

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-L

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

78K0/KY2-Lマイクロコントローラ
 78K0/KA2-Lマイクロコントローラ
 78K0/KB2-Lマイクロコントローラ
 78K0/KC2-L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 ... 4
 - 2.1 回路イメージ ... 4
 - 2.2 マイコン以外の使用デバイス ... 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 ... 9
 - 4.2 ベクタ・テーブルの設定 ... 15
 - 4.3 ROM/RAMサイズの設定 ... 16
 - 4.4 スタック・ポインタの設定 ... 17
 - 4.5 ウォッチドッグ・タイマの設定と制御 ... 18
 - 4.6 クロックの設定 ... 19
 - 4.7 ポートの設定... 27
 - 4.8 メイン処理... 34
- 第5章 関連資料 ... 36
- 付録A プログラム・リスト ... 37
- 付録B 78K0/KC2-Lの44ピン製品を使用する場合 ... 55
- 付録C 改版履歴 ... 56

- 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に、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

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

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

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

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

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社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。

M8E0710J

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、入出力ポートの設定など、78K0/Kx2-L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

<オプション・バイトでの設定>

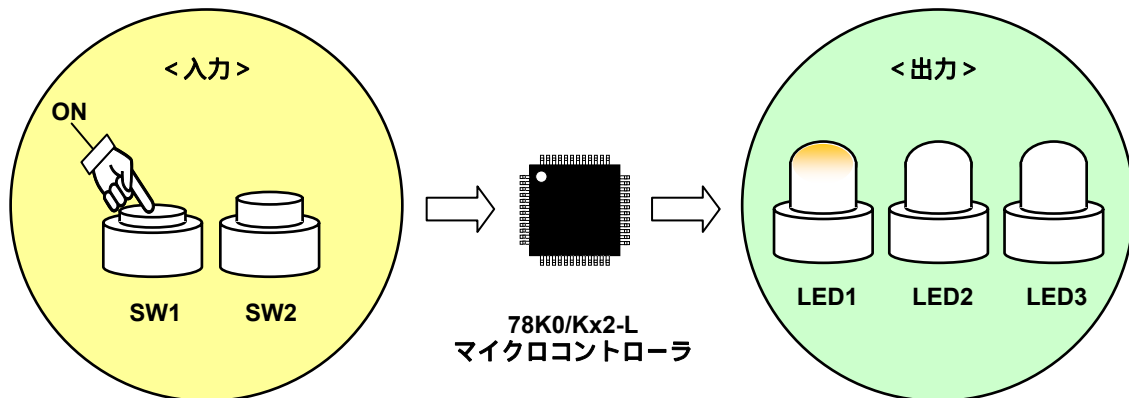
- ・低速内蔵発振器の動作をソフトウェアにより停止可能に設定
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振クロック周波数を8 MHzに設定
- ・LVIデフォルト・スタート機能停止

<リセット解除後の初期化処理での設定>

- ・ROM/RAMサイズの設定
- ・CPUクロックを高速内蔵発振クロック動作に設定（4 MHz）
- ・低速内蔵発振器の停止
- ・入出力ポートの設定
- ・使用しない周辺ハードウェアの動作禁止

(2) メイン処理動作の内容

78K0/Kx2-L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

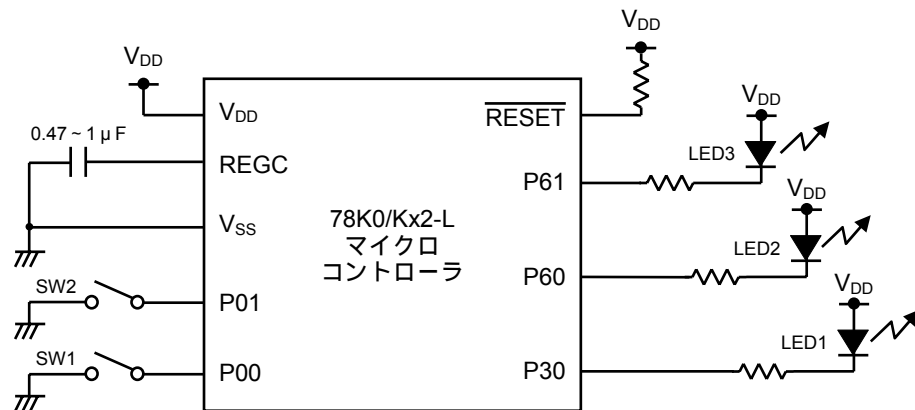
注意 デバイス使用上の注意事項については、[78K0/Kx2-L ユーザーズ・マニュアル](#)を参照してください。

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マイコン以外の使用デバイスを説明します。

2.1 回路イメージ

回路イメージを次に示します。



注意1. 1.8 V V_{DD} 5.5 Vの電圧範囲で使用してください。

2. AV_{REF} 端子は V_{DD} に直接接続してください。
3. AV_{SS} 端子はGNDに直接接続してください（78K0/KC2-L, 78K0/KB2-Lマイクロコントローラのみ）。
4. REGCはコンデンサ（0.47 ~ 1 μ F）を介し、 V_{SS} に接続してください。
5. 78K0/KY2-Lと78K0/KA2-Lの V_{SS} は、A/Dコンバータのグランド電位と兼用しています。 V_{SS} を必ず安定しているGNDに接続してください。
6. 回路イメージ中に記載のない未使用端子は以下のように処理してください。
 - ・出力ポート : 出力モードに設定し、オープン（未接続）にしてください
 - ・入力ポート : 個別に抵抗を介して、 V_{DD} または V_{SS} に接続してください
7. このサンプル・プログラムでは、P121/X1/TOOLC0端子、およびP122/X2/EXCLK/TOOLD0端子をオンチップ・デバッグ用に使用します。

2.2 マイコン以外の使用デバイス

マイコン以外に使用するデバイスを次に示します。

(1) 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)	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	注
main.c (C言語版)			
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 (ウォッチドッグ・タイマの設定，低速内蔵発振器の設定，高速内蔵発振クロック周波数の選択などを行います)		
Kx2-L_Init.prw	統合開発環境 PM+用ワーク・スペース・ファイル		
Kx2-L_Init.prj	統合開発環境 PM+用プロジェクト・ファイル		

注 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。

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+で使用するファイルを同封

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。

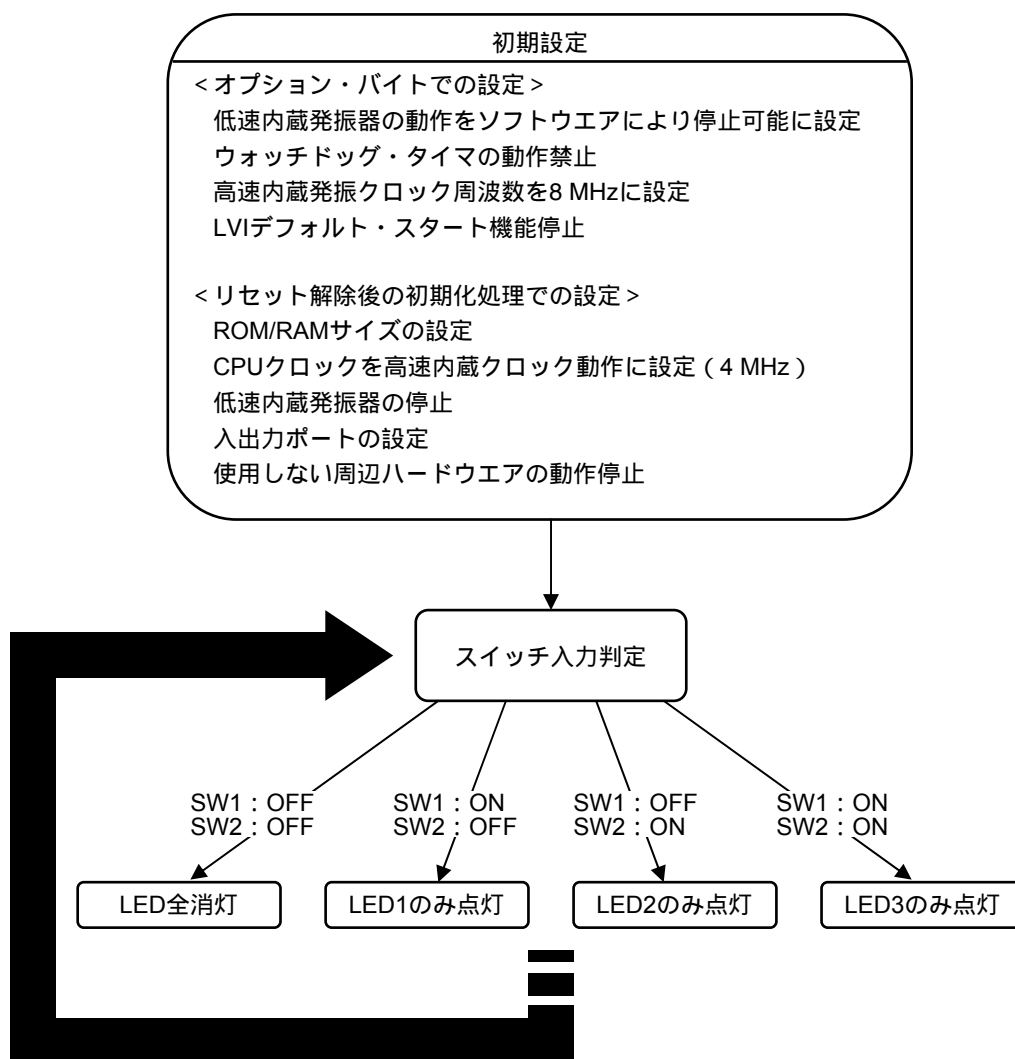
- ・入力ポート（スイッチ入力用）：P00, P01
- ・出力ポート（LED点灯用）：P30, P60, P61

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。

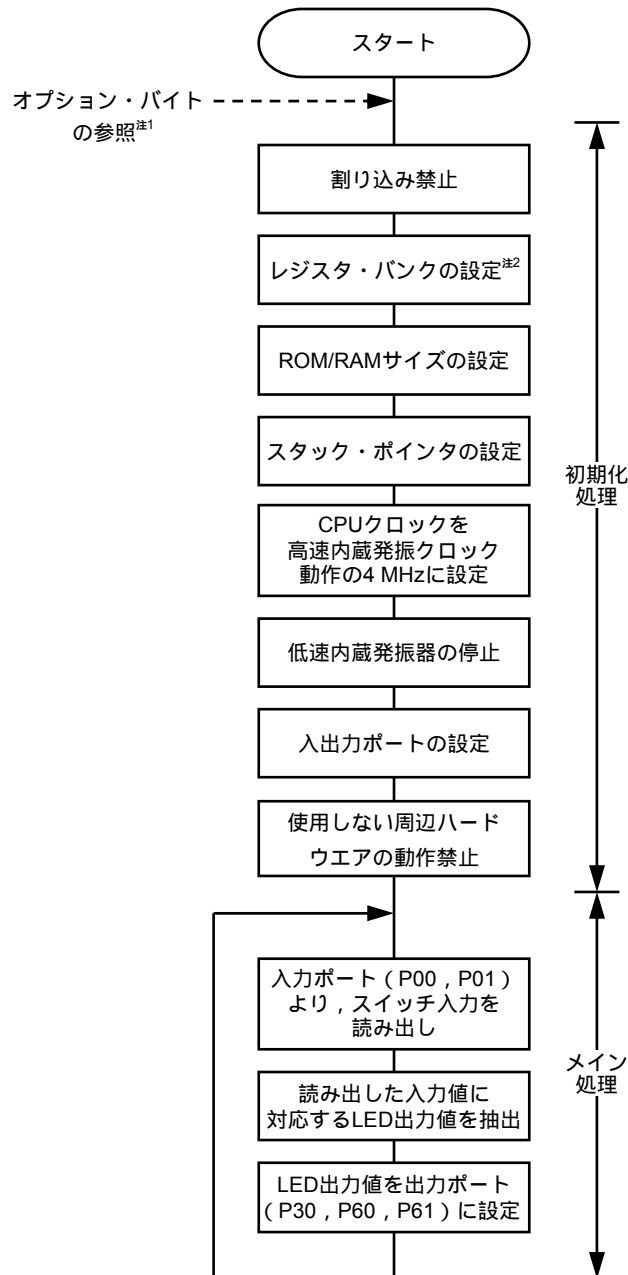
初期設定完了後は、2つのスイッチ入力（SW1, SW2）の組み合わせに応じて、3つのLED（LED1, LED2, LED3）の点灯を制御します。

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。



注1. 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・バイトで以下の設定を行います。

- ・低速内蔵発振器の動作をソフトウェアにより停止可能に設定
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振クロック周波数を8 MHzに設定
- ・LVIデフォルト・スタート機能停止

2. 78K0/Kx2-Lシリーズの汎用レジスタは、4レジスタ・バンク構成になっていますので、通常処理で使用するレジスタと割り込み時で使用するレジスタをバンクごとに切り替えることにより、効率のよいプログラムを作成できます。なお、このサンプル・プログラムでは、レジスタ・バンク0のみを使用します。

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、ROM/RAMサイズの設定、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、入出力ポートの設定、およびメイン処理について説明します。

C言語によるプログラムを実行させるには、システムへ組み込むためのROM化処理、ユーザ・プログラム（main関数）の起動などを行うプログラムが必要となります。このプログラムのことをスタートアップ・ルーチンと呼びます。このスタートアップ・ルーチンは一般的に、マイコンがリセット（初期化）後、最初に動くプログラムで、CPU、メモリ、I/Oポートなどのハードウェア初期設定や、メイン処理ルーチンを動かすための初期設定を行います。プログラムは基本的に、このスタートアップ・ルーチンから始まり、その次にメイン・ルーチン処理が実行され、その後サブルーチン処理や割り込み処理と続いて実行されます。

このサンプル・プログラムのC言語版では、hdwinit関数によりクロック関連の設定や周辺ハードウェアの初期設定などを行っており、hdwinit関数を実行したあとにmain関数を実行するので、メイン処理はmain関数に記述します。アセンブリ言語版ではリセット（初期化）後、ベクタ・テーブルの0000H番地に書かれたIRESET番地からプログラムが実行され、C言語版のhdwinit関数と同じように、クロック関連の設定や周辺ハードウェアの初期設定などを行い、メイン処理に入ります。

なお、このサンプル・プログラムでは、使用しない周辺ハードウェアは動作禁止としています。使用時には、用途や機能に合わせて、各レジスタを設定してください。

スタートアップ・ルーチンの詳細については、[CC78K0 操作編 ユーザーズ・マニュアル](#)のスタートアップ・ルーチンの章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、[78K0/Kx2-L 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、[78K/0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。

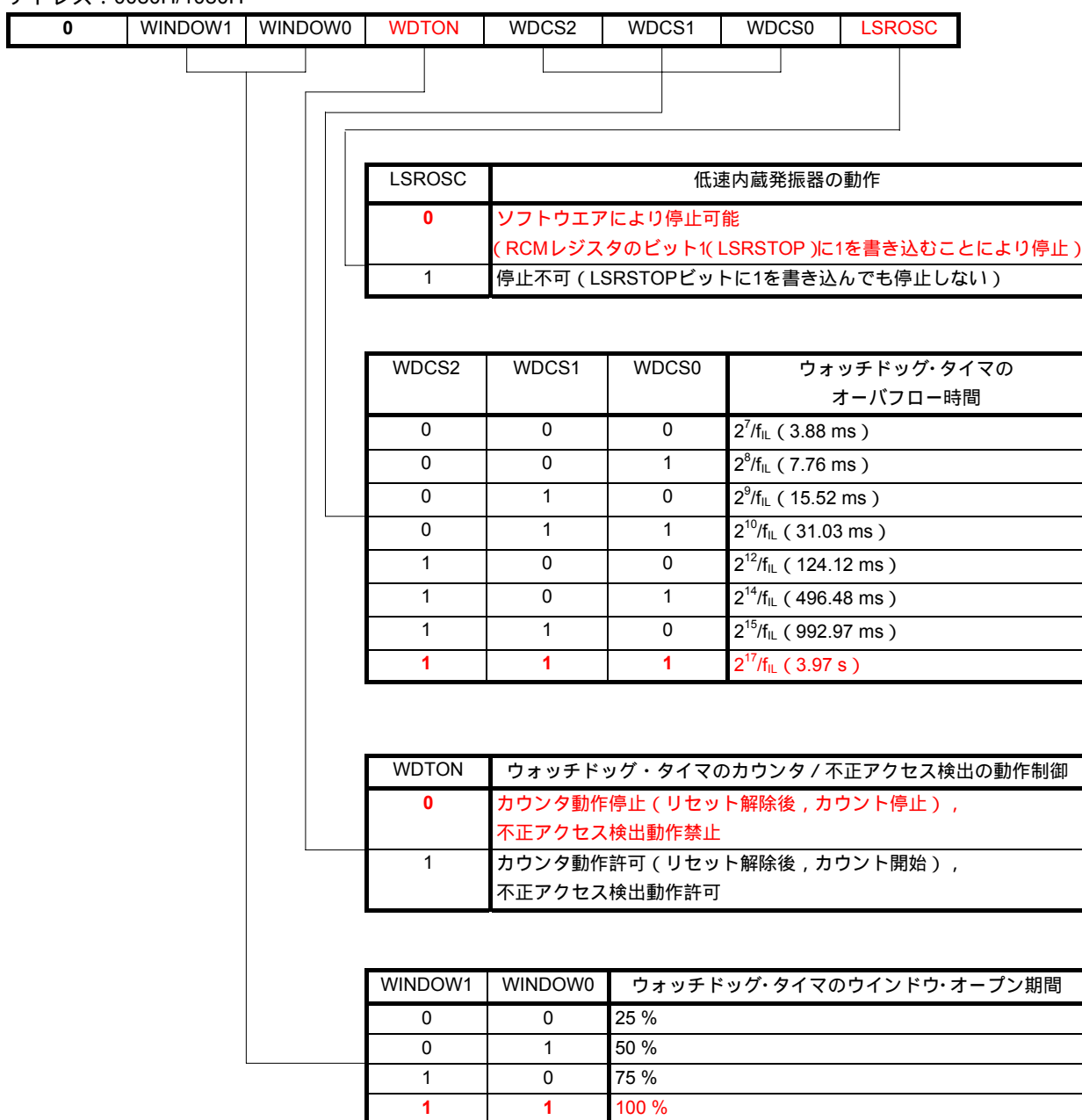
4.1 オプション・バイトの設定

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) 低速内蔵発振器の発振制御
- (2) ウォッチドッグ・タイマのインターバル時間の設定
- (3) ウォッチドッグ・タイマのカウンタ動作設定
- (4) ウォッチドッグ・タイマのウインドウ・オープン期間の設定
- (5) LVIデフォルト・スタートの動作制御
- (6) 高速内蔵発振クロック周波数の選択
- (7) オンチップ・デバッグ動作制御

図4 - 1 オプション・バイトのフォーマット (アドレス : 0080H/1080H)

アドレス : 0080H/1080H^注



注 ブート・スワップ使用時は、0080Hと1080Hが切り替わるので、1080Hにも0080Hと同じ値を設定してください。

注意1. WDCS2 = WDCS1 = WDCS0 = 0かつWINDOW1 = WINDOW0 = 0の組み合わせは設定禁止です。

2. フラッシュ・メモリのセルフ・プログラミング時およびEEPROMエミュレーション時でも、ウォッチドッグ・タイマの動作は継続します。ただし、これらの処置中には割り込みの受け付け時間が遅れるので、遅延を考慮し、オーバフロー時間およびウインドウ・サイズを設定してください。

3. LSROSC = 0 (ソフトウェアにより停止可能) の場合、内蔵発振モード・レジスタ (RCM) のビット0 (LSRSTOP) の設定に関係なく、HALT/STOPモード時では、ウォッチドッグ・タイマにカウント・クロックは供給されません。

ただし、8ビット・タイマH1が低速内蔵発振クロックで動作している場合は、HALT/STOPモード時でも、8ビット・タイマH1にカウント・クロックが供給されます。

4. ビット7には必ず0を書き込んでください。

備考1. f_{IL} : 低速内蔵発振クロック周波数

2. () 内は $f_{IL} = 33 \text{ kHz (MAX.)}$ の場合

3. 図の赤字部分がサンプル・プログラムでの設定値となります。

図4 - 2 オプション・バイトのフォーマット (アドレス : 0081H/1081H)

アドレス : 0081H/1081H^{注1,2}

0	0	0	0	0	0	0	LVISTART
---	---	---	---	---	---	---	----------

LVISTART	LVIデフォルト・スタートの動作制御
0	リセット解除時または電源立ち上げ時に、LVIがデフォルトでオフ (LVIデフォルト・スタート機能停止)
1	リセット解除時または電源立ち上げ時に、LVIがデフォルトでオン (LVIデフォルト・スタート機能動作)

- 注1. LVISTARTは、専用フラッシュ・メモリ・プログラマによる書き込みでのみ設定可能です。セルフ・プログラミング、およびセルフ・プログラミング中のブート・スワップ動作では設定できません。ただし、ブート・スワップ時は、0081Hと1081Hが切り替わるので、あらかじめ1081Hにも0081Hと同じ値を設定してください。
2. LVIデフォルト・スタートの設定内容を変更する場合は、フラッシュ・メモリの一括消去(チップ消去)後に、再度0081Hに値を設定してください。指定したブロックのメモリ消去後の設定変更は無効となります。

注意 ビット7-1には必ず0を書き込んでください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

図4 - 3 オプション・バイトのフォーマット (アドレス : 0082H/1082H)

アドレス : 0082H/1082H^注

0	0	0	0	0	0	0	R4M8MSEL
---	---	---	---	---	---	---	----------

R4M8MSEL	高速内蔵発振クロック周波数の選択
0	8 MHz (TYP.)
1	4 MHz (TYP.)

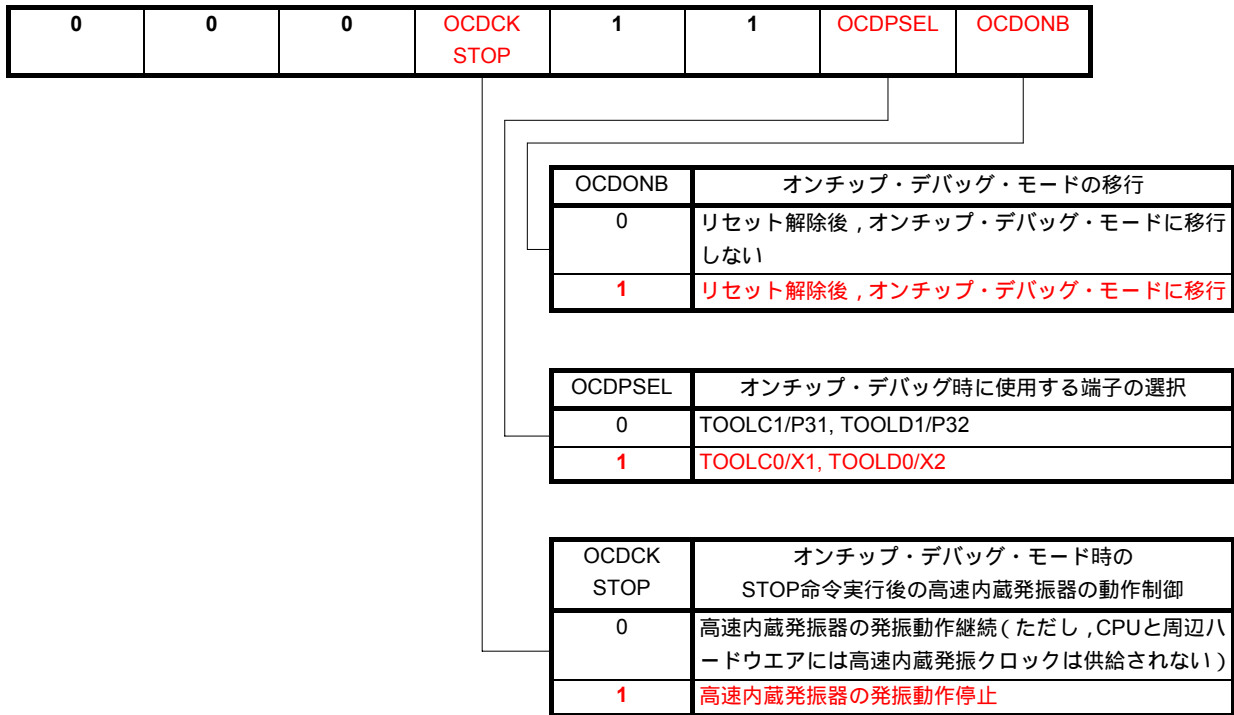
注 ブート・スワップ時は、0082Hと1082Hが切り替わるので、あらかじめ1082Hにも0082Hと同じ値を設定してください。

注意 ビット7-1には必ず0を書き込んでください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

図4 - 4 オプション・バイトのフォーマット (アドレス : 0083H/1083H)

アドレス : 0083H/1083H^注



注 ブート・スワップ時は, 0083Hと1083Hが切り替わるので, あらかじめ1083Hにも0083Hと同じ値を設定してください。

注意 ビット7-5には0を, ビット3, 2には1を必ず書き込んでください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

図4-5 オプション・バイトのフォーマット (アドレス : 0084H/1084H)

アドレス : 0084H/1084H^注

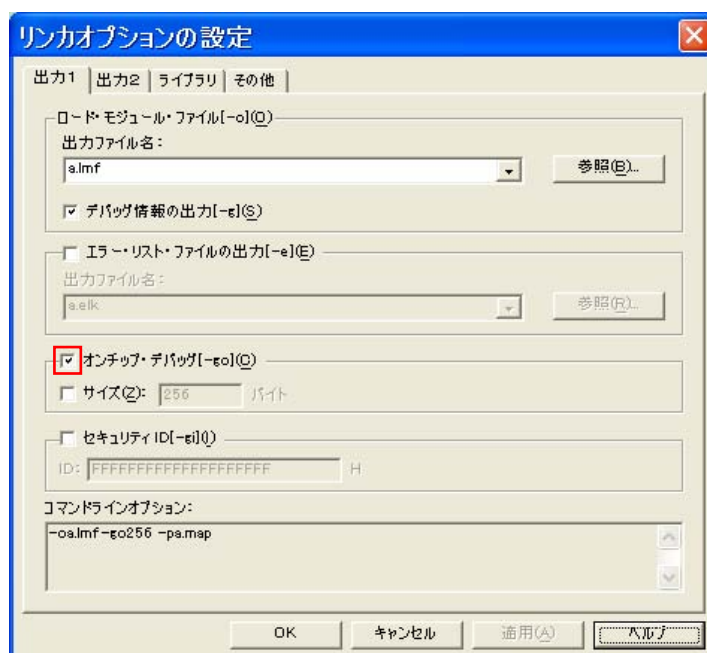


OCDEN1	OCDEN0	オンチップ・デバッグ動作制御
0	0	動作禁止
0	1	設定禁止
1	0	動作許可, オンチップ・デバッグ・セキュリティID 認証失敗時にフラッシュ・メモリのデータを消去し ない
1	1	動作許可, オンチップ・デバッグ・セキュリティID 認証失敗時にフラッシュ・メモリのデータを消去す る

注 ブート・スワップ時は, 0084Hと1084Hが切り替わるので, あらかじめ1084Hにも0084Hと同じ値を設定してください。

注意1. ビット7-2には必ず0を書き込んでください。

2. オンチップ・デバッグ機能を使用するためには, オプション・バイトの設定に加え, リンカ・オプションの設定が必要となります。オンチップ・デバッグ機能を使用する場合, 次に示すように, PM+の [ツール] → [リンカ・オプションの設定] にて, オンチップ・デバッグ [-go] をチェックしてください。なお, オンチップ・デバッグ機能についての詳細は, [78K0/Kx2-L ユーザーズ・マニュアル](#)の第26章 オンチップ・デバッグ機能を参照してください。



備考 図の赤字部分がサンプル・プログラムでの設定値となります。

前述したオプション・バイトの設定値をソフトウェアで記述すると次のようになります。

```
TOPTIONB      CSEG      AT      0080H
               DB      01101110B
               DB      00000000B
               DB      00000000B
               DB      00011111B
               DB      00000010B
```

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。

```
TOPTIONB      CSEG      AT      0080H
               DB      01101110B
               DB      00000000B
               DB      00000000B
               DB      00011111B
               DB      00000010B

END
```



【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、[RA78K0 言語編 ユーザーズ・マニュアル](#)を参照してください。

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。

なお、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、設定は不要です。

【設定例】 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）

```

XVECT1      CSEG   AT      0000H      アドレス      機能名
           DW      RESET_START      ;0000H      RESET入力, POC, LVI, WDT
XVECT2      CSEG   AT      0004H
           DW      IINIT              ;0004H      INTLVI
           DW      IINIT              ;0006H      INTP0
           DW      IINIT              ;0008H      INTP1
           DW      IINIT              ;000AH     INTP2
           ... (略) ...
           DW      IINIT              ;003AH     INTP10
           DW      IINIT              ;003CH     INTP11
           DW      IINIT              ;003EH     BRK
           ... (略) ...
;*****
;
;      不要な割り込み要因による割り込み処理
;
;*****
XMAIN      CSEG   UNIT
IINIT:
;      不要な割り込みが発生した場合、ここに分岐します。
;      ここでは何も処理をしないで元の処理に戻ります

      RETI

```

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、すべて「IINIT」を設定します（上記）。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「IINIT」（上記）に分岐させ、不要な割り込みとして何も処理をさせずに割り込みから復帰させます。



【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、[CC78K0 言語編 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。

4.3 ROM/RAMサイズの設定

78K0/Kx2-Lシリーズは、モデルによりROM容量、内部高速RAM容量が異なるため、初期化処理内でメモリ・サイズ切り替えレジスタ（IMS）を設定する必要があります。

プロジェクトのデバイス設定（デバイス・ファイル設定）は、デフォルトでは次のようになっています。

```
78K0/KY2-L : μ PD78F0557
78K0/KA2-L : μ PD78F0567
78K0/KB2-L : μ PD78F0578
78K0/KC2-L : μ PD78F0588
```

サンプル・ソース内にすべてのモデルにおけるIMSの設定値が記述されているので、使用するマイコンに応じて設定値を切り替えてご使用ください。

【例】μ PD78F0586を使用する場合

・アセンブリ言語

```

;-----
;          ROM/RAMサイズの設定
;-----
;          モデルにより設定値が異なるので注意してください。
;          使用モデルの設定を有効にしてください。（デフォルトではuPD78F0588）
;-----
; を削除  ;uPD78F0581,uPD78F0586使用時の設定
;          ;MOV    IMS,    #042H          ;ROM/RAMサイズの設定
;
;          ;uPD78F0582,uPD78F0587使用時の設定
;          ;MOV    IMS,    #004H          ;ROM/RAMサイズの設定
;
; を追加  ;uPD78F0583,uPD78F0588使用時の設定
;          ;MOV    IMS,    #0C8H          ;ROM/RAMサイズの設定

```

μ PD78F0586を使用する場合、『;uPD78F0583,uPD78F0588使用時の設定』の1行下の『MOV』の前に『;』を追加して、行をコメント行にすることで無効にします。次に、『;uPD78F0581,uPD78F0586使用時の設定』の1行下の『MOV』の前の『;』を削除して命令行を有効にしてください。これで、IMSの設定をμ PD78F0586用に設定することができます。

・C言語

```

/*-----
/*          ROM/RAMサイズの設定
/*-----
/*          モデルにより設定値が異なるので注意してください。
/*          使用モデルの設定を有効にしてください。（デフォルトではuPD78F0588）
/*-----*/
/* と */ /* uPD78F0581,uPD78F0586使用時の設定 */
/* を削除 /*IMS = 0x42;*/          /* ROM/RAMサイズの設定 */
/*
/*          /* uPD78F0582,uPD78F0587使用時の設定 */
/*          /*IMS = 0x04;*/          /* ROM/RAMサイズの設定 */
/*
/* と */ /* uPD78F0583,uPD78F0588使用時の設定 */
/* を追加 /*IMS = 0xC8;*/          /* ROM/RAMサイズの設定 */

```

μ PD78F0586を使用する場合、『/* uPD78F0583,uPD78F0588使用時の設定 */』の1行下の『IMS = 0xC8;』の前に『/*』、後に『*/』を追加して、行をコメント行にすることで無効にします。次に、『/* uPD78F0581,uPD78F0586使用時の設定 */』の1行下の『IMS = 0x42;』の前の『/*』と後の『*/』を削除して、命令行を有効にしてください。これで、IMSの設定をμ PD78F0586用に設定することができます。

4.4 スタック・ポインタの設定

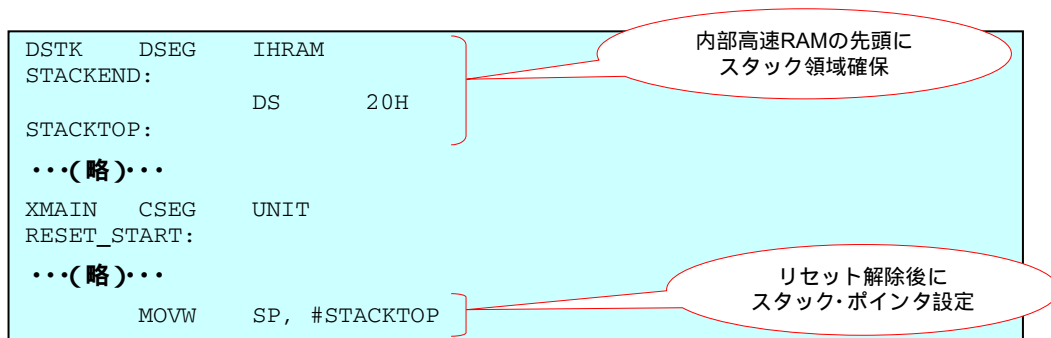
スタック領域は、プログラム・カウンタ、レジスタ値、PSW（プログラム・ステータス・ワード）などのデータを、一時的に格納するメモリ領域です。スタック領域は、内部高速RAMにのみ指定できます。このスタック領域の先頭アドレスはスタック・ポインタで設定し、スタック領域を確保します。

スタック領域は、次の命令実行および割り込み時に使用します。

- ・PUSH, CALL, CALLT, CALLF 割り込み : スタック領域へデータ退避
- ・POP, RET, RETI : スタック領域からデータ復帰

スタック領域の確保は、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・ルーチンにてスタック領域が自動的に確保されるため、設定は不要です。

【例】 内部高速RAMの先頭の32バイトを、スタック領域として使用する場合（サンプル・プログラムの設定と同内容）



上記の記述により、内部高速RAMの先頭の32バイトをスタック領域として確保することが可能となります。

なお、内部高速RAMの先頭アドレスはデバイスによって異なりますので、スタックとして確保される領域のアドレスは次のようになります。

- 78K0/KY2-Lの場合 : 0FD80H - 0FD9FH
- 78K0/KA2-Lの場合 : 0FD00H - 0FD1FH
- 78K0/KB2-Lの場合 : 0FC00H - 0FC1FH
- 78K0/KC2-Lの場合 : 0FB00H - 0FB1FH

このサンプル・プログラムではDSEG疑似命令IHRAM^注を使用することで、絶対的なアドレスを記述することなく内部高速RAMの先頭を指定しています。

注 詳細は[RA78K0 言語編 ユーザーズ・マニュアル](#)を参照してください。

4.5 ウォッチドッグ・タイマの設定と制御

ウォッチドッグ・タイマの設定は、オプション・バイトで行います。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。

なお、ウォッチドッグ・タイマを使用する場合（WDTON=1の場合）、ウォッチドッグ・タイマの制御は、ウォッチドッグ・タイマ・イネーブル・レジスタ（WDTE）で行います。WDTEにACHを書き込むことにより、ウォッチドッグ・タイマのカウンタをクリアし、再びカウントを開始します。リセット信号により、WDTEは9AH[※]になります。

注 オプション・バイト（0080H）のWDTONの設定値によって、WDTEのリセット値が異なります。

WDTONの設定	WDTEのリセット値
0（ウォッチドッグ・タイマのカウント動作禁止）	1AH
1（ウォッチドッグ・タイマのカウント動作許可）	9AH

- 注意1. WDTEにACH以外の値を書き込んだ場合、内部リセット信号を発生します。ただし、ウォッチドッグ・タイマのソース・クロックが停止している場合は、ウォッチドッグ・タイマのソース・クロックが再び動作開始した時点で、内部リセット信号を発生します。
- WDTEに1ビット・メモリ操作命令を実行した場合、内部リセット信号を発生します。ただし、ウォッチドッグ・タイマのソース・クロックが停止している場合は、ウォッチドッグ・タイマのソース・クロックが再び動作開始した時点で、内部リセット信号を発生します。
 - WDTEのリード値は、“9AH / 1AH”（書き込んだ値（“ACH”）とは異なる値）になります。



【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。



【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、クロック周波数選択やポート入出力設定などの初期設定は、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、[CC78K0 言語編 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。

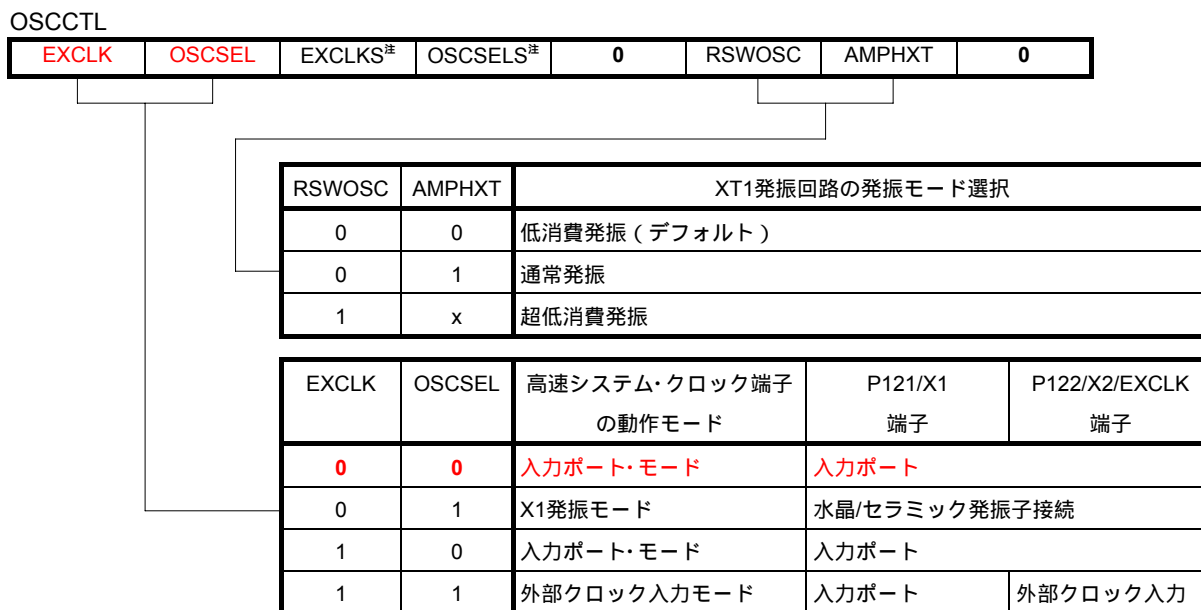
4.6 クロックの設定

CPUクロック (f_{CPU}) と周辺ハードウェアへの供給クロック (f_{PRS}) は、メイン・システム・クロック (f_{XP}) を分周して、生成されます。

(1) クロック動作モードの選択

クロック動作モード選択レジスタ (OSCCTL) でクロック動作モードを選択します。

図4 - 6 クロック動作モード選択レジスタ (OSCCTL) のフォーマット



注 EXCLKS, OSCSELSはサブシステム・クロック搭載デバイス (78K0/KC2-L) のみ有効です。なお、詳細は [\(3\) サブシステム・クロック端子の動作モードの設定](#) を参照してください。

注意1. EXCLKとOSCSELを別の値に書き換える場合、メインOSCコントロール・レジスタ (MOC) のビット7 (MSTOP) が1 (X1発振回路停止またはEXCLK端子からの外部クロック無効) であることを必ず確認してください。

2. 次のビットには必ず0を書き込んでください。

78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-Lの場合 : ビット5-0

78K0/KC2-Lの場合 : ビット3, 0

3. XT1発振回路は低消費電力を実現するために、増幅度が低い回路になっています。設計の際は、次の点に注意してください。

- ・XT1端子, XT2端子と発振子との配線は極力短くし、寄生容量、配線抵抗を小さくしてください。特に超低消費発振 (RSWOSC = 1) を選択している場合はご注意ください。
- ・XT1発振回路の周辺には、できるかぎりV_{SS}と同電位のグランド・パターンを配置してください。
- ・XT1端子, XT2端子と発振子の信号線は他の信号と交差させないでください。また、変化する大電流が流れる線と接近させないでください。

4. 注意事項の詳細は、[78K0/Kx2-L ユーザーズ・マニュアル](#)の第5章 クロック発生回路を参照してください。

備考1. x : don't care

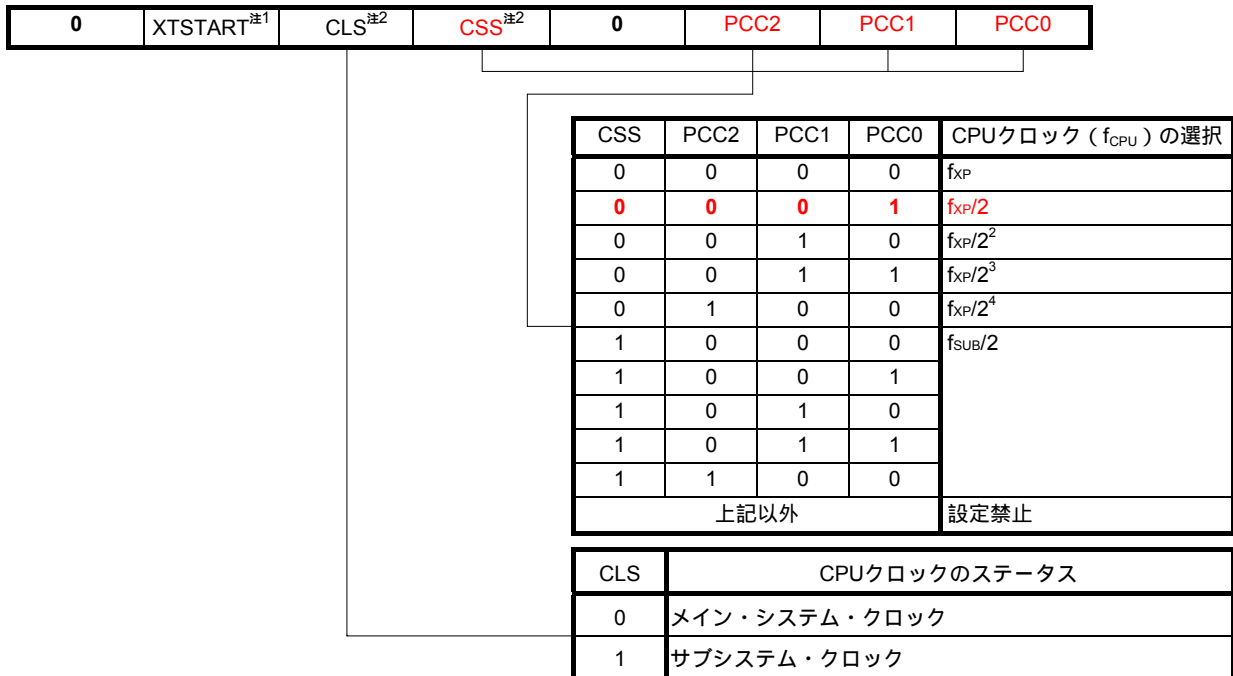
2. 図の赤字部分がサンプル・プログラムでの設定値となります。

(2) CPUクロック (f_{CPU}) の選択, 分周比の設定

プロセッサ・クロック・コントロール・レジスタ (PCC) でCPUクロック (f_{CPU}) の選択, 分周比を設定します。

図4 - 7 プロセッサ・クロック・コントロール・レジスタ (PCC) のフォーマット

PCC



注1. XTSTARTはサブシステム・クロック搭載デバイス (78K0/KC2-L) のみ有効です。なお, 詳細は [\(3\) サブシステム・クロック端子の動作モードの設定](#) を参照してください。

2. CLS, CSSはサブシステム・クロック搭載デバイス (78K0/KC2-L) のみ有効です。

注意1. 次のビットには必ず0を書き込んでください。

78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-Lの場合 : ビット7-3

78K0/KC2-Lの場合 : ビット7, 3

2. PCCの分周比の設定では, 周辺ハードウェア・クロック (f_{PRS}) は分周されません。

3. ビット5は, Read Onlyです。

備考1. f_{XP} : メイン・システム・クロック周波数

2. f_{SUB} : サブシステム・クロック周波数

3. 図の赤字部分がサンプル・プログラムでの設定値となります。

78K0/Kx2-Lマイクロコントローラの一番速い命令はCPUクロック2クロックで実行されます。したがって、CPUクロック (f_{CPU}) と最小命令実行時間の関係は、次のようになります。

CPUクロック (f_{CPU})	最小命令実行時間： $2/f_{CPU}$			
	メイン・システム・クロック			サブシステム ・クロック ^{注2}
	高速システム・クロック ^{注1}	高速内蔵発振クロック ^{注1}		
	10 MHz動作時	8 MHz (TYP.) 動作時	4 MHz (TYP.) 動作時	32.768 kHz動作時
f_{XP}	0.2 μ s (TYP.)	0.25 μ s (TYP.)	0.5 μ s (TYP.)	-
$f_{XP}/2$	0.4 μ s (TYP.)	0.5 μ s (TYP.)	1.0 μ s (TYP.)	-
$f_{XP}/2^2$	0.8 μ s (TYP.)	1.0 μ s (TYP.)	2.0 μ s (TYP.)	-
$f_{XP}/2^3$	1.6 μ s (TYP.)	2.0 μ s (TYP.)	4.0 μ s (TYP.)	-
$f_{XP}/2^4$	3.2 μ s (TYP.)	4.0 μ s (TYP.)	8.0 μ s (TYP.)	-
$f_{SUB}/2$	-	-	-	122.1 μ s

- 注1. CPUクロックに供給するメイン・システム・クロックの設定（高速システム・クロック / 高速内蔵発振クロック）は、メイン・クロック・モード・レジスタ（MCM）で行います。
2. サブシステム・クロック搭載デバイス（78K0/KC2-L）のみ。

(3) サブシステム・クロック端子の動作モードの設定[※]

プロセッサ・クロック・コントロール・レジスタ (PCC) のビット6 (XTSTART) とクロック動作モード選択レジスタ (OSCCTL) のビット5, 4 (EXCLKS, OSCSELS) を組み合わせて, サブシステム・クロック端子の動作モードを設定します。

注 サブシステム・クロック搭載デバイス (78K0/KC2-L) のみ。

XTSTART, EXCLKS, およびOSCSELSの設定値とサブシステム・クロック端子の動作モードの関係は, 次のようになります。

PCC ビット5 XTSTART	OSCCTL		サブシステム・クロック端子 の動作モード	P123/XT1端子	P124/XT2/EXCLKS端子
	ビット5 EXCLKS	ビット4 OSCSELS			
0	0	0	入力ポート・モード	入力ポート	
0	0	1	XT1発振モード	水晶発振子接続	
0	1	0	入力ポート・モード	入力ポート	
0	1	1	外部クロック入力モード	入力ポート	外部クロック入力
1	x	x	XT1発振モード	水晶発振子接続	

注意 XTSTART, EXCLKSとOSCSELSを別の値に書き換える場合, プロセッサ・クロック・コントロール・レジスタ (PCC) のビット5 (CLS) が0 (メイン・システム・クロックでCPU動作) であることを確認してください。

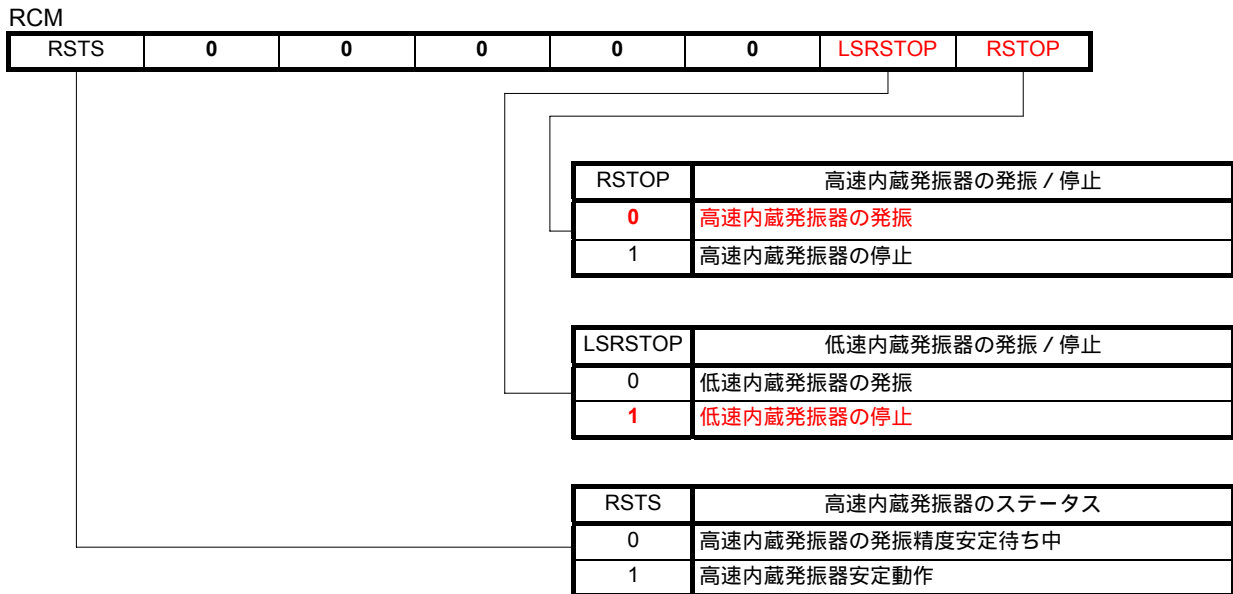
備考1. x : don't care

2. 図の赤字部分がサンプル・プログラムでの設定値となります。

(4) 内蔵発振器の動作モードの設定

内蔵発振モード・レジスタ(RCM)で高速内蔵発振器および低速内蔵発振器の動作モードを設定します。

図4 - 8 内蔵発振モード・レジスタ (RCM) のフォーマット



- 注意1. リセット解除直後は00Hですが、高速内蔵発振器の発振精度安定待ち後に、自動的に80Hに切り替わりま
す。
2. ビット7は、Read Onlyです。
 3. ビット6-2には必ず0を書き込んでください。
 4. RSTOPに1を設定するとき、必ずCPUクロックが高速内蔵発振クロック以外で動作していることを確認
してください。具体的には、次のいずれかの条件です。
 - a. 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-Lの場合
 - ・MCS = 1のとき (CPUクロックが高速システム・クロックで動作)
 - b. 78K0/KC2-Lの場合
 - ・MCS = 1, CLS = 0のとき (CPUクロックが高速システム・クロックで動作)
 - ・CLS = 1のとき (CPUクロックがサブシステム・クロックで動作)
- また、高速内蔵発振クロックで動作している周辺ハードウェアを停止してから、RSTOPに1を設定して
ください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

(5) 高速システム・クロックの動作モードの設定

メインOSCコントロール・レジスタ (MOC) で高速システム・クロックの動作モードを設定します。

図4 - 9 メインOSCコントロール・レジスタ (MOC) のフォーマット

MOC

MSTOP	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---

MSTOP	高速システム・クロックの動作制御	
	X1発振モード時	外部クロック入力モード時
0	X1発振回路動作	EXCLK端子からの外部クロック有効
1	X1発振回路停止	EXCLK端子からの外部クロック無効

注意1. MSTOPに1を設定するとき、必ずCPUクロックが高速システム・クロック以外で動作していることを確認してください。具体的には、次のいずれかの条件です。

- a. 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-Lの場合
 - ・MCS = 0のとき (CPUクロックが高速内蔵発振クロックで動作)
- b. 78K0/KC2-Lの場合
 - ・MCS = 0, CLS = 0のとき (CPUクロックが高速内蔵発振クロックで動作)
 - ・CLS = 1のとき (CPUクロックがサブシステム・クロックで動作)

また、高速システム・クロックで動作している周辺ハードウェアを停止してから、MSTOPに1を設定してください。

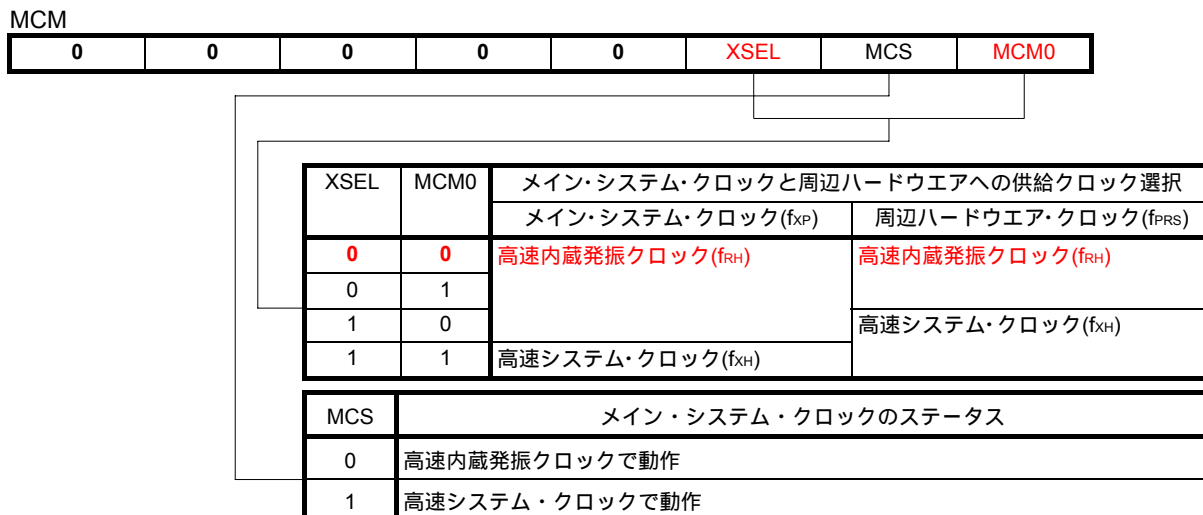
- 2. クロック動作モード選択レジスタ (OSCCTL) のビット6 (OSCSEL) が0のとき (入力ポート・モード), MSTOPに0を設定しないでください。
- 3. 周辺ハードウェア・クロックを停止すると、周辺ハードウェアは動作不可となります。周辺ハードウェア・クロック停止後に再開する場合は、周辺ハードウェアを初期化してください。
- 4. ビット6-0には必ず0を書き込んでください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

(6) メイン・システム・クロックと周辺ハードウェアへのクロック供給の選択

メイン・クロック・モード・レジスタ (MCM) でCPUクロックに供給するメイン・システム・クロックの選択と、周辺ハードウェア・クロックに供給するクロックの選択をします。

図4 - 10 メイン・クロック・モード・レジスタ (MCM) のフォーマット



注意1. ビット1はRead Onlyです。

2. XSELはリセット解除後、1回だけ設定が可能です。
3. CPUクロックがサブシステム・クロックで動作しているとき、MCM0を書き換えしないでください。
4. 次の周辺機能には、XSELとMCM0の設定によらず、f_{PRS}以外のクロックが供給されます。
 - ・ウォッチドッグ・タイマ (低速内蔵発振クロックで動作)
 - ・8ビット・タイマH1のカウント・クロックに「fil」、「fil/2⁶」または「fil/2¹⁵」を選択時 (低速内蔵発振クロックで動作)
 - ・クロック・ソースに外部クロックを選択している周辺ハードウェア (ただし、TM00の外部カウント・クロック選択時 (TI000端子の有効エッジ) は除く)

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

(7) リアルタイム・カウンタの制御クロックの制御

周辺イネーブル・レジスタ0 (PER0)^{注1}でリアルタイム・カウンタ^{注1}の制御クロックを制御します。

図4 - 11 周辺イネーブル・レジスタ0 (PER0) のフォーマット

PER0

RTCEN	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---

RTCEN	リアルタイム・カウンタ (RTC) の制御クロックの制御
0	制御クロック供給停止 ^{注2} ・リアルタイム・カウンタで使用するSFRへのライト不可 (リード可) ・リアルタイム・カウンタのカウンタ動作は継続可能
1	制御クロック供給 ・リアルタイム・カウンタで使用するSFRへのリード/ライト可

注1. 78K0/KC2-Lのみ。

- リアルタイム・カウンタの制御クロックの供給を停止することで、さらに低消費電力化を実現します。

注意1. RTCENを0にクリアすることにより、供給が停止される制御クロックは、CPUからリアルタイム・カウンタ (RTC) で使用するレジスタ (RTCC0レジスタなど) にライト・アクセスするとき使用するクロックです。RTCENを0にクリアしても、RTCの動作クロック (f_{SUB}) は停止しません。

- ビット6-0には必ず0を設定してください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります。

(1) から (7) の設定値をソフトウェアで記述すると、次のようになります。

・アセンブリ言語の場合

```
MOV    OSCCTL, #00000000B
MOV    PCC,    #00000001B
MOV    RCM,    #00000010B
MOV    MOC,    #10000000B
MOV    MCM,    #00000000B
MOV    PER0,   #00000000B
```

・C言語の場合

```
OSCCTL = 0b00000000;
PCC     = 0b00000001;
RCM     = 0b00000010;
MOC     = 0b10000000;
MCM     = 0b00000000;
PER0    = 0b00000000;
```

4.7 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。

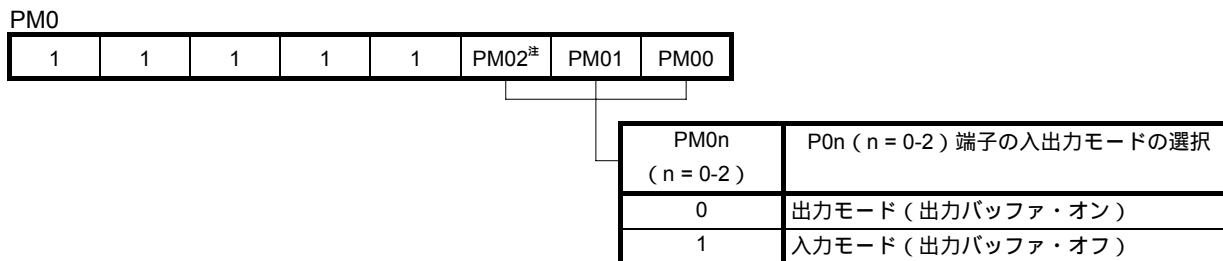
	78K0/KY2-L	78K0/KA2-L	78K0/KB2-L	78K0/KC2-L	
				44ピン製品	48ピン製品
ポート0	P00, P01	P00, P01	P00, P01	P00, P01	P00-P02
ポート1	-	-	P10-P17	P10-P17	P10-P17
ポート2	P20-P23	P20-P25	P20-P23	P20-P27	P20-P27
ポート3	P30	P30-P32	P30-P33	P30-P33	P30-P33
ポート4	-	-	-	P40, P41	P40-P42
ポート6	P60, P61	P60, P61	P60, P61	P60-P63	P60-P63
ポート7	-	-	-	P70-P73	P70-P75
ポート12	P121-P122, P125	P121-P122, P125	P120-P122, P125	P120-P125	P120-P125

(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、PM0を例にして、説明します。

図4 - 12 ポート・モード・レジスタ0 (PM0) のフォーマット



注 78K0/KC2-Lの48ピン製品のみ。

注意 次のビットには必ず1を設定してください

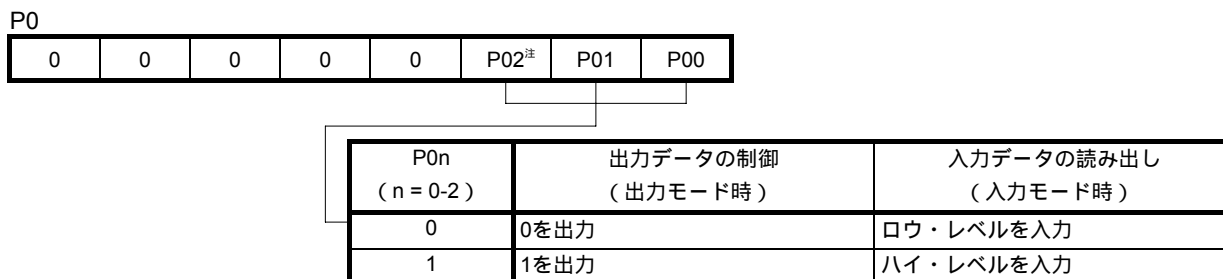
- ・ 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-L, 78K0/KC2-L (44ピン製品) の場合 : ビット7-2
- ・ 78K0/KC2-L (48ピン製品) の場合 : ビット7-3

(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、P0を例にして、説明します。

図4 - 13 ポート・レジスタ0 (P0) のフォーマット



注 78K0/KC2-Lの48ピン製品のみ。

注意 次のビットには必ず0を設定してください

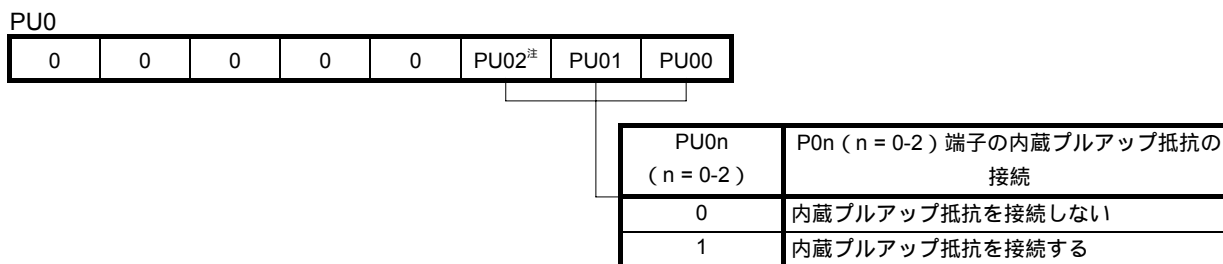
- ・ 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-L, 78K0/KC2-L (44ピン製品) の場合 : ビット7-2
- ・ 78K0/KC2-L (48ピン製品) の場合 : ビット7-3

(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どうかを設定します。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、PU0を例にして、説明します。

図4 - 14 プルアップ抵抗オプション・レジスタ0 (PU0) のフォーマット



注 78K0/KC2-Lの48ピン製品のみ。

注意 次のビットには必ず0を設定してください

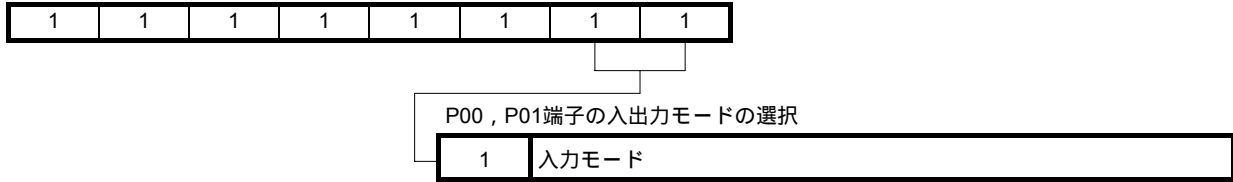
- ・ 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-L, 78K0/KC2-L (44ピン製品) の場合 : ビット7-2
- ・ 78K0/KC2-L (48ピン製品) の場合 : ビット7-3

【例 1】 スイッチ入力用に使用するため、P0を次のように設定する。

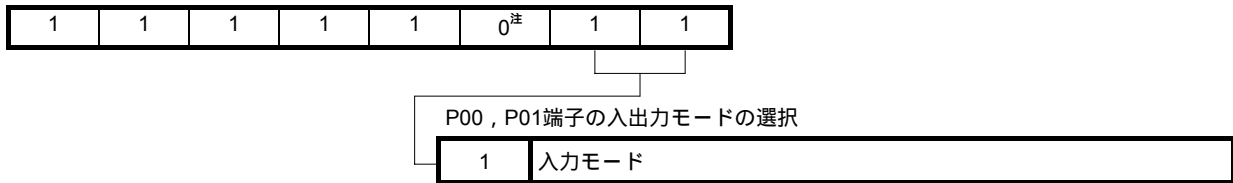
- ・ P00, P01を入力ポートに設定
- ・ P00, P01に内蔵プルアップ抵抗を接続する
(サンプル・プログラムの設定と同内容)

PM0

・ 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-L, 78K0/KC2-L (44ピン製品) の場合

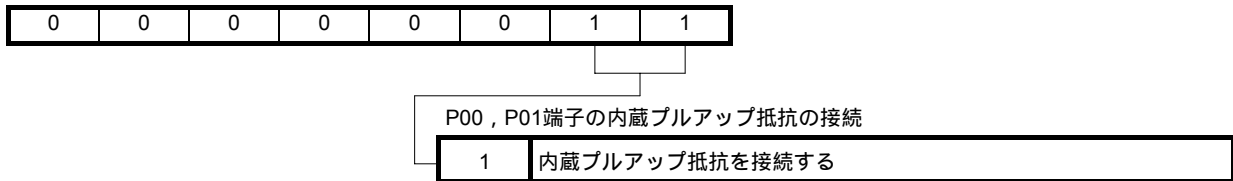


・ 78K0/KC2-Lの48ピン製品の場合



注 未使用端子は出力ポートとします。

PU0



なお、このサンプル・プログラムではスイッチ入力 (SW1, SW2) をロウ・アクティブで使用しますので、スイッチ入力用ポート (P00, P01) にはスイッチがオンのときにロウ・レベル (0)、スイッチがオフのときにハイ・レベル (1) が入力されます。

スイッチ入力 (SW1, SW2) とスイッチ入力用ポート (P00, P01) の関係は次のとおりです。

スイッチ入力		スイッチ入力用ポート	
SW1	SW2	P00	P01
オン	オン	0	0
オフ	オン	1	0
オン	オフ	0	1
オフ	オフ	1	1

ソフトウェアを記述すると、次のようになります。

・78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-L, 78K0/KC2-L (44ピン製品)の場合

[アセンブリ言語]

```
MOV    P0,      #00000000B
MOV    PM0,     #11111111B
MOV    PU0,     #00000011B
```

[C言語]

```
P0      = 0b00000000;
PM0     = 0b11111111;
PU0     = 0b00000011;
```

・78K0/KC2-L (48ピン製品)の場合

[アセンブリ言語]

```
MOV    P0,      #00000000B
MOV    PM0,     #11111011B
MOV    PU0,     #00000011B
```

[C言語]

```
P0      = 0b00000000;
PM0     = 0b11111011;
PU0     = 0b00000011;
```


【例 2】 LED出力用に使用するため，P3, P6を次のように設定する。

- P30, P60, P61を出力ポートに設定
- P30, P60, P61の出力ラッチをハイ・レベル出力に設定する
(サンプル・プログラムの設定と同内容)

PM3

• 78K0/KY2-Lの場合

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

P30端子の入出力モードの選択

0	出力モード
---	-------

• 78K0/KA2-Lの場合

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

P30端子の入出力モードの選択

0	出力モード
---	-------

• 78K0/KB2-L, 78K0/KC2-Lの場合

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

P30端子の入出力モードの選択

0	出力モード
---	-------

注 未使用端子は出力ポートとします。

PM6

• 78K0/KY2-L, 78K0/KA2-L, 78K0/KB2-Lの場合

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

P60, P61端子の入出力モードの選択

0	出力モード
---	-------

• 78K0/KC2-Lの場合

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

P60, P61端子の入出力モードの選択

0	出力モード
---	-------

注 未使用端子は出力ポートとします。

P3

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

P30端子の出力ラッチのレベル選択

1	ハイ・レベル出力
---	----------

P6

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

P60, P61端子の出力ラッチのレベル選択

1	ハイ・レベル出力
---	----------

なお,このサンプル・プログラムではLED点灯(LED1, LED2, LED3)をロウ・アクティブで使用しますので,LED出力用ポート(P30, P60, P61)の出力値が0のときに点灯,1のときに消灯となります。

LED出力用ポート(P30, P60, P61)とLED点灯(LED1, LED2, LED3)の関係は次のとおりです。

LED出力用ポート			LED点灯		
P30	P60	P61	LED1	LED2	LED3
0	1	1	点灯	消灯	消灯
1	0	1	消灯	点灯	消灯
1	1	0	消灯	消灯	点灯
1	1	1	消灯	消灯	消灯

ソフトウェアを記述すると、次のようになります。

・78K0/KY2-Lの場合

[アセンブリ言語]

```
MOV    P3,    #00000001B
MOV    PM3,   #11111110B
MOV    P6,    #00000011B
MOV    PM6,   #11111100B
```

[C言語]

```
P3     = 0b00000001;
PM3    = 0b11111110;
P6     = 0b00000011;
PM6    = 0b11111100;
```

・78K0/KA2-Lの場合

[アセンブリ言語]

```
MOV    P3,    #00000001B
MOV    PM3,   #11111000B
MOV    P6,    #00000011B
MOV    PM6,   #11111100B
```

[C言語]

```
P3     = 0b00000001;
PM3    = 0b11111000;
P6     = 0b00000011;
PM6    = 0b11111100;
```

・78K0/KB2-Lの場合

[アセンブリ言語]

```
MOV    P3,    #00000001B
MOV    PM3,   #11110000B
MOV    P6,    #00000011B
MOV    PM6,   #11111100B
```

[C言語]

```
P3     = 0b00000001;
PM3    = 0b11110000;
P6     = 0b00000011;
PM6    = 0b11111100;
```

・78K0/KC2-Lの場合

[アセンブリ言語]

```
MOV    P3,    #00000001B
MOV    PM3,   #11110000B
MOV    P6,    #00000011B
MOV    PM6,   #11110000B
```

[C言語]

```
P3     = 0b00000001;
PM3    = 0b11110000;
P6     = 0b00000011;
PM6    = 0b11110000;
```

4.8 メイン処理

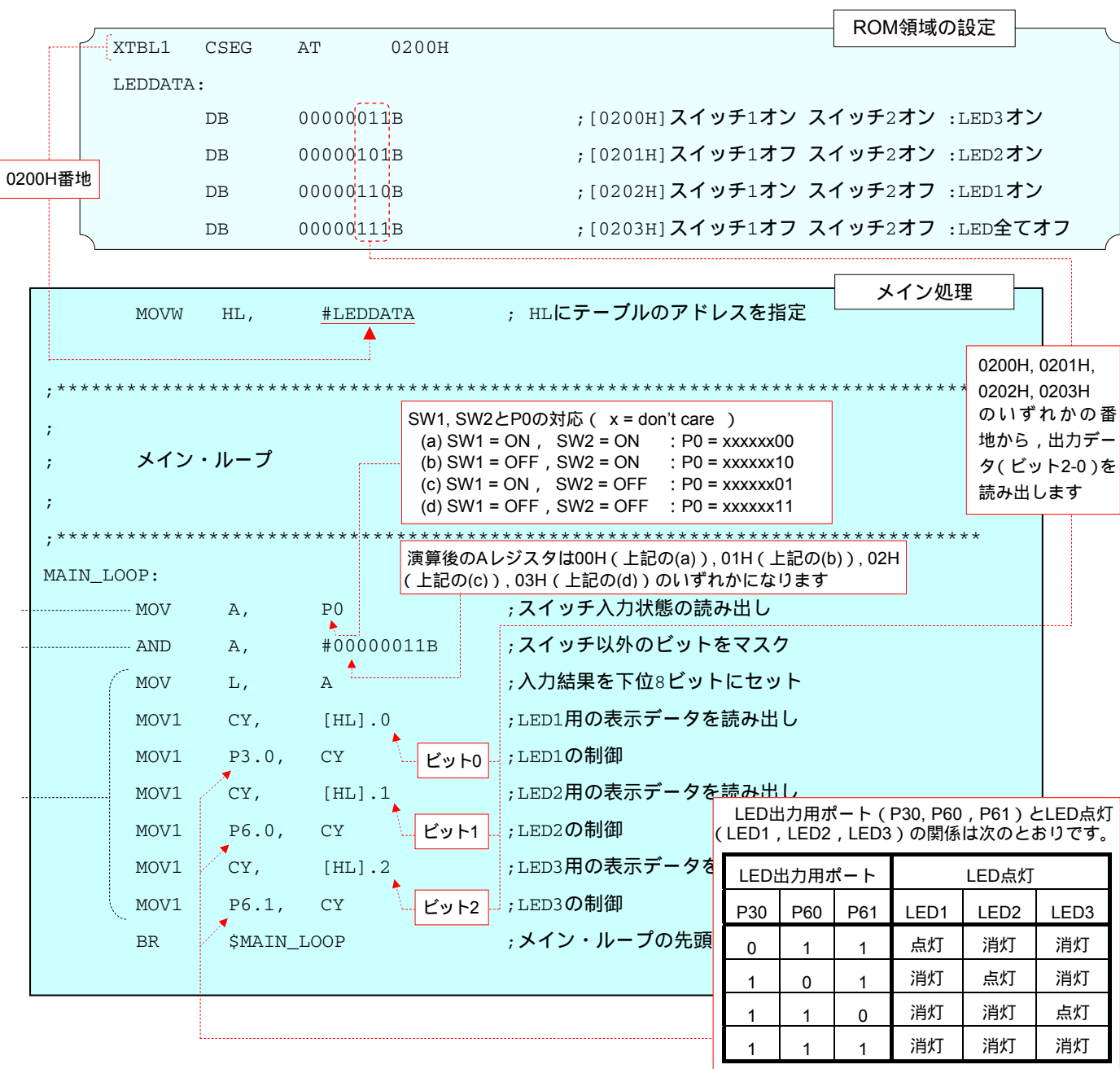
アセンブリ言語のメイン処理では、次の動作を行います。

P0のデータを読み出します

読み出した8ビット・データのうち、スイッチ入力用ポート（P00, P01）以外のビットを0にします

0200H-0203H番地（「LEDDATA」テーブル）から、P00, P01の入力レベルの組み合わせに対応する出力データを読み出し、1ビットずつP30, P60, P61に出力します

この動作により、P00, P01に接続しているスイッチ（SW1, SW2）の入力の組み合わせのみを判断することが可能になります。なお、このサンプル・プログラムでは、スイッチをロウ・アクティブで使用しますので、P00, P01にはスイッチがオンのときにロウ・レベル（0）、スイッチがオフのときにハイ・レベル（1）が入力されます。



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 aLedOut[4]
        = {0b00000011,0b00000101,0b00000110,0b00000111}; /* LED出力用のテーブル */
    unsigned char ucSwitchBuffer; /* スイッチ入力データ保存領域 */

    while(1){
        /* 有効スイッチ情報取得 */
        ucSwitchBuffer = ( P0 & 0b00000011 );

        /* テーブルから表示データを読み出して表示 */
        P3 = ( aLedOut[ucSwitchBuffer] & 0b00000001 ); /* LED1出力 */
        P6 = ( ( aLedOut[ucSwitchBuffer] >> 1 ) & 0b00000011 ); /* LED2,LED3出力 */
    }
}

```

{ }内に4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。

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	P00, P01	ucSwitchBuffer	aLedOut	LED点灯
SW1 = ON, SW2 = ON	P00 = 0, P01 = 0	0b00000000	0b00000011	LED3のみ点灯
SW1 = OFF, SW2 = ON	P00 = 1, P01 = 0	0b00000001	0b00000101	LED2のみ点灯
SW1 = ON, SW2 = OFF	P00 = 0, P01 = 1	0b00000010	0b00000110	LED1のみ点灯
SW1 = OFF, SW2 = OFF	P00 = 1, P01 = 1	0b00000011	0b00000111	全LED消灯

第5章 関連資料

関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

資料名		和文 / 英文
78K0/Kx2-L 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

付録A プログラム・リスト

プログラム・リスト例として、78K0/KC2-L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      78K0/KC2-Lシリーズ
;
;*****
; 78K0/KC2-Lシリーズ      サンプル・プログラム (初期設定)
;*****
; LED点灯のスイッチ制御編
;*****
; 【履歴】
; 2009.1.--      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、入出力ポートの設定などを行う
;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。
;
; <初期設定の主な内容>
; (オプション・バイトでの設定)
; ・低速内蔵発振器の動作をソフトウェアにより停止可能に設定
; ・ウォッチドッグ・タイマの動作禁止
; ・高速内蔵発振クロック周波数を8MHzに設定
; ・LVIデフォルト・スタート機能停止
; (リセット解除後の初期化処理での設定)
; ・ROM/RAMサイズの設定
; ・CPUクロックを高速内蔵発振クロック動作に設定(4MHz)
; ・低速内蔵発振器の停止
; ・入出力ポートの設定
; ・使用しない周辺ハードウェアの動作禁止
;
;
; <スイッチ入力とLED点灯>
;
; +
```

```

;   | スイッチ1 | スイッチ2 | LED1 | LED2 | LED3 |
;   | (P00)   | (P01)   | (P30) | (P60) | (P61) |
;   |         |         |         |         |         |
;   | OFF    | OFF    | OFF   | OFF   | OFF   |
;   | ON     | OFF    | ON    | OFF   | OFF   |
;   | OFF    | ON     | OFF   | ON    | OFF   |
;   | ON     | ON     | OFF   | OFF   | ON    |
;   +                                     +

```

スイッチがONのときポートの入力値は0, OFFのとき1となります。

LEDはポートの出力値が1のときOFF(消灯), 0のときON(点灯)となります。

< 入出力ポートの設定 >

入力ポート : P00-P01

出力ポート : P30, P60-P61

未使用のポートで出力に設定できるもの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

=====

```

XVECT1          CSEG  AT    0000H
  DW  RESET_START          ;0000H  RESET入力, POC, LVI, WDT
XVECT2          CSEG  AT    0004H
  DW  IINIT                ;0004H  INTLVI
  DW  IINIT                ;0006H  INTP0
  DW  IINIT                ;0008H  INTP1
  DW  IINIT                ;000AH  INTP2
  DW  IINIT                ;000CH  INTP3
  DW  IINIT                ;000EH  INTP4
  DW  IINIT                ;0010H  INTP5
  DW  IINIT                ;0012H  INTSRE6
  DW  IINIT                ;0014H  INTSR6
  DW  IINIT                ;0016H  INTST6
  DW  IINIT                ;0018H  INTCSI10
  DW  IINIT                ;001AH  INTTMH1
  DW  IINIT                ;001CH  INTTMH0
  DW  IINIT                ;001EH  INTTM50
  DW  IINIT                ;0020H  INTTM000

```



```

DW      IINIT          ;0022H INTTM010
DW      IINIT          ;0024H INTAD
DW      IINIT          ;0026H INTP6
DW      IINIT          ;0028H INTRTCI
DW      IINIT          ;002AH INTTM51
DW      IINIT          ;002CH INTKR
DW      IINIT          ;002EH INTRTC
DW      IINIT          ;0030H INTP7
DW      IINIT          ;0032H INTP8
DW      IINIT          ;0034H INTIICA0
DW      IINIT          ;0036H INTCSI11
DW      IINIT          ;0038H INTP9
DW      IINIT          ;003AH INTP10
DW      IINIT          ;003CH INTP11
DW      IINIT          ;003EH BRK

```

=====

```

;
;
;   ROMの定義
;

```

=====

```

XTBL1 CSEG   AT      0200H

```

```

LEDDATA:

```

```

DB      00000011B      ;[0200H]スイッチ1オン スイッチ2オン :LED3オン
DB      00000101B      ;[0201H]スイッチ1オフ スイッチ2オン :LED2オン
DB      00000110B      ;[0202H]スイッチ1オン スイッチ2オフ :LED1オン
DB      00000111B      ;[0203H]スイッチ1オフ スイッチ2オフ :LED全てオフ

```

=====

```

;
;
;   スタック領域の確保
;

```

=====

```

DSTK DSEG   IHRAM

```

```

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

```

```

;
;
;   不要な割り込み要因による割り込み処理
;

```

```

;*****
;
XMAIN CSEG    UNIT
IINIT:
;    不要な割り込みが発生した場合、ここに分岐します。
;    ここでは何も処理をしないで元の処理に戻ります

    RETI

;*****
;
;    リセット解除後の初期化処理
;
;*****
RESET_START:

;-----
;    割り込み禁止
;-----
    DI                                ;割り込み禁止

;-----
;    レジスタ・バンクの設定
;-----
    SEL    RBO                        ;レジスタ・バンク設定

;-----
;    ROM/RAMサイズの設定
;-----
;    モデルにより設定値が異なるので注意してください。
;    使用モデルの設定を有効にしてください。（デフォルトではuPD78F0588）
;-----
;uPD78F0581,uPD78F0586使用時の設定
;MOV    IMS,    #042H                ;ROM/RAMサイズの設定

;uPD78F0582,uPD78F0587使用時の設定
;MOV    IMS,    #004H                ;ROM/RAMサイズの設定

;uPD78F0583,uPD78F0588使用時の設定
MOV    IMS,    #0C8H                ;ROM/RAMサイズの設定

;-----
;    スタック・ポインタの設定
;-----

```

MOVW SP, #STACKTOP ;スタック・ポインタを設定

; クロック周波数の設定

; 高速内蔵発振クロックで動作が行えるように設定します。

MOV OSCCTL, #0000000B ;クロック動作モード

; |||+||+----- 必ず0に設定
; ||| ++----- RSWOSC/AMPHXT
; ||| [XT1発振回路の発振モード選択]
; ||| 00: 低消費発振
; ||| 01: 通常発振
; ||| 1x: 超低消費発振
; ||+----- EXCLKS/OSCELS
; || [サブシステム・クロック端子の動作設定]
; || (P123/XT1,P124/XT2/EXCLKS)
; || XTSTARTと合わせて000で入出力ポートに設定
; ++----- EXCLK/OSCESEL
; [高速システム・クロック端子の動作設定]
; (P121/X1,P122/X2/EXCLK)
; 00: 入力ポート
; 01: X1発振モード
; 10: 入力ポート
; 11: 外部クロック入力モード

MOV PCC, #0000001B ;CPUクロック (fCPU)の選択

; |||+|+----- CSS/PCC2/PCC1/PCC0
; ||| | [CPUクロック (fCPU)の選択]
; ||| | 0000: fXP
; ||| | 0001: fXP/2
; ||| | 0010: fXP/2²
; ||| | 0011: fXP/2³
; ||| | 0100: fXP/2⁴
; ||| | 1000: fSUB/2
; ||| | 1001: fSUB/2
; ||| | 1010: fSUB/2
; ||| | 1011: fSUB/2
; ||| | 1100: fSUB/2
; ||| | (上記以外:設定禁止)
; ||| +----- 必ず0に設定
; ||+----- CLS
; || [CPUクロックのステータス]

```

;          |+----- XTSTART
;          |          [サブシステム・クロック端子の動作設定]
;          |          EXCLKS, OSCSELSと組み合わせて設定
;          +----- 必ず0に設定

MOV      RCM, #0000010B ;内蔵発振器の動作モード選択
;
;          |||||+----- RSTOP
;          |||||          [高速内蔵発振器の発振 / 停止]
;          |||||          0:高速内蔵発振器の発振
;          |||||          1:高速内蔵発振器の停止
;
;          |||||+----- LSRSTOP
;          |||||          [低速内蔵発振器の発振 / 停止]
;          |||||          0:低速内蔵発振器の発振
;          |||||          1:低速内蔵発振器の停止
;
;          |++++----- 必ず0に設定
;          +----- RSTS
;          [高速内蔵発振器のステータス]

MOV      MOC, #1000000B ;高速システム・クロックの動作モード選択
;
;          |++++----- 必ず0に設定
;          +----- MSTOP
;          [高速システム・クロックの動作制御]
;          0:X1発振回路動作/EXCLK端子からの
;          外部クロック有効
;          1:X1発振回路停止/EXCLK端子からの
;          外部クロック無効

MOV      MCM, #0000000B ;供給クロック選択
;
;          ||||+|+----- XSEL/MCM0:
;          |||| |          [メイン・システム, 周辺ハードウェアへの
;          |||| |          供給クロック]
;          |||| |          00: メイン・システム・クロック (fXP)
;          |||| |          = 高速内蔵発振クロック (fIH)
;          |||| |          周辺ハードウェア・クロック (fPRS)
;          |||| |          = 高速内蔵発振クロック (fIH)
;          |||| |          01: メイン・システム・クロック (fXP)
;          |||| |          = 高速内蔵発振クロック (fIH)
;          |||| |          周辺ハードウェア・クロック (fPRS)
;          |||| |          = 高速内蔵発振クロック (fIH)
;          |||| |          10: メイン・システム・クロック (fXP)
;          |||| |          = 高速内蔵発振クロック (fIH)
;          |||| |          周辺ハードウェア・クロック (fPRS)
;          |||| |          = 高速システム・クロック (fIH)

```

```

;          |||| |          11: メイン・システム・クロック (fXP)
;          |||| |          = 高速システム・クロック (fIH)
;          |||| |          周辺ハードウェア・クロック (fPRS)
;          |||| |          = 高速システム・クロック (fIH)
;          |||| +----- MCS
;          ||||           [メイン・システム・クロックのステータス]
;          +++++----- 必ず0に設定

MOV   PER0, #0000000B ;リアルタイム・カウンタの制御クロックの制御
;          |+++++----- 必ず0に設定
;          +----- RTCCN:
;          [リアルタイム・カウンタの制御クロック]
;          0: 制御クロック供給停止
;          1: 制御クロック供給

-----
;   ポート0の設定
-----
MOV   P0,   #0000000B ;P00-P02の出力ラッチLow
MOV   PM0,  #11111011B ;P00-P01を入力ポートに設定
;          ;P02を出力ポートに設定
MOV   PU0,  #00000011B ;P00-P01の内蔵プルアップ抵抗を接続する
;          ;P02の内蔵プルアップ抵抗を接続しない
;          ;P00:スイッチ1入力用に使用する
;          ;P01:スイッチ2入力用に使用する
;          ;P02:未使用

-----
;   ポート1の設定
-----
MOV   ADPC1, #00000111B ;P10-P12をデジタル入出力に設定
MOV   P1,   #00000000B ;P10-P17の出力ラッチLow
MOV   PM1,  #00000000B ;P10-P17を出力ポートに設定
;          ;P10-P17:未使用

-----
;   ポート2の設定
-----
MOV   ADPC0, #11111111B ;P20-P27をデジタル入出力に設定
MOV   P2,   #00000000B ;P20-P27の出力ラッチLow
MOV   PM2,  #00000000B ;P20-P27を出力ポートに設定
;          ;P20-P27:未使用

```

```

;-----
;   ポート3の設定
;-----
MOV   P3,    #0000001B    ;P30の出力ラッチHigh
                                ;P31-P33の出力ラッチLow
MOV   PM3,   #11110000B   ;P30-P33を出力ポートに設定
                                ;P30:LED1出力用に使用する
                                ;P31-P33:未使用
;-----
;   ポート4の設定
;-----
MOV   P4,    #00000000B   ;P40-P42の出力ラッチLow
MOV   PM4,   #11111000B   ;P40-P42を出力ポートに設定
                                ;P40-P42:未使用
;-----
;   ポート6の設定
;-----
MOV   P6,    #00000011B   ;P60-P61の出力ラッチHigh
                                ;P62-P63の出力ラッチLow
MOV   PM6,   #11110000B   ;P60-P63を出力ポートに設定
                                ;P60:LED2出力用に使用する
                                ;P61:LED3出力用に使用する
                                ;P62-P63:未使用
;-----
;   ポート7の設定
;-----
MOV   P7,    #00000000B   ;P70-P75の出力ラッチLow
MOV   PM7,   #11000000B   ;P70-P75を出力ポートに設定
                                ;P70-P75:未使用
;-----
;   ポート12の設定
;-----
MOV   P12,   #00000000B   ;P120の出力ラッチLow
MOV   PM12,  #11111110B   ;P120を出力ポートに設定
                                ;P120-P125:未使用
;-----
;   使用しない周辺ハードウェアの動作禁止
;-----

```

```
;16ビット・タイマ/イベント・カウンタ00
MOV    TMC00, #00000000B    ;動作禁止

;8ビット・タイマ/イベント・カウンタ50, 51
MOV    TMC50, #00000000B    ;タイマ50 動作禁止
MOV    TMC51, #00000000B    ;タイマ51 動作禁止

;8ビット・タイマH0, H1
MOV    TMHMD0, #00000000B   ;タイマH0 カウント動作停止
MOV    TMHMD1, #00000000B   ;タイマH1 カウント動作停止

;リアルタイム・カウンタ
MOV    RTCC0, #00000000B    ;カウンタ動作停止

;クロック出力制御回路
MOV    CKS, #00000000B     ;クロック分周回路動作停止

;A/Dコンバータ
MOV    ADMO, #00000000B    ;A/D変換動作停止

;オペアンプ
MOV    AMP0M, #00000000B   ;オペアンプ0 動作停止
MOV    AMP1M, #00000000B   ;オペアンプ1 動作停止

;シリアル・インタフェースUART6
MOV    ASIM6, #00000001B   ;動作禁止

;シリアル・インタフェースIICA
MOV    IICACTL0, #00000000B ;動作禁止

;シリアル・インタフェースCSI10, CSI11
MOV    CSIM10, #00000000B   ;CSI10 動作禁止
MOV    CSIM11, #00000000B   ;CSI11 動作禁止

;低電圧検出回路
MOV    LVIM, #00000000B    ;動作禁止

;割り込み機能
MOVW   MK0, #0FFFFH        ;全割り込み禁止
MOVW   MK1, #0FFFFH        ;
MOV    EGPCTL0, #00000000B  ;全外部割り込みのエッジ検出禁止
MOV    EGPCTL1, #00000000B  ;
```

```

;キー割り込み機能
MOV    KRM,    #00000000B    ;全キー割り込み禁止

;-----
;    レジスタ値の初期化
;-----
MOVW   HL,     #LEDDATA     ;LED出力用のテーブル・アドレスを指定

;-----
;    割り込み許可
;    (割り込みを使用する場合はこのタイミングで許可します。)
;-----
;    EI                      ;割り込みを許可する場合は
;                             ;コメントアウトを外します。

BR     MAIN_LOOP            ;メイン・ループへ

;*****
;
;    メイン・ループ
;
;*****
MAIN_LOOP:
MOV    A,      P0           ;スイッチ入力状態の読み出し
AND    A,      #00000011B   ;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
MOV1   CY,     [HL].0      ;LED1用の表示データを読み出し
MOV1   P3.0,   CY          ;LED1の制御
MOV1   CY,     [HL].1      ;LED2用の表示データを読み出し
MOV1   P6.0,   CY          ;LED2の制御
MOV1   CY,     [HL].2      ;LED3用の表示データを読み出し
MOV1   P6.1,   CY          ;LED3の制御
BR     $MAIN_LOOP         ;メイン・ループの先頭へ

end

```


main.c (C言語版)

/*****

NEC Electronics 78K0/KC2-Lシリーズ

78K0/KC2-Lシリーズ サンプル・プログラム (初期設定)

LED点灯のスイッチ制御編

【履歴】

2009.1.-- 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、入出力ポート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。

<初期設定の主な内容>

(オプション・バイトでの設定)

- ・低速内蔵発振器の動作をソフトウェアにより停止可能に設定
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振クロック周波数を8MHzに設定
- ・LVIデフォルト・スタート機能停止

(リセット解除後の初期化处理での設定)

- ・ROM/RAMサイズの設定
- ・CPUクロックを高速内蔵発振クロック動作に設定(4MHz)
- ・低速内蔵発振器の停止
- ・入出力ポートの設定
- ・使用しない周辺ハードウェアの動作禁止

<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	
	(P00)		(P01)		(P30)		(P60)		(P61)	
	OFF		OFF		OFF		OFF		OFF	
	ON		OFF		ON		OFF		OFF	
	OFF		ON		OFF		ON		OFF	
	ON		ON		OFF		OFF		ON	

+

+

スイッチがONのときポートの入力値は0, OFFのとき1となります。
 LEDはポートの出力値が1のときOFF(消灯), 0のときON(点灯)となります。

< 入出力ポートの設定 >

入力ポート : P00-P01

出力ポート : P30, P60-P61

未使用のポートで出力に設定できるもの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 DI           /* DI命令を記述可能にする */
#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 */
```

リセット解除後の初期化処理

*****/

```
void hdwinit( void )
```

```
{
```

/*-----

割り込み禁止

-----*/

```
DI();              /* 割り込み禁止 */
```

/*-----

ROM/RAMサイズの設定

モデルにより設定値が異なるので注意してください。

使用モデルの設定を有効にしてください。(デフォルトではuPD78F0588)

-----*/

```
/* uPD78F0581, uPD78F0586使用時の設定 */
```

```
/* IMS = 0x42; */      /* ROM/RAMサイズの設定 */
```

```

/* uPD78F0582,uPD78F0587使用時の設定 */
/* IMS = 0x04;*/          /* ROM/RAMサイズの設定 */

/* uPD78F0583,uPD78F0588使用時の設定 */
IMS = 0xC8;          /* ROM/RAMサイズの設定 */

/*-----
クロック周波数の設定
-----
高速内蔵発振クロックで動作が行えるように設定します。
-----*/

OSCCCTL = 0b00000000; /* クロック動作モード */
/*      |||+||+---- 必ず0に設定 */
/*      ||| +----- RSWOSC/AMPHXT */
/*      |||          [XT1発振回路の発振モード選択] */
/*      |||          00: 低消費発振 */
/*      |||          01: 通常発振 */
/*      |||          1x: 超低消費発振 */
/*      ||+----- EXCLKS/OSCSELS */
/*      ||          [サブシステム・クロック端子の動作設定] */
/*      ||          (P123/XT1,P124/XT2/EXCLKS) */
/*      ||          XTSTARTと合わせて000で入出力ポートに設定 */
/*      ++----- EXCLK/OSCSEL */
/*          [高速システム・クロック端子の動作設定] */
/*          (P121/X1,P122/X2/EXCLK) */
/*          00: 入力ポート */
/*          01: X1発振モード */
/*          10: 入力ポート */
/*          11: 外部クロック入力モード */

PCC = 0b00000001; /* CPUクロック(fCPU)の選択 */
/*      |||+|+----- CSS/PCC2/PCC1/PCC0 */
/*      ||| |          [CPUクロック(fCPU)の選択] */
/*      ||| |          0000: fXP */
/*      ||| |          0001: fXP/2 */
/*      ||| |          0010: fXP/2^2 */
/*      ||| |          0011: fXP/2^3 */
/*      ||| |          0100: fXP/2^4 */
/*      ||| |          1000: fSUB/2 */
/*      ||| |          1001: fSUB/2 */
/*      ||| |          1010: fSUB/2 */
/*      ||| |          1011: fSUB/2 */

```

```

/*      ||| |      1100:fSUB/2 */
/*      ||| |      (上記以外:設定禁止) */
/*      ||| +----- 必ず0に設定 */
/*      ||+----- CLS */
/*      ||      [CPUクロックのステータス] */
/*      |+----- XTSTART */
/*      |      [サブシステム・クロック端子の動作設定] */
/*      |      EXCLKS,OSCSLSと組み合わせて設定 */
/*      +----- 必ず0に設定 */

RCM = 0b00000010; /* 内蔵発振器の動作モード選択 */
/*      |||||+---- RSTOP */
/*      |||||      [高速内蔵発振器の発振/停止] */
/*      |||||      0:高速内蔵発振器の発振 */
/*      |||||      1:高速内蔵発振器の停止 */
/*      |||||+----- LSRSTOP */
/*      |||||      [低速内蔵発振器の発振/停止] */
/*      |||||      0:低速内蔵発振器の発振 */
/*      |||||      1:低速内蔵発振器の停止 */
/*      |+++++----- 必ず0に設定 */
/*      +----- RSTS */
/*      [高速内蔵発振器のステータス]

MOC = 0b10000000; /* 高速システム・クロックの動作モード選択 */
/*      |+++++----- 必ず0に設定 */
/*      +----- MSTOP */
/*      [高速システム・クロックの動作制御] */
/*      0:X1発振回路動作/EXCLK端子からの外部クロック有効 */
/*      1:X1発振回路停止/EXCLK端子からの外部クロック無効

MCM = 0b00000000; /* 供給クロック選択 */
/*      |||||+|+---- XSEL/MCM0 */
/*      ||||| |      [メイン・システム, 周辺ハードウェアへの供給クロック] */
/*      ||||| |      00: メイン・システム・クロック(fXP) */
/*      ||||| |      = 高速内蔵発振クロック(fIH) */
/*      ||||| |      周辺ハードウェア・クロック(fPRS) */
/*      ||||| |      = 高速内蔵発振クロック(fIH) */
/*      ||||| |      01: メイン・システム・クロック(fXP) */
/*      ||||| |      = 高速内蔵発振クロック(fIH) */
/*      ||||| |      周辺ハードウェア・クロック(fPRS) */
/*      ||||| |      = 高速内蔵発振クロック(fIH) */
/*      ||||| |      10: メイン・システム・クロック(fXP) */
/*      ||||| |      = 高速内蔵発振クロック(fIH)

```

```

/*      ||||| |          周辺ハードウェア・クロック (fPRS) */
/*      ||||| |          = 高速システム・クロック (fIH) */
/*      ||||| |          11: メイン・システム・クロック (fXP) */
/*      ||||| |          = 高速システム・クロック (fIH) */
/*      ||||| |          周辺ハードウェア・クロック (fPRS) */
/*      ||||| |          = 高速システム・クロック (fIH) */
/*      ||||| +----- MCS */
/*      |||||          [メイン・システム・クロックのステータス] */
/*      +----- 必ず0に設定 */

PERO   = 0b00000000; /* リアルタイム・カウンタの制御クロックの制御 */
/*      |+++++----- 必ず0に設定 */
/*      +----- RTCEN: */
/*      [リアルタイム・カウンタの制御クロック] */
/*      0: 制御クロック供給停止 */
/*      1: 制御クロック供給 */

```

```

/*-----
ポート0の設定
-----*/

```

```

P0     = 0b00000000; /* P00-P02の出力ラッチLow */
PM0    = 0b11111011; /* P00-P01を入力ポートに設定 */
/*      P02を出力ポートに設定 */
PU0    = 0b00000011; /* P00-P01の内蔵プルアップ抵抗を接続する */
/*      P02の内蔵プルアップ抵抗を接続しない */
/*      P00: スイッチ1入力用に使用する */
/*      P01: スイッチ2入力用に使用する */
/*      P02: 未使用 */

```

```

/*-----
ポート1の設定
-----*/

```

```

ADPC1  = 0b00000111; /* P10-P12をデジタル入出力に設定 */
P1     = 0b00000000; /* P10-P17の出力ラッチLow */
PM1    = 0b00000000; /* P10-P17を出力ポートに設定 */
/*      P10-P17: 未使用 */

```

```

/*-----
ポート2の設定
-----*/

```

```

ADPC0  = 0b11111111; /* P20-P27をデジタル入出力に設定 */
P2     = 0b00000000; /* P20-P27の出力ラッチLow */
PM2    = 0b00000000; /* P20-P27を出力ポートに設定 */

```

/* P20-P27:未使用 */

/*-----*/

ポート3の設定

-----*/

```
P3      = 0b00000001; /* P30の出力ラッチHigh */
                          /* P31-P33の出力ラッチLow */
PM3     = 0b11110000; /* P30-P33を出力ポートに設定 */
                          /* P30:LED1出力用に使用する */
                          /* P31-P33:未使用 */
```

/*-----*/

ポート4の設定

-----*/

```
P4      = 0b00000000; /* P40-P42の出力ラッチLow */
PM4     = 0b11111000; /* P40-P42を出力ポートに設定 */
                          /* P40-P42:未使用 */
```

/*-----*/

ポート6の設定

-----*/

```
P6      = 0b00000011; /* P60-P61の出力ラッチHigh */
                          /* P62-P63の出力ラッチLow */
PM6     = 0b11110000; /* P60-P63を出力ポートに設定 */
                          /* P60:LED2出力用に使用する */
                          /* P61:LED3出力用に使用する */
                          /* P62-P63:未使用 */
```

/*-----*/

ポート7の設定

-----*/

```
P7      = 0b00000000; /* P70-P75の出力ラッチLow */
PM7     = 0b11000000; /* P70-P75を出力ポートに設定 */
                          /* P70-P75:未使用 */
```

/*-----*/

ポート12の設定

-----*/

```
P12     = 0b00000000; /* P120の出力ラッチLow */
PM12    = 0b11111110; /* P120を出力ポートに設定 */
                          /* P120-P125:未使用 */
```

/*-----*/

使用しない周辺ハードウェアの動作禁止

-----*/

/* 16ビット・タイマ/イベント・カウンタ00 */

TMC00 = 0b00000000; /* 動作禁止 */

/* 8ビット・タイマ/イベント・カウンタ50, 51 */

TMC50 = 0b00000000; /* タイマ50 動作禁止 */

TMC51 = 0b00000000; /* タイマ51 動作禁止 */

/* 8ビット・タイマH0, H1 */

TMHMD0 = 0b00000000; /* タイマH0 カウント動作停止 */

TMHMD1 = 0b00000000; /* タイマH1 カウント動作停止 */

/* リアルタイム・カウンタ */

RTCC0 = 0b00000000; /* カウンタ動作停止 */

/* クロック出力制御回路 */

CKS = 0b00000000; /* クロック分周回路動作停止 */

/* A/Dコンバータ */

ADMO = 0b00000000; /* A/D変換動作停止 */

/* オペアンプ */

AMP0M = 0b00000000; /* オペアンプ0 動作停止 */

AMP1M = 0b00000000; /* オペアンプ1 動作停止 */

/* シリアル・インタフェースUART6 */

ASIM6 = 0b00000001; /* 動作禁止 */

/* シリアル・インタフェースIICA */

IICACTL0 = 0b00000000; /* 動作禁止 */

/* シリアル・インタフェースCSI10, CSI11 */

CSIM10 = 0b00000000; /* CSI10 動作禁止 */

CSIM11 = 0b00000000; /* CSI11 動作禁止 */

/* 低電圧検出回路 */

LVIM = 0b00000000; /* 動作禁止 */

/* 割り込み機能 */

MK0 = 0xFFFF; /* 全割り込み禁止 */

MK1 = 0xFFFF;

EGPCTL0 = 0b00000000; /* 全外部割り込みのエッジ検出禁止 */

```
EGPCTL1 = 0b00000000;

/* キー割り込み機能 */
KRM      = 0b00000000; /* 全キー割り込み禁止 */

/*-----
割り込み許可
(割り込みを使用する場合はこのタイミングで許可します。)
-----*/
/* EI(); /* 割り込みを許可する場合は /*
/* コメントアウトを外します。 */

}

/*****

メイン・ループ

*****/
void main(void)
{
    const unsigned char aLedOut[4]
    = {0b00000011,0b00000101,0b00000110,0b00000111}; /* LED出力用のテーブル */
    unsigned char ucSwitchBuffer; /* スイッチ入力データ保存領域 */

    while(1){
        /* 有効スイッチ情報取得 */
        ucSwitchBuffer = ( P0 & 0b00000011 );

        /* テーブルから表示データを読み出して表示 */
        P3 = ( aLedOut[ucSwitchBuffer] & 0b00000001 ); /* LED1の制御 */
        P6 = ( ( aLedOut[ucSwitchBuffer] >> 1 ) & 0b00000011 ); /* LED2,LED3の制御 */
    }
}
```


付録B 78K0/KC2-Lの44ピン製品を使用する場合

78K0/KC2-Lのサンプル・プログラムは、すべて48ピン製品用となっています。78K0/KC2-Lのサンプル・プログラムを44ピン製品用に使用する場合、次のように変更してください。

(1) ポートの初期設定

- ・ポート0の設定

ポート・モード・レジスタ0 (PM0) のビット2への設定値を“0”から“1”に変更してください。

- ・ポート4の設定

ポート・モード・レジスタ4 (PM4) のビット2への設定値を“0”から“1”に変更してください。

- ・ポート7の設定

ポート・モード・レジスタ7 (PM7) のビット5, 4への設定値を“00”から“11”に変更してください。

(2) 使用しない周辺ハードウェアの動作禁止

クロック出力選択レジスタ (CKS) の設定を行っている命令文を削除してください。

付録C 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