

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（初期設定）

LED点灯のスイッチ制御編

この資料は、サンプル・プログラムの「初期設定」の動作概要と、マイコンの基本的な初期設定を説明したものです。サンプル・プログラムでは、クロック周波数の選択や入出力ポートの選択など、マイコンの基本的な初期設定を行ったあとに、2つのスイッチ入力と3つのLED点灯の制御を行います。

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
第2章 回路図 ...	4
2.1 回路図 ...	4
2.2 周辺ハードウェア ...	4
第3章 ソフトウェアについて ...	5
3.1 ファイル構成 ...	5
3.2 使用する内蔵周辺機能 ...	6
3.3 初期設定と動作概要 ...	6
3.4 フロー・チャート ...	7
第4章 設定方法について ...	8
4.1 オプション・バイトの設定 ...	8
4.2 ベクタ・テーブルの設定 ...	12
4.3 スタック・ポインタの設定 ...	13
4.4 ウォッチドッグ・タイマの設定 ...	14
4.5 クロックの設定 ...	17
4.6 ポートの設定 ...	23
4.7 メイン処理 ...	27
第5章 システム・シミュレータ SM+での動作確認 ...	29
5.1 サンプル・プログラムのビルド ...	29
5.2 SM+での動作 ...	31
第6章 関連資料 ...	35
付録A プログラム・リスト ...	36
付録B 改版履歴 ...	45

資料番号 U18752JJ2V0AN00（第2版）

発行年月 July 2008 NS

- 本資料に記載されている内容は2008年7月現在のものです、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がその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

第1章 概 要

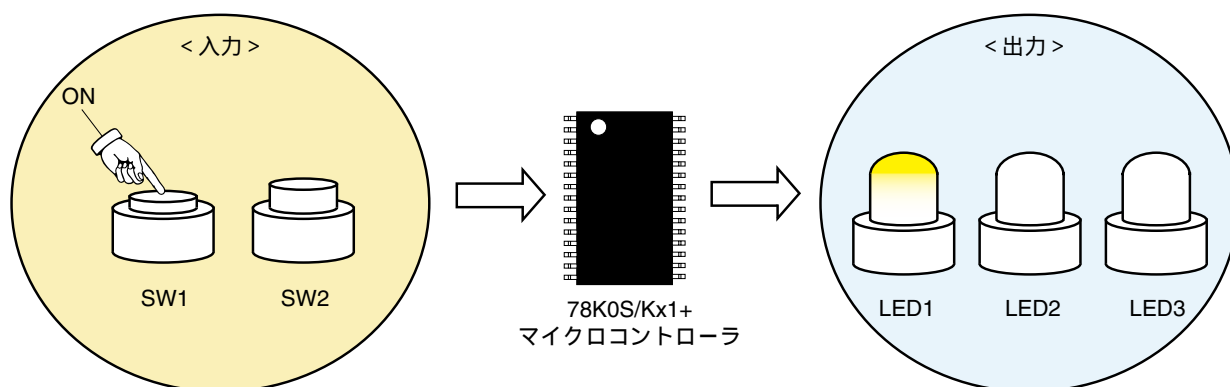
このサンプル・プログラムでは、オプション・バイトの設定、クロック周波数の選択、ポート入出力の設定など、78K0S/Kx1+マイクロコントローラの基本的な初期設定を行います。また、初期設定完了後のメイン処理動作では、2つのスイッチ入力により、3つのLED点灯を制御します。

(1) 初期設定の主な内容

- ・システム・クロック・ソースとして、高速内蔵発振器を選択
- ・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

(2) メイン処理動作の内容

78K0S/Kx1+マイクロコントローラにて、スイッチ入力(SW1, SW2)を検出し、LED点灯(LED1, LED2, LED3)を制御します。



スイッチ入力		LED出力		
SW1	SW2	LED1	LED2	LED3
OFF	OFF	OFF	OFF	OFF
ON	OFF	ON	OFF	OFF
OFF	ON	OFF	ON	OFF
ON	ON	OFF	OFF	ON

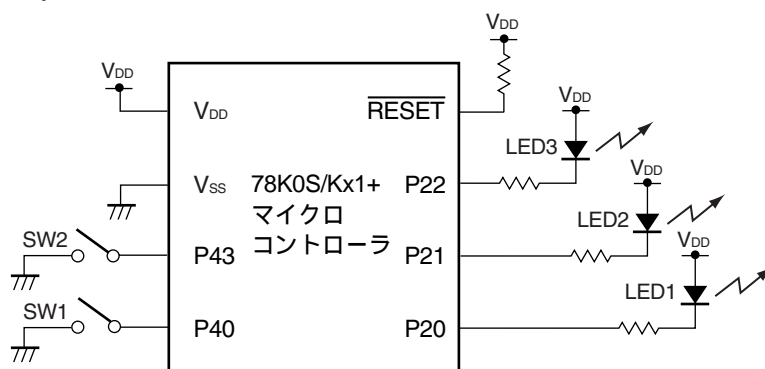
注意 デバイス使用上の注意事項については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

第2章 回路図

この章では、このサンプル・プログラムで使用する場合の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。



- 注意1. AV_{REF}端子はV_{DD}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. AV_{SS}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AV_{REF}, AV_{SS}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 (SW1, SW2)

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)			
initial.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
initial.prj	統合開発環境 PM+用プロジェクト・ファイル			
initial.pri initial.prs initial.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	
initial0.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	
initial0.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。

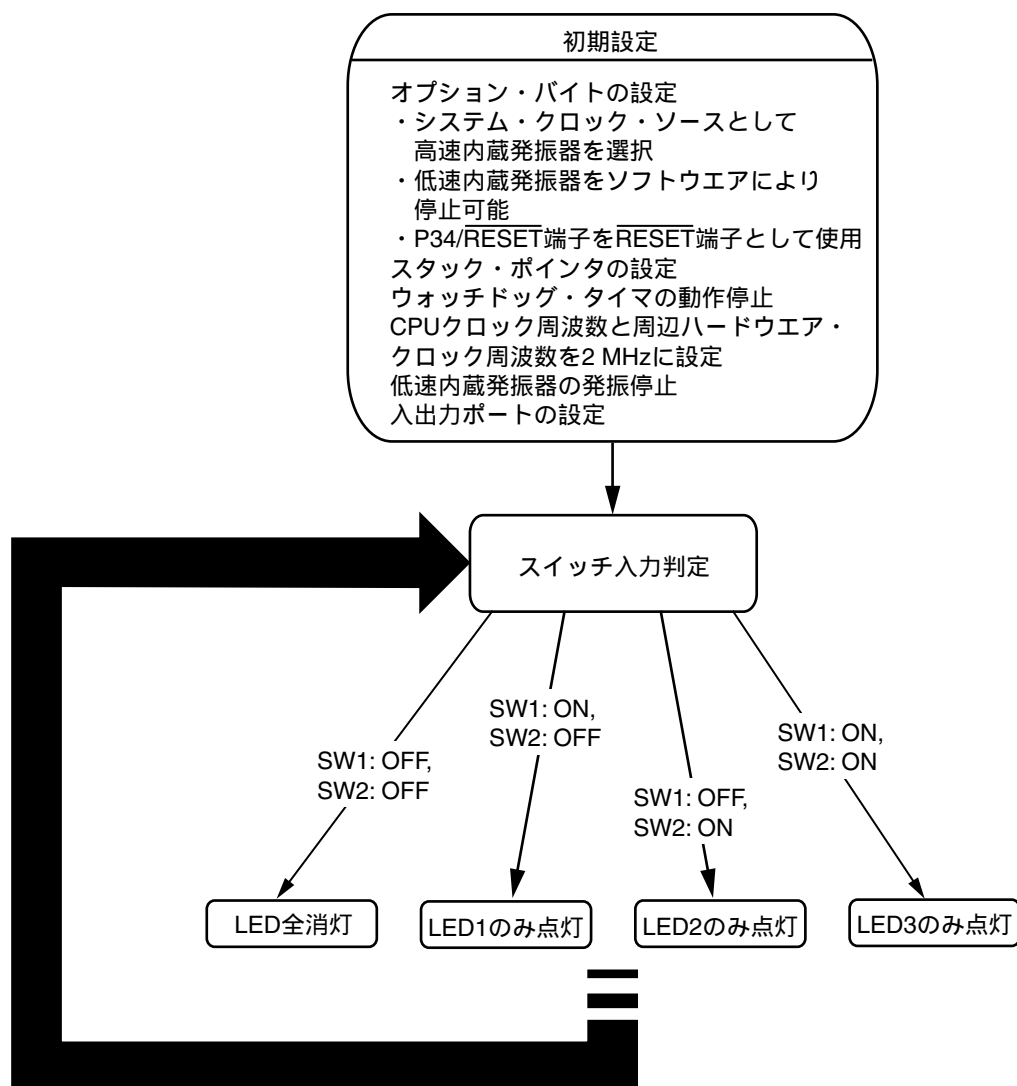
- ・入力ポート（スイッチ入力用）：P40, P43
- ・出力ポート（LED点灯用）：P20, P21, P22

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などを行います。

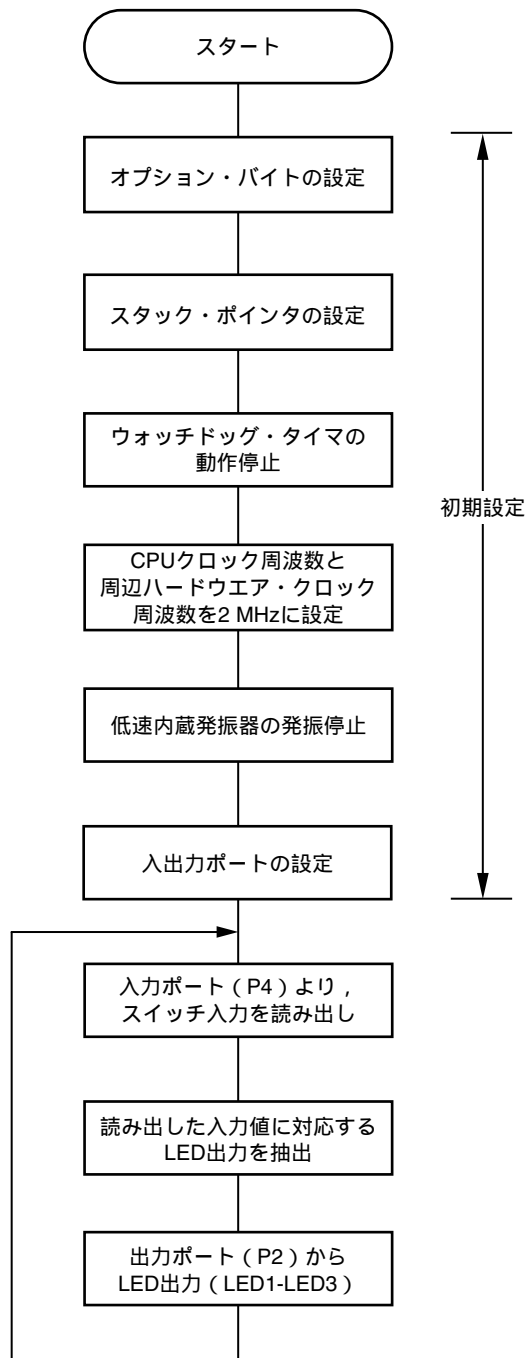
初期設定完了後は、2つのスイッチ入力（SW1, SW2）の組み合わせに応じて、3つのLED（LED1, LED2, LED3）の点灯を制御します。

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。



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。



第4章 設定方法について

この章では、オプション・バイト、ベクタ・テーブル、スタック・ポインタ、ウォッチドッグ・タイマ、クロック周波数、ポートの設定、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、各製品のユーザーズ・マニュアル（[78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)）を参照してください。

アセンブラ命令については、[78K/0Sシリーズ 命令編 ユーザーズ・マニュアル](#)を参照してください。

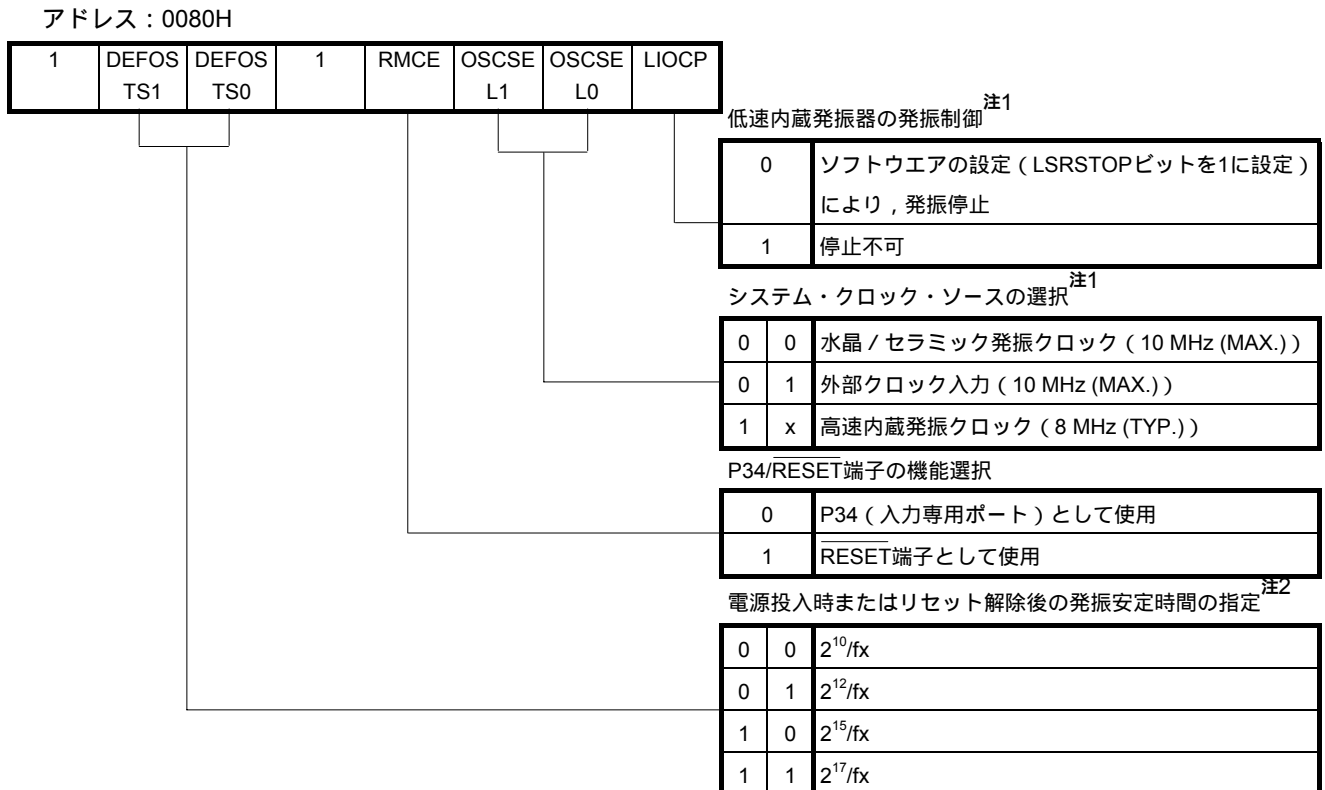
4.1 オプション・バイトの設定

オプション・バイトの設定は、必須です。オプション・バイトで、次の項目を設定します。

- (1) システム・クロック・ソースの選択
- (2) 低速内蔵発振器の発振制御
- (3) P34/RESET端子の機能選択
- (4) 電源投入時またはリセット解除後の発振安定時間の指定（水晶/セラミック発振使用時のみ）

このサンプル・プログラムでは、後述の【例 1】の内容で、オプション・バイトを設定しています。

図4-1 オプション・バイトのフォーマット



注1. 低速内蔵発振器の発振停止やウォッチドッグ・タイマの動作停止、クロック周波数の設定は、リセット解除後に行います。詳細については、[4.4 ウォッチドッグ・タイマの設定](#)、[4.5 クロックの設定](#)を参照してください。

（注2，注意，備考は次ページにあります）

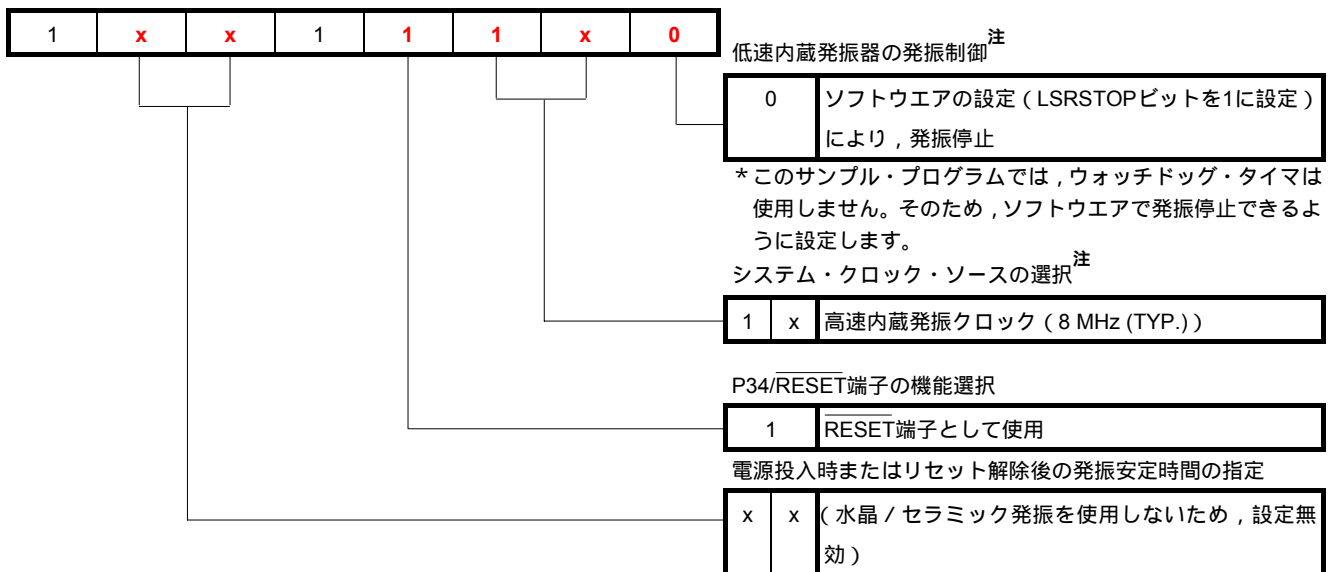
注2. システム・クロックに高速内蔵発振クロックまたは外部クロック入力を選択した場合，発振安定時間は不要のため，設定は無効（don't care）となります。

注意 ビット4, 7には必ず1を設定してください。

備考 x : don't care

【例 1】 高速内蔵発振クロック使用，RESET端子として使用，低速内蔵発振器の停止可（サンプル・プログラムの設定と同内容）

アドレス : 0080H



注 低速内蔵発振器の発振停止やウォッチドッグ・タイマの動作停止，クロック周波数の設定は，リセット解除後に行います。詳細については，[4.4 ウォッチドッグ・タイマの設定](#)，[4.5 クロックの設定](#)を参照してください。

オプション・バイトの設定値は，「1xx111x0（x: don't care，ビット4, 7は1に必ず設定）」となります。プロテクト・バイトの設定（以降この資料では，すべてのブロックへの書き込みを許可に設定）と合わせて，ソフトウェアを記述すると，次のようになります（下記の例では「x」を0に設定）。

```
OPBT    CSEG    AT      0080H
        DB      10011100B
        DB      11111111B
```

C言語を使用する場合は，次のようなアセンブリ言語のソース・ファイル（ファイル名：「*.asm（*：任意）」）を準備し，プロジェクトのソース・ファイルに指定して，他のソース・ファイル（main.c）と一緒にビルドしてください。

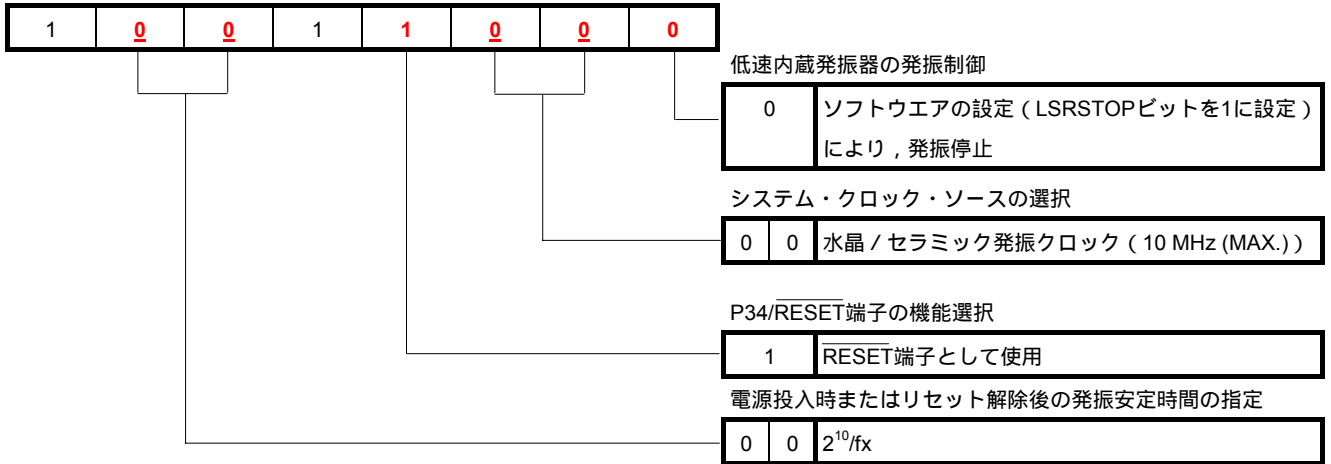
```
OPBT    CSEG    AT      0080H
        DB      10011100B
        DB      11111111B
        end
```

備考 プロテクト・バイト（アドレス：0081H）は，書き込みおよびブロック消去の禁止領域を設定するバイトで，その設定内容は，セルフ・プログラミング時のみ有効となります。詳細は，各製品のユーザー・マニュアル（[78K0S/KU1+](#)，[78K0S/KY1+](#)，[78K0S/KA1+](#)，[78K0S/KB1+](#)）を参照してください。

【例 2】 水晶/セラミック発振クロック使用，RESET端子として使用，低速内蔵発振器の停止可，発振安定時間を最短 ($2^{10}/f_x$)

(下線部以外は，例1と同内容)

アドレス：0080H



オプション・バイトの設定値は，「10011000 (ビット4, 7は1に必ず設定)」となります。プロテクト・バイトの設定と合わせて，ソフトウェアを記述すると，次のようになります。

OPBT	CSEG	AT	0080H
	DB		10011000B
	DB		11111111B



【コラム】 CSEG (Code Segment), DSEG (Data Segment), BSEG (Bit Segment) とは

CSEG, DSEG, BSEGは，命令やデータなどの生成されたコードをどこに配置するかを示す疑似命令です。これらの疑似命令以降に記述された命令やデータは，CSEG疑似命令ではROM領域に，DSEG疑似命令ではRAM領域に，BSEG疑似命令ではRAMのsaddr領域へ配置されます。

例えば，オプション・バイトの設定内容を内部ROM (フラッシュ・メモリ) の0080Hから配置する場合は，まず，CSEG疑似命令とAT属性を使用し，アドレスに0080Hを指定します。そのあとに，DB疑似命令で0080H番地以降に設定したい値を定義し，それをアセンブリ言語にて記述したプログラム内に記述してください。

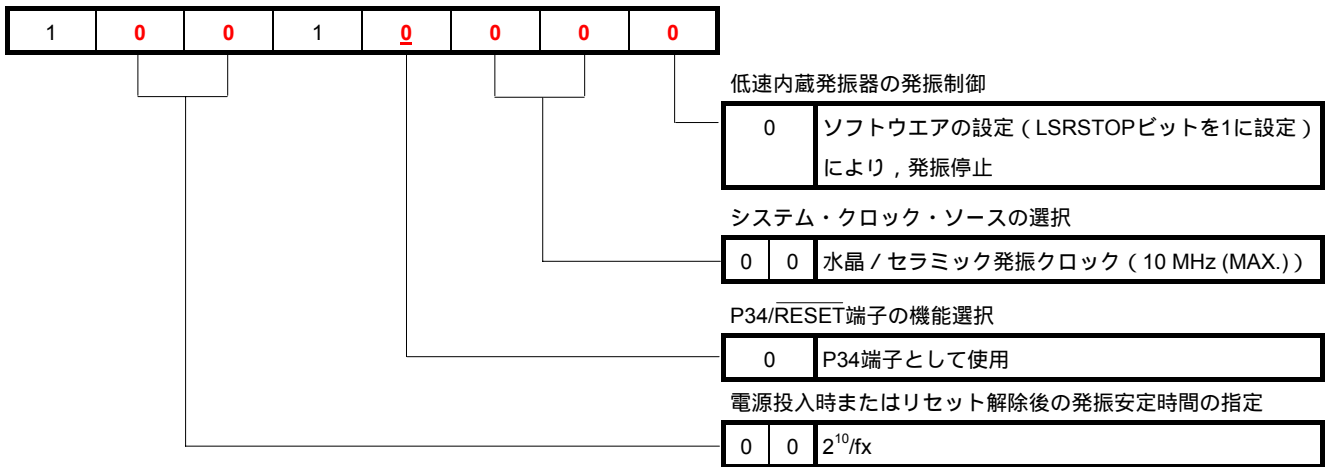
CSEG疑似命令で指定したROM領域のみ，DB, DW疑似命令は使用可能です。DSEG, BSEG疑似命令で指定したRAM領域では，DB, DW疑似命令の記述はエラーにはなりませんが，使用しないでください。この場合，オブジェクトは生成され，MINICUBE2 (オンチップ・デバッグ・エミュレータ) やSM+ (システム・シミュレータ) では記述された命令やデータをRAM領域に展開するため，デバッグ動作を行うことができます。しかし，実際のデバイスでは，それらをRAM領域に展開できないため，動作できなくなります。

CSEG, DSEG, BSEG疑似命令の詳細については，[RA78K0Sの言語編のユーザーズ・マニュアル](#)を参照してください。

【例 3】 水晶/セラミック発振クロック使用，P34端子として使用，低速内蔵発振器の停止可，発振安定時間を最短 ($2^{10}/f_x$)

(下線部以外は，例2と同内容)

アドレス：0080H



オプション・バイトの設定値は，「10010000 (ビット4, 7は1に必ず設定)」となります。プロテクト・バイトの設定と合わせて，ソフトウェアを記述すると，次のようになります。

OPBT	CSEG	AT	0080H
	DB		10010000B
	DB		11111111B

4.2 ベクタ・テーブルの設定

ベクタ・テーブル領域は、リセットや各割り込み要求発生により分岐するときのプログラム・スタート・アドレスを格納する領域です。

このサンプル・プログラムでは、割り込み処理は行わないため、リセット・スタート時に使用するリセット・ベクタのみ設定します。この設定は、アセンブリ言語で記述するときに必要な設定です。C言語で記述する場合は、スタートアップ・ルーチンにてリセット・ベクタが自動的に設定されるため、設定は不要です。

ベクタ・テーブルや割り込みについての設定方法および設定例に関しては、「[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](#)」を参照してください。

【例 1】 リセット・スタート時に使用するリセット・ベクタのみ設定（サンプル・プログラムの設定と同内容）

XVCT	CSEG	AT	0000H	アドレス	機能名
→	DW	RESET_START		; (00)	RESET
	DW	RESET_START		; (02)	--
	DW	RESET_START		; (04)	--
	DW	RESET_START		; (06)	INTLVI
	DW	RESET_START		; (08)	INTP0
	DW	RESET_START		; (0A)	INTP1
	DW	RESET_START		; (0C)	INTTMH1
	DW	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

リセット解除後、プログラムは、リセット・ベクタで指定したアドレス（上記では、の「RESET_START」）からスタートします。

このサンプル・プログラムでは、ベクタ・テーブル・アドレスの0000H以外は使用しません。残りのベクタ・テーブル・アドレスには、0000Hと同様に、すべて「RESET_START」を設定しています。このように設定することにより、万が一、割り込みが発生した場合でも、「RESET_START」に分岐し、リセット解除後と同じ処理が行われます。



【コラム】#pragma指令とは

#pragma指令は、C言語で使用する前処理命令で、ソース・プログラムの冒頭に記述します。

主な#pragma指令は、次のとおりです。

- #pragma sfr : SFR領域に関する操作をCソース・レベルで記述可能
- #pragma ei : EI命令をCソース・レベルで記述可能
- #pragma di : DI命令をCソース・レベルで記述可能
- #pragma nop : NOP命令をCソース・レベルで記述可能（CPUを動作させずにクロックを進めることが可能）
- #pragma interrupt : 割り込み関数をCソース・レベルで記述可能

#pragma指令の詳細は、「[CC78K0Sの言語編のユーザーズ・マニュアル](#)」の拡張機能の章を参照してください。

4.4 ウォッチドッグ・タイマの設定

WDTMで、ウォッチドッグ・タイマの動作クロックの選択とオーバフロー時間を選択します。

このサンプル・プログラムでは、暴走検出用として、ウォッチドッグ・タイマを使用しないため、後述の【例 1】の内容で、WDTMを設定しています。

注意 ウォッチドッグ・タイマの動作クロックおよびオーバフロー時間の設定は、必ず初期設定の中で行ってください。

図4-2 ウォッチドッグ・タイマ・モード・レジスタ (WDTM) のフォーマット

WDTM

0	1	1	WDCS	WDCS	WDCS	WDCS	WDCS
			4	3	2	1	0

オーバフロー時間の設定 ^{注1}

			低速内蔵発振クロック 動作時	システム・クロック 動作時
0	0	0	$2^{11}/f_{RL}$	$2^{13}/f_x$
0	0	1	$2^{12}/f_{RL}$	$2^{14}/f_x$
0	1	0	$2^{13}/f_{RL}$	$2^{15}/f_x$
0	1	1	$2^{14}/f_{RL}$	$2^{16}/f_x$
1	0	0	$2^{15}/f_{RL}$	$2^{17}/f_x$
1	0	1	$2^{16}/f_{RL}$	$2^{18}/f_x$
1	1	0	$2^{17}/f_{RL}$	$2^{19}/f_x$
1	1	1	$2^{18}/f_{RL}$	$2^{20}/f_x$

動作クロックの選択 ^{注2}

0	0	低速内蔵発振クロック (f _{RL})
0	1	システム・クロック (f _x)
1	x	ウォッチドッグ・タイマ動作停止

注1. ウォッチドッグ・タイマ動作停止を選択した場合、オーバフロー時間の設定は無効 (don't care) となります。

2. ウォッチドッグ・タイマ動作を停止する場合、またはウォッチドッグ・タイマの動作クロックにシステム・クロックを使用する場合は、オプション・バイトで、「ソフトウェアの設定 (LSRSTOPビットを1に設定) により、発振停止」に必ず設定してください。詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

注意 ビット7, 6, 5にはそれぞれ0, 1, 1を必ず設定してください。

備考 x : don't care

【例 1】 ウォッチドッグ・タイマ停止 (サンプル・プログラムの設定と同内容)

WDTM

0	1	1	1	x	x	x	x
---	---	---	---	---	---	---	---

オーバーフロー時間の設定

x	x	x	(ウォッチドッグ・タイマ動作停止のため、設定無効)
---	---	---	---------------------------

動作クロックの選択

1	x	ウォッチドッグ・タイマ動作停止 ^注
---	---	------------------------------

*このサンプル・プログラムでは、暴走検出用として、ウォッチドッグ・タイマを使用しないため、動作停止に設定します。

注 ウォッチドッグ・タイマ動作を停止する場合は、オプション・バイトで、「ソフトウェアの設定 (LSRSTOPビットを1に設定) により、発振停止」に必ず設定してください。詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

WDTMの設定値は、「0111xxxx (x: don't care, ビット7, 6, 5はそれぞれ0, 1, 1に必ず設定)」となります (下記の例では、ビット3の「x」を0、ビット2-0の「x」を1に設定)。

・アセンブリ言語の場合

```
MOV    WDTM,    #01110111B
```

・C言語の場合

```
WDTM = 0b01110111
```

【例 2】 ウォッチドッグ・タイマの動作クロックに低速内蔵発振クロック (f_{RL}) 使用、オーバーフロー時間を最大周期 ($2^{18}/f_{RL}$) (リセット解除後のWDTMの値と同内容)

WDTM

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

オーバーフロー時間の設定

1	1	1	$2^{18}/f_{RL}$
---	---	---	-----------------

動作クロックの選択

0	0	低速内蔵発振クロック (f_{RL})
---	---	-------------------------

上記の内容で使用する場合、WDTMの設定値は、リセット解除後のWDTMの値と同じ値になるため、プログラムでの設定は不要です。

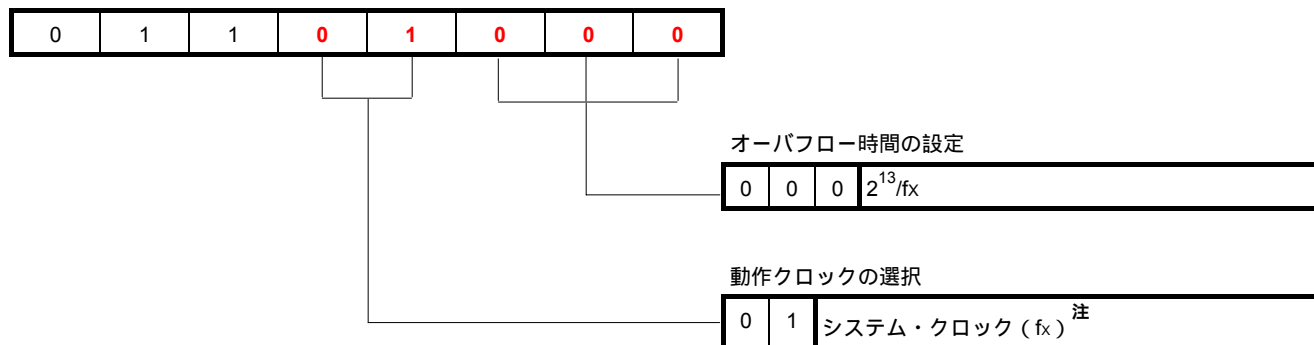


【コラム】2進数値の表記

2進数値を表記する場合、アセンブリ言語では2進数値の後ろに「B」または「Y」を、C言語では2進数値の前に「0b」または「0B」を付加してください。

【例 3】 ウォッチドッグ・タイマの動作クロックにシステム・クロック (fx) 使用, オーバフロー時間を最小周期 ($2^{13}/fx$)

WDTM



注 ウォッチドッグ・タイマの動作クロックにシステム・クロックを使用する場合は, オプション・バイトで, 「ソフトウェアの設定 (LSRSTOPビットを1に設定) により, 発振停止」に必ず設定してください。詳細は, [4.1 オプション・バイトの設定](#)を参照してください。

WDTMの設定値は, 「01101000 (ビット7, 6, 5はそれぞれ0, 1, 1に必ず設定)」となります。

・アセンブリ言語の場合

```
MOV    WDTM,    #01101000B
```

・C言語の場合

```
WDTM = 0b01101000
```



【コラム】hdwinit関数とmain関数

C言語でプログラムを作成する場合, CPUリセット直後に, 周辺装置(SFR)の初期化を行うために, hdwinit関数を呼び出します。したがって, ウォッチドッグ・タイマの設定やクロック周波数選択などの初期設定は, 基本的にhdwinit関数に記述します。

hdwinit関数を呼び出したあとに, main関数を呼び出しますので, メイン処理はmain関数に記述します。

main関数からは, hdwinit関数を呼び出さないでください。この場合, hdwinit関数は2度実行され, 1度のみしか許可されていないウォッチドッグ・タイマの設定が, 2度実行されてしまいます。その結果, 2度目の実行時に内部リセット信号が発生され, プログラムが初期設定から進まなくなります。

詳細については, [CC78K0Sの言語編のユーザーズ・マニュアル](#), 当社HPのFAQの「[初期設定](#)」「[プログラミング](#)」の『[最初に実行する内容について](#)』を参照してください。

4.5 クロックの設定

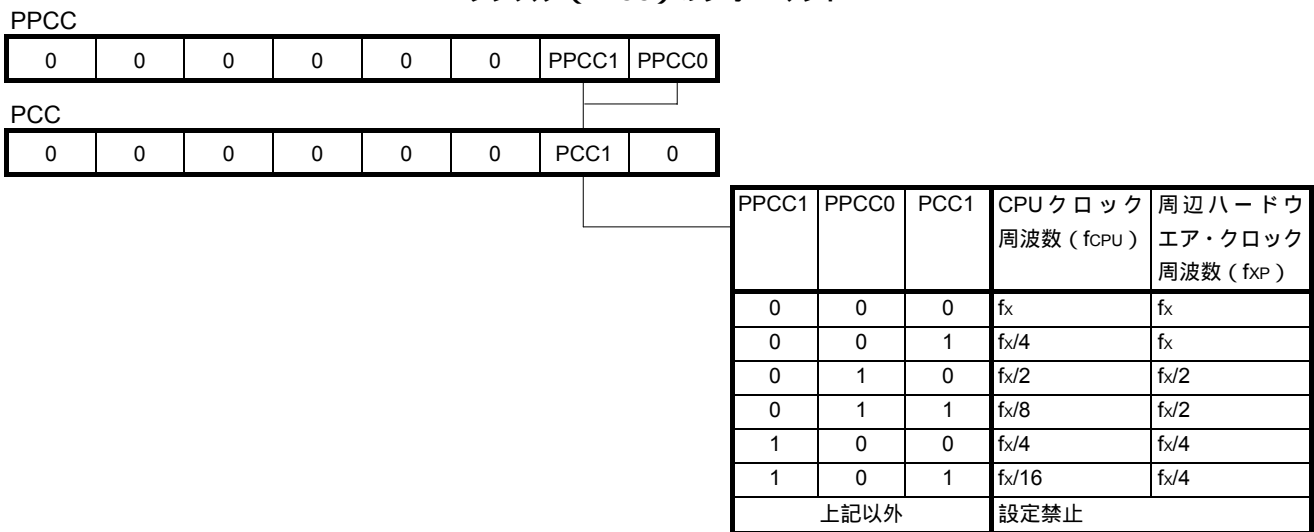
(1) クロック周波数の設定

CPUクロック周波数 (f_{CPU}) と周辺ハードウェアへの供給クロック周波数 (f_{XP}) は、オプション・バイトで設定したシステム・クロックを分周して、生成されます。

PPCCとPCCで、CPUクロック周波数 (f_{CPU}) と周辺ハードウェアへの供給クロック周波数 (f_{XP}) を選択します。

このサンプル・プログラムでは、 $f_{CPU} = f_{XP} = 2$ MHzになるように、後述の【例 1】の内容で、PPCCとPCCを設定しています。

図4-3 プロセッサ・クロック・コントロール・レジスタ (PCC) とプリプロセッサ・クロック・コントロール・レジスタ (PPCC) のフォーマット



注意1. PPCCのビット2-7とPCCのビット0, 2-7には必ず0を必ず設定してください。

2. 電源電圧により、使用できるクロック周波数の範囲が異なります。

発振子	条件	CPUクロック周波数 (f_{CPU})	周辺ハードウェア・クロック周波数 (f_{XP})
セラミック発振子, 水晶振動子, 外部クロック	4.0 ~ 5.5 V	125 kHz f_{CPU} 10 MHz	500 kHz f_{XP} 10 MHz
	3.0 ~ 4.0 V	125 kHz f_{CPU} 6 MHz	
	2.7 ~ 3.0 V	125 kHz f_{CPU} 5 MHz	
	2.0 ~ 2.7 V ^注	125 kHz f_{CPU} 2 MHz	500 kHz f_{XP} 5 MHz
高速内蔵発振器	4.0 ~ 5.5 V	500 kHz (TYP.) f_{CPU} 8 MHz (TYP.)	2 MHz (TYP.) f_{XP} 8 MHz (TYP.)
	2.7 ~ 4.0 V	500 kHz (TYP.) f_{CPU} 4 MHz (TYP.)	
	2.0 ~ 2.7 V ^注	500 kHz (TYP.) f_{CPU} 2 MHz (TYP.)	2 MHz (TYP.) f_{XP} 4 MHz (TYP.)

注 標準品と (A) 水準品は、パワーオン・クリア (POC) 回路の検出電圧 (V_{POC}) が $2.1 V \pm 0.1 V$ のため、2.2 ~ 5.5 V の電圧範囲で使用してください。

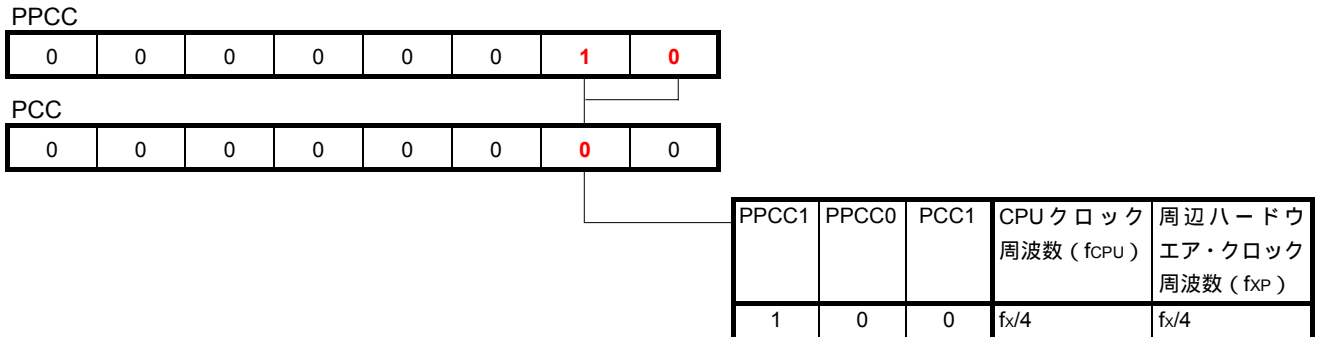
(A2) 水準品は、パワーオン・クリア (POC) 回路の検出電圧 (V_{POC}) が 2.26 V (MAX.) のため、2.26 ~ 5.5 V の電圧範囲で使用してください。

備考 f_x : システム・クロック周波数

使用するシステム・クロックは、オプション・バイトで設定します。詳細については、[4.1 オプション・バイトの設定](#)を参照してください。

【例 1】 $f_{CPU} = f_{XP} = f_x/4$

(サンプル・プログラムの設定と同内容。このサンプル・プログラムでは、オプション・バイトの設定で、システム・クロック (f_x) に「高速内蔵発振クロック (8 MHz (TYP.))」を選択しているため、CPUクロック周波数 (f_{CPU}) と周辺ハードウェア・クロック周波数 (f_{XP}) は、2 MHz (= 8 MHz/4) になります。)



備考 f_x : システム・クロック周波数

使用するシステム・クロックは、オプション・バイトで設定します。詳細については、[4.1 オプション・バイトの設定](#)を参照してください。

PPCCの設定値は「00000010 (ビット2-7は必ず0に設定)」、PCCの設定値は「00000000 (ビット0, 2-7は必ず0に設定)」となります。

標準品と(A)水準品の場合、パワーオン・クリア(POC)回路の検出電圧 (V_{POC}) は2.1 V \pm 0.1 Vです。POC解除時点で、「電源電圧 2.0 V」の条件(このサンプル・プログラムのCPUクロック周波数の設定条件)を満たすため、このサンプル・プログラム動作時には、CPUは正常に動作します。(A2)水準品の場合も同様です。

・アセンブリ言語の場合

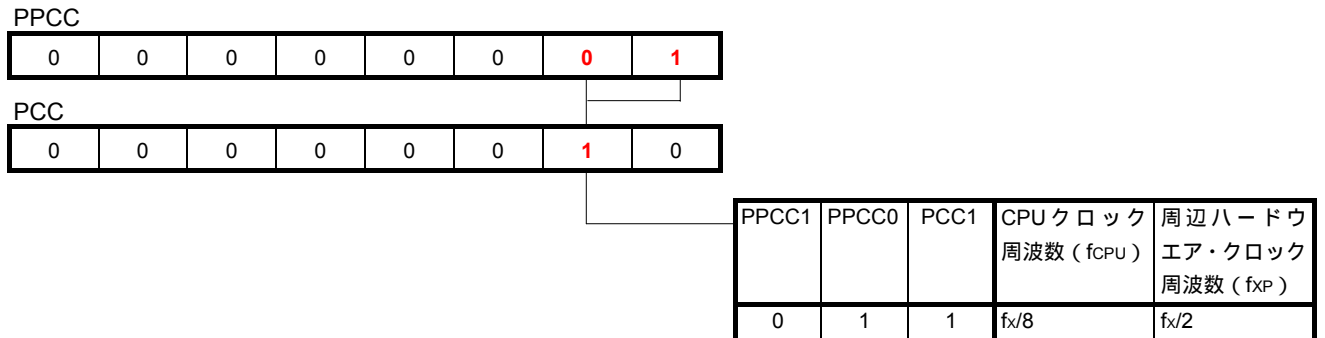
```
MOV    PPCC,    #00000010B
MOV    PCC,     #00000000B
```

・C言語の場合

```
PPCC = 0b00000010
PCC  = 0b00000000
```

【例 2】 $f_{CPU} = f_x/8$, $f_{XP} = f_x/2$

(オプション・バイトの設定でシステム・クロック (f_x) に「高速内蔵発振クロック (8 MHz (TYP.))」を選択している場合、CPUクロック周波数 (f_{CPU}) は1 MHz (= 8 MHz/8) に、周辺ハードウェア・クロック周波数 (f_{XP}) は、4 MHz (= 8 MHz/2) となります。)



備考 f_x : システム・クロック周波数

使用するシステム・クロックは、オプション・バイトで設定します。詳細については、[4.1 オプション・バイトの設定](#)を参照してください。

PPCCの設定値は「00000001 (ビット2-7は必ず0に設定)」, PCCの設定値は「00000010 (ビット0, 2-7は必ず0に設定)」となります。

電源電圧により、使用できるクロック周波数の範囲が異なるため、PPCCとPCCを設定する場合は、注意が必要です (詳細については、[図4-3 プロセッサ・クロック・コントロール・レジスタ \(PCC\) とブリプロセッサ・クロック・コントロール・レジスタ \(PPCC\) のフォーマット](#)を参照)。

・アセンブリ言語の場合

```
MOV    PPCC,    #00000001B
MOV    PCC,     #00000010B
```

・C言語の場合

```
PPCC = 0b00000001
PCC  = 0b00000010
```

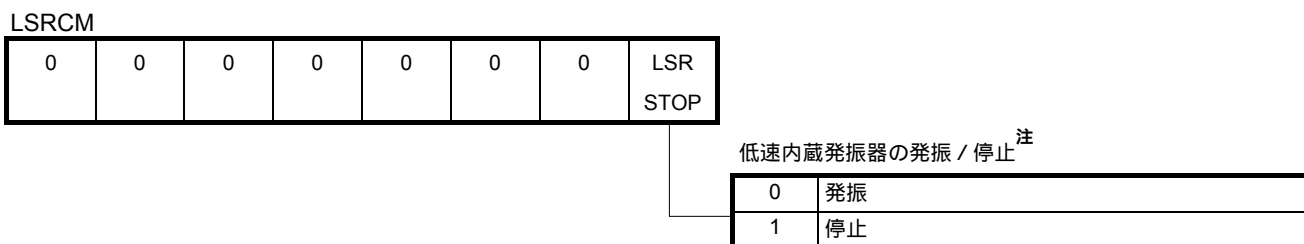
(2) 低速内蔵発振器の設定

低速内蔵発振クロックは、ウォッチドッグ・タイマや8ビット・タイマH1のクロック・ソースとして使用します。

LSRCMで、低速内蔵発振器の発振 / 停止を設定します。

このサンプル・プログラムでは、低速内蔵発振クロックは使用しないため、後述の【例 1】の内容で、LSRCMを設定しています。

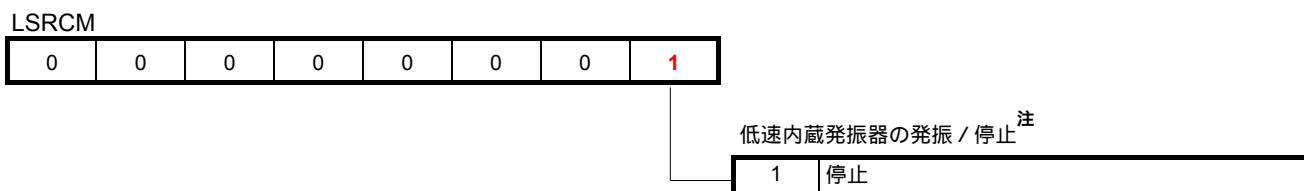
図4 - 4 低速内蔵発振モード・レジスタ (LSRCM) のフォーマット



注 低速内蔵発振器を停止する場合は、オプション・バイトで、「ソフトウェアの設定 (LSRSTOPビットを1に設定) により、発振停止」に必ず設定してください。詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

注意 LSRCMのビット1-7には必ず0を必ず設定してください。

【例 1】 低速内蔵発振器の発振停止 (サンプル・プログラムの設定と同内容)



注 低速内蔵発振器を停止する場合は、オプション・バイトで、「ソフトウェアの設定 (LSRSTOPビットを1に設定) により、発振停止」に必ず設定してください。詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

LSRCMの設定値は、「00000001 (ビット7-1は0に必ず設定)」となります。

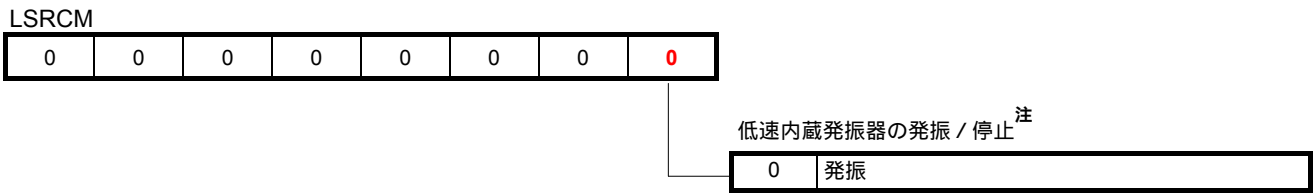
・アセンブリ言語の場合

```
MOV    LSRCM,    #00000001B
```

・C言語の場合

```
LSRCM = 0b00000001
```

【例 2】 低速内蔵発振器の発振



注 オプション・バイトで、低速内蔵発振器の発振を「停止不可」に設定した場合、LSRCMの設定は無効 (don't care) になります。詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

LSRCMの設定値は、「00000000 (ビット7-1は0に必ず設定)」となります。

・アセンブリ言語の場合

```
MOV    LSRCM,    #00000000B
```

・C言語の場合

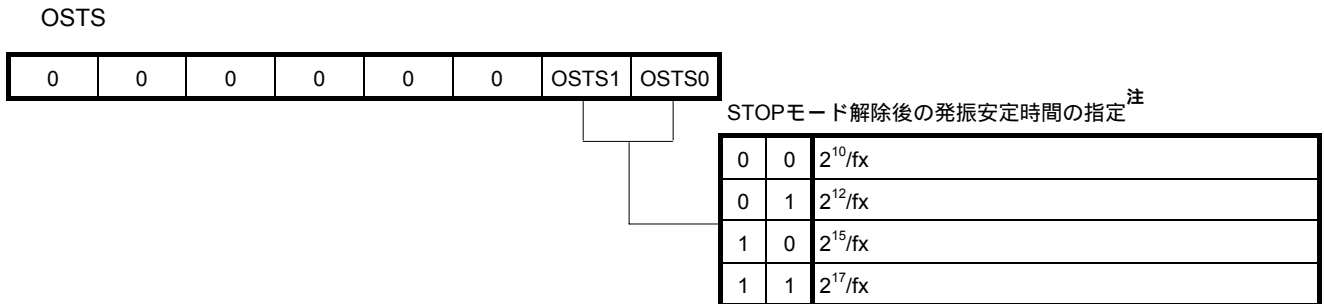
```
LSRCM = 0b00000000
```

(3) STOPモード解除後の発振安定時間の設定 (システム・クロック・ソースに水晶/セラミック発振使用時)

OSTSで、STOPモード解除後の発振安定時間を設定します。OSTSの設定は、システム・クロック・ソースに水晶/セラミック発振使用時のみ有効です。

このサンプル・プログラムでは、高速内蔵発振クロックを使用するため、OSTSの設定は行っていません。

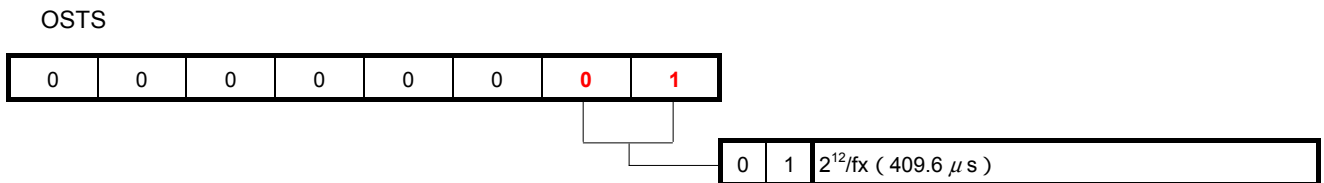
図4 - 5 発振安定時間選択レジスタ (OSTS) のフォーマット



注 システム・クロックに高速内蔵発振クロックまたは外部クロック入力を選択した場合、OSTSの設定は不要です。

注意 ビット7-2には必ず0を設定してください。

【例 1】 水晶/セラミック発振クロック使用時 ($f_x = 10 \text{ MHz}$) に、発振安定時間を $409.6 \mu\text{s}$ に設定



OSTSの設定値は、「00000001 (ビット7-2は0に必ず設定)」となります。

・アセンブリ言語の場合

```
MOV    OSTS,    #00000001B
```

・C言語の場合

```
OSTS = 0b00000001
```


4.6 ポートの設定

注意 各製品により、内蔵するポートは異なるため、設定するポートも異なります。

	78K0S/KA1+	78K0S/KB1+	78K0S/KU1+	78K0S/KY1+
ポート0	-	P00-P03	-	-
ポート2	P20-P23	P20-P23	P20-P23	P20-P23
ポート3	P30, P31, P34 ^注	P30-P33, P34 ^注	P32, P34 ^注	P32, P34 ^注
ポート4	P40-P45	P40-P47	P40, P43	P40-P47
ポート12	P121-P123	P120-P123	-	-
ポート13	P130	P130	-	-

注 P34はRESETと兼用しています。使用する機能の選択は、オプション・バイトの設定で行います。
 詳細は、[4.1 オプション・バイトの設定](#)を参照してください。

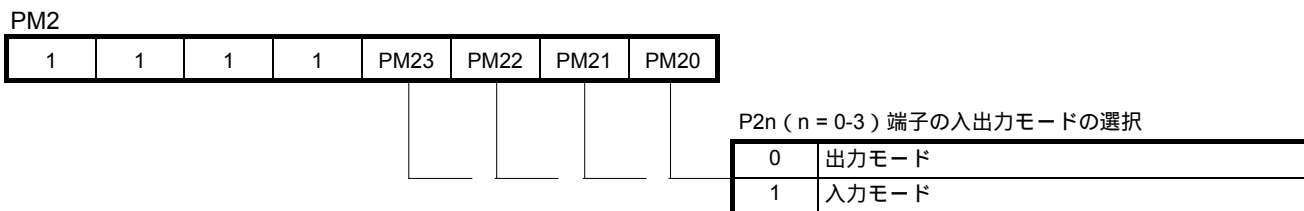
(1) ポートの入力/出力の設定

PMxxで、ポートを入力ポートまたは出力ポートとして使用するかを設定します。リセット解除後は、入力ポートに設定されます。

PMxxのフォーマットは、PM2を例にして、説明します。

このサンプル・プログラムでは、ポート2を後述の【例 1】、ポート4を【例 2】の内容で設定しています。

図4 - 6 ポート・モード・レジスタ2 (PM2) のフォーマット



注意 ビット7-4には必ず1を設定してください。

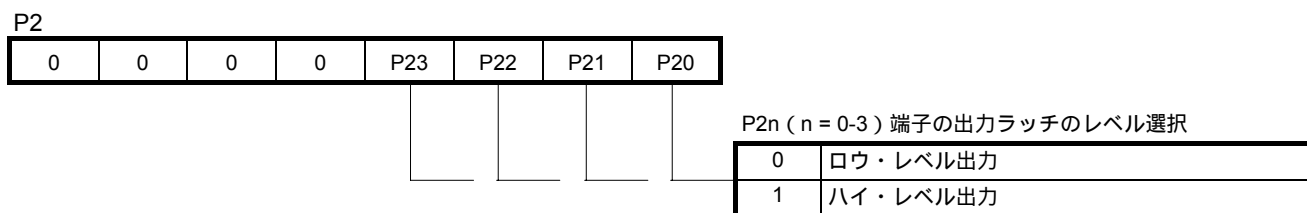
(2) 出力ポートの出力ラッチの設定

Pxxで、出力ポートの出力ラッチをハイ・レベルまたはロウ・レベルにするかを設定します。リセット解除後は、ロウ・レベル出力に設定されます。

Pxxのフォーマットは、P2を例にして、説明します。

このサンプル・プログラムでは、ポート2を後述の【例 1】の内容で設定しています。

図4 - 7 ポート・レジスタ2 (P2) のフォーマット



注意 ビット7-4には必ず0を設定してください。

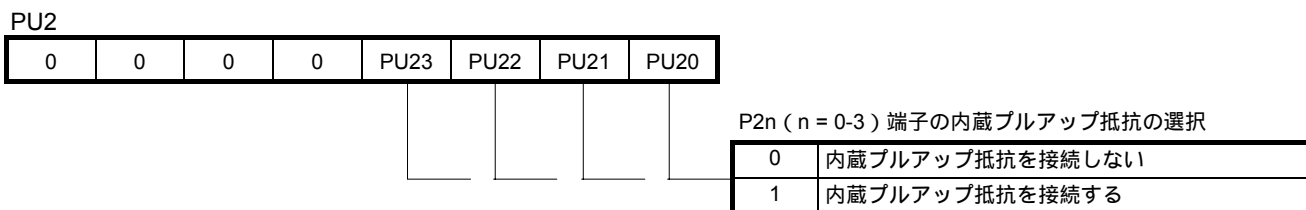
(3) 入力ポートの内蔵プルアップ抵抗接続の設定

PUxxで、入力ポートに内蔵プルアップ抵抗を接続するか否かを設定します。リセット解除後は、内蔵プルアップ抵抗に接続しません。

PUxxのフォーマットは、PU2を例にして、説明します。

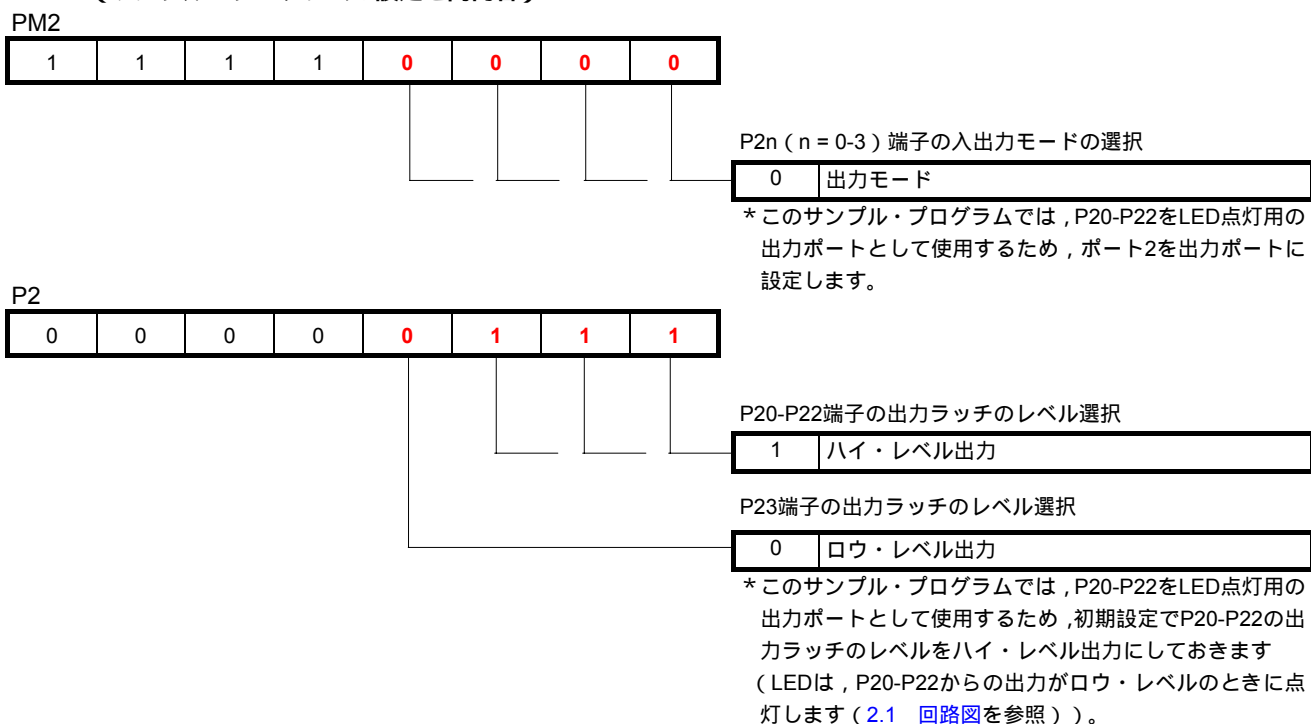
このサンプル・プログラムでは、ポート4を後述の【例 2】の内容で設定しています。

図4 - 8 プルアップ抵抗オプション・レジスタ2 (PU2) のフォーマット



注意 ビット7-4には必ず0を設定してください。

- 【例 1】
- ・ P20-P23を出力ポートに設定
 - ・ P20-P22の出力ラッチをハイ・レベル出力に、P23の出力ラッチをロウ・レベル出力に設定 (サンプル・プログラムの設定を同内容)



PM2の設定値は「11110000 (ビット7-4は1に必ず設定)」、P2の設定値は「00001111 (ビット7-4は0に必ず設定)」となります。

・アセンブリ言語の場合

```
MOV PM2, #11110000B
MOV P2, #00000111B
```

・C言語の場合

```
PM2 = 0b11110000
P2 = 0b00000111
```

- 【例 2】 ・ P40, P43を入力ポートに設定
 ・ P40, P43に内蔵プルアップ抵抗接続を設定
 (サンプル・プログラムの設定を同内容)

PM4

・ 78K0S/KB1+, 78K0S/KY1+の場合

x	x	x	x	1	x	x	1
---	---	---	---	---	---	---	---

P40, P43端子の入出力モードの選択

1	入力モード
---	-------

・ 78K0S/KA1+の場合

注 0	注 0	x	x	1	x	x	1
--------	--------	---	---	---	---	---	---

P40, P43端子の入出力モードの選択

1	入力モード
---	-------

注 ビット7, 6は, リセット解除後に1となります。初期設定で必ず0に設定してください。

・ 78K0S/KU1+の場合

注 0	注 0	注 0	注 0	1	注 0	注 0	1
--------	--------	--------	--------	---	--------	--------	---

P40, P43端子の入出力モードの選択

1	入力モード
---	-------

*このサンプル・プログラムでは, P40, P43をスイッチ入力用のポートとして使用するため, P40, P43を入力ポートに設定します。

注 ビット7-4, 2, 1は, リセット解除後に1となります。必ず初期設定で0を設定してください。

PU4

・ 78K0S/KB1+, 78K0S/KY1+の場合

x	x	x	x	1	x	x	1
---	---	---	---	---	---	---	---

P40, P43端子の内蔵プルアップ抵抗の選択

1	内蔵プルアップ抵抗を接続する
---	----------------

・ 78K0S/KA1+の場合

0	0	x	x	1	x	x	1
---	---	---	---	---	---	---	---

P40, P43端子の内蔵プルアップ抵抗の選択

1	内蔵プルアップ抵抗を接続する
---	----------------

注意 ビット7, 6には必ず0を設定してください。

・ 78K0S/KU1+の場合

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

P40, P43端子の内蔵プルアップ抵抗の選択

1	内蔵プルアップ抵抗を接続する
---	----------------

注意 ビット7-4, 2, 1には必ず0を設定してください。

*このサンプル・プログラムでは, P40, P43の入力レベルを制御するために, P40, P43に内蔵プルアップ抵抗を接続します (スイッチがOFFのときにハイ・レベル入力, ONのときにロウ・レベル入力になります (2.1 回路図を参照))。

PM4の設定値は, 78K0S/KB1+, 78K0S/KY1+の場合では「xxxx1xx1 (x: don't care)」、78K0S/KA1+の場合では「00xx1xx1 (x: don't care)」、78K0S/KU1+の場合では「00001001」となります。また, PU4の設定値は, 78K0S/KB1+, 78K0S/KY1+の場合では「xxxx1xx1 (x: don't care)」、78K0S/KA1+の場合では「00xx1xx1 (x: don't care)」、78K0S/KU1+の場合では「00001001」となります (下記の例では, PM4とPU4の「x」を0に設定)。

・アセンブリ言語の場合

```
MOV PM4, #00001001B  
MOV PU4, #00001001B
```

・C言語の場合

```
PM4 = 0b00001001  
PU4 = 0b00001001
```

4.7 メイン処理

アセンブリ言語のメイン処理では、次の動作を行います。

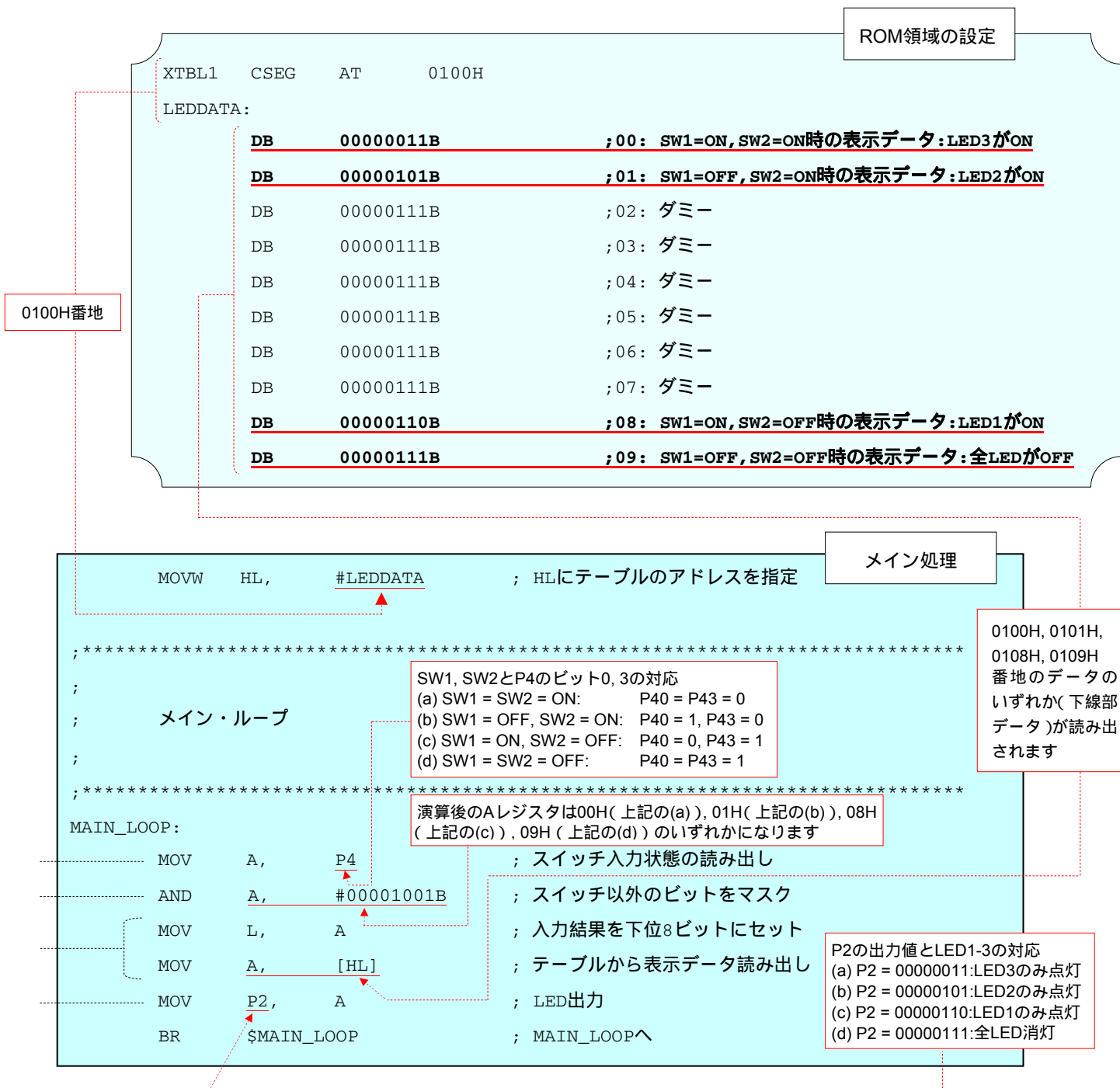
P4のデータを読み込みます

読み込んだ8ビット・データのうち、入力ポート（P40, P43）以外のビットを0にします

P40, P43の入力レベルの組み合わせに対応する出力データを、0100H-0109H番地（「LEDDATA」テーブル）より取り出します。

取り出したデータをP2に出力します

この動作により、スイッチ（SW1とSW2）に接続しているP40, P43の入力レベルの組み合わせのみを判断することが可能になります。また、ここでは、0100H-0109H番地のうち、0100H, 0101H, 0108H, 0109H番地のデータのみを使用します（入力レベルの組み合わせが4パターンのため）。



C言語のメイン処理も，アセンブリ言語と同様な動作を行います。

C言語の場合は，入力データと出力データの対応を，配列で設定します。

```

/*****
メイン・ループ
*****/
void main(void){

    const unsigned char OUTDATA[10] =
        {0x03,0x05,0x07,0x07,0x07,0x07,0x07,0x07,0x06,0x07};
        /* 表示データパターン用配列 */

    unsigned char INDATA;
        /* スイッチ入力データ変数 */

    while(1){
        INDATA = P4 & 0b00001001;
        /* 有効スイッチ情報取得 */
        P2 = OUTDATA[INDATA];
        /* テーブルから表示データを読み出して表示 */
    }
}


```

{ }内に10個のデータがあると定義し，その中に出力データを設定しています。ただし，このメイン処理では，下線部のみ出力データとして使用します（下線部以外はダミー・データ）。

入力データと出力データの対応は次のようになります。


スイッチ入力	P40, P43	INDATA	OUTDATA	LED点灯
SW1 = ON, SW2 = ON	P40 = 0, P43 = 0	0b00000000	0x03	LED3のみ点灯
SW1 = OFF, SW2 = ON	P40 = 1, P43 = 0	0b00000001	0x05	LED2のみ点灯
SW1 = ON, SW2 = OFF	P40 = 0, P43 = 1	0b00001000	0x06	LED1のみ点灯
SW1 = OFF, SW2 = OFF	P40 = 1, P43 = 1	0b00001001	0x07	全LED消灯

第5章 システム・シミュレータ SM+での動作確認

この章では、のアイコンを選択してダウンロードした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。

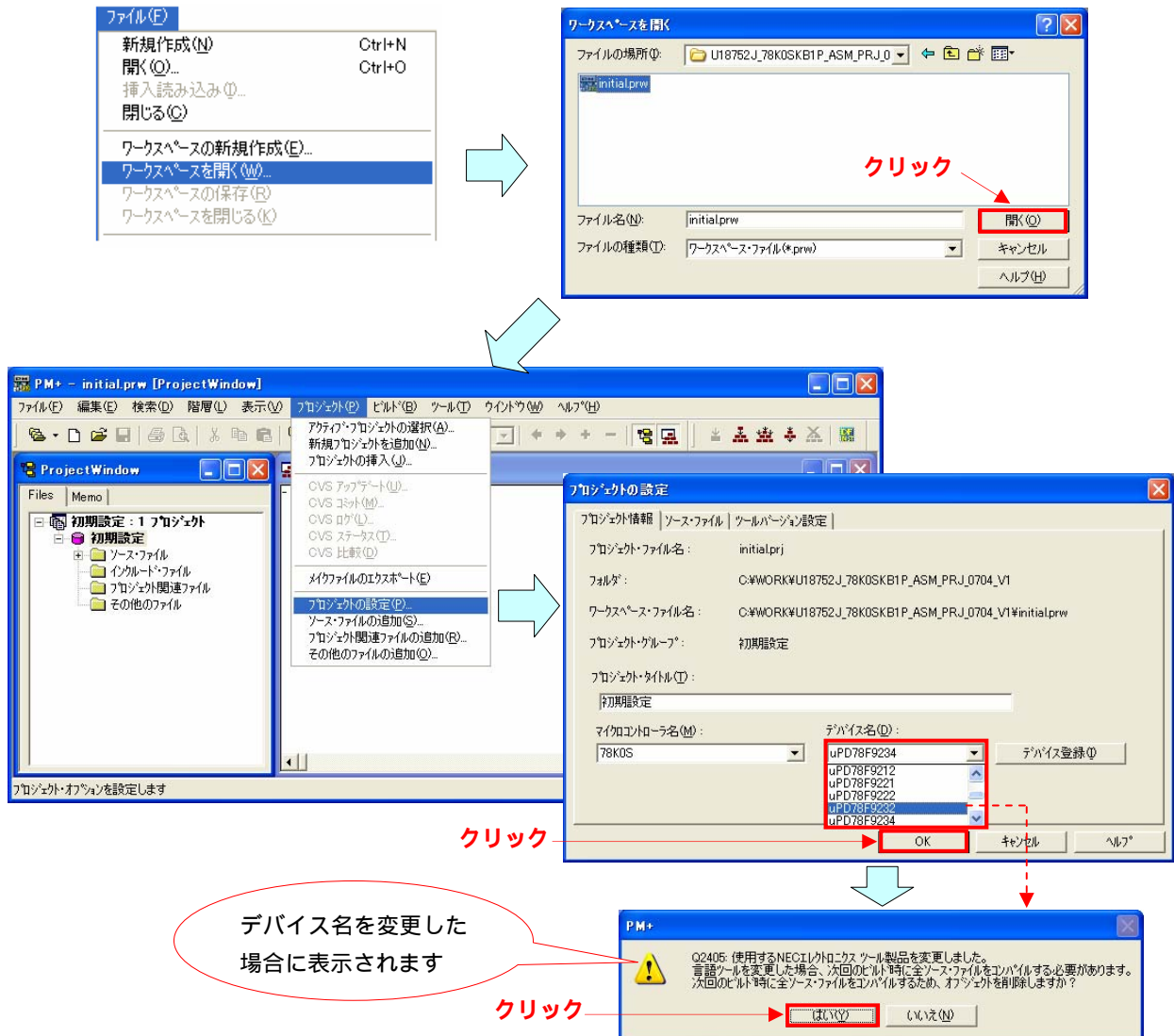
[コンパイラオプションの設定]ダイアログが開いたら、「スタートアップ・ルーチン」タグを選択してください。


「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）。

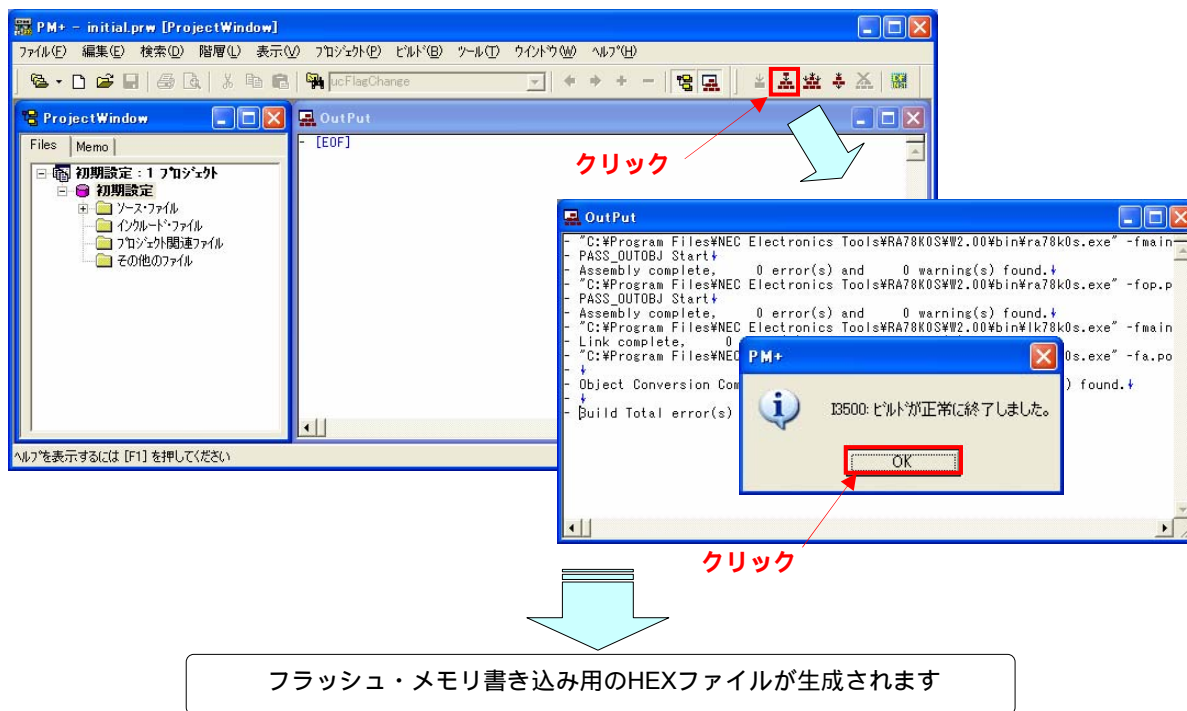
「標準ライブラリ固定領域を使用する」のチェックを外すと、標準ライブラリ固定領域として確保されていた118バイトのRAM領域が使用可能になりますが、標準ライブラリ（getchar関数やmalloc関数など）を使用できなくなります。

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、「initial.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。



- (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。




5.2 SM+での動作

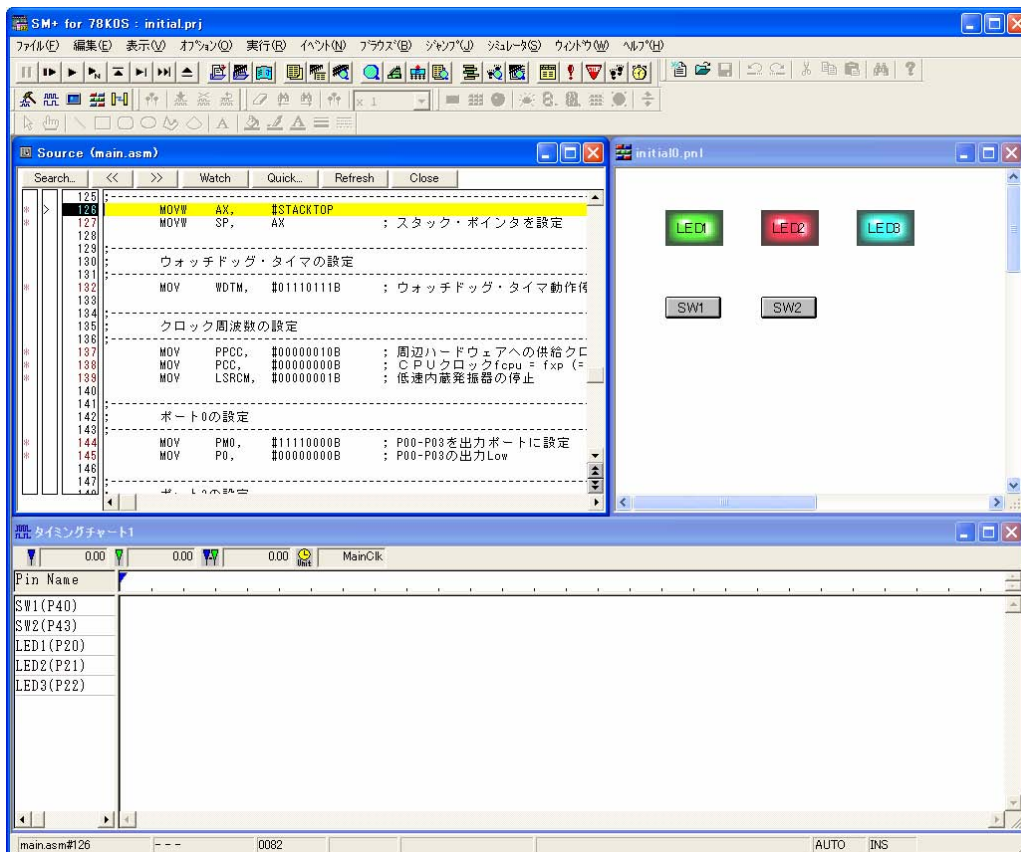
ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。


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。

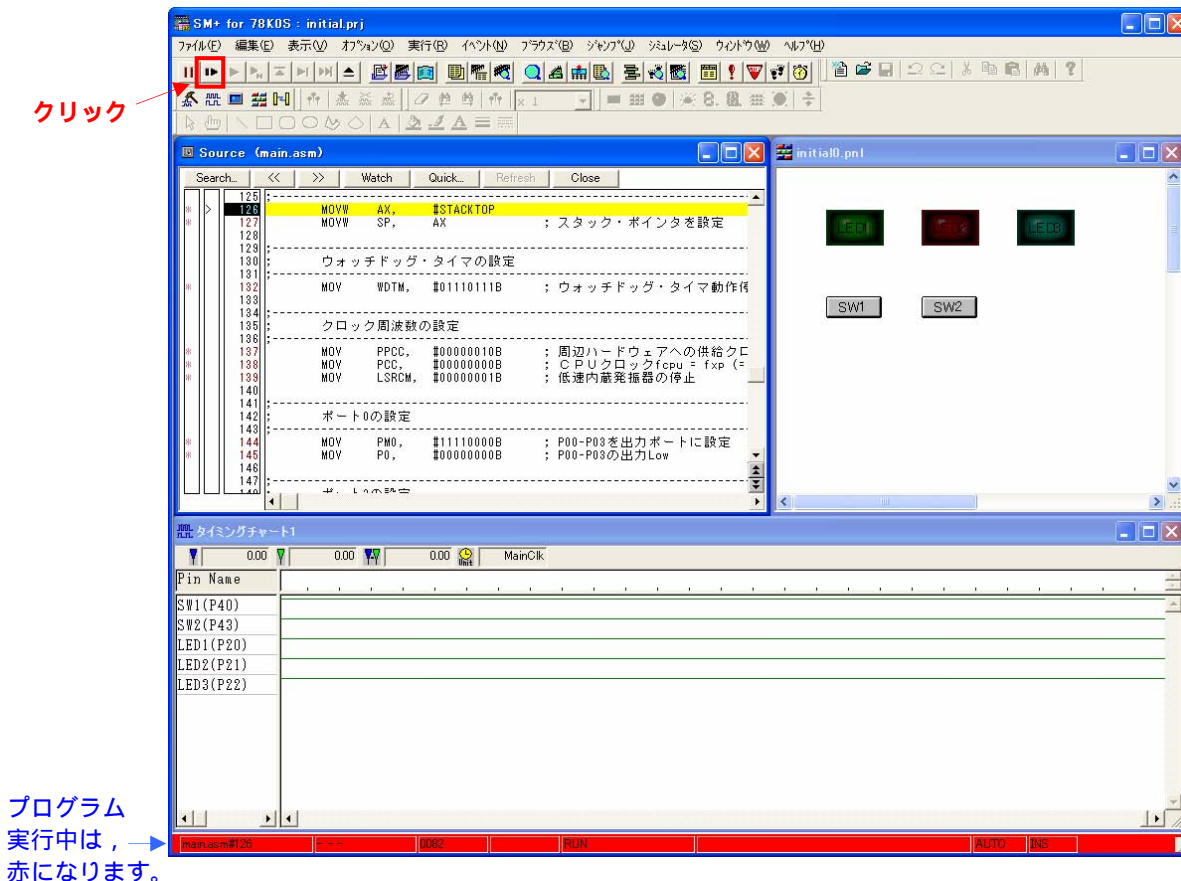
- (1) SM+ for 78K0S/Kx1+ W1.02をPM+ Ver6.30の環境で使用するために、次のサイトにあるPDFファイルを参照して、「外部ツールの登録」を行い、SM+を起動してください。

・ <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）。

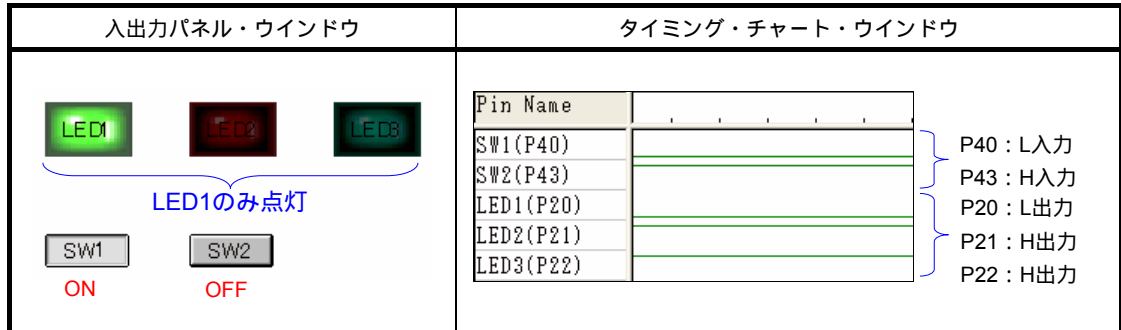


- (3) （「リスタート」ボタン）をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。

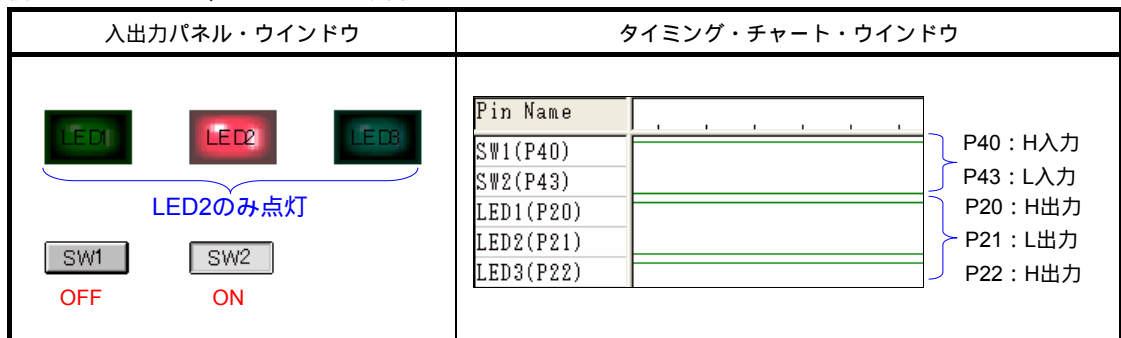


- (4) プログラム実行中に、入出力パネル・ウィンドウ上の [SW1][SW2] ボタンをクリックしてください。
 [SW1][SW2] ボタンの組み合わせにより、入出力パネル・ウィンドウ上の [LED1] ~ [LED3] の点灯およびタイミング・チャート・ウィンドウ上の波形が変化することを確認してください。

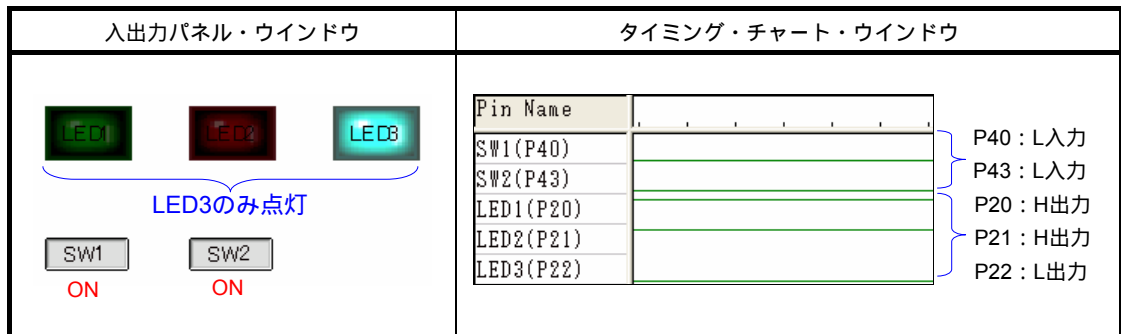
例1. SW1 : ON , SW2 : OFFの場合



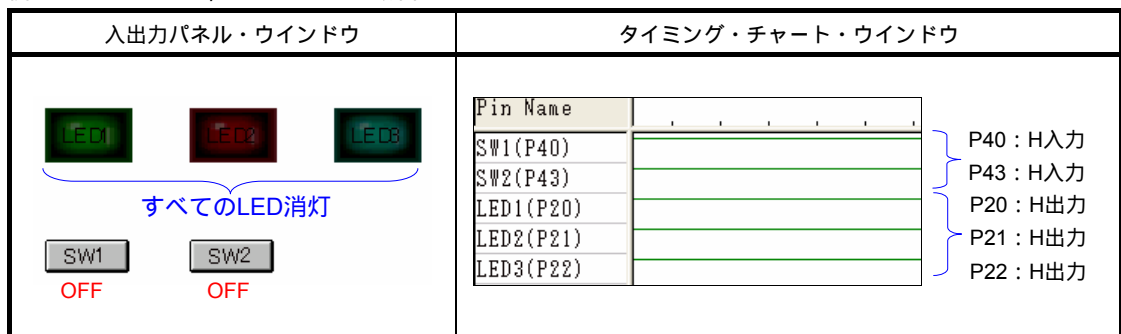
例2. SW1 : OFF , SW2 : ONの場合



例3. SW1 : ON , SW2 : ONの場合



例4. SW1 : OFF , SW2 : OFFの場合



備考 H : ハイ・レベル , L : ロウ・レベル

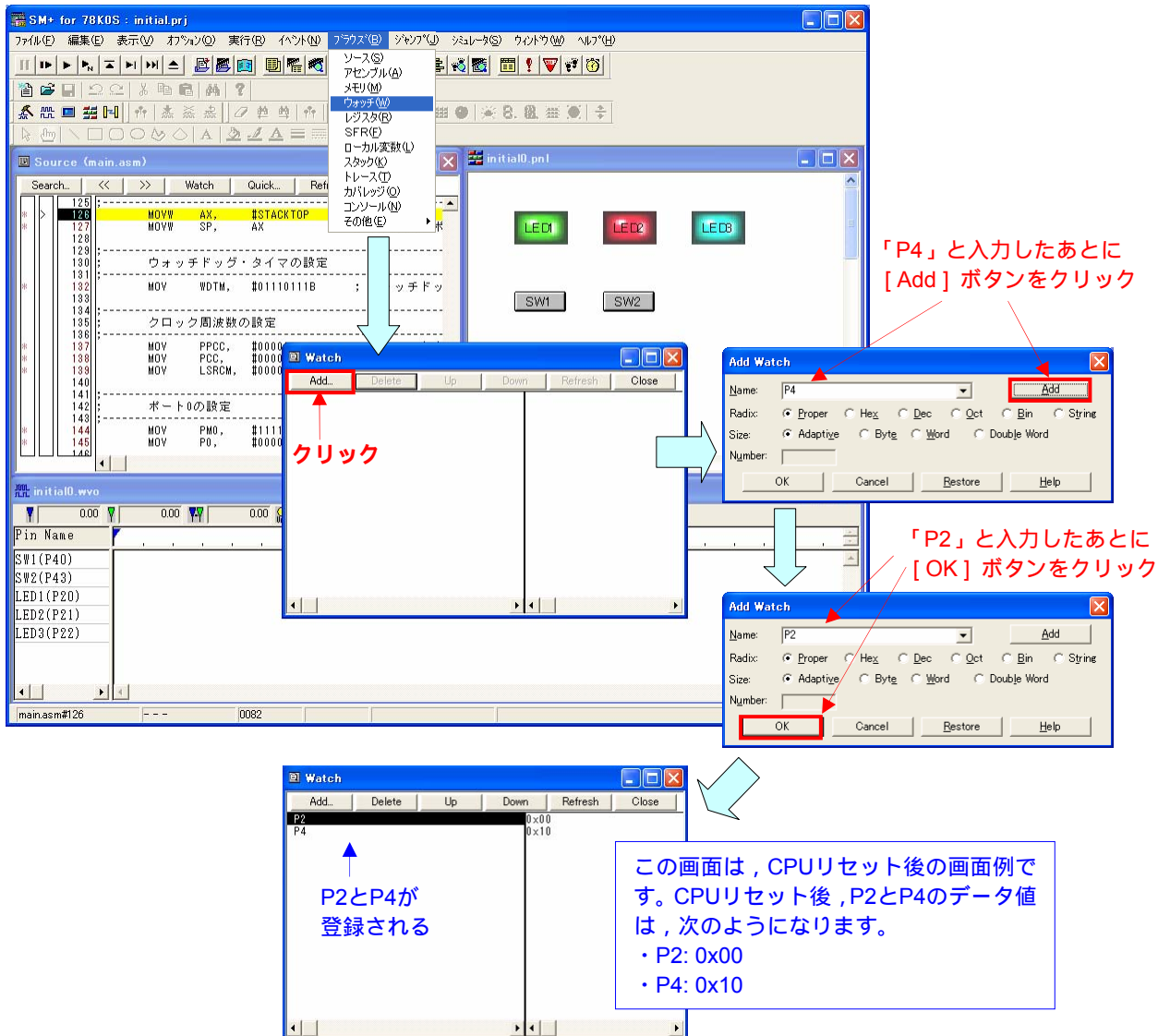
【補 足】SM+のウォッチ機能を使用することにより、ポート2, 4のデータ値の変化を確認することができます。

[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「P4」と入力し、[add] ボタンをクリックすると、[Watch] ウィンドウに、「P4」が登録されます(このとき、[Add watch] ウィンドウは開いたままです)。

次に、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 [SW1] [SW2] ボタンをクリックしてください。[SW1] [SW2] ボタンの組み合わせにより、[Watch] ウィンドウ上のP2とP4のデータ値が変化することを確認してください。

SW1とSW2の組み合わせ	[Watch] ウィンドウのデータ値
SW1: ON, SW2: OFF	P2: 0x06, P4: 0x08
SW1: OFF, SW2: ON	P2: 0x05, P4: 0x01
SW1: ON, SW2: ON	P2: 0x03, P4: 0x00
SW1: OFF, SW2: OFF	P2: 0x07, P4: 0x09

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+ アプリケーション・ノート	サンプル・プログラム スタートアップ・ガイド	PDF
	サンプル・プログラム (割り込み) スイッチ入力による外部割り込み編	PDF

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
;*****
;
;   初期設定
;*****
;
; 【履歴】
;
;   2007.4.-- 新規作成
;*****
;
;
; 【概要】
;
;
; 本サンプルプログラムでは、クロック周波数の選択、ポート入出力の設定などを行うこと
; で、マイコンの基本的な初期設定を行う。初期設定完了後のメイン・ループでは、2つの
; スイッチ入力を検出して3つのLED点灯を制御する。
;
;
; <初期設定の主な設定内容>
;
;
;   ・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・タイマの停止
;   ・CPUクロック周波数を2MHzに設定
;   ・低速内蔵発振器の停止
;   ・ポート・モードの設定（入力ポートor出力ポート）
;   ・プルアップ抵抗接続の設定（入力ポートのみ）
;   ・出力ラッチの設定（出力ポートのみ）
;
;
;
; <スイッチ入力とLED点灯>
;
```

```

;      +                                     +
;      | スイッチ 1 | スイッチ 2 | LED 1 | LED 2 | LED 3 |
;      | (P40)    | (P43)    | (P20) | (P21) | (P22) |
;      |          |          |          |          |          |
;      | OFF     | OFF     | OFF   | OFF   | OFF   |
;      | ON      | OFF     | ON    | OFF   | OFF   |
;      | OFF     | ON      | OFF   | ON    | OFF   |
;      | ON      | ON      | OFF   | OFF   | ON    |
;      +                                     +
;
;
; 【ポート入出力の設定】
;
; 入力ポート：P40, P43
; 出力ポート：P00-P03, P20-P23, P30-P33, P41, 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)  INTP0
      DW    RESET_START      ;(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

```

```

;=====
;
;   ROMの定義
;
;=====
XTBL1  CSEG      AT      0100H
LEDDATA:
    DB      0000011B      ;00: SW1=ON,SW2=ON時の表示データ:LED3がON
    DB      00000101B     ;01: SW1=OFF,SW2=ON時の表示データ:LED2がON
    DB      00000111B     ;02: ダミー
    DB      00000111B     ;03: ダミー
    DB      00000111B     ;04: ダミー
    DB      00000111B     ;05: ダミー
    DB      00000111B     ;06: ダミー
    DB      00000111B     ;07: ダミー
    DB      00000110B     ;08: SW1=ON,SW2=OFF時の表示データ:LED1がON
    DB      00000111B     ;09: SW1=OFF,SW2=OFF時の表示データ:全LEDがOFF

;=====
;
;   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      ; 低速内蔵発振器の停止

;-----
;   ポート0の設定
;-----
MOV     PM0,    #11110000B      ; P00-P03を出力ポートに設定
MOV     P0,     #00000000B      ; P00-P03の出力Low

;-----
;   ポート2の設定
;-----
MOV     PM2,    #11110000B      ; P20-P23を出力ポートに設定
MOV     P2,     #00000111B      ; P20-P22の出力High、P23の出力Low

;-----
;   ポート3の設定
;-----
MOV     PM3,    #11110000B      ; P30-P33を出力ポートに設定
MOV     P3,     #00000000B      ; P30-P33の出力Low

;-----
;   ポート4の設定
;-----
MOV     PM4,    #00001001B      ; P40,P43を入力ポートに、P41,P42,P44-P47を出力ポートに設定
MOV     PU4,    #00001001B      ; P40,P43に内蔵プルアップ抵抗を使用
MOV     P4,     #00000000B      ; P41,P42,P44-P47の出力Low

;-----
;   ポート12の設定
;-----
MOV     PM12,   #11110000B      ; P120-P123を出力ポートに設定
MOV     P12,    #00000000B      ; P120-P123の出力Low

```

```

;-----
;   ポート13の設定
;-----
MOV     P13,    #00000001B      ; P130の出力High

;-----
;   レジスタ値の初期化
;-----
MOVW    HL,     #LEDDATA ; HLにテーブルのアドレスを指定

;*****
;
;   メイン・ループ
;*****
MAIN_LOOP:
MOV     A,      P4              ; スイッチ入力状態の読み出し
AND     A,      #00001001B     ; スイッチ以外のビットをマスク
MOV     L,      A              ; 入力結果を下位 8 ビットにセット
MOV     A,      [HL]          ; テーブルから表示データ読み出し
MOV     P2,     A              ; L E D 出力
BR      $MAIN_LOOP            ; MAIN_LOOPへ

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

初期設定

【履歴】

2007.4.-- 新規作成

【概要】

本サンプルプログラムでは、クロック周波数の選択、ポート入出力の設定などを行うことで、マイコンの基本的な初期設定を行う。初期設定完了後のメイン・ループでは、2つのスイッチ入力を検出して3つのLED点灯を制御する。

<初期設定の主な設定内容>

- ・ウォッチドッグ・タイマの停止
- ・CPUクロック周波数を2MHzに設定
- ・低速内蔵発振器の停止
- ・ポート・モードの設定(入力ポートor出力ポート)
- ・プルアップ抵抗接続の設定(入力ポートのみ)
- ・出力ラッチの設定(出力ポートのみ)

<スイッチ入力とLED点灯>

+						+				
	スイッチ1		スイッチ2		LED1		LED2		LED3	
	(P40)		(P43)		(P20)		(P21)		(P22)	
	OFF		OFF		OFF		OFF		OFF	
	ON		OFF		ON		OFF		OFF	
	OFF		ON		OFF		ON		OFF	
	ON		ON		OFF		OFF		ON	
+										+

【ポート入出力の設定】

入力ポート：P40, P43

出力ポート：P00-P03, P20-P23, P30-P33, P41, 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 */

/******

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; /* 低速内蔵発振器の停止 */

/*-----

ポート0の設定

-----*/

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

P0 = 0b00000000; /* P00-P03の出力Low */

/*-----

ポート2の設定

-----*/

```

PM2 = 0b11110000;          /* P20-P23を出力ポートに設定 */
P2  = 0b00000111;          /* P20-P22の出力High、P23の出力Low */

/*-----*/
      ポート3の設定
/*-----*/

PM3 = 0b11110000;          /* P30-P33を出力ポートに設定 */
P3  = 0b00000000;          /* P30-P33の出力Low */

/*-----*/
      ポート4の設定
/*-----*/

PM4 = 0b00001001;          /* P40,P43を入力ポートに、P41,P42,P44-P47を出力ポートに
設定 */
PU4 = 0b00001001;          /* P40,P43に内蔵プルアップ抵抗を使用 */
P4  = 0b00000000;          /* P41,P42,P44-P47の出力Low */

/*-----*/
      ポート12の設定
/*-----*/

PM12 = 0b11110000;          /* P120-P123を出力ポートに設定 */
P12  = 0b00000000;          /* P120-P123の出力Low */

/*-----*/
      ポート13の設定
/*-----*/

P13 = 0b00000001;          /* P130の出力High */
}

/*****

      メイン・ループ

*****/

void main(void){

    const unsigned char  OUTDATA[10] = {0x03,0x05,0x07,0x07,0x07,0x07,0x07,0x07,0x06,0x07};
    /* 表示データパターン用配列 */
    unsigned char  INDATA;          /* スイッチ入力データ変数 */

    while(1){
        INDATA = P4 & 0b00001001;  /* 有効スイッチ情報取得 */
        P2 = OUTDATA[INDATA];          /* テーブルから表示データを読み出して表示 */
    }
}

```

```

    }
}

```

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	May 2007	-	-
第2版	July 2008	p.12	4.2 ベクタ・テーブルの設定 ・サンプル・プログラムの「割り込み（仮称，作成中）」を「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」に変更
		p.29	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4月現在）を（2008年7月現在）に変更
		pp.29-31	5.1 サンプル・プログラムのビルドを変更
		p.31	5.2 SM+での動作 ・（1）を追加
		p.35	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。
