

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

78K0S/Kx1+

サンプル・プログラム（割り込み）

スイッチ入力による外部割り込み編

この資料は、サンプル・プログラムの動作概要や使用方法、および割り込み機能の設定方法や活用方法について説明したものです。サンプル・プログラムでは、スイッチ入力の立ち下がりエッジを検出して割り込みを発生させ、スイッチの入力回数に応じたLED点灯パターンを表示します。

対象デバイス

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

目次

第1章 概要 ... 3
1.1 初期設定の主な内容 ... 3
1.2 メイン・ループ以降の内容 ... 4
第2章 回路図 ... 5
2.1 回路図 ... 5
2.2 周辺ハードウェア ... 5
第3章 ソフトウェアについて ... 6
3.1 ファイル構成 ... 6
3.2 使用する内蔵周辺機能 ... 7
3.3 初期設定と動作概要 ... 7
3.4 フロー・チャート ... 8
第4章 設定方法について ... 9
4.1 割り込みの設定 ... 9
4.2 割り込み中の処理 ... 15
第5章 システム・シミュレータ SM+での動作確認 ... 18
5.1 サンプル・プログラムのビルド ... 18
5.2 SM+での動作 ... 20
第6章 関連資料 ... 25
付録A プログラム・リスト ... 26
付録B 改版履歴 ... 37

- 本資料に記載されている内容は2008年7月現在のものです、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、一切その責を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 本資料に記載された回路、ソフトウェアおよびこれらに関する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

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

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

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

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

(注)

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

第1章 概 要

このサンプル・プログラムでは、割り込み機能の使用例を示しています。スイッチ入力の立ち下がりエッジを検出して割り込み処理を行い、スイッチの入力回数に応じたLED点灯パターンを表示します。

1.1 初期設定の主な内容

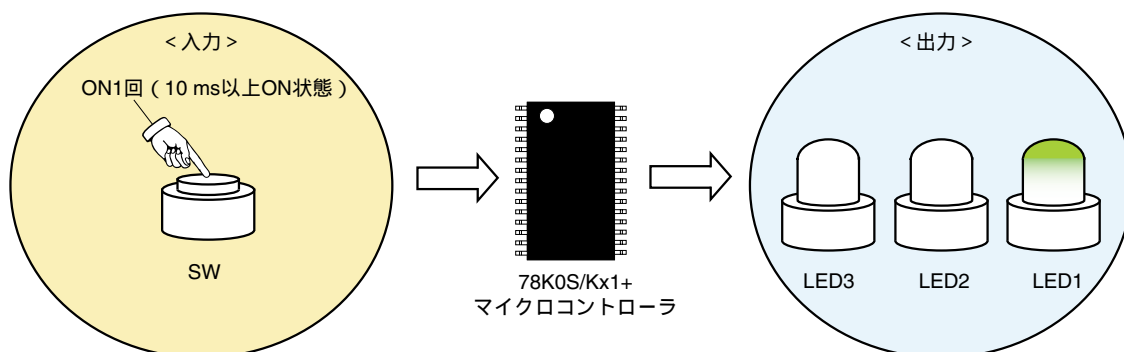
初期設定の主な内容は、次のとおりです。

- ・システム・クロック・ソースとして、高速内蔵発振器を選択^注
- ・ウォッチドッグ・タイマの動作停止
- ・CPUクロック周波数と周辺ハードウェア・クロック周波数を2 MHzに設定
- ・入出力ポートの設定
- ・INTP1（外部割り込み）の有効エッジを立ち下がりエッジに設定
- ・割り込み許可

注 オプション・バイトで設定します

1.2 メイン・ループ以降の内容

スイッチ入力によるINTP1端子の立ち下がりエッジを検出し、割り込み処理を行います。割り込み処理では、INTP1端子の立ち下がりエッジを検出してから約10 ms経過後に、スイッチがONであることを確認し、LEDの点灯パターンを変化させます。約10 ms経過後に、スイッチがOFFである場合は、チャタリングであると判定し、LEDの点灯パターンを変化させません。



スイッチの入力回数 ^注	LED出力		
	LED3	LED2	LED1
0	OFF	OFF	OFF
1	OFF	OFF	ON
2	OFF	ON	OFF
3	OFF	ON	ON
4	ON	OFF	OFF
5	ON	OFF	ON
6	ON	ON	OFF
7	ON	ON	ON

注 8回目以降は、0回目からの点灯パターンの繰り返しになります。

注意 デバイス使用上の注意事項については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。



【コラム】チャタリングとは

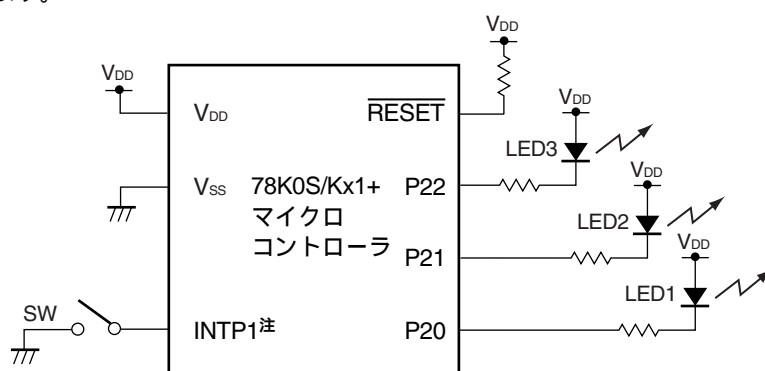
スイッチが切り替わった直後に、接点が機械的にばたつくことにより、電気信号がONとOFFを繰り返す現象のことです。

第2章 回路図

この章では、このサンプル・プログラムで使用する回路図および周辺ハードウェアを説明します。

2.1 回路図

回路図を次に示します。



注 INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ

INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

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

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

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

2.2 周辺ハードウェア

使用する周辺ハードウェアを次に示します。

(1) スイッチ (SW)

LED点灯制御用の入力として、スイッチを使用します。

(2) LED (LED1, LED2, LED3)




スイッチ入力に対応した出力として、LEDを使用します。

第3章 ソフトウェアについて

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

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル		
				
main.asm (アセンブリ言語版) ----- main.c (C言語版)	マイコンのハードウェア初期化処理とメイン処理のソース・ファイル	注1	注1	
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル (システム・クロック・ソースなどを設定)			
int.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
int.prj	統合開発環境 PM+用プロジェクト・ファイル			
int.pri int.prs int.prm	システム・シミュレータ SM+ for 78K0S/Kx1+用プロジェクト・ファイル		注2	
int0.pnl	システム・シミュレータ SM+ for 78K0S/Kx1+用入出力パネル・ファイル (周辺ハードウェア動作を確認するために使用)		注2	
int0.wvo	システム・シミュレータ SM+ for 78K0S/Kx1+用タイミング・チャート・ファイル (波形を確認するために使用)			

注1. アセンブリ言語版には「main.asm」、C言語版には「main.c」が同封されています。

2. 78K0S/KU1+マイクロコントローラには、同封されていません。

備考



: ソース・ファイルのみ同封



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



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

3.2 使用する内蔵周辺機能

このサンプル・プログラムでは、マイコンに内蔵する次の周辺機能を使用します。

- ・スイッチ入力 : INTP1^注 (外部割り込み)
- ・LED出力 : P20, P21, P22 (出力ポート)

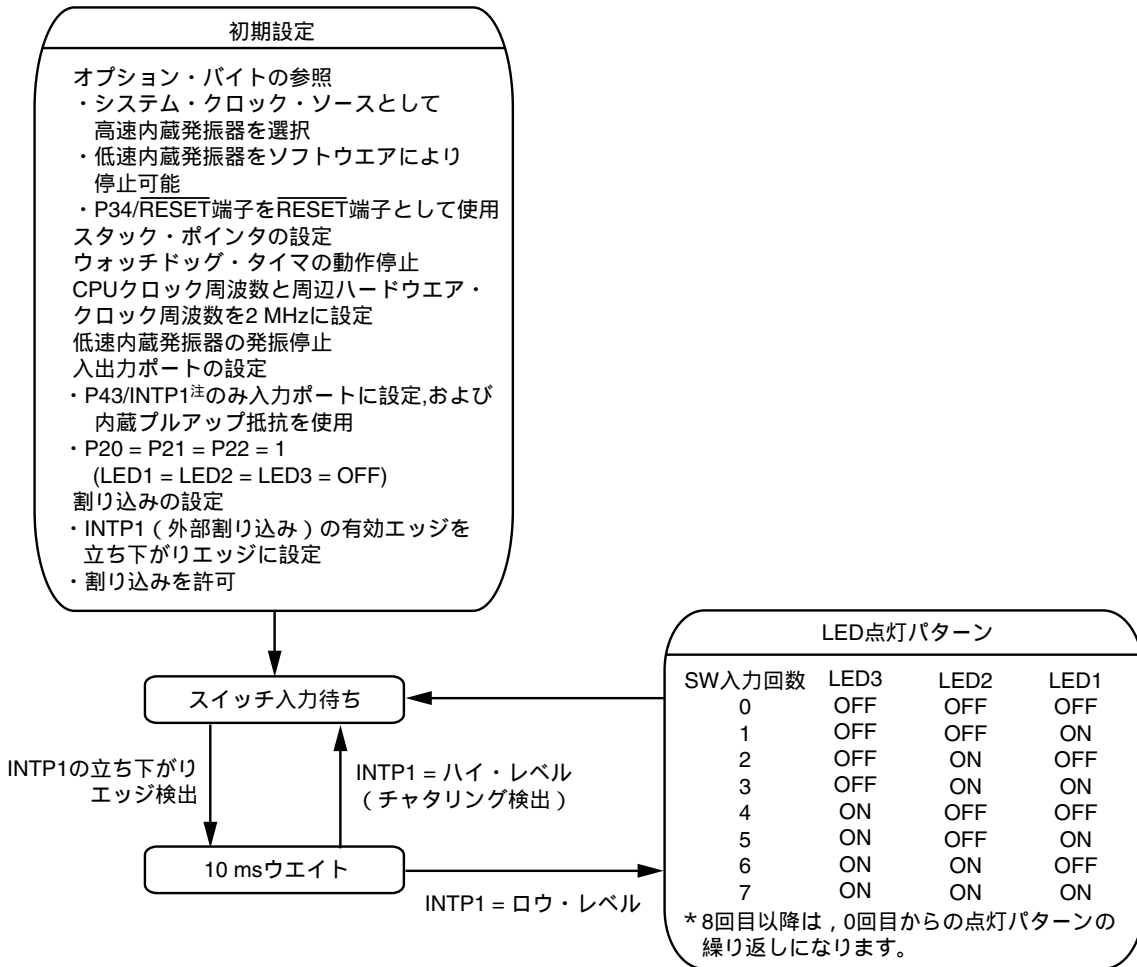
注 INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、クロック周波数の選択や、入出力ポートの設定、割り込みの設定などを行います。

初期設定完了後は、スイッチ入力 (SW) の立ち下がりエッジを検出して割り込み処理を行い、スイッチの入力回数に応じて、3つのLED (LED1, LED2, LED3) の点灯を制御します。

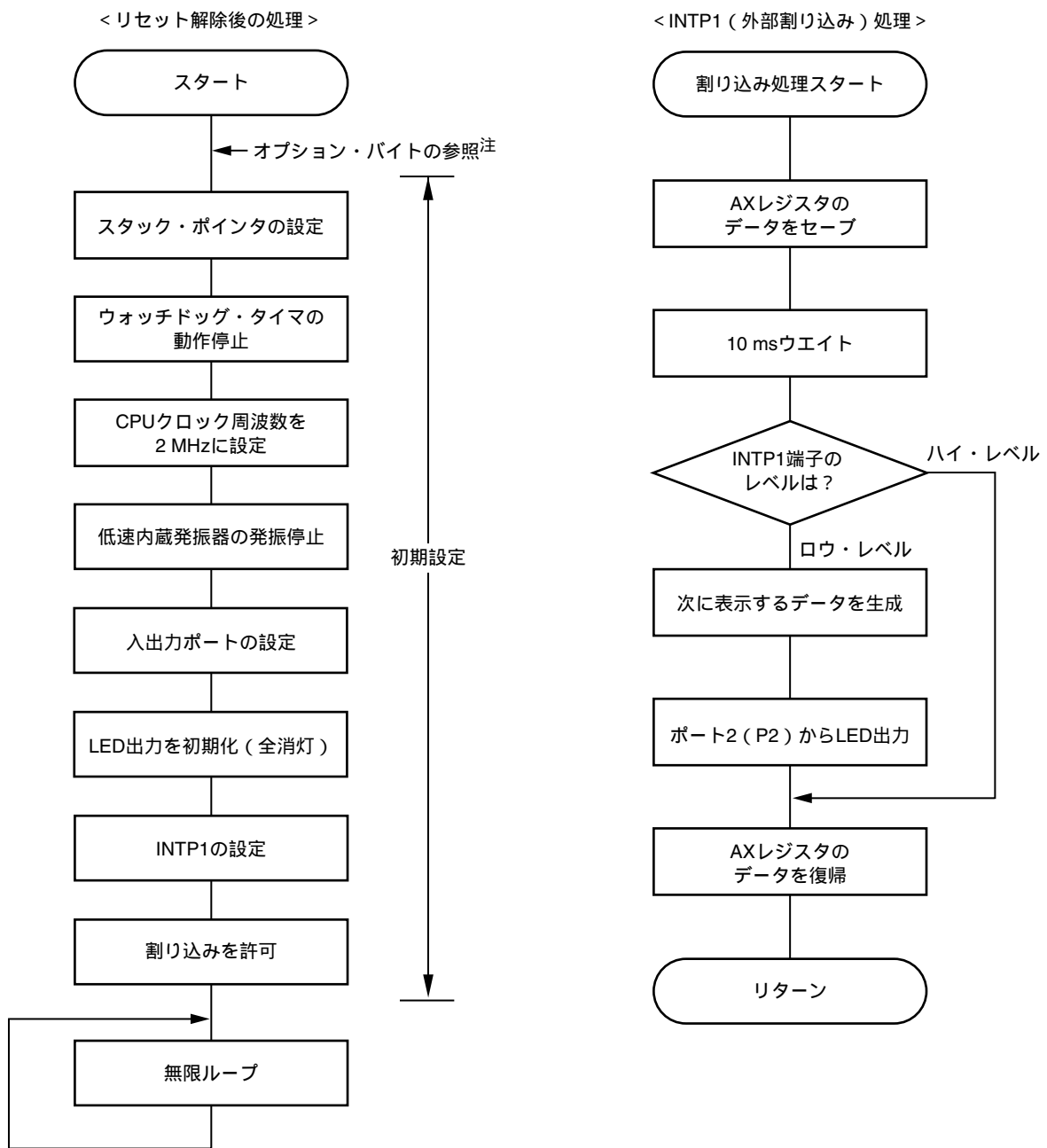
詳細については、次の状態遷移図 (ステート・チャート) に示します。



注 INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

3.4 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。



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

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

第4章 設定方法について

この章では、割り込みの設定、および割り込み処理について説明します。

その他の初期設定については、[78K0S/Kx1+ サンプル・プログラム \(初期設定\) LED点灯のスイッチ制御編 アプリケーション・ノート](#)を参照してください。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#))を参照してください。

アセンブラ命令については、[78K0S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 割り込みの設定

割り込み機能は次の4種類のレジスタで制御します。

- ・割り込み要求フラグ・レジスタ0, 1^注 (IF0, IF1^注)
- ・割り込みマスク・フラグ・レジスタ0, 1^注 (MK0, MK1^注)
- ・外部割り込みモード・レジスタ0, 1^注 (INTM0, IMTM1^注)
- ・プログラム・ステータス・ワード (PSW)

各割り込み要求に対する割り込み要求フラグ、割り込みマスク・フラグ名称は次のようになります。

割り込み要求信号名	割り込み要求フラグ	割り込みマスク・フラグ
INTLVI	LVIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010
INTAD	ADIF	ADMK
INTP2 ^注	PIF2 ^注	PMK2 ^注
INTP3 ^注	PIF3 ^注	PMK3 ^注
INTTM80 ^注	TMIF80 ^注	TMMK80 ^注
INTSRE6 ^注	SREIF6 ^注	SREMK6 ^注
INTSR6 ^注	SRIF6 ^注	SRMK6 ^注
INTST6 ^注	STIF6 ^注	STMK6 ^注

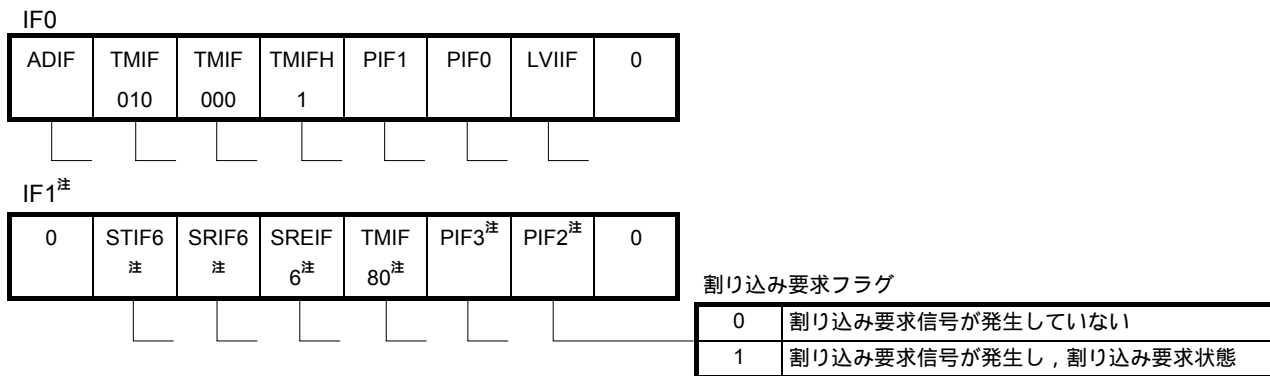
注 78K0S/KA1+, 78K0S/KB1+マイクロコントローラのみ

(1) 割り込み要求フラグの設定

割り込み要求フラグは、対応する割り込み要求の発生または命令の実行によりセット（1）され、割り込み要求受け付け時、リセット実行時、または命令の実行によりクリア（0）されるフラグです。

割り込み機能を使用する場合は、割り込み要求発生前に、使用する割り込み要求フラグをクリア（0）してください。

図4 - 1 割り込み要求フラグ・レジスタ0, 1^注（IF0, IF1^注）のフォーマット

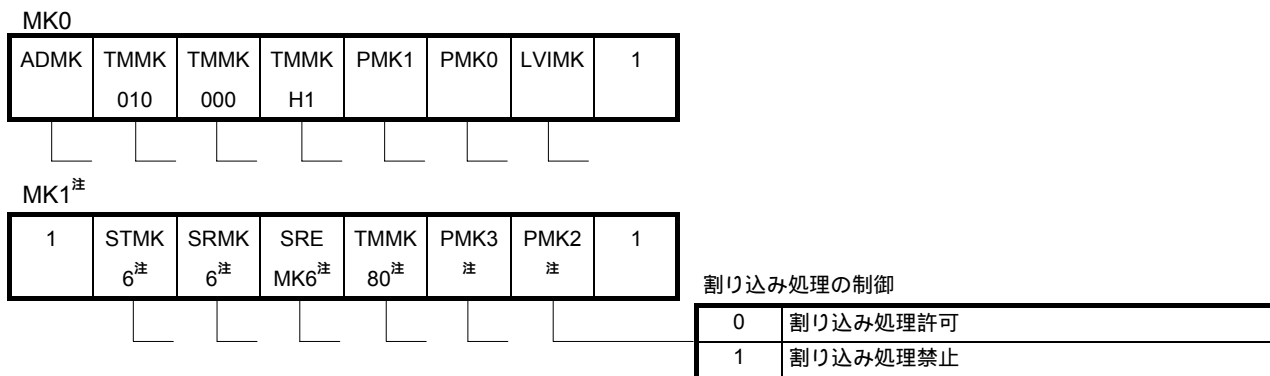


(2) 割り込み処理の許可 / 禁止の設定

割り込みマスク・フラグで、対応する割り込み処理の許可（0） / 禁止（1）を設定します。

割り込み機能を使用する場合は、割り込み要求発生前に、使用する割り込みマスク・フラグをクリア（0）してください。

図4 - 2 割り込みマスク・フラグ・レジスタ0, 1^注（MK0, MK1^注）のフォーマット

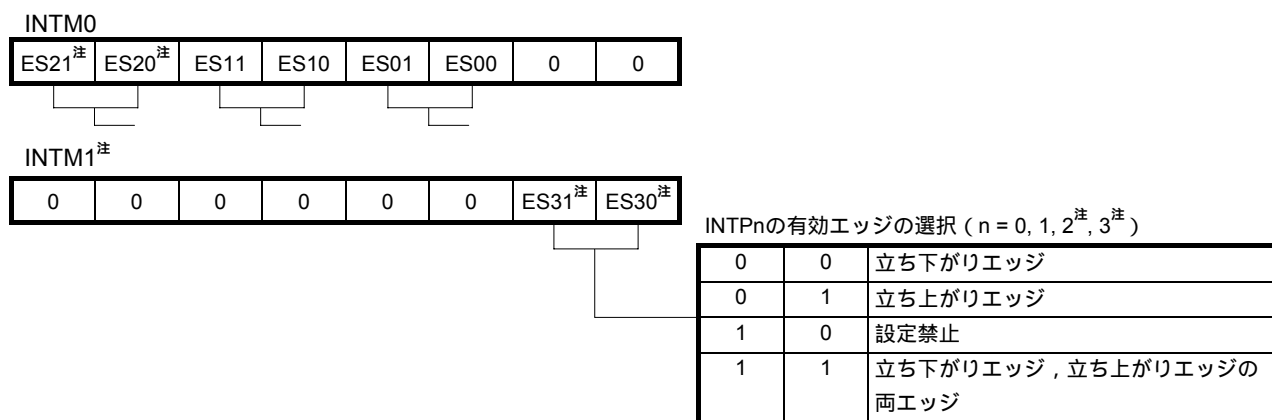


注 78K0S/KA1+, 78K0S/KB1+マイクロコントローラのみ

(3) 外部割り込みの有効エッジの設定

INTM0, INTM1^注で, INTP0, INTP1, INTP2^注, INTP3^注 (外部割り込み)の有効エッジを設定します。

図4-3 外部割り込みモード・レジスタ0, 1^注 (INTM0, INTM1^注)のフォーマット



外部割り込みと有効エッジの設定フラグの組み合わせを, 次に示します。

設定フラグ		外部割り込み
ES31 ^注	ES30 ^注	INTP3 ^注
ES21 ^注	ES20 ^注	INTP2 ^注
ES11	ES10	INTP1
ES01	ES00	INTP0

注 78K0S/KA1+, 78K0S/KB1+マイクロコントローラのみ

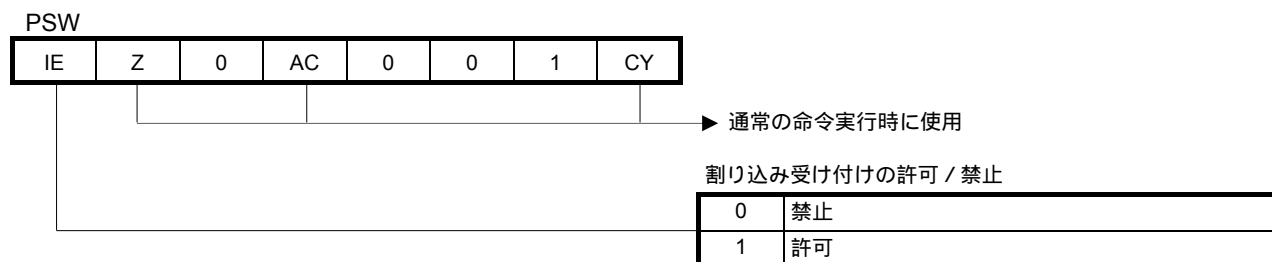
(4) 割り込みの許可/禁止の設定

プログラム・ステータス・ワード (PSW) のIEフラグで, 割り込みの許可 (1) / 禁止 (0) を設定します。IEフラグは, 専用命令 (EI: 割り込み許可, DI: 割り込み禁止) により, 操作可能です。

割り込みマスク・フラグがクリア (0), かつIEフラグがセット (1) されているときに, 割り込み要求が発生し, 割り込み要求フラグがセット (1) された場合に, 割り込みが受け付けられます。

割り込み受け付け時には, PSWは自動的にスタックに退避され, IEフラグはクリア (0) されます。

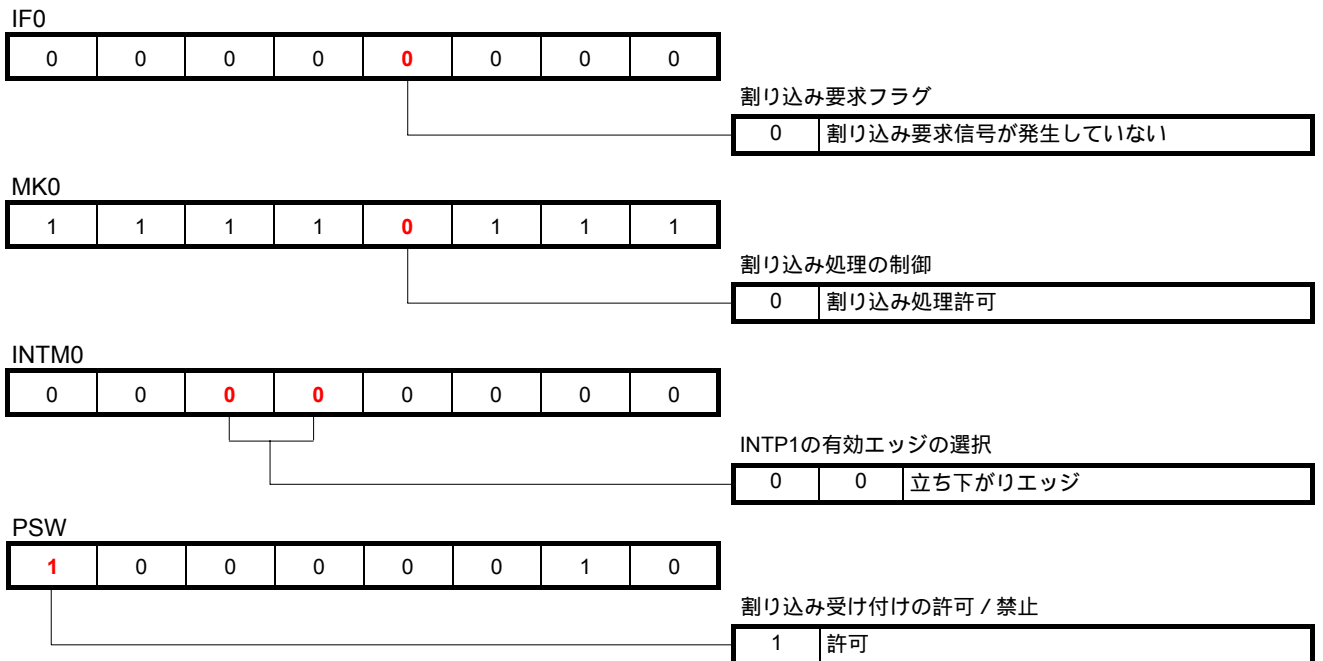
図4-4 プログラム・ステータス・ワード (PSW) のフォーマット



【例】 INTP1の立ち下がりエッジを検出して、割り込み処理を行う（サンプル・プログラムの設定と同内容）

レジスタの設定

- ・INTP1に対応する割り込み要求フラグ（PIF1）を0にクリア
- ・INTP1に対応する割り込みマスク・フラグ（PMK1）を0にクリア
- ・INTP1の有効エッジを「立ち下がりエッジ」に設定
- ・EI命令を実行し、割り込みを許可



備考 INTP1と兼用しているポート（P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ，P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ）は、「入力ポート」に設定してください。また、このサンプル・プログラムでは、マイコンの外側でプルアップ抵抗を接続しない回路構成のため、「内蔵プルアップ抵抗を使用する」に設定してください。ただし、この内蔵プルアップ抵抗の使用については、外部割り込みを使用する場合の必須の設定ではありません。

ポートの設定については、[78K0S/Kx1+ サンプル・プログラム（初期設定） LED点灯のスイッチ制御編 アプリケーション・ノート](#)を参照してください。

アセンブリ言語のプログラム例

(78K0S/KB1+マイクロコントローラのサンプル・プログラムの設定と同内容)

```

XVCT  CSEG  AT      0000H
      DW    RESET_START      ; (00)  RESET
      DW    RESET_START      ; (02)  --
      DW    RESET_START      ; (04)  --
      DW    RESET_START      ; (06)  INTLVI
      DW    RESET_START      ; (08)  INTP0
      DW    INTERRUPT_P1    ; (0A)  INTP1
      DW    RESET_START      ; (0C)  INTTMH1
      DW    RESET_START      ; (0E)  INTTM000
      DW    RESET_START      ; (10)  INTTM010
      DW    RESET_START      ; (12)  INTAD
      DW    RESET_START      ; (14)  --
      DW    RESET_START      ; (16)  INTP2
      DW    RESET_START      ; (18)  INTP3
      DW    RESET_START      ; (1A)  INTTM80
      DW    RESET_START      ; (1C)  INTSRE6
      DW    RESET_START      ; (1E)  INTSR6
      DW    RESET_START      ; (20)  INTST6
      .
      .
      .
XMAIN CSEG  UNIT
RESET_START:
      .
      .
      .
      MOV   P4,      #00000000B      ; P40-P47の出力ラッチLow
      MOV   PU4,     #00001000B      ; P43に内蔵プルアップ抵抗を使用
      MOV   PM4,     #00001000B      ; P43を入力ポートに、P40-P42,
      .                                     ; P44-P47を出力ポートに設定
      .
      .
      .
      MOV   INTM0,  #00000000B      ; INTP1の有効エッジ = 立下りエッジ
      CLR1  PIF1                    ; 無効割り込み要求をクリアしておく
      CLR1  PMK1                    ; INTP1割り込みマスク解除
      EI                             ; ベクタ割り込み許可
      .
      .
      .
MAIN_LOOP:
      NOP
      BR   $MAIN_LOOP      ; MAIN_LOOPへ

INTERRUPT_P1:
      PUSH  AX
      .
      .
      .
      RETI
    
```

DW疑似命令で、初期値(「INTERRUPT_P1」)を記述することにより、割り込み発生時には、「INTERRUPT_P1」のシンボルから割り込み処理開始

INTP1端子と兼用しているP43を入力ポートに設定

INTP1の設定

割り込みを許可

INTP1の有効エッジ(立ち下がりエッジ)検出により、「INTERRUPT_P1」から割り込み処理開始

C言語のプログラム例

(78K0S/KB1+マイクロコントローラのサンプル・プログラムの設定と同内容)

```

#pragma interrupt INTP1 fn_intp1 /* 割り込み関数宣言:INTP1 */
.
.
.
void hdwinit(void) {
.
.
.
P4 = 0b00000000; /* P40-P47の出力ラッチLow */
PU4 = 0b00001000; /* P43に内蔵プルアップ抵抗を使用 */
PM4 = 0b00001000; /* P43を入力ポートに、P40-P42, */
/* P44-P47を出力ポートに設定 */
.
.
.
INTM0 = 0b00000000; /* INTP1の有効エッジ = 立下りエッジ */
PIF1 = 0; /* 無効割り込み要求をクリアしておく */
PMK1 = 0; /* INTP1割り込みマスク解除 */
return;
}

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

while (1) {
NOP();
NOP();
}

__interrupt void fn_intp1() {
unsigned int unChat; /* チャタリング除去タイマ用16ビット変数 */
.
.
.
return;
}
    
```

前処理指令 (#pragma指令)にて、INTP1の割り込み関数(この例では「fn_intp1」)を宣言し、その割り込み関数を__interrupt修飾子で宣言することにより、割り込み発生時には、__interrupt修飾子で宣言した割り込み関数から割り込み処理開始

INTP1端子と兼用しているP43を入力ポートに設定

INTP1の設定

割り込みを許可

INTP1の有効エッジ(立ち下がりエッジ)検出により、「fn_intp1」から割り込み処理開始

4.2 割り込み中の処理

割り込み中の処理について、次に説明します。

(1) アセンブリ言語の場合

アセンブリ言語の割り込み中の処理では、次の動作を行います。

スイッチ入力にて割り込み要求が発生し、割り込み処理が開始されます。

AXレジスタのデータをスタックへ退避します

10 msウエイトします。

PIF1 (INTP1の割り込み要求フラグ) をクリア (0) します

INTP1/P43^注の端子状態を確認します

・INTP1/P43^注 = ロウ・レベル

・INTP1/P43^注 = ハイ・レベル (チャタリング検出)

現在出力中のP2のデータを読み込みます

読み込んだデータをデクリメント (-1) します

のデータのうち、ビット0-2以外の値を0にします

のデータをP2に出力します。

スタックのデータをAXレジスタに復帰します

割り込み処理から復帰します

注 INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ

INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

備考1. 上述の ~ と、次ページの ~ は対応しています。

2. 次ページのプログラム例は、78K0S/KB1+マイクロコントローラのサンプル・プログラムと同内容です。

RAM領域の設定

```

XRAM  DSEG  SADDR
CNT_1: DS    1          ; メジャー・ループ用
CNT_2: DS    1          ; マイナー・ループ用
    
```

割り込み処理で使用するRAMの作業領域(1バイト×2)を確保

割り込み処理

```

INTERRUPT_P1:
    PUSH    AX          ; AXレジスタのデータをスタックへ退避
;----- チャタリング対策の10msウェイト -----
    MOV     CNT_1, #28  ; メジャー・ループ用のカウント値を代入
    NOP
LOOP_1:
    MOV     CNT_2, #70  ; マイナー・ループ用のカウント値を代入
LOOP_2:
    NOP
    DBNZ   CNT_2, $LOOP_2 ; マイナー・ループ
    DBNZ   CNT_1, $LOOP_1 ; メジャー・ループ
    CLR1   PIF1        ; INTP1割り込み要求をクリア
;----- チャタリング検出の判定 -----
    BT     P4.3, $END_INTP1 ; スイッチ入力がないと分岐する
;----- LED点灯処理 -----
    MOV    A, P2        ; 現在の出力値の読み出し
    DEC   A            ; Aレジスタの値を-1
    AND   A, #00000111B ; ビット0-2以外をマスク
    MOV   P2, A        ; LED出力
END_INTP1:
    POP   AX          ; AXレジスタのデータを復帰
    RETI             ; 割り込み処理から復帰
    
```

命令のクロック数

- (a) 6クロック
- (b) 2クロック
- (c) 6クロック
- (d) 2クロック
- (e) 8クロック
- (f) 8クロック

INTP1/P43 = ロウ・レベル

INTP1/P43 = ハイ・レベル(チャタリング検出)

入力データと出力データの対応は次のようになります。

スイッチの入力回数 ^注	P22, P21, P20の出力データ	LED点灯
0	P22 = P21 = P20 = 1	7 全LED消灯
1	P22 = P21 = 1, P20 = 0	6 LED3, LED2: 消灯, LED1: 点灯
2	P22 = 1, P21 = 0, P20 = 1	5 LED3: 消灯, LED2: 点灯, LED1: 消灯
3	P22 = 1, P21 = P20 = 0	4 LED3: 消灯, LED2, LED1: 点灯
4	P22 = 0, P21 = P20 = 1	3 LED3: 点灯, LED2, LED1: 消灯
5	P22 = 0, P21 = 1, P20 = 0	2 LED3: 点灯, LED2: 消灯, LED1: 点灯
6	P22 = P21 = 0, P20 = 1	1 LED3, LED2: 点灯, LED1: 消灯
7	P22 = P21 = P20 = 0	0 全LED点灯

注 8回目以降は、0回目からの点灯パターンの繰り返しになります。

備考1. 上述の ウェイト時間(10ms)は、次の計算式で求めることができます。

・ 1クロック = 1/2 MHz = 0.5 μs

・ のクロック数 = (a) + (b) + { (c) + ((d) + (e)) × 70 + (f) } × 28

↑ (d) + (e)を70回繰り返す

(c) ~ (f)を28回繰り返す

・ のウェイト時間 = [(a) + (b) + { (c) + ((d) + (e)) × 70 + (f) } × 28] × 0.5 μs

$$= [6 + 2 + \{ 6 + (2 + 8) \times 70 + 8 \} \times 28] \times 0.5 \mu s$$

$$= 20,000 \times 0.5 \mu s = 10,000 \mu s = 10 \text{ ms}$$

2. 上述の ~ と、前ページの ~ は対応しています。

(2) C言語の場合

C言語の割り込み処理も、アセンブリ言語と同様な動作を行います。

備考 次のプログラム例は、78K0S/KB1+マイクロコントローラのサンプル・プログラムと同内容です。

割り込み処理

```

__interrupt void fn_intp1(){
    unsigned int unChat;                /* チャタリング除去タイマ用16ビット変数 */

    for (unChat = 0; unChat < 278; unChat++){
        NOP();                          /* 約10msウェイト(チャタリング除去用) */
    }

    PIF1 = 0;                           /* INTp1割り込み要求をクリア */
    if (!P4.3){                          /* 10ms以上SWオンの場合の処理 */
        P2 = (P2 - 1) & 0b00000111;    /* SW入力回数に応じたLED出力 */
    }

    return;
}


```

入力データと出力データの対応は次のようになります。

スイッチの入力回数 ^注	P22, P21, P20の出力データ	LED点灯
0	P22 = P21 = P20 = 1	7 全LED消灯
1	P22 = P21 = 1, P20 = 0	6 LED3, LED2: 消灯, LED1: 点灯
2	P22 = 1, P21 = 0, P20 = 1	5 LED3: 消灯, LED2: 点灯, LED1: 消灯
3	P22 = 1, P21 = P20 = 0	4 LED3: 消灯, LED2, LED1: 点灯
4	P22 = 0, P21 = P20 = 1	3 LED3: 点灯, LED2, LED1: 消灯
5	P22 = 0, P21 = 1, P20 = 0	2 LED3: 点灯, LED2: 消灯, LED1: 点灯
6	P22 = P21 = 0, P20 = 1	1 LED3, LED2: 点灯, LED1: 消灯
7	P22 = P21 = P20 = 0	0 全LED点灯


注 8回目以降は、0回目からの点灯パターンの繰り返しになります。

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アセンブリ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 78K0S/Kx1+でどのように動作するかを説明します。

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

5.1 サンプル・プログラムのビルド

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

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

【コラム】ビルドのエラー


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

[ツール] [コンパイラオプションの設定]を選択してください。

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

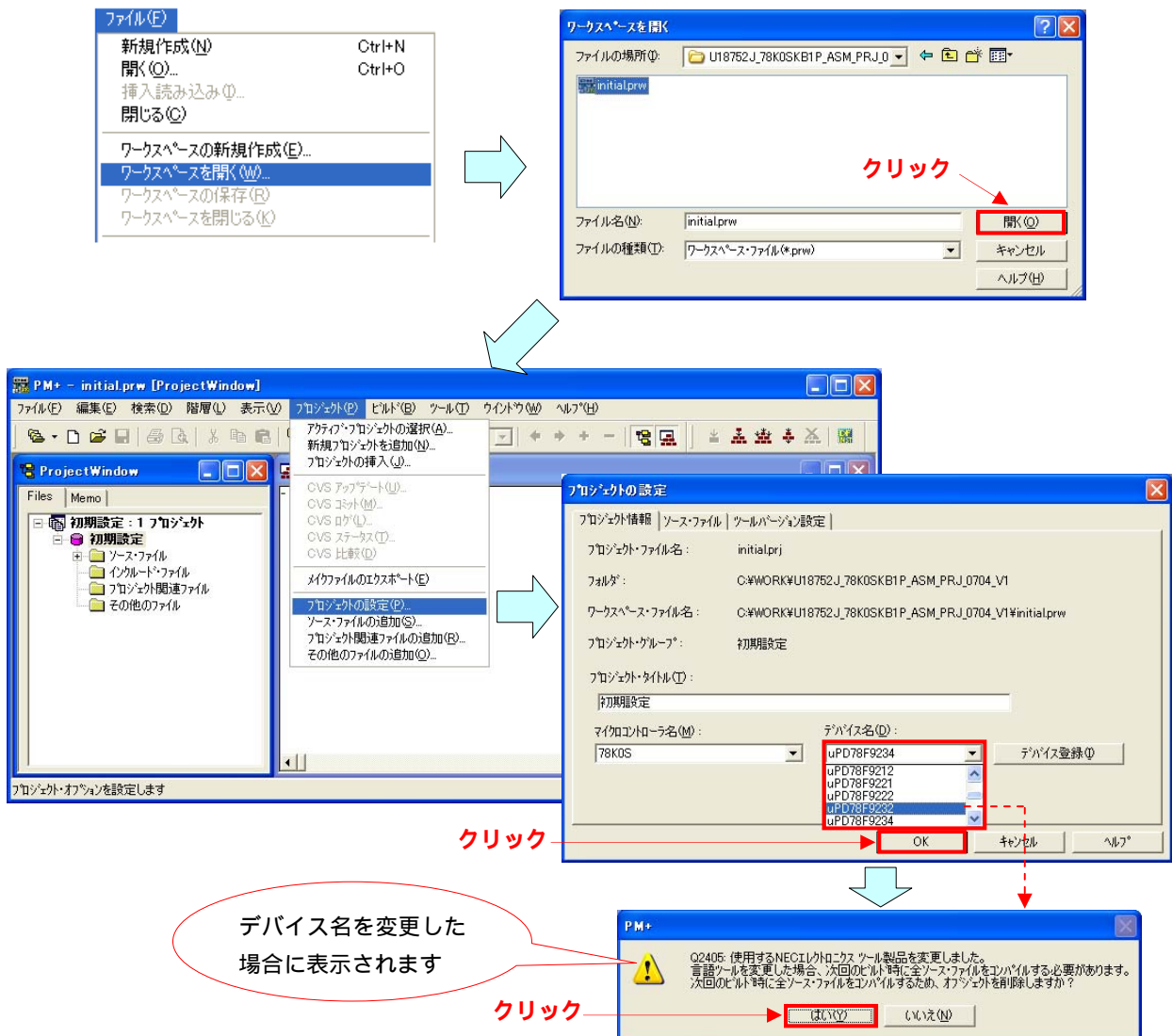
「標準ライブラリ固定領域を使用する」のチェックを外してください（それ以外のチェックは、そのまま）。


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

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

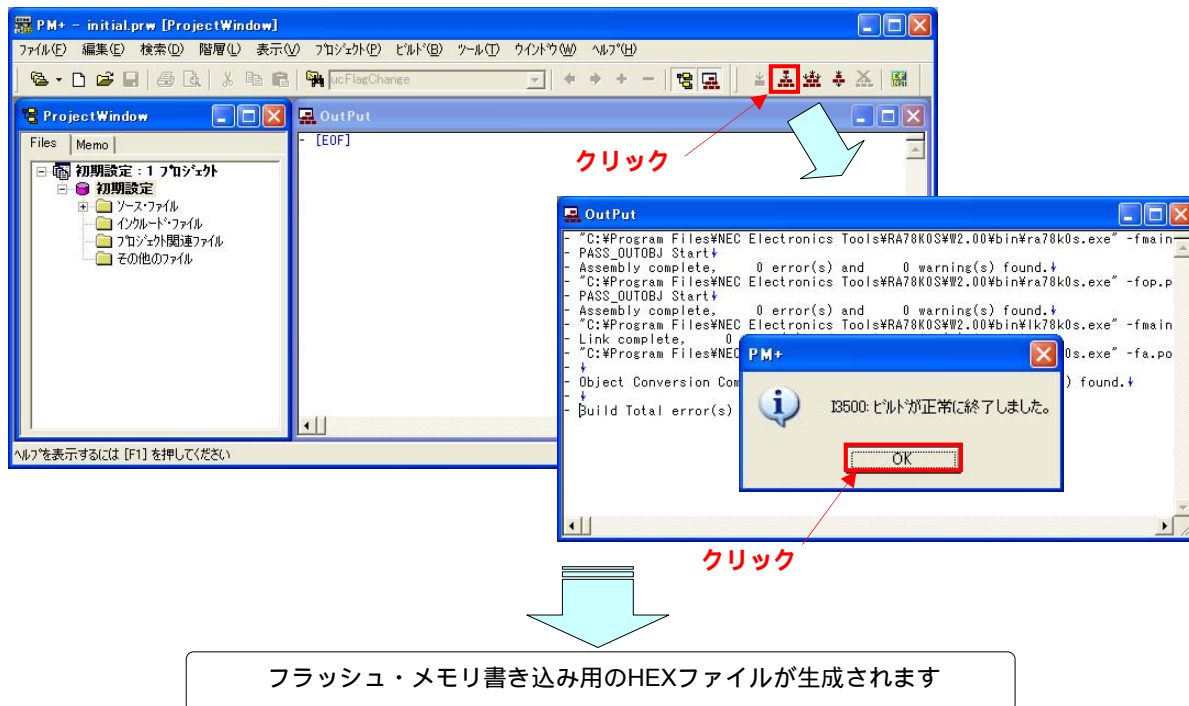
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「int.prw」を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が立ち上がったら、使用するデバイス名を選択（デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択）し、[OK] ボタンをクリックしてください。

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



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

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




5.2 SM+での動作

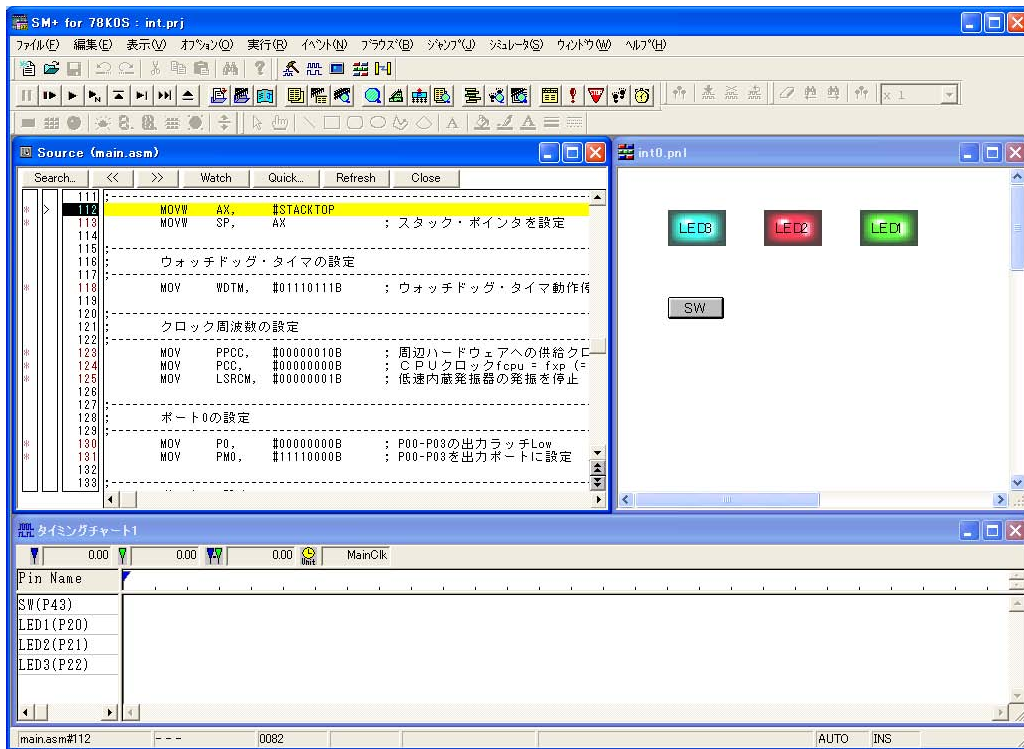
ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。


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

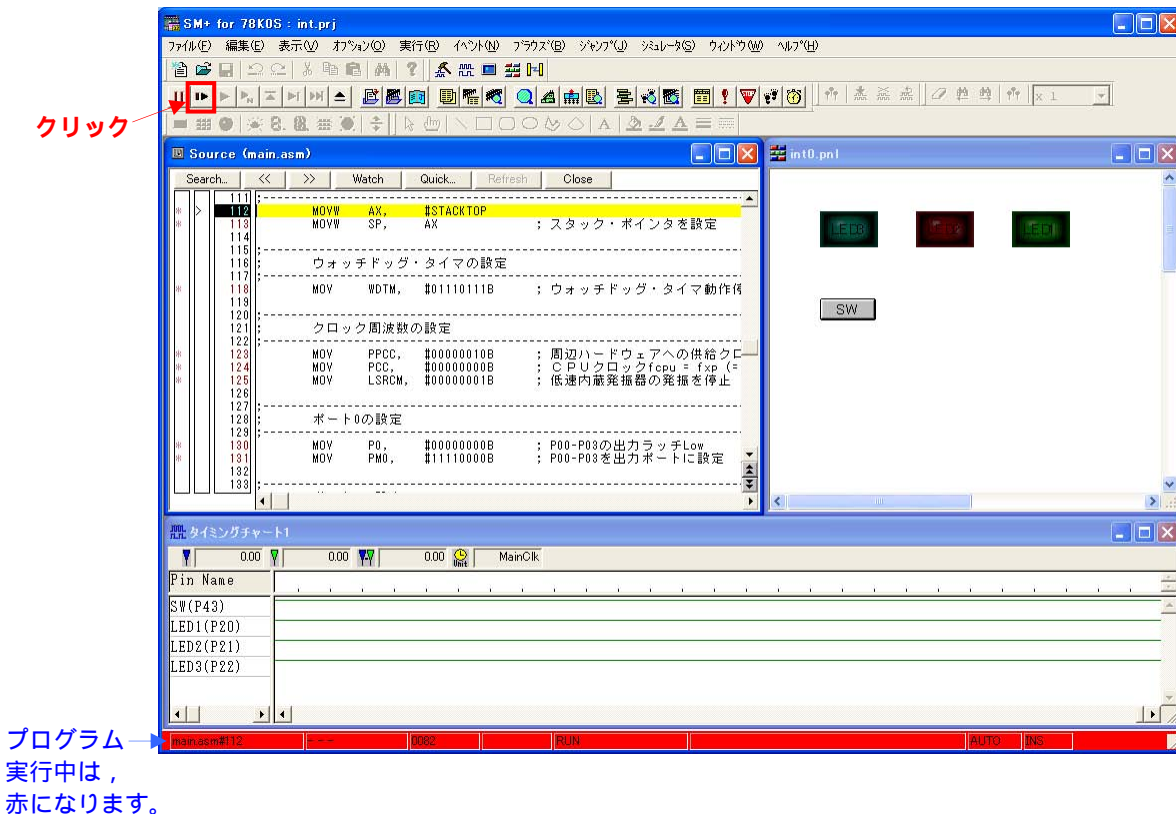
- (1) SM+ for 78K0S/Kx1+ W1.02をPM+ Ver6.30の環境で使用するために、次のサイトにあるPDFファイルを参照して、「外部ツールの登録」を行い、SM+を起動してください。

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

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


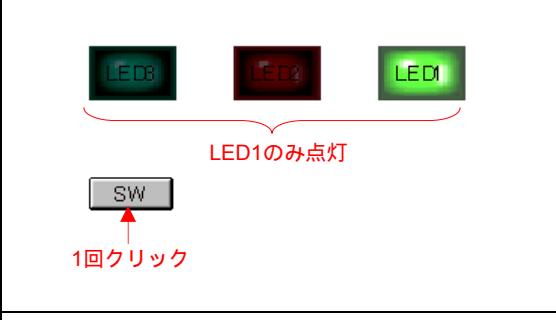
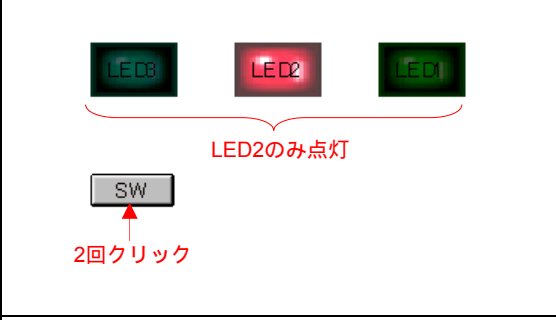
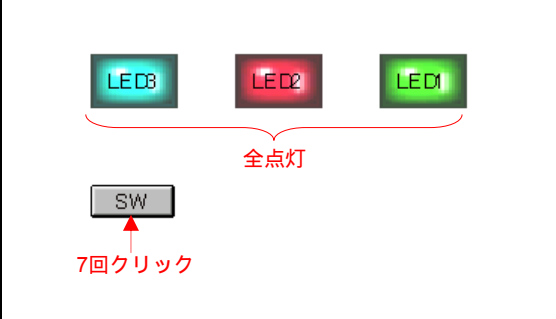


- (3) （「リスタート」ボタン）をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。



- (4) プログラム実行中に，入出力パネル・ウインドウ上の [SW] ボタンをクリックしてください。
 [SW] ボタンの入力回数により，入出力パネル・ウインドウ上の [LED1] ~ [LED3] の点灯およびタイミ
 ング・チャート・ウインドウ上の波形が変化することを確認してください。

注

入出力パネル・ウインドウ	タイミング・チャート・ウインドウ										
 <p>全消灯</p> <p>クリックしない</p>	<table border="1"> <thead> <tr> <th>Pin Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>SW (P43)</td> <td></td> </tr> <tr> <td>LED1 (P20)</td> <td></td> </tr> <tr> <td>LED2 (P21)</td> <td></td> </tr> <tr> <td>LED3 (P22)</td> <td></td> </tr> </tbody> </table> <p>P20 : H出力 P21 : H出力 P22 : H出力</p>	Pin Name		SW (P43)		LED1 (P20)		LED2 (P21)		LED3 (P22)	
Pin Name											
SW (P43)											
LED1 (P20)											
LED2 (P21)											
LED3 (P22)											
 <p>LED1のみ点灯</p> <p>1回クリック</p>	<table border="1"> <thead> <tr> <th>Pin Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>SW (P43)</td> <td></td> </tr> <tr> <td>LED1 (P20)</td> <td></td> </tr> <tr> <td>LED2 (P21)</td> <td></td> </tr> <tr> <td>LED3 (P22)</td> <td></td> </tr> </tbody> </table> <p>P20 : L出力 P21 : H出力 P22 : H出力</p>	Pin Name		SW (P43)		LED1 (P20)		LED2 (P21)		LED3 (P22)	
Pin Name											
SW (P43)											
LED1 (P20)											
LED2 (P21)											
LED3 (P22)											
 <p>LED2のみ点灯</p> <p>2回クリック</p>	<table border="1"> <thead> <tr> <th>Pin Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>SW (P43)</td> <td></td> </tr> <tr> <td>LED1 (P20)</td> <td></td> </tr> <tr> <td>LED2 (P21)</td> <td></td> </tr> <tr> <td>LED3 (P22)</td> <td></td> </tr> </tbody> </table> <p>P20 : H出力 P21 : L出力 P22 : H出力</p>	Pin Name		SW (P43)		LED1 (P20)		LED2 (P21)		LED3 (P22)	
Pin Name											
SW (P43)											
LED1 (P20)											
LED2 (P21)											
LED3 (P22)											
.										
 <p>全点灯</p> <p>7回クリック</p>	<table border="1"> <thead> <tr> <th>Pin Name</th> <th></th> </tr> </thead> <tbody> <tr> <td>SW (P43)</td> <td></td> </tr> <tr> <td>LED1 (P20)</td> <td></td> </tr> <tr> <td>LED2 (P21)</td> <td></td> </tr> <tr> <td>LED3 (P22)</td> <td></td> </tr> </tbody> </table> <p>P20 : L出力 P21 : L出力 P22 : L出力</p>	Pin Name		SW (P43)		LED1 (P20)		LED2 (P21)		LED3 (P22)	
Pin Name											
SW (P43)											
LED1 (P20)											
LED2 (P21)											
LED3 (P22)											

注 8回目以降は，0回目からの点灯パターンの繰り返しになります。

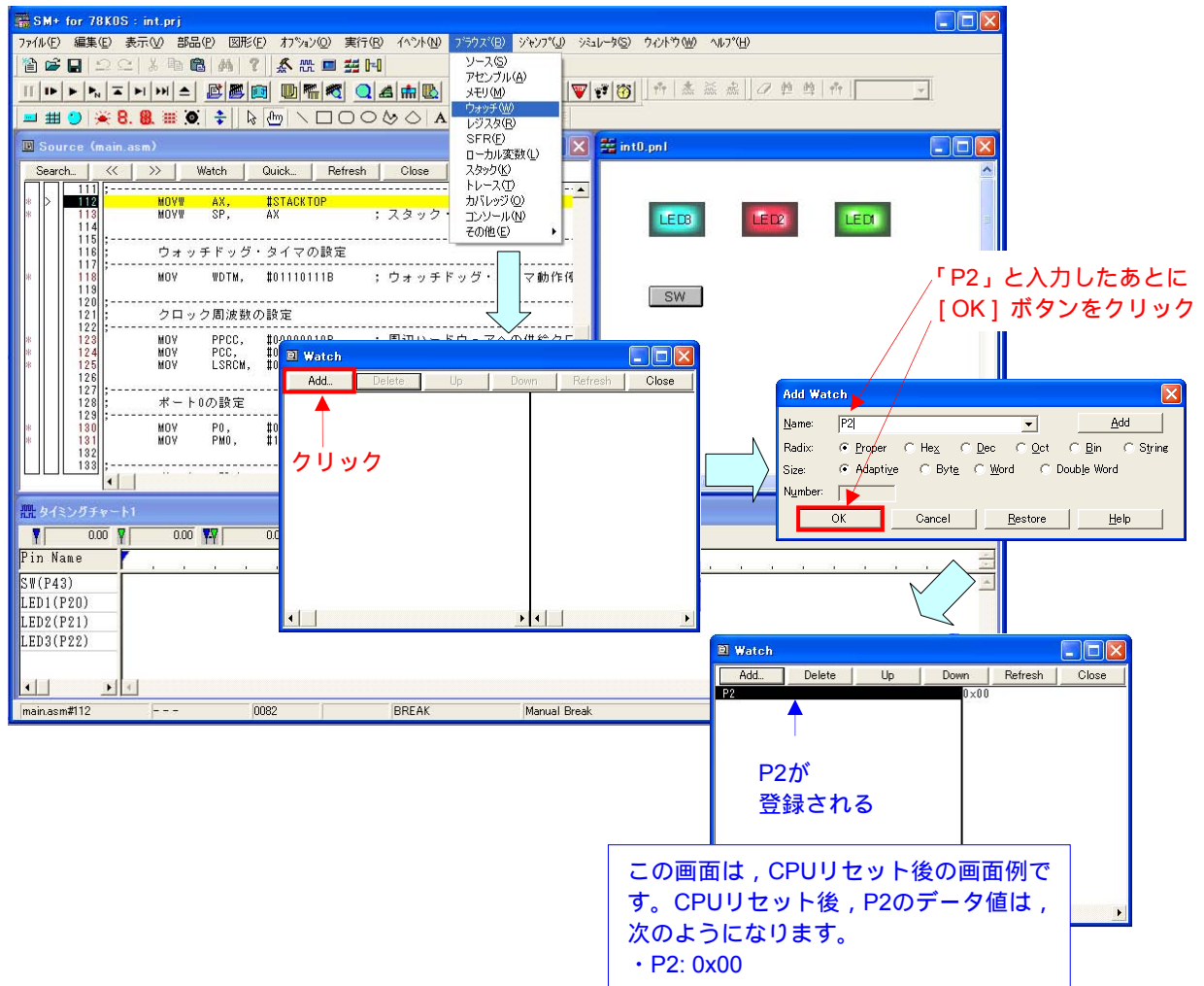
備考 H：ハイ・レベル，L：ロウ・レベル

【補足1】 SM+のウォッチ機能を使用することにより，ポート2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「P2」と入力し，[OK] ボタンをクリックすると，[Watch] ウィンドウに，「P2」が登録され，[Add watch] ウィンドウが閉じられます。




プログラムを実行し，入出力パネル・ウィンドウ上の [SW] ボタンをクリックしてください。

[SW] ボタンの入力回数により，[Watch] ウィンドウ上のP2のデータ値が変化することを確認してください。

[SW] ボタンの入力回数 ^注	[Watch] ウィンドウのデータ値
0	P2: 0x07
1	P2: 0x06
2	P2: 0x05
3	P2: 0x04
4	P2: 0x03
5	P2: 0x02
6	P2: 0x01
7	P2: 0x00

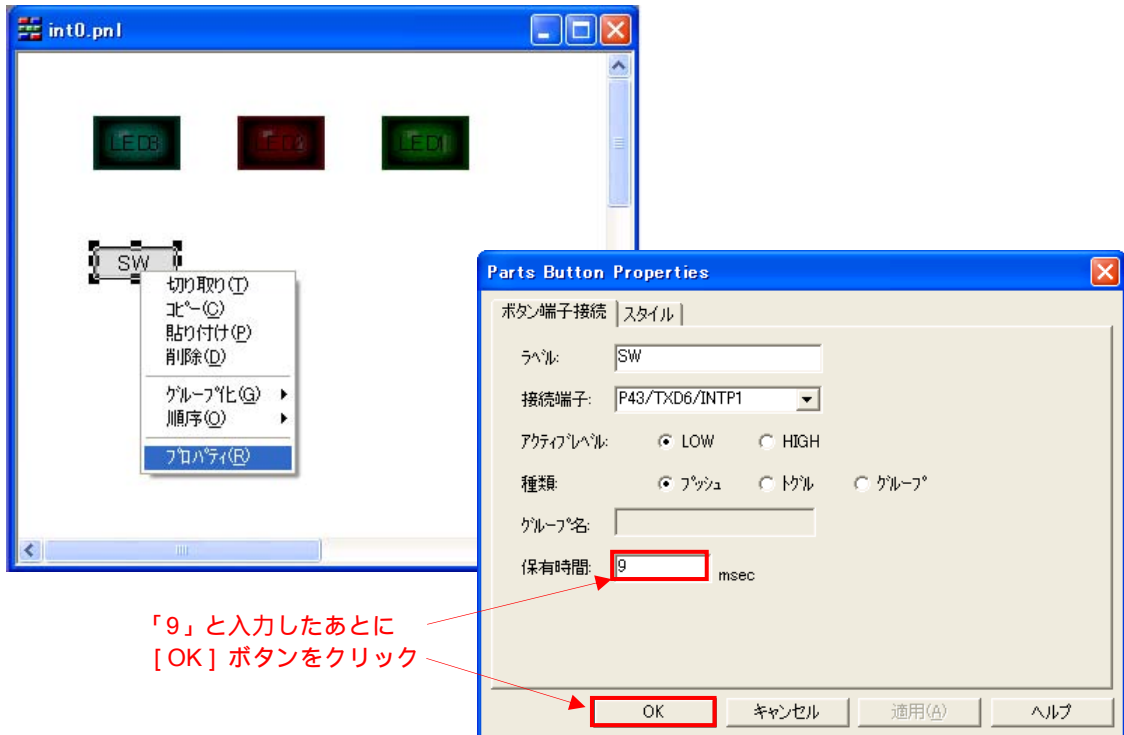
注 8回目以降は，0回目からの点灯パターンの繰り返しになります。

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


ツール・バーの  を選択してください。

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

保有時間に「9」を入力し，[OK] ボタンをクリックしてください。



「9」と入力したあとに
[OK] ボタンをクリック

ツール・バーの  を選択してください。

プログラムを実行し，[SW] ボタンをクリックしてください。ボタンの保有時間が9 msのため，[SW] ボタンをクリックしても，チャタリングであると判定され，LED点灯パターンは変化しません。

第6章 関連資料

資料名		和文 / 英文
78K0S/KU1+ ユーザーズ・マニュアル		PDF
78K0S/KY1+ ユーザーズ・マニュアル		PDF
78K0S/KA1+ ユーザーズ・マニュアル		PDF
78K0S/KB1+ ユーザーズ・マニュアル		PDF
78K0Sシリーズ 命令編 ユーザーズ・マニュアル		PDF
RA78K0S アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
CC78K0S Cコンパイラ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		PDF
SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル		PDF
フラッシュ書き込み簡単マニュアル (MINICUBE2編) インフォメーション	78K0S/KU1+	PDF
	78K0S/KY1+	PDF
	78K0S/KA1+	PDF
	78K0S/KB1+	PDF
78K0S/Kx1+	サンプル・プログラム スタートアップ・ガイド	PDF
アプリケーション・ノート	サンプル・プログラム (初期設定) LED点灯のスイッチ制御編	PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0S/KB1+マイクロコントローラのソース・プログラムを次に示します。

```
main.asm (アセンブリ言語版)
;*****
;
; NEC Electronics      78K0S/KB1+シリーズ
;
;*****
; 78K0S/KB1+シリーズ      サンプル・プログラム
;*****
; 割り込み
;*****
; 【履歴】
; 2007.6.--      新規作成
;*****
;
; 【概要】
;
;本サンプルプログラムは、割り込み機能の使用例を示すものである。
;スイッチ入力の立ち下がりエッジを検出して割り込みを発生させ、スイッチ入力回数に
;応じたLED点灯パターンを表示する。ここで、割り込み直後に10msのチャタリング除去
;時間を設けているので、10ms以内のチャタリングをスイッチ入力としてカウントする
;ことはない。
;
;
; <主な設定内容>
;
; ・ベクタ・テーブルの設定
; ・スタック・ポインタの設定
; ・ウォッチドッグ・タイマの動作停止
; ・CPUクロック周波数を2MHzに設定
; ・外部割り込みINTP1の有効エッジ：立ち下りエッジに設定
; ・スイッチ入力時のチャタリング検出時間 = 10ms;
;
; <スイッチ入力回数とLED点灯パターン>
;
```

```

; +-----+
; | SW入力回数 | LED3 | LED2 | LED1 |
; | (P43)      | (P22) | (P21) | (P20) |
; |-----|-----|
; | 0回      | OFF  | OFF  | OFF  |
; | 1回      | OFF  | OFF  | ON   |
; | 2回      | OFF  | ON   | OFF  |
; | 3回      | OFF  | ON   | ON   |
; | 4回      | ON   | OFF  | OFF  |
; | 5回      | ON   | OFF  | ON   |
; | 6回      | ON   | ON   | OFF  |
; | 7回      | ON   | ON   | ON   |
; +-----+
; # 8回目以降は0回からの繰り返し
;
;
; 【ポート入出力の設定】
;
; 入力ポート：P43
; 出力ポート：P00-P03, P20-P23, P30-P33, P40-P42, P44-P47, P120-P123, P130
; 未使用のポートは全て出力ポートに設定しておく
;
;*****
;=====
;
; ベクタ・テーブルの設定
;
;=====
XVCT CSEG AT 0000H
      DW RESET_START ;(00) RESET
      DW RESET_START ;(02) --
      DW RESET_START ;(04) --
      DW RESET_START ;(06) INTLVI
      DW RESET_START ;(08) INTPO
      DW INTERRUPT_P1 ;(0A) INTP1
      DW RESET_START ;(0C) INTTMH1
      DW RESET_START ;(0E) INTTM000
      DW RESET_START ;(10) INTTM010
      DW RESET_START ;(12) INTAD
      DW RESET_START ;(14) --
      DW RESET_START ;(16) INTP2

```

```

DW    RESET_START      ;(18)  INTP3
DW    RESET_START      ;(1A)  INTTM80
DW    RESET_START      ;(1C)  INTSRE6
DW    RESET_START      ;(1E)  INTSR6
DW    RESET_START      ;(20)  INTST6

;=====
;
;   RAMの定義
;
;=====
XRAM  DSEG    SADDR
CNT_1:    DS      1      ; メジャー・ループ用
CNT_2:    DS      1      ; マイナー・ループ用

;=====
;
;   スタック領域の確保
;
;=====
XSTK  DSEG    AT      0FEE0H
STACKEND:
        DS      20H      ; スタック領域を32バイト確保
STACKTOP:      ; スタック領域の先頭アドレス = FF00H

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

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

```

```
-----  
;  
; クロック周波数の設定  
-----  
MOV    PPCC,    #00000010B    ; 周辺ハードウェアへの供給クロック fxp = fx/4 (= 2MHz)  
MOV    PCC,     #00000000B    ; CPUクロック fcpu = fxp (= 2MHz)  
MOV    LSRCM,   #00000001B    ; 低速内蔵発振器の発振を停止  
  
-----  
;  
; ポート0の設定  
-----  
MOV    P0,      #00000000B    ; P00-P03の出力ラッチLow  
MOV    PM0,     #11110000B    ; P00-P03を出力ポートに設定  
  
-----  
;  
; ポート2の設定  
-----  
MOV    P2,      #00000111B    ; P20-P22の出力ラッチHigh(LED1-LED3消灯)、P23の出力ラッチLow  
MOV    PM2,     #11110000B    ; P20-P23を出力ポートに設定  
  
-----  
;  
; ポート3の設定  
-----  
MOV    P3,      #00000000B    ; P30-P33の出力ラッチLow  
MOV    PM3,     #11110000B    ; P30-P33を出力ポートに設定  
  
-----  
;  
; ポート4の設定  
-----  
MOV    P4,      #00000000B    ; P40-P47の出力ラッチLow  
MOV    PU4,     #00001000B    ; P43に内蔵プルアップ抵抗を使用  
MOV    PM4,     #00001000B    ; P43を入力ポートに、P40-P42,P44-P47を出力ポートに設定  
  
-----  
;  
; ポート12の設定  
-----  
MOV    P12,     #00000000B    ; P120-P123の出力ラッチLow  
MOV    PM12,    #11110000B    ; P120-P123を出力ポートに設定  
  
-----  
;  
; ポート13の設定  
-----  
MOV    P13,     #00000001B    ; P130の出力High
```

```

;-----
;   割り込みの設定
;-----
MOV     INTMO, #00000000B    ; INTP1の有効エッジ = 立下りエッジ
CLR1    PIF1                ; 無効割り込み要求をクリアしておく
CLR1    PMK1                ; INTP1割り込みマスク解除

EI      ; ベクタ割り込み許可

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

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

;----- チャタリング対策の10msウェイト -----
MOV     CNT_1, #28         ; メジャー・ループ用のカウント値を代入
NOP
LOOP_1:
MOV     CNT_2, #70         ; マイナー・ループ用のカウント値を代入
LOOP_2:
NOP
DBNZ   CNT_2, $LOOP_2     ; マイナー・ループ
DBNZ   CNT_1, $LOOP_1     ; メジャー・ループ

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

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

```



```
;----- LED点灯処理 -----  
MOV    A,    P2          ; 現在の出力値の読み出し  
DEC    A              ; Aレジスタの値を-1  
AND    A,    #00000111B ; ビット0-2以外をマスク  
MOV    P2,    A        ; LED出力  
  
END_INT1:  
POP    AX              ; AXレジスタのデータを復帰  
RET I                  ; 割り込み処理から復帰  
  
end
```

main.c (C言語版)

/******

NEC Electronics 78K0S/KB1+シリーズ

78K0S/KB1+シリーズ サンプル・プログラム

割り込み

【履歴】

2007.6.-- 新規作成

【概要】

本サンプルプログラムは、割り込み機能の使用例を示すものである。
 スイッチ入力の立ち下がりエッジを検出して割り込みを発生させ、スイッチ入力回数に応じたLED点灯パターンを表示する。ここで、割り込み直後に10msのチャタリング除去時間を設けているので、10ms以内のチャタリングをスイッチ入力としてカウントすることはない。

< 主な設定内容 >

- ・ 割り込みで起動される関数の宣言: INTP1 -> fn_intp1()
- ・ ウォッチドッグ・タイマの動作停止
- ・ CPUクロック周波数を2MHzに設定
- ・ 外部割り込みINTP1の有効エッジ: 立ち下りエッジに設定
- ・ スイッチ入力時のチャタリング検出時間 = 10ms

< スイッチ入力回数とLED点灯パターン >

SW入力回数	LED3	LED2	LED1
(P43)	(P22)	(P21)	(P20)
0回	OFF	OFF	OFF
1回	OFF	OFF	ON
2回	OFF	ON	OFF
3回	OFF	ON	ON
4回	ON	OFF	OFF

```
| 5回 | ON | OFF | ON |
| 6回 | ON | ON | OFF |
| 7回 | ON | ON | ON |
```

+-----+

8回目以降は0回からの繰り返し

【ポート入出力の設定】

入力ポート：P43

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

未使用のポートは全て出力ポートに設定しておく

*****/

/*=====

前処理指令（#pragma指令）

=====*/

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

*****/

リセット解除後の初期化処理

*****/

```
void hdwinit(void){
```

/*-----

ウォッチドッグ・タイマの設定

-----*/

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

/*-----

クロック周波数の設定

-----*/

```
PPCC = 0b00000010; /* 周辺ハードウェアへの供給クロック fxp = fx/4 (= 2MHz) */
```

```
PCC = 0b00000000; /* CPUクロック fcpu = fxp (= 2MHz) */
```

LSRCM = 0b00000001; /* 低速内蔵発振器の発振を停止 */

/*-----*/

ポート0の設定

-----*/

P0 = 0b00000000; /* P00-P03の出力ラッチLow */

PM0 = 0b11110000; /* P00-P03を出力ポートに設定 */

/*-----*/

ポート2の設定

-----*/

P2 = 0b00000111; /* P20-P22の出力ラッチHigh(LED1-LED3消灯)、P23の出力ラッチLow */

PM2 = 0b11110000; /* P20-P23を出力ポートに設定 */

/*-----*/

ポート3の設定

-----*/

P3 = 0b00000000; /* P30-P33の出力ラッチLow */

PM3 = 0b11110000; /* P30-P33を出力ポートに設定 */

/*-----*/

ポート4の設定

-----*/

P4 = 0b00000000; /* P40-P47の出力ラッチLow */

PU4 = 0b00001000; /* P43に内蔵プルアップ抵抗を使用 */

PM4 = 0b00001000; /* P43を入力ポートに、P40-P42,P44-P47を出力ポートに設定 */

/*-----*/

ポート12の設定

-----*/

P12 = 0b00000000; /* P120-P123の出力ラッチLow */

PM12 = 0b11110000; /* P120-P123を出力ポートに設定 */

/*-----*/

ポート13の設定

-----*/

P13 = 0b00000001; /* P130の出力High */

/*-----*/

割り込みの設定

-----*/

INTM0 = 0b00000000; /* INTP1の有効エッジ = 立下りエッジ */

PIF1 = 0; /* 無効割り込み要求をクリアしておく */

```
    PMK1 = 0;                                /* INTP1割り込みマスク解除 */

    return;
}

/*****

    メイン・ループ

*****/

void main(void){

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

    while (1){
        NOP();
        NOP();
    }
}

/*****

    外部割り込みINTP1

*****/

__interrupt void fn_intp1(){
    unsigned int unChat;    /* チャタリング除去タイマ用16ビット変数 */

    for (unChat = 0; unChat < 278; unChat++){    /* 約10msウェイト(チャタリング除去用) */
        NOP();
    }

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

    if (!P4.3){                                /* 10ms以上SWオンの場合の処理 */
        P2 = (P2 - 1) & 0b00000111;    /* SW入力回数に応じたLED出力 */
    }

    return;
}
```

op.asm (アセンブリ言語版とC言語版共通)

```

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

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

end

```

付録B 改版履歴

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

版 数	発行年月	改版箇所	改版内容
第1版	June 2007	-	-
第2版	July 2008	p.18	第5章 システム・シミュレータ SM+での動作確認 ・注意文中の（2007年6月現在）を（2008年7月現在）に変更
		pp.18-20	5.1 サンプル・プログラムのビルドを変更
		p.20	5.2 SM+での動作 ・（1）を追加
		p.25	第6章 関連資料 ・フラッシュ書き込み簡単マニュアル（MINICUBE2編） インフォメーションを追加 ・78K0S/Kx1 サンプル・プログラム スタートアップ・ガイド アプリケーション・ノートにリンク先を追加

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

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