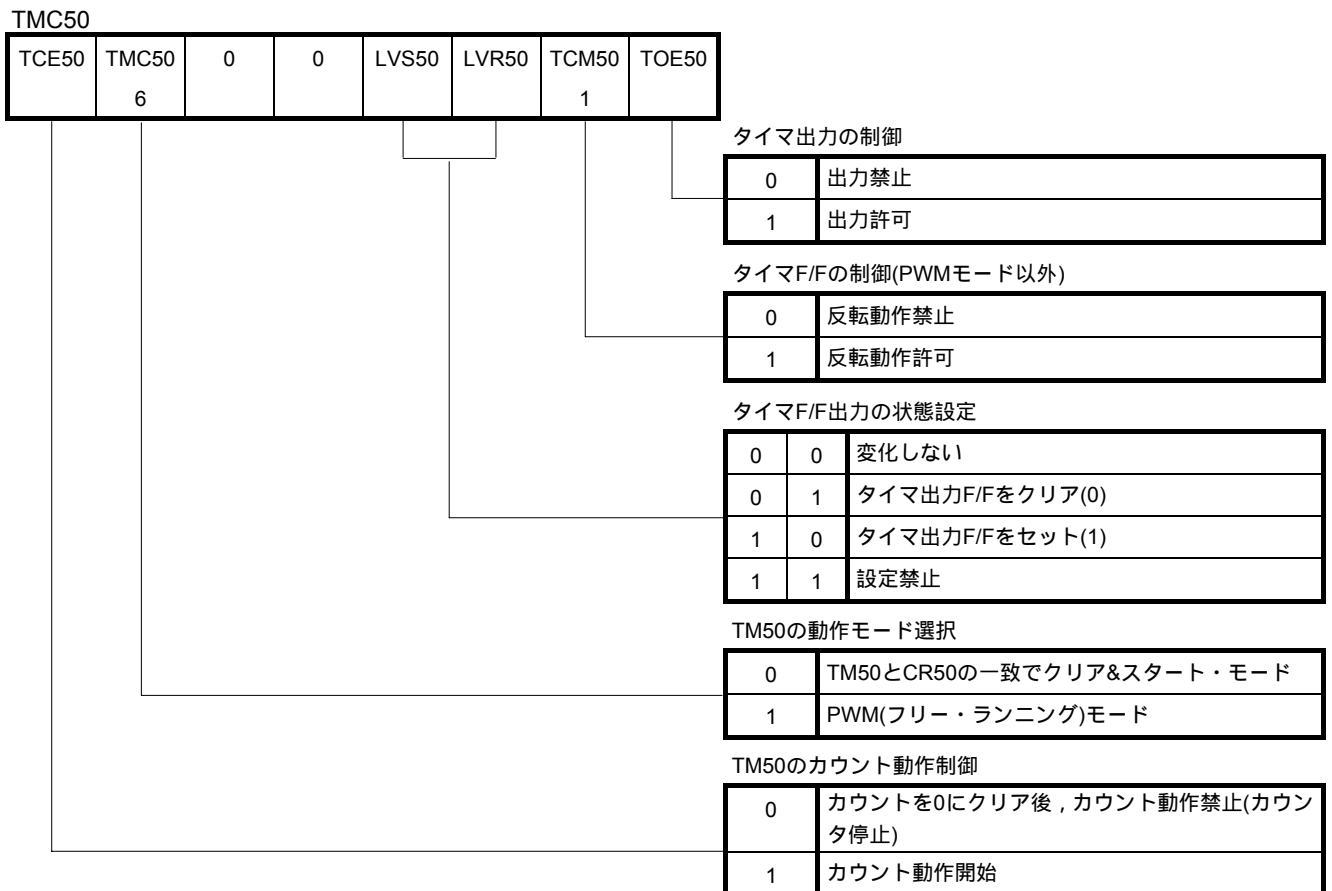
- 注意1. TCL50を同一データ以外に書き換える場合は、いったんタイマ動作を停止させてから書き換えてください。
2. ビット3-7には、必ず0を設定してください。

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。

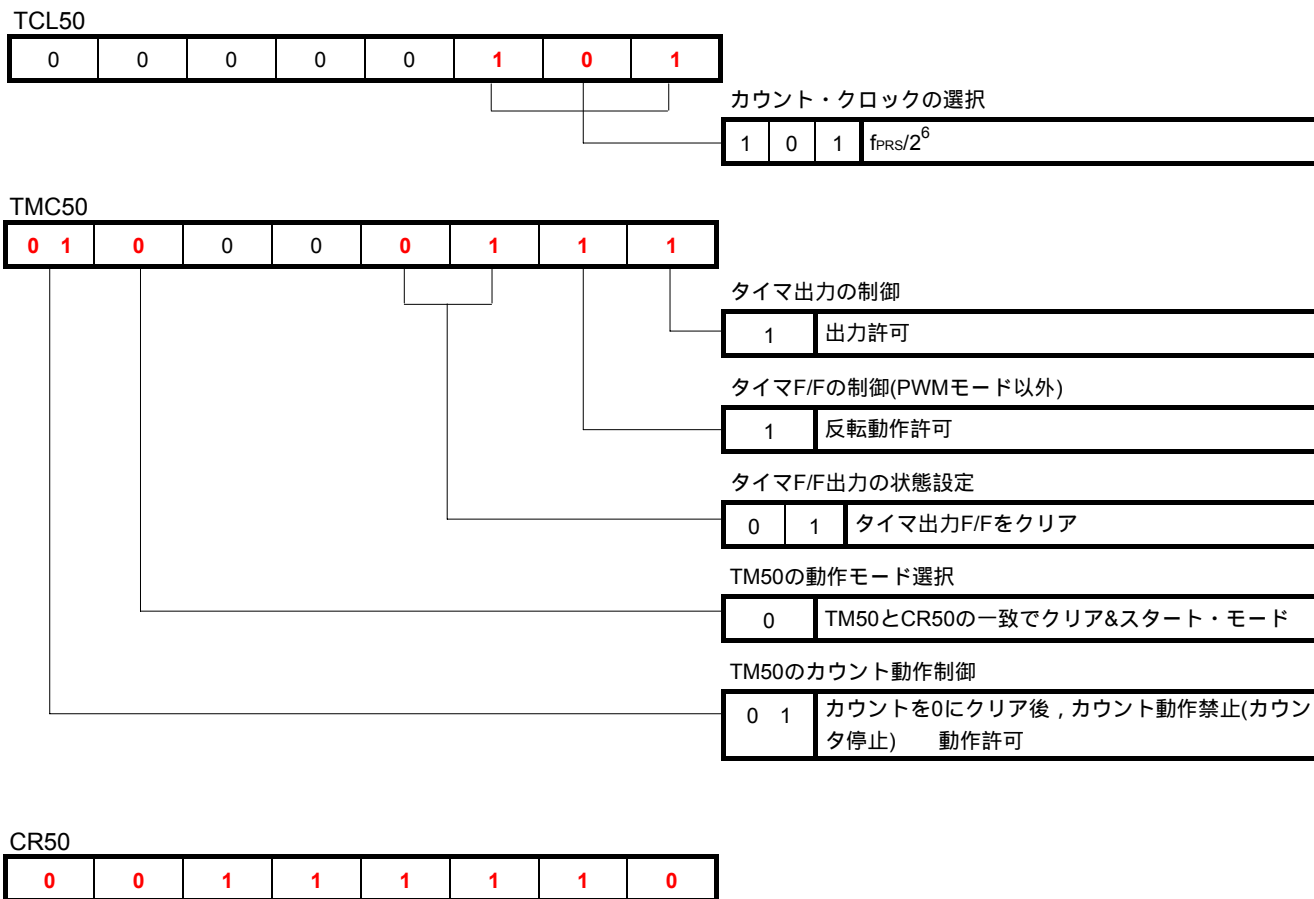
図4-8 8ビット・タイマ・コンペア・レジスタ50



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 $f_{PRS}/2^6$ に設定
 - ・コンペア・レジスタに00Hを設定
- (サンプル・プログラムの設定と同内容)



下記の計算式よりコンペア値を計算し、方形波周波数を初期値1 [kHz] に設定します。

$$\text{コンペア・レジスタ設定値} = (\text{タイマ・クロック周波数} / \text{方形波周波数}) / 2$$

【C言語】 (Kx2_Square.c)

```

TCL50 = 0b00000101; /* タイマ・クロックをfPRS/2^6に設定 */
CR50  = 62;         /* コンペア・レジスタに初期値を設定 */
TMC50 = 0b00000010;
TMC50 = 0b00000001;
TMC50 = 0b00000111; /* タイマ出力設定（許可） */
TCE50 = 1;         /* タイマの動作許可 */
      .
      .
      .
    
```

TMC50の設定手順です。

【アセンブリ言語】 (Kx2_Square.asm)

```

MOV    TCL50, #00000101B ;タイマ・クロックをfPRS/2^6に設定
MOV    CR50,  #62        ;コンペア・レジスタに初期値を設定
MOV    TMC50, #00000010B
MOV    TMC50, #00000001B
MOV    TMC50, #00000111B ;タイマ出力設定（出力許可）
SET1   TCE50            ;タイマの動作許可
      .
      .
      .
    
```

TMC50の設定手順です。

【8ビット・タイマH0】

方形波出力は、次の項目を設定します。

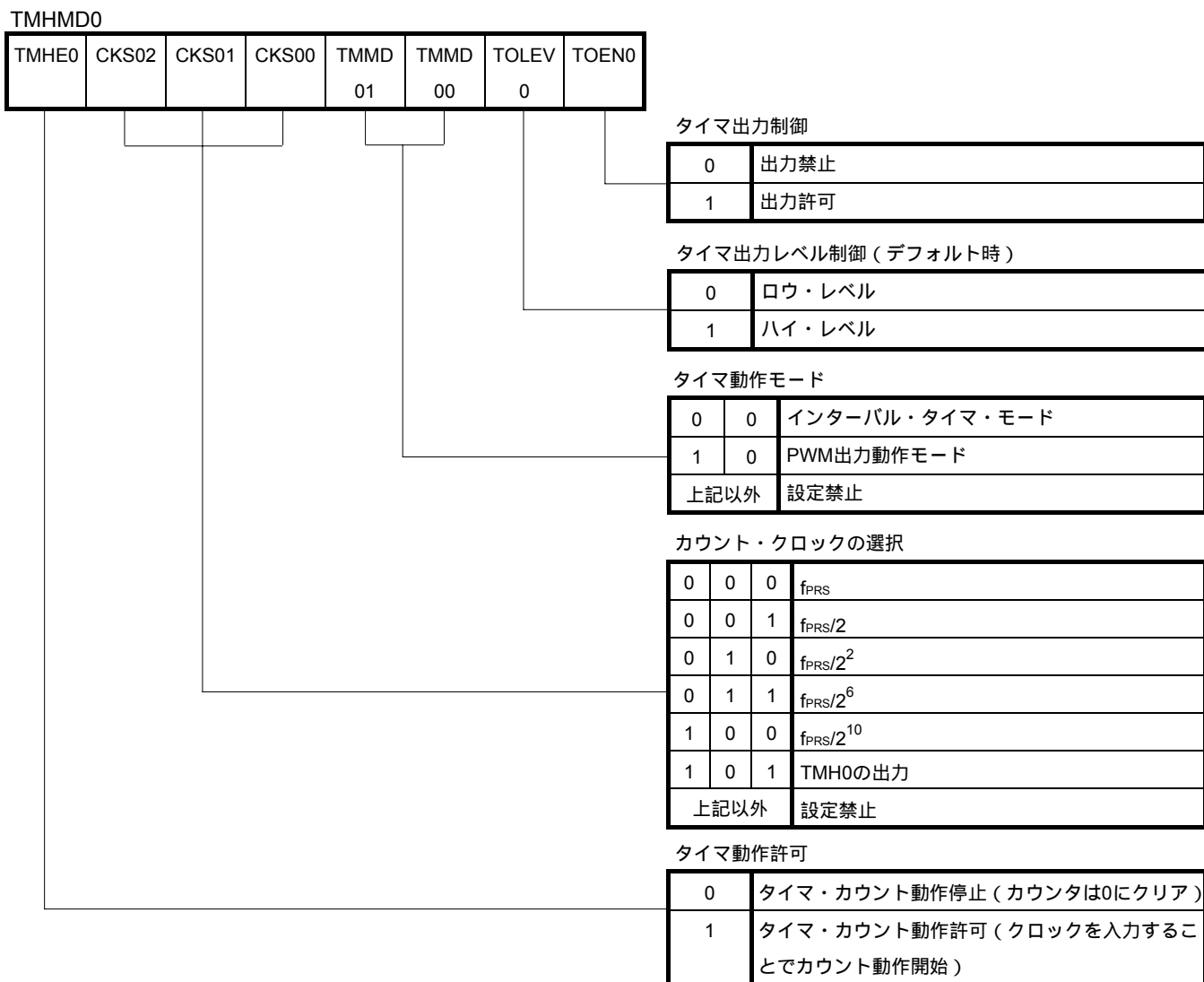
- (1) タイマ・クロック選択, タイマ・モード設定, タイマ出力設定 (出力許可)
- (2) タイマ・コンペア・レジスタ (CMP00) の設定

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のカウント・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

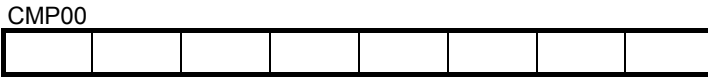


注意 TMHE0 = 1 のとき, TMHMD0の他のビットを設定することは禁止です。
ただし, リフレッシュ (同値書き込み) することは可能です。

(2) 8ビット・タイマ・コンペア・レジスタ(CMP00)の設定

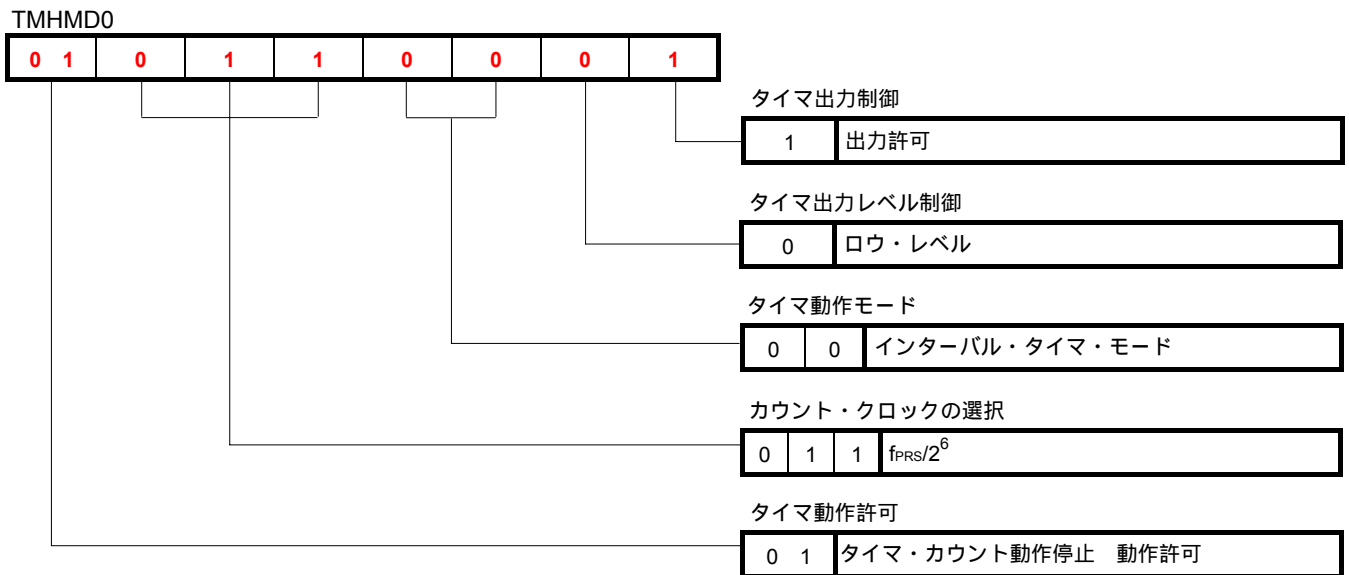
8ビット・タイマH0のカウンタ値と比較し、2つの値が一致したときに割り込みを発生します。

図4 - 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 $f_{PRS}/2^6$
 ・コンペア・レジスタに00Hを設定
 (サンプル・プログラムの設定と同内容)



下記の計算式よりコンペア値を計算し、方形波周波数を初期値1 [kHz] に設定します。

$$\text{コンペア・レジスタ設定値} = (\text{タイマ・クロック周波数} / \text{方形波周波数}) / 2$$

【C言語】 (Kx2_Square.c)

```
TMHMD0 = 0b00110001; /* 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
                       タイマ出力を許可 */
CMP00 = 62; /* コンペア・レジスタに初期値を設定 */
TMHE0 = 1; /* タイマの動作許可 */
.
.
.
```

【アセンブリ言語】 (Kx2_Square.asm)

```
MOV     TMHMD0, #00110001B ;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
                       ;タイマ出力を許可
MOV     CMP00, #62 ;コンペア・レジスタに初期値を設定
SET1    TMHE0 ;タイマの動作許可
.
.
.
```

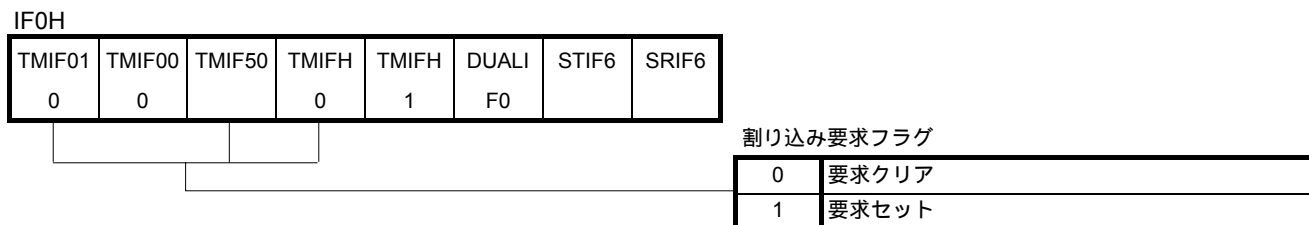
4.3 割り込みの設定

(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

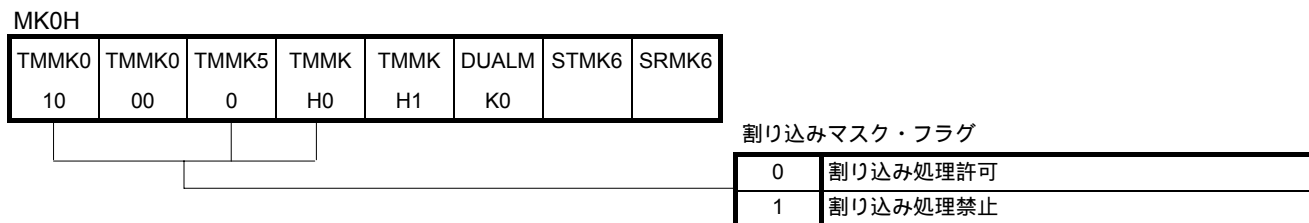


(2) 割り込みマスクの設定

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】
- ・INTTM001割り込み要求 (TMIF001) をクリア
 - ・INTTM001割り込みマスク (TMMK001) をクリア (割り込み処理許可)
- (サンプル・プログラムTM01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_Square.c)

```

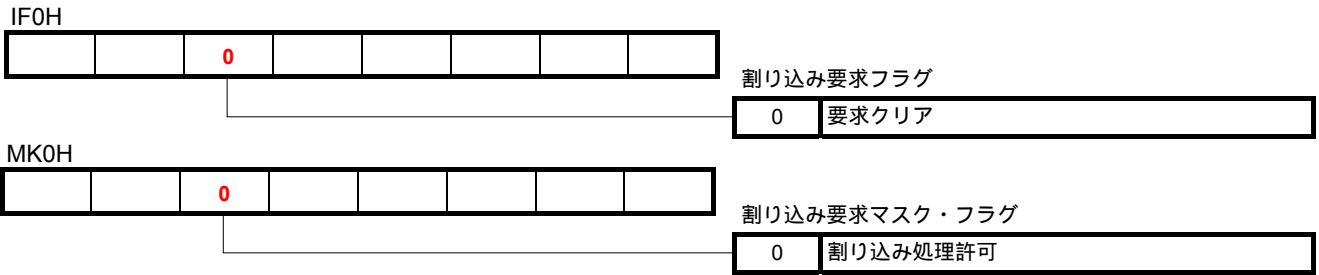
TMIF001 = 0; /* INTTM001割り込み要求クリア */
TMMK001 = 0; /* INTTM001割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Square.asm)

```

CLR1    TMIF001    ;INTTM001割り込み要求クリア
CLR1    TMMK001    ;INTTM001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_Square.c)

```

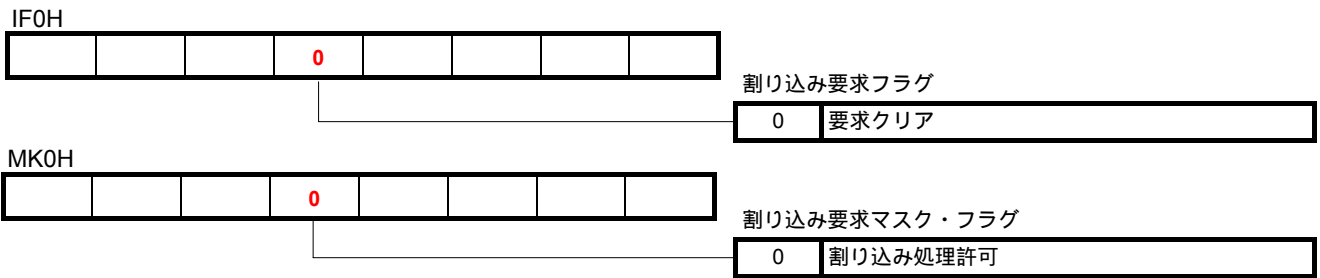
TMIF50 = 0; /* INTTM50割り込み要求クリア */
TMMK50 = 0; /* INTTM50割り込みマスク解除 */
.
.
.
    
```

【アセンブリ言語】 (Kx2_Square.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_Square.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_Square.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、P06/TO01, P17/TO50, P15/TOH0を出力ポートとして使用します。ここでは、TM01版のP06/TO01の設定内容を例にとり、説明します。TM50版, TMH0版はこの説明では省略します。

なお、サンプル・プログラムでは後述の【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0 (PM0) のフォーマット

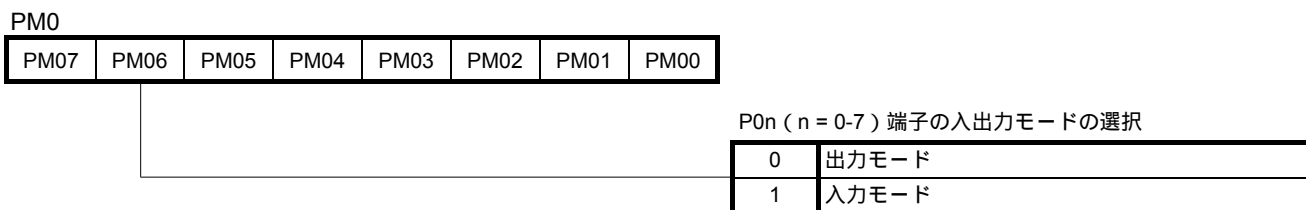
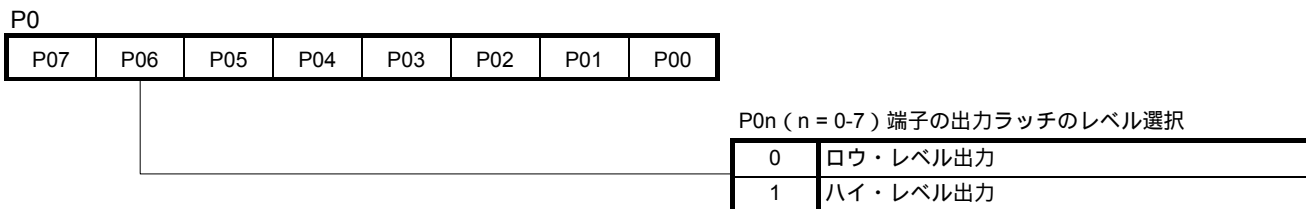
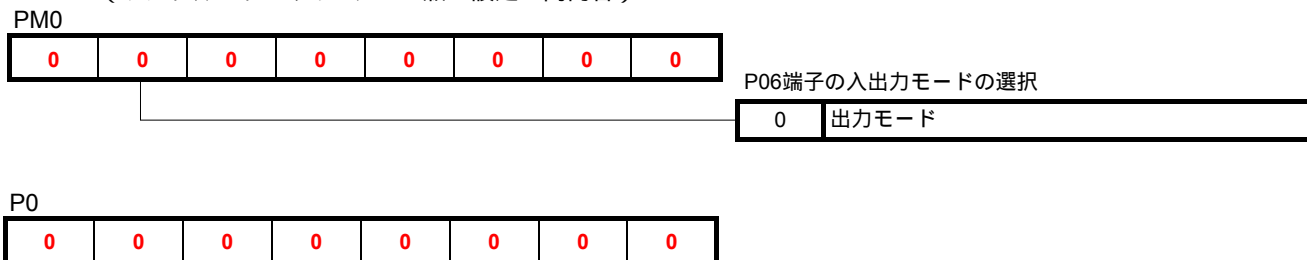


図4 - 14 ポート・レジスタ0 (P0) のフォーマット



【例】 ・ P06を出力ポートに設定

(サンプル・プログラムTM01版の設定と同内容)



PM0の設定値は「00000000」, P0の設定値は「00000000」とします。

【C言語】

```
PM0 = 0b00000000;
P0 = 0b00000000;
```

【アセンブリ言語】

```
MOV PM0, #00000000B
MOV P0, #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_Square.c)

- ・ 方形波出力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、方形波出力を継続するNOP命令のみの割り込み待ちループ処理になります。

```

/*****
;   メイン・ループ
;*****/
void main(void){
    fn_InitTimer();
    EI();
    while (1){
        NOP();
    }
}

```

Callout 1: 方形波出力の初期化を行い、処理を開始します。

Callout 2: 割り込みを許可します。

Callout 3: メイン処理内ではNOP命令のみで方形波出力をし続けます。

【アセンブリ言語】 (Kx2_Square.asm)

- ・ 割り込み処理でレジスタ・バンク1の汎用レジスタをグローバル変数として使用するので、メイン・ループに入る前にレジスタ・バンク1の汎用レジスタを初期化します。
- (・ TMH0版のみ割り込み回数をカウントするのでカウンタの初期化をします。)
- ・ メイン処理内では何もせず、方形波出力を継続するNOP命令のみのループ処理になります。

```

;*****/
;   メイン・ループ
;*****/
SEL      RB1
MOV      RINTCNT, #2
MOVW    AX, #62H
MOV      B, #CDATANO
SEL      RB0
EI
MMAIN:
NOP
BR      MMAIN

```

Callout 1: レジスタ・バンクを1に切り替えます。

Callout 2: TMH0版のみ割り込み回数カウンタを2で初期化します。

Callout 3: 割り込み処理にて使用する汎用レジスタを初期化します。

Callout 4: レジスタ・バンクを0に戻します。

Callout 5: 割り込みを許可します。

Callout 6: メイン処理内ではNOP命令のみで方形波出力をし続けます。

アセンブリ言語での方形波出力の初期化は、メイン処理に入る前に、方形波出力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 変数・定数の定義

ここでは、本サンプル・プログラムで使用する変数および定数の定義について説明します。

【C言語】 (Kx2_Square.c)

C言語版では次のような定義をします。

- ・データ・テーブルに定義するコンペア・レジスタ設定値のデータ数をカウントする変数を定義します。
(・TMH0版のみ割り込み回数をカウントするので、カウンタを定義します。)
- ・方形波出力の周波数を変更するためのコンペア・レジスタに設定する値をROMに定義します。
(周波数を1 [kHz] ~ 2 [kHz] の範囲で変更させる64個のデータを定義します。)

```

/*=====
;      使用する変数・定数の定義
;=====*/
sreg unsigned int g_unIntCnt = 2;
sreg unsigned int g_unDataNo = 64-1;

const unsigned int g_unDataTable[64] =
{ 62,61,60,59,58,57,56,55,54,53,52,51,50,49,48,47,
  46,45,44,43,42,41,40,39,38,37,36,35,34,33,32,31,
  31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,
  47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62};
    
```

TMH0版のみ割り込み回数カウンタ定義し,2で初期化します。

TMH0版のみ - 1します。

データ数カウンタを定義します。

コンペア・レジスタ設定値のデータを定義します。

【アセンブリ言語】 (Kx2_Square.asm)

アセンブリ言語版では次のような定義をします。

- ・データ・テーブルに定義するコンペア・レジスタ設定値のデータ数を定義します。
- ・方形波出力の周波数を変更するためのコンペア・レジスタに設定する値をROMに定義します。
(周波数を1 [kHz] ~ 2 [kHz] の範囲で変更させる64個のデータを定義します。)
- (・TMH0版のみ割り込み回数をカウントするので、カウンタを定義します。)

```

;=====
;      データ数の定義
;=====
CDATANO      EQU      64
.
.

;=====
;      ROMの定義 (方形波出力データテーブル)
;=====
XROM      CSEG AT      0100H
TDATATABLE:
    DB      62,61,60,59,58,57,56,55,54,53,52,51,50,49,48,47
    DB      46,45,44,43,42,41,40,39,38,37,36,35,34,33,32,31
    DB      31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46
    DB      47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62
.
.

;=====
;      使用する変数の定義
;=====
DINTCNT      DSEG SADDR
RINTCNT: DS      1
    
```

データ数を定義します。
(割り込み内で汎用レジスタを用い、データ数をカウントします。)

コンペア・レジスタ設定値のデータを定義します。

TMH0版のみ割り込み回数カウンタ定義します。

4.7 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_Square.c)

- ・コンペア・レジスタの値を更新して、周波数を変化させます。

なお、ここでは例としてTMH0版を紹介しています。

```

/*****
;          1周期完了での割り込み処理
;*****/
__interrupt void fn_intTimerSquare(void)
{
    if(g_unIntCnt--; == 0){ /* 割り込みカウンタをデクリメントして0ならば処理実行*/

        g_unDataNo--;      /* データ変数をデクリメント */

        TMHE0 = 0;        /* タイマ動作禁止 */
        CMP00 = g_unDataTable[g_unDataNo]; /* 新しいインターバルを設定 */
        TMHE0 = 1;        /* タイマ動作再開 */

        if(g_unDataNo == 0){
            /* データ変数が0ならば処理実行 */
            g_unDataNo = 64;
            /* データ変数を初期化 */
        }

        g_unIntCnt = 2;
        /* 割り込み回数カウンタを初期化 */
    }
}
    
```

ここでは、TMH0を紹介していますが、使用するタイマによってこの設定が異なります。

赤字の処理においては、TMH0版のみの処理となります。
(割り込み2回ごとにコンペア・レジスタの値を更新するため)

【アセンブリ言語】 (Kx2_Square.asm)

- ・コンペア・レジスタの値を更新して，周波数を変化させます。
- ・割り込み処理では，以下のレジスタ・バンク1の汎用レジスタをグローバル変数として使用します。

AX : 次のデータ
 B : データ・カウンタ
 HL : データ・テーブルの先頭アドレス

なお，ここでは例としてTMH0版を紹介しています。

```

;*****;
;      1周期完了での割り込み処理
;*****
IINTSQUARE:
    SEL     RB1                ;割り込みではレジスタバンク1を使用

    DBNZ   RINTCNT, $HINTEND   ;割り込み回数カウンタをデクリメントして
                                ; 0になったら初期化する

    CLR1   TMHE0              ;タイマ動作禁止
    MOV    CMP00, A           ;新しいインターバルを設定
    SET1   TMHE0              ;タイマ動作再開

    DBNZ   B, $HINTNEXT       ;データ数をデクリメントして0になったら
                                ;初期化する

    MOV    B, #CDATAN0        ;データ数初期化

HINTNEXT:
    MOVW   HL, #TDATATABLE-1

    MOV    A, [HL+B]          ;次のデータ読み出し

    MOV    RINTCNT, #2        ;割り込み回数カウンタを初期化

HINTEND:
    RETI                       ;割り込みから復帰
    
```

ここでは，TMH0を紹介していますが，使用するタイマによってこの設定が異なります。

赤字の処理においては，TMH0版のみの処理となります。
 (割り込み2回ごとにコンペア・レジスタの値を更新するため)

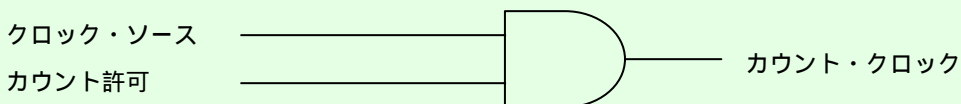
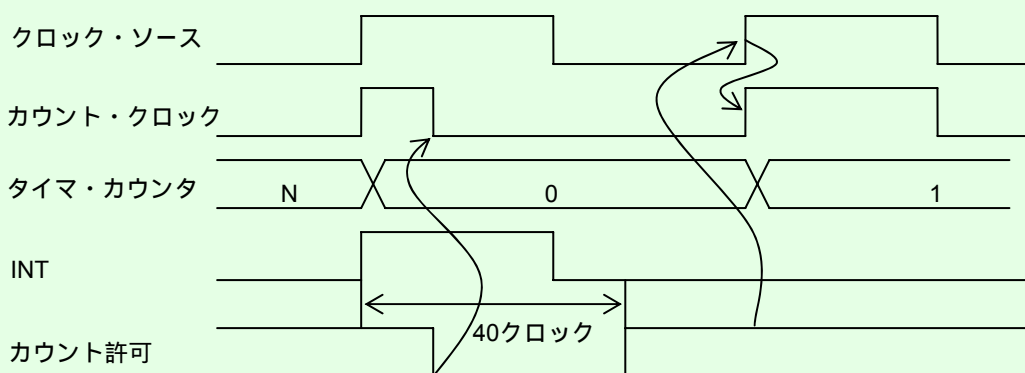


【コラム】タイマ動作中の動作停止 / 動作再開について

タイマ・カウンタは動作停止中にはクリアされ、0になっています。また、タイマ・カウンタに入力されているカウント・クロックはロウ・レベルになっています。次のカウント・アップの予定タイミングまでにタイマを再起動することができれば、カウントの誤差は発生しないことになります。このときの動作を以下に説明します。


タイマHのブロック図ではカウント・クロックは以下の図に抜粋したようにセクタからのクロック・ソースがカウント許可信号でANDされています。そのため、割り込み処理でカウントを禁止することで、カウント・クロックはロウ・レベルになります。再度、カウント許可にすることで、カウント・クロックは動作可能になり、次のクロック・ソースの立ち上がりで0 1とカウント・アップします(このタイミングではクロック・ソースがロウ・レベルでカウント許可しています。クロック・ソースがハイ・レベルの場合にはカウント・クロックは直ぐに立ち上がったとしても、タイマ・カウンタにはまだクリアがかかっているため、ここではカウント・アップせず、次のカウント・クロックの立ち上がりでカウントします)。

このように、カウント・クロックを適切に選べば、カウント停止の影響を残さずに処理可能です。




タイマHブロック図より

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

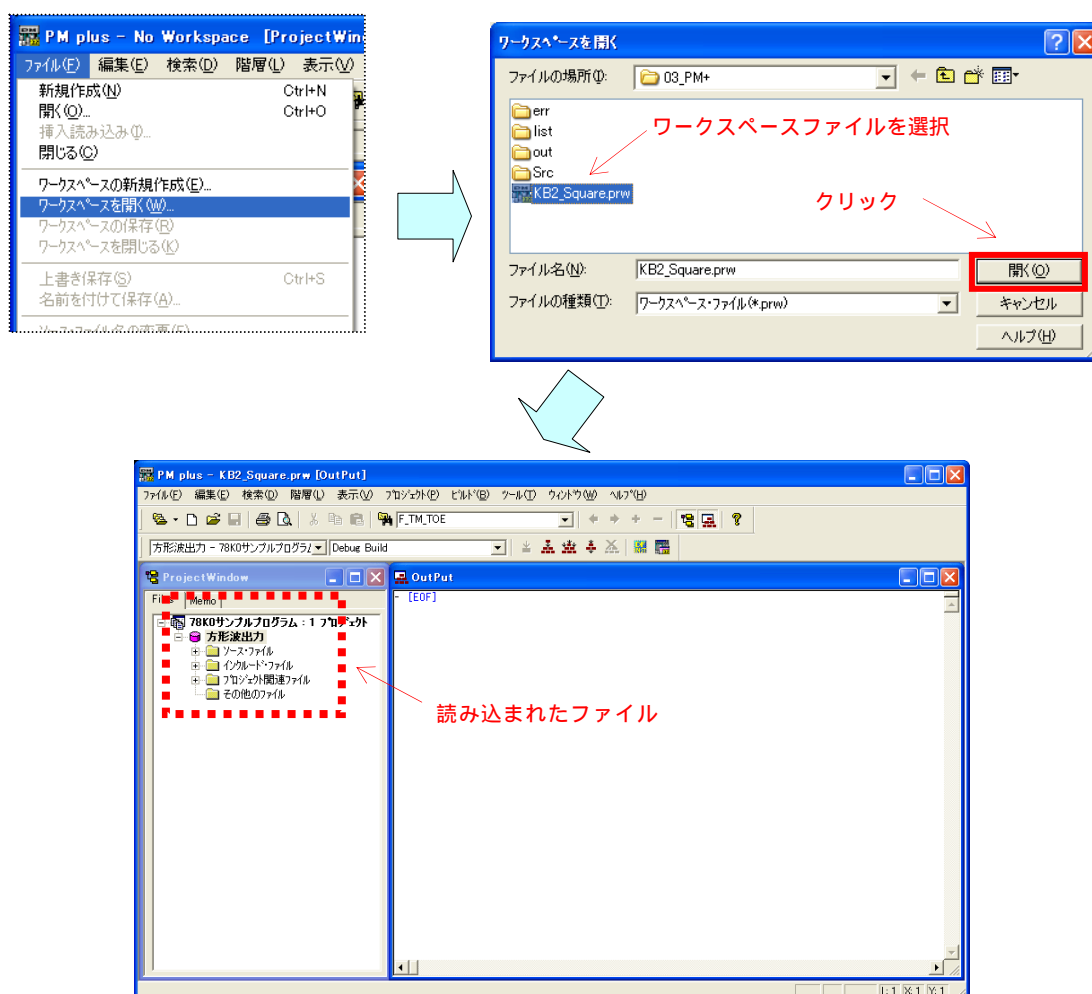
サンプル・プログラムをSM+ for 78K0/Kx2 (以降、「SM+」と表記します)で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、でダウンロードしたC言語用のファイルを用いて、統合開発環境 PM+にてビルドしてから、SM+を起動するまでの動作の一例を説明します。

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。

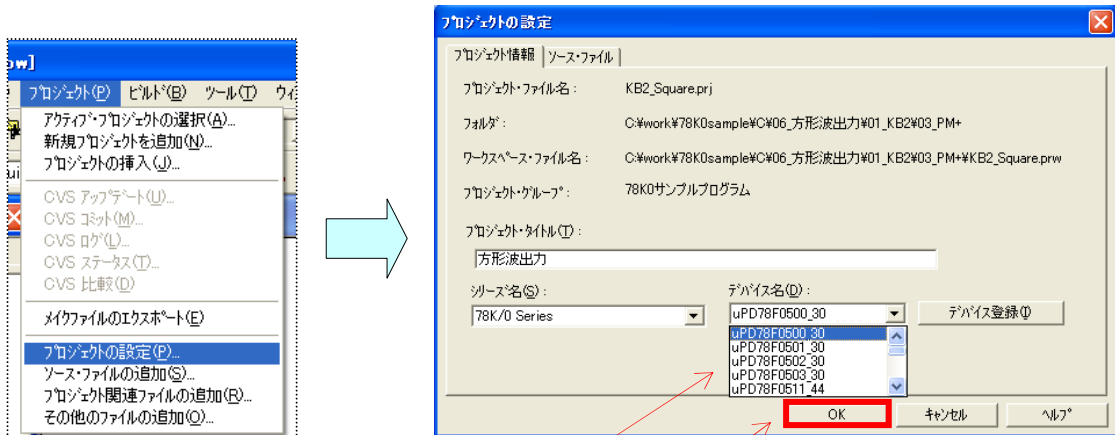
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「Kx2_Square.prw」[※]を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。

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_Square.prw」



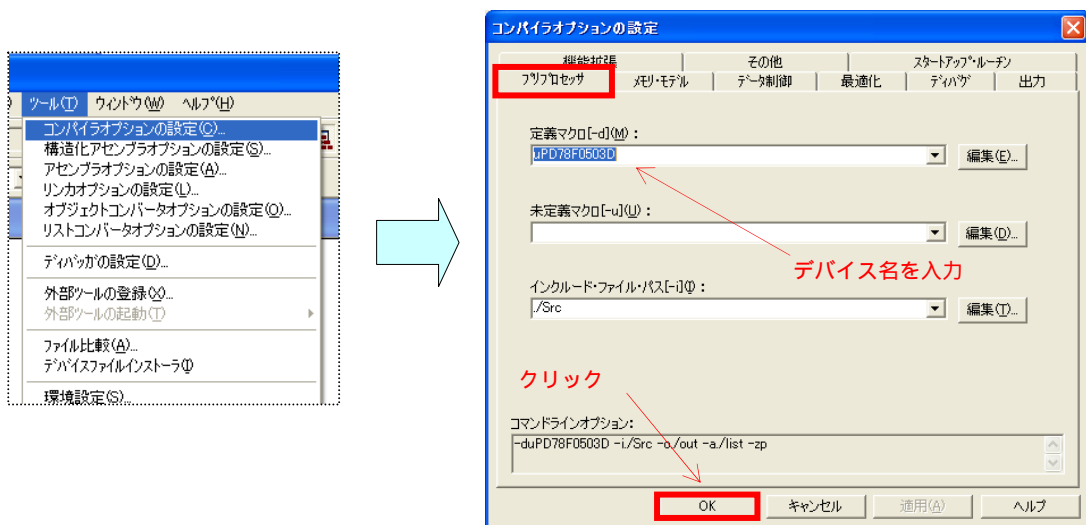
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が表示されたら、使用するデバイス名を選択（デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択）し、[OK] ボタンをクリックしてください。



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。

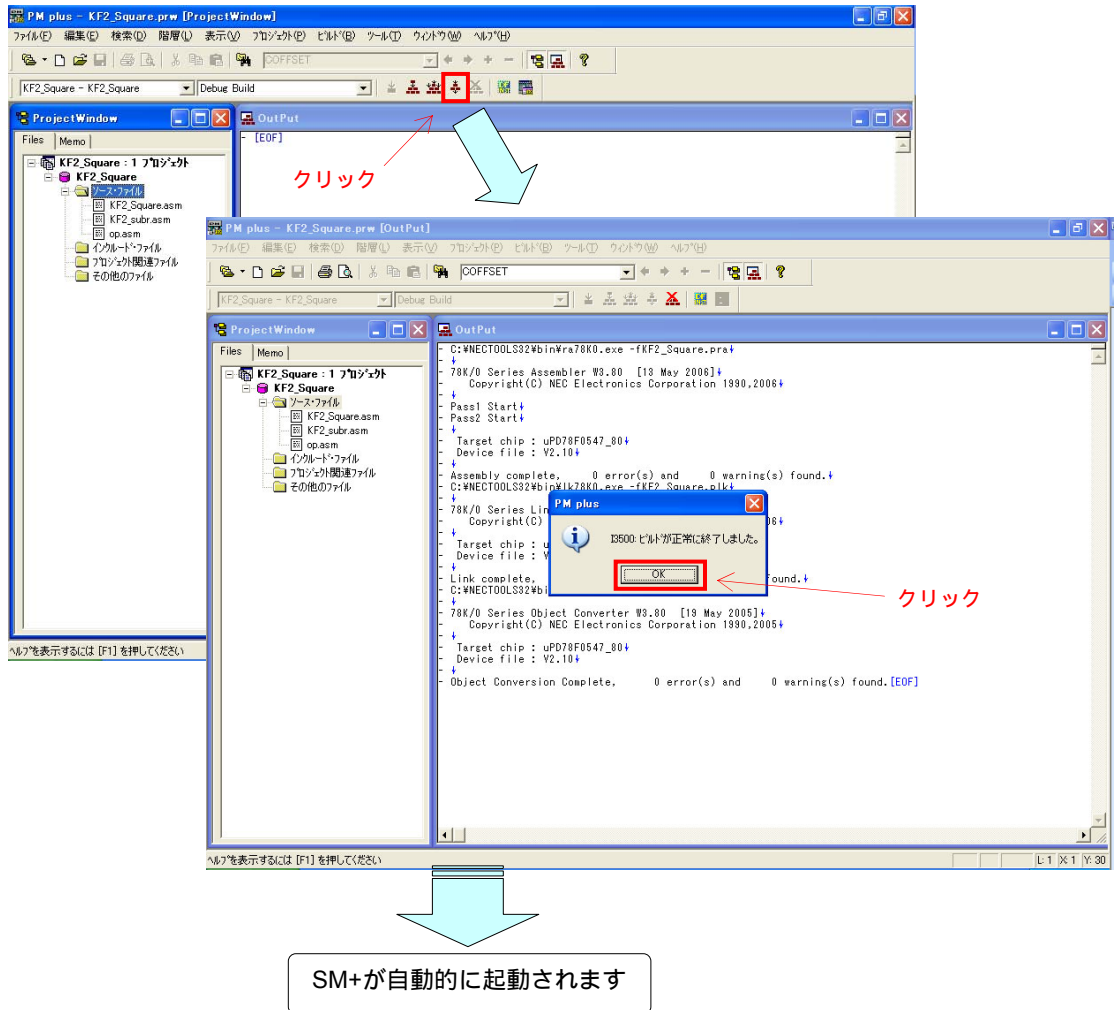
- (4) [ツール] [コンパイラオプションの設定] を選択してください。[コンパイラオプションの設定] 画面が表示されたら、[プリプロセッサ] タグページが表示されているのを確認し、その中の定義マクロ欄に使用するデバイス名を入力し、[OK]をクリックします。入力するデバイス名は、「Kx2_Res.h」ヘッダファイルの先頭部のコメントを確認してください（以下は78K0/KB2の場合）。



デバイス名を入力

クリック

- (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_Square.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。

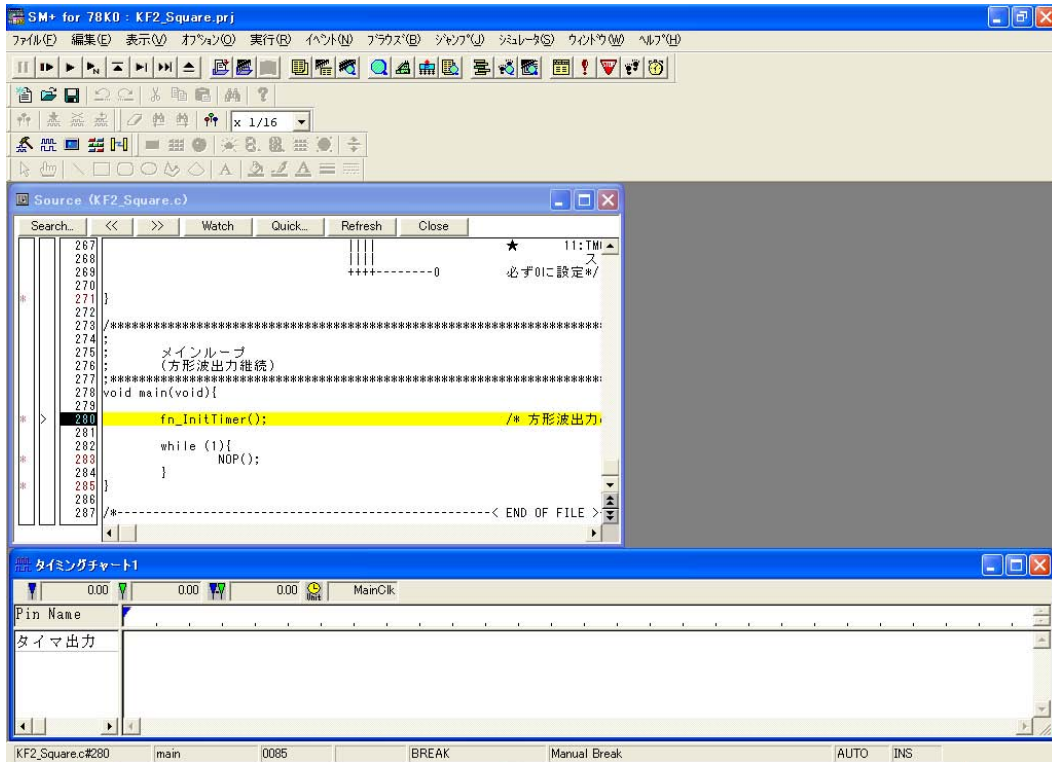



5.2 SM+での動作

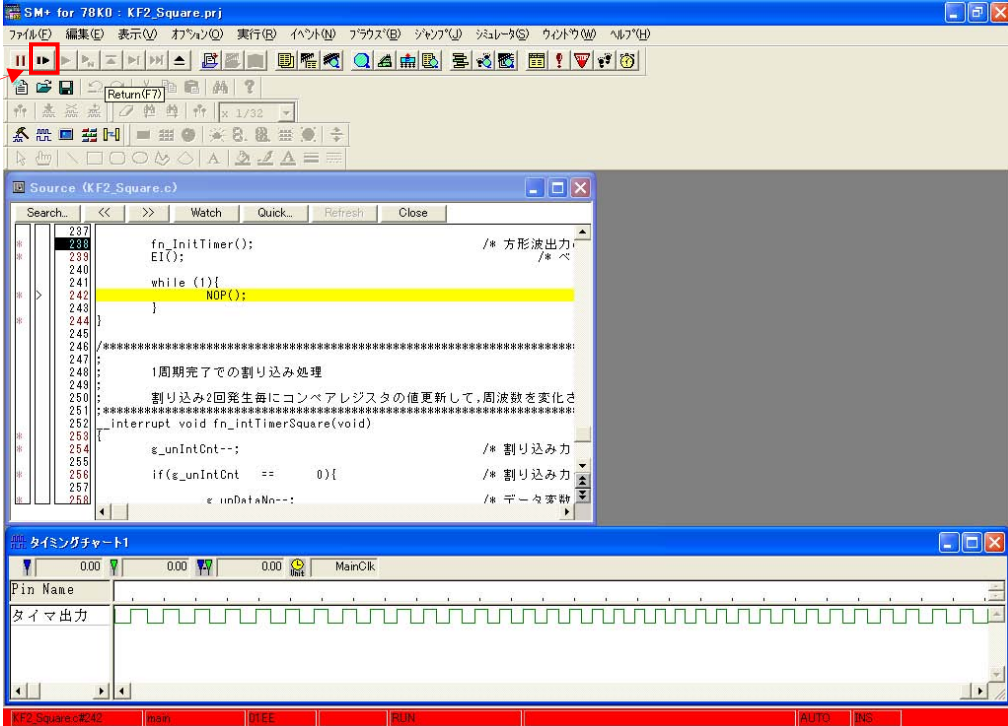
ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。

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。



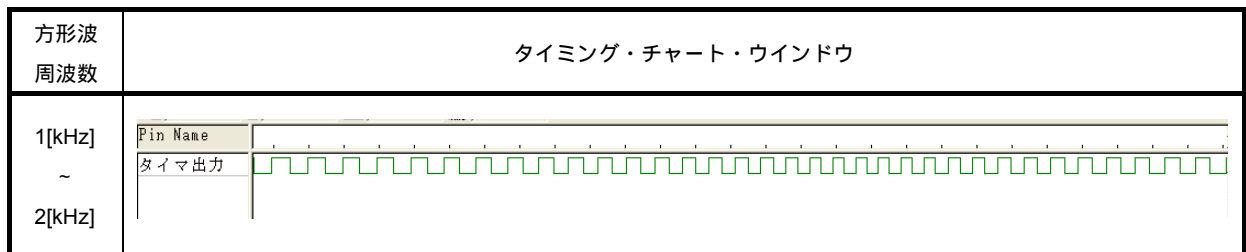
- (2)  (「リスタート」ボタン)をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。



クリック

プログラム実行中は、 赤になります。

- (3) プログラムを実行すると、タイミング・チャート上にポートの出力結果が表示されるのがわかります。



プログラムを実行させるとタイマ出力の周波数が、上記のように一定周期で変動していくのがわかります。

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。

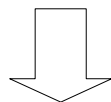
(1) オプション・バイトの設定

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
    
```

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

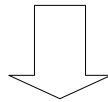
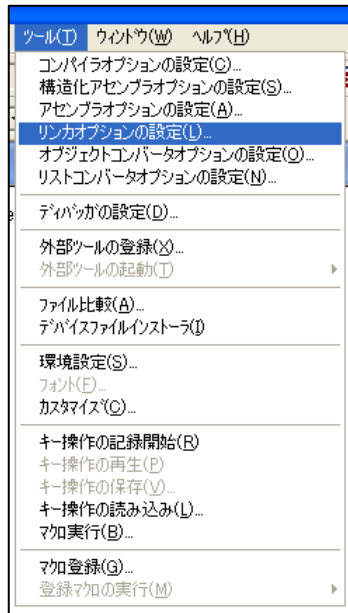
TVECT_TBL1    CSEG      AT      0004H
              DW      IINIT      ;0004H INTLVI
    
```

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。

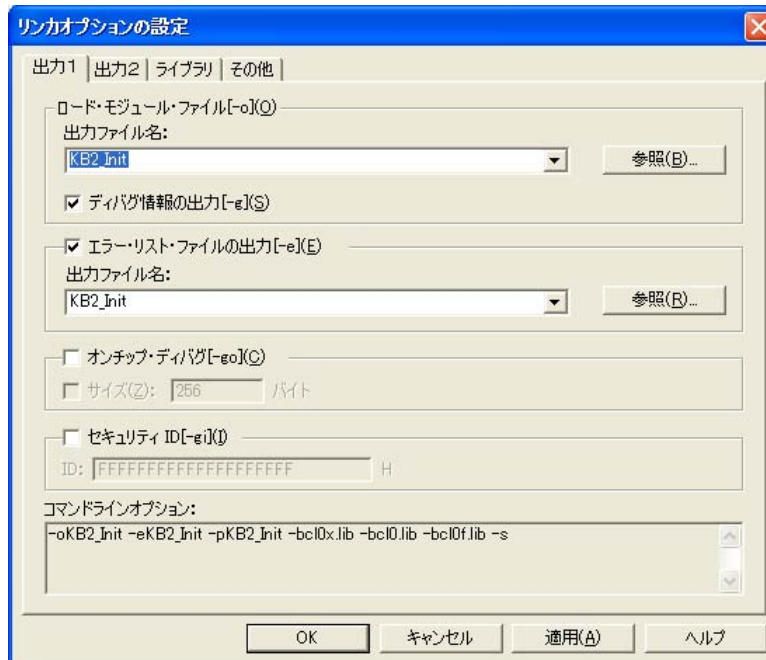
C言語版では不要です。

(3) リンカの設定

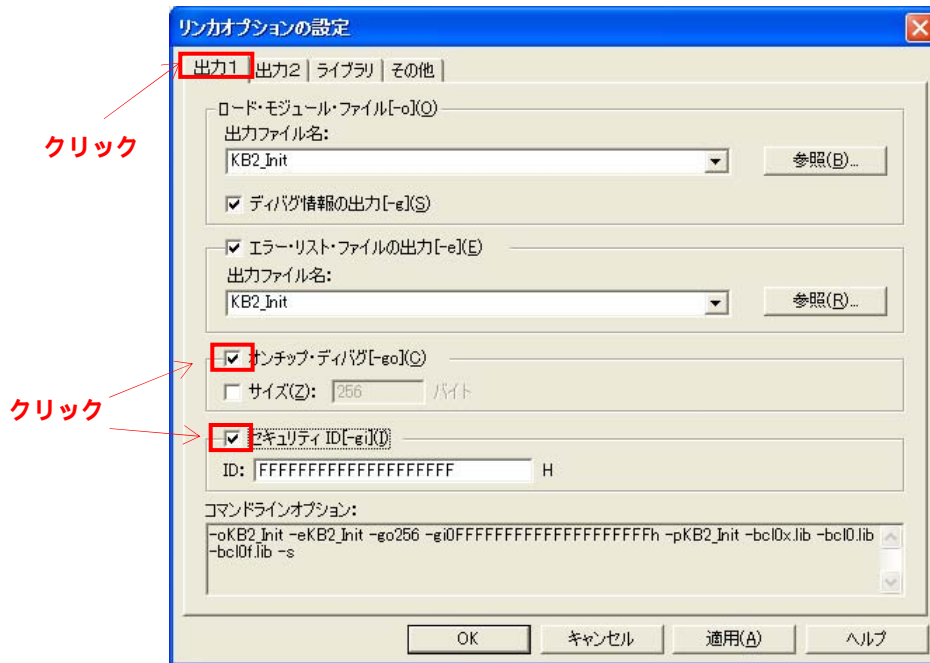
オンチップ・デバッグを行う場合、ビルドの際、リンカの設定を行う必要があります。
PM+の「ツール」メニューから「リンカオプションの設定」を選択してください。



「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 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