

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

## 78K0R/Kx3

### 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使用する周辺の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使用する周辺の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

78K0R/KE3マイクロコントローラ  
 78K0R/KF3マイクロコントローラ  
 78K0R/KG3マイクロコントローラ  
 78K0R/KH3マイクロコントローラ  
 78K0R/KJ3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 使用する周辺の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 ...	13
4.3 スタック・ポインタの設定 ...	14
4.4 クロックの設定 ...	15
4.5 ポートの設定 ...	22
4.6 メイン処理 ...	26
第5章 関連資料 ...	28
付録A プログラム・リスト ...	29
付録B 改版履歴 ...	42

- 本資料に記載されている内容は2008年12月現在のもので、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、一切その責を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。また、当社製品は耐放射線設計については行っておりません。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

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

特別水準：輸送機器（自動車、電車、船舶等）、交通信号機器、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

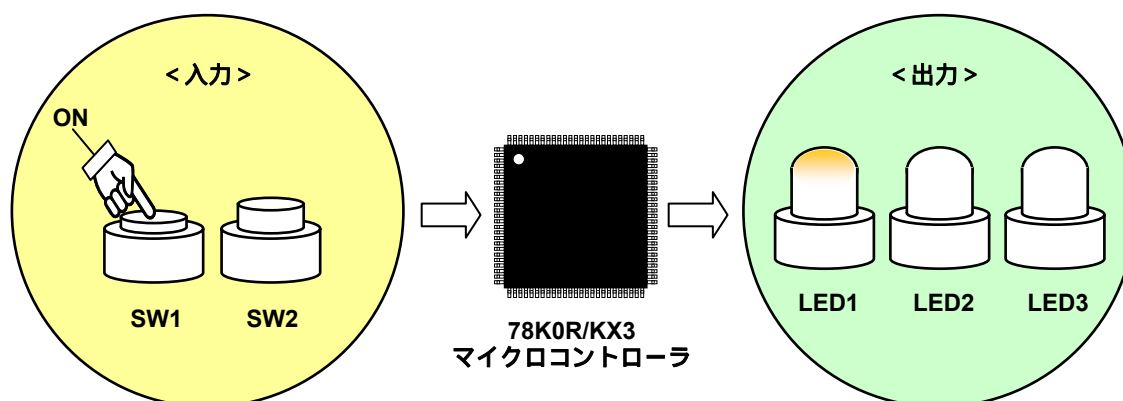
このサンプル・プログラムでは、クロック周波数の選択、ポート入出力の設定など、78K0R/Kx3マイクロコントローラの基本的な使用する周辺の初期設定を行います。また、使用する周辺の初期設定完了後のメイン処理動作では、2つのスイッチ入力により、3つのLED点灯を制御します。

## (1) 使用する周辺の初期設定の主な内容

- 割り込みの禁止
- CPU / 周辺ハードウェア・クロックを高速内蔵発振クロック動作の2MHzに設定
- ポート・モードの設定 (入力ポート / 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)

## (2) メイン処理動作の内容

78K0R/Kx3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

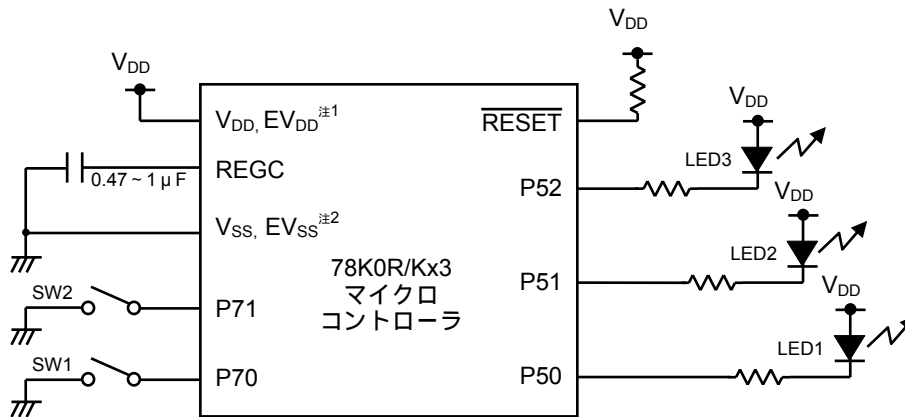
**注意** デバイス使用上の注意事項については、各製品のユーザズ・マニュアル (78K0R/KE3, 78K0R/KF3, 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3) を参照してください。

## 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<sub>SS</sub>端子はV<sub>SS</sub>, EV<sub>SS</sub>と同電位にし、GNDに直接接続してください。AV<sub>REF</sub>端子は、1.8V AV<sub>REF</sub> V<sub>DD</sub>の電圧範囲で使用してください。
2. EV<sub>DD</sub>は、V<sub>DD</sub>と同電位にしてください。
3. REGCはコンデンサ(0.47~1μF)を介し、V<sub>SS</sub>, EV<sub>SS</sub>に接続してください。
4. 回路図中の端子、X1/X2とXT1/XT2発振子端子およびAV<sub>REF</sub>, AV<sub>SS</sub>端子以外の未使用のポート機能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。

注1. 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3マイクロコントローラはEV<sub>DD0</sub>とEV<sub>DD1</sub>になります。

注2. 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3マイクロコントローラはEV<sub>SS0</sub>とEV<sub>SS1</sub>になります。

注3. 78K0R/KF3, 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3マイクロコントローラはAV<sub>REF0</sub>とAV<sub>REF1</sub>になります。

### 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・プログラムの使用する周辺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、メイン処理と割り込み処理のソース・ファイル	注	注
main.c (C言語版)			
Uart_LED.prw	統合開発環境 PM+用ワーク・スペース・ファイル		
Uart_LED.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。

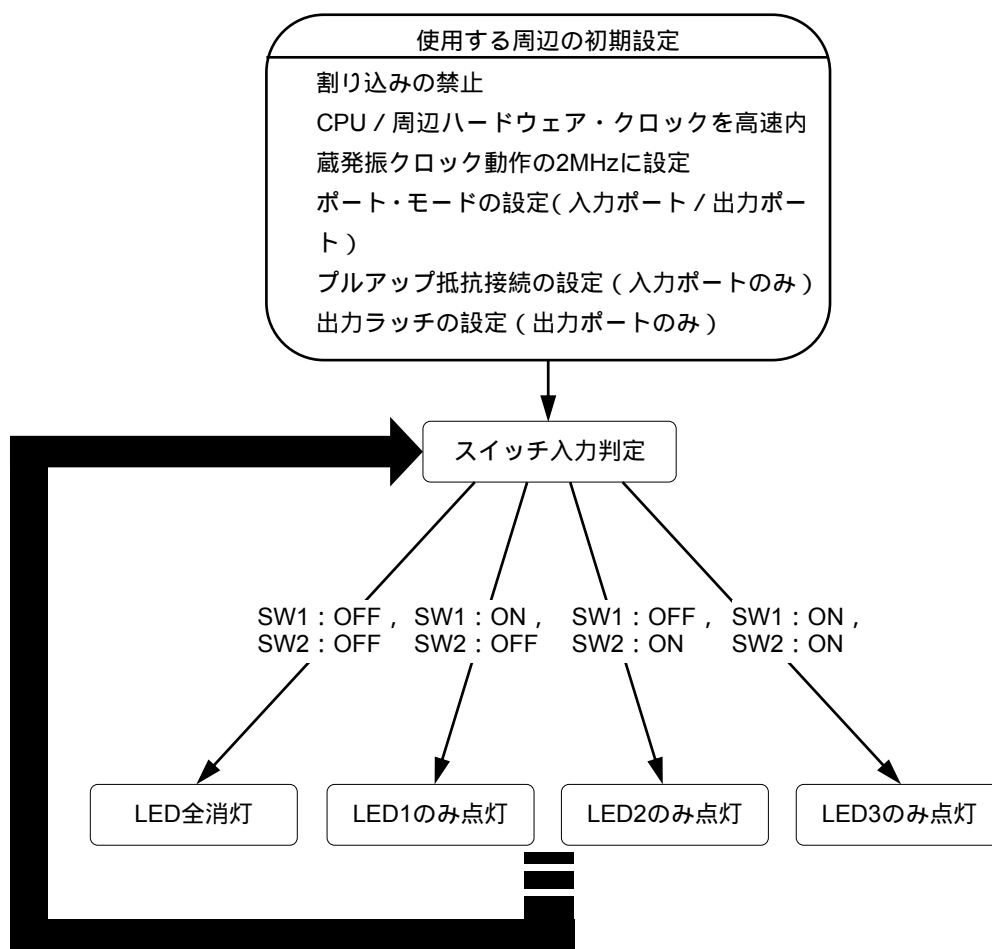
- ・入力ポート（スイッチ入力用） : P70, P71
- ・出力ポート（LED点灯用） : P50, P51, P52

## 3.3 使用する周辺の初期設定と動作概要

このサンプル・プログラムでは、使用する周辺の初期設定にて、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。

使用する周辺の初期設定完了後は、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。

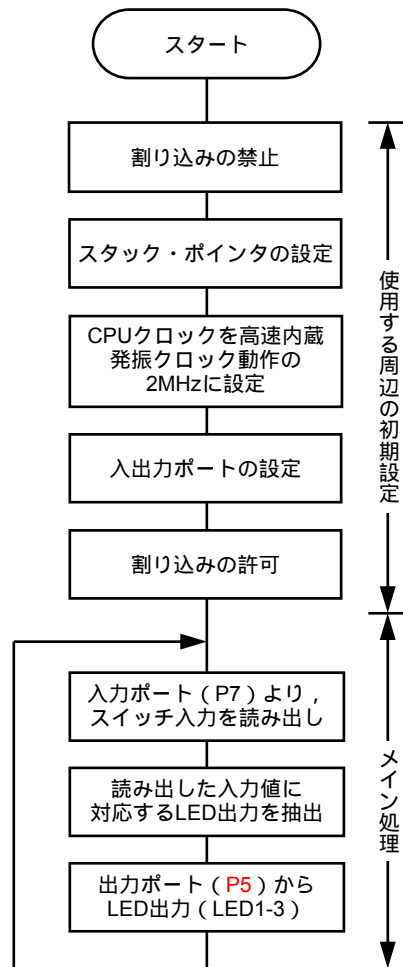
詳細については、次の状態遷移図（ステート・チャート）に示します。





### 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。



**注意** オプション・バイトは、RA78K0Rのリンク・オプションにて設定してください。設定の仕方については、RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアルを参照してください。

オプション・バイトは、次の内容が設定されます。

- ・ウォッチドッグ・タイマの動作
- ・リセット解除時（電源立ち上げ時）のLVIの設定
- ・オンチップ・デバッグの動作制御

## 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、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(78K0R/KE3, 78K0R/KF3, 78K0R/KG3, 78K0R/KH3, 78K0R/KJ3)を参照してください。

アセンブラ命令については、78K0Rマイクロコントローラ 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、RA78K0Rのリンカ・オプションにて設定してください。オプション・バイトで、次の項目を設定します。

- (1) ユーザ・オプション・バイト
  - ・ ウォッチドッグ・タイマの設定
  - ・ リセット解除時(電源立ち上げ時)のLVIの設定
- (2) オンチップ・デバッグ・ユーザ・オプション・バイト
  - オンチップ・デバッグの動作制御

図4-1-1 ユーザ・オプション・バイトのフォーマット(000C0H/010C0H)(1/2)

アドレス: 000C0H/010C0H<sup>注1</sup>

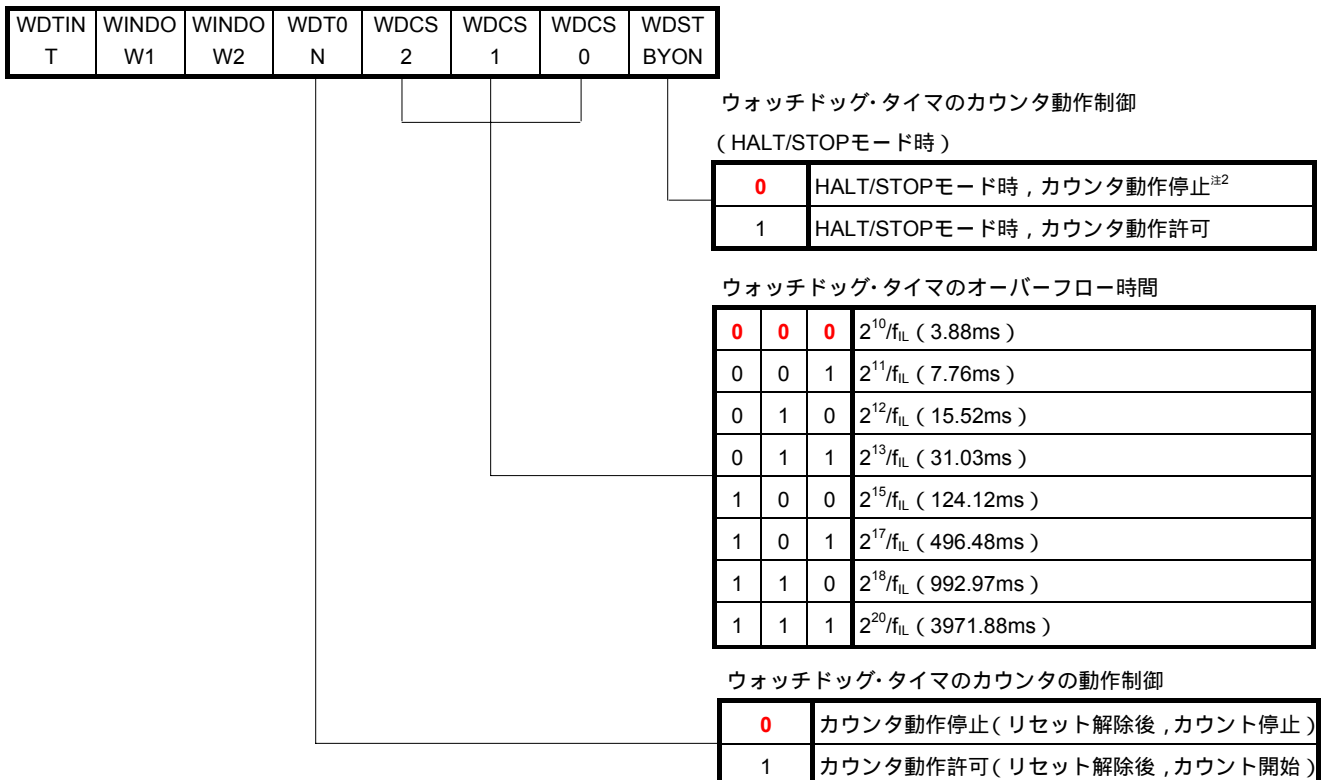


図4 - 1 - 1 ユーザ・オプション・バイトのフォーマット (000C0H/010C0H) (2/2)

アドレス : 000C0H/010C0H<sup>注1</sup>

WDTIN	WINDO	WINDO	WDT0	WDCS	WDCS	WDCS	WDST
T	W1	W2	N	2	1	0	BYON

ウォッチドッグ・タイマのウィンドウ・オープン期間<sup>注2</sup>

0	0	25%
0	1	50%
1	0	75%
1	1	100%

ウォッチドッグ・タイマのインターバル割り込みの使用 / 不使用

0	インターバル割り込みを使用しない
1	オーバーフロー時間の75%到達時にインターバル割り込みを発生する

- 注1. ブート・スワップ時は, 000C0Hと010C0Hが切り替わるので, 010C0Hにも000C0Hと同じ値を設定してください。
- 注2. WDSTBYON = 0のときは, WINDOW1, WINDOW0の値に関係なく, ウィンドウ・オープン期間100%となります。

注意 表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 - 1 - 2 ユーザ・オプション・バイトのフォーマット (000C1H/010C1H)

アドレス : 000C1H/010C1H<sup>注</sup>

1	1	1	1	1	1	1	LVI OFF
---	---	---	---	---	---	---	---------

電源立ち上げ時のLVIの設定

0	リセット解除時(電源立ち上げ時), LVIがデフォルトでON (LVIデフォルト・スタート機能動作)
1	リセット解除時(電源立ち上げ時), LVIがデフォルトでOFF (LVIデフォルト・スタート機能停止)

- 注. ブート・スワップ時は, 000C1Hと010C1Hが切り替わるので, 010C1Hにも000C1Hと同じ値を設定してください。

注意 表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 - 1 - 3 ユーザ・オプション・バイトのフォーマット (000C2H/010C2H)

アドレス : 000C2H/010C2H<sup>注</sup>

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

注. 000C2Hは予約領域なので、必ずFFHを設定してください。またブート・スワップ時は、000C2Hと010C2Hが切り替わるので、010C2HにもFFHを設定してください。

図4 - 1 - 4 オンチップ・デバッグ・オプション・バイトのフォーマット (000C3H/010C3H)

アドレス : 000C3H/010C3H<sup>注</sup>

OCDE NSET	0	0	0	0	1	0	OCDE RDS
--------------	---	---	---	---	---	---	-------------

オンチップ・デバッグ動作制御

0	0	オンチップ・デバッグ動作禁止
0	1	設定禁止
1	0	オンチップ・デバッグ動作許可、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
1	1	オンチップ・デバッグ動作許可、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

注. ブート・スワップ時は、000C3Hと010C3Hが切り替わるので、010C3Hにも000C3Hと同じ値を設定してください。

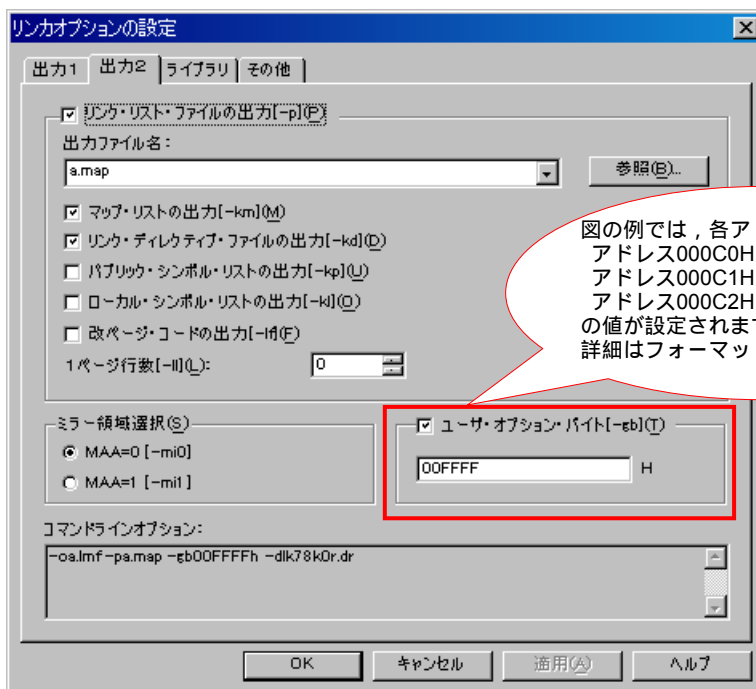
注意 表の赤字部分がサンプル・プログラムでの設定値となります。本サンプル・プログラムでは未使用のため、特に設定は行っていません。

## リンカ・オプションによる設定

### (1) ユーザ・オプション・バイトの設定

PM+の「ツール リンカオプションの設定」にて、「出力2」のタブを選択し、オプション・バイトに値を設定する場にチェックを有効にし、値を指定します。

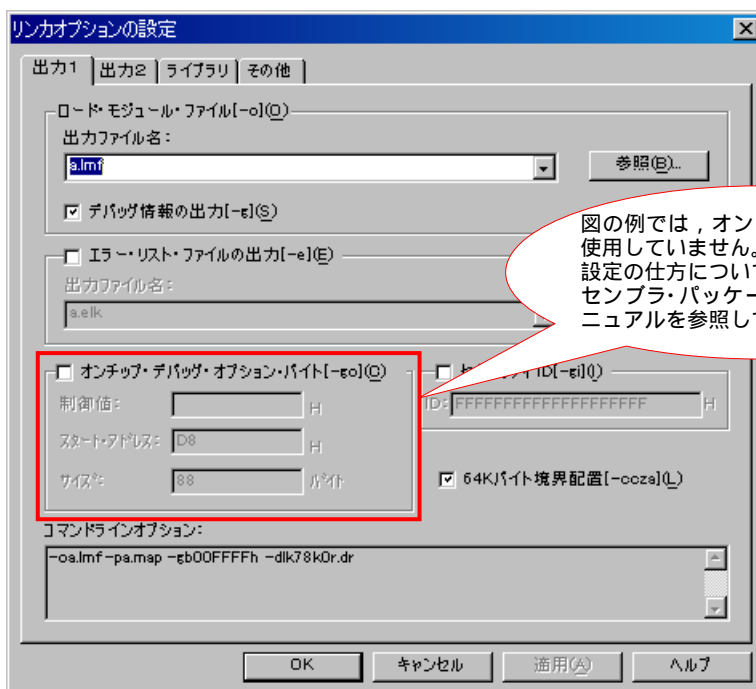
図4-1-5 [リンカオプションの設定] ダイアログ ([出力2] タブ選択時)



### (2) オンチップ・デバッグ・オプション・バイトの設定

PM+の「ツール リンカオプションの設定」にて、「出力2」のタブを選択します。

図4-1-5 [リンカオプションの設定] ダイアログ ([出力1] タブ選択時)



【オンチップ・デバッグ・オプション・バイトの設定例】

オンチップ・デバッグ動作許可，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，内部ROM終了アドレスが0x3FFFF。

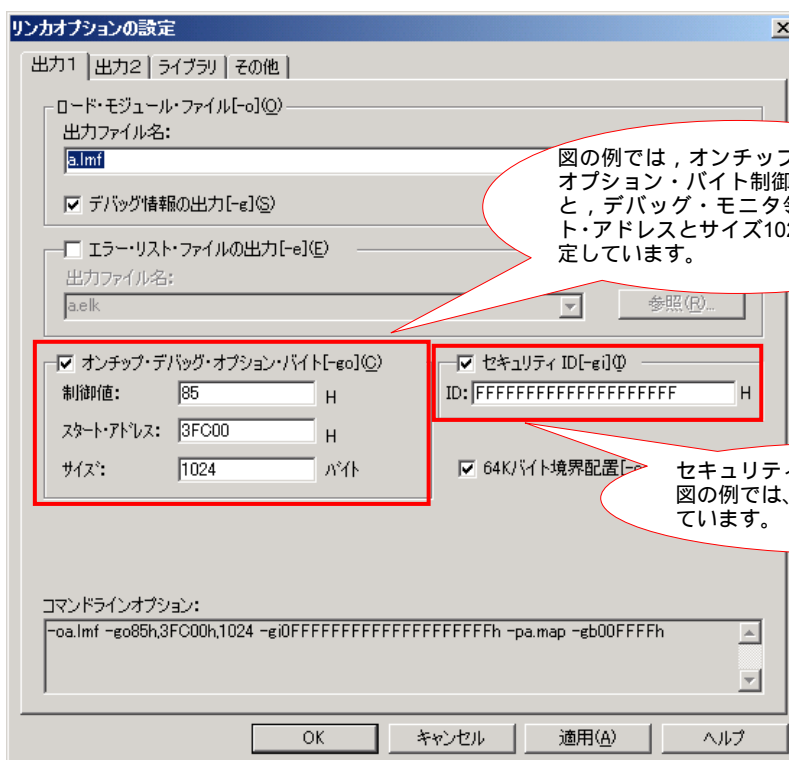
アドレス：000C3H/010C3H<sup>注</sup>

OCDE NSET	0	0	0	0	1	0	OCDE RDS
--------------	---	---	---	---	---	---	-------------

オンチップ・デバッグ動作制御

0	0	オンチップ・デバッグ動作禁止
0	1	設定禁止
1	0	オンチップ・デバッグ動作許可，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
1	1	オンチップ・デバッグ動作許可，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

注. ブート・スワップ時は，000C3Hと010C3Hが切り替わるので，010C3Hにも000C3Hと同じ値を設定してください。



詳しい設定方法については，RA78K0R アセンブラ・パッケージ 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。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、他のサンプル・プログラムを参照してください。

### 【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）

			アドレス	機能名
TVCT1	CSEG	AT	000000H	
	DW	RESET_START	; (00)	RESET入力, POC, LVI, WDT, TRAP
TVCT2	CSEG	AT	000004H	
	DW	RESET_START	; (04)	INTWDTI
	DW	RESET_START	; (06)	INTLVI
	DW	RESET_START	; (08)	INTP0
	DW	RESET_START	; (0A)	INTP1
	DW	RESET_START	; (0C)	INTP2
	DW	RESET_START	; (0E)	INTP3
	DW	RESET_START	; (10)	INTP4
	DW	RESET_START	; (12)	INTP5
	DW	RESET_START	; (14)	INTST3
	DW	RESET_START	; (16)	INTSR3
	DW	RESET_START	; (18)	INTSRE3
	DW	RESET_START	; (1A)	INTDMA0
	DW	RESET_START	; (1C)	INTDMA1
	DW	RESET_START	; (1E)	INTST0/INTCSI00
	DW	RESET_START	; (20)	INTSR0/INTCSI01
	DW	RESET_START	; (22)	INTSRE0
	DW	RESET_START	; (24)	INTST1/INTCSI10/INTIIC10
	DW	RESET_START	; (26)	INTSR1
	DW	RESET_START	; (28)	INTSRE1
	DW	RESET_START	; (2A)	INTIIC0
	DW	RESET_START	; (2C)	INTTM00
	DW	RESET_START	; (2E)	INTTM01
	DW	RESET_START	; (30)	INTTM02
	DW	RESET_START	; (32)	INTTM03
	DW	RESET_START	; (34)	INTAD
	DW	RESET_START	; (36)	INTRTC
	DW	RESET_START	; (38)	INTRTCI
	DW	RESET_START	; (3A)	INTKR
	DW	RESET_START	; (3C)	INTST2/INTCSI20/INTIIC20
	DW	RESET_START	; (3E)	INTSR2
	DW	RESET_START	; (40)	INTSRE2
	DW	RESET_START	; (42)	INTTM04
	DW	RESET_START	; (44)	INTTM05
	DW	RESET_START	; (46)	INTTM06
	DW	RESET_START	; (48)	INTTM07
	DW	RESET_START	; (4A)	INTP6
	DW	RESET_START	; (4C)	INTP7
	DW	RESET_START	; (4E)	INTP8
	DW	RESET_START	; (50)	INTP9
	DW	RESET_START	; (52)	INTP10
	DW	RESET_START	; (54)	INTP11
TBRK	CSEG	AT	00007EH	
	DW	RESET_START	; (7E)	BRK

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。

### 4.3 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領域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, 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 1】 RAM領域のFFEC0H- FFEDFH (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)

DSTK	DSEG	AT	0FFEC0H	}	RAM領域に スタック領域確保
STACKEND:	DS	20H			
STACKTOP:					
XMAIN	CSEG	UNIT		}	リセット解除後に スタック・ポインタ設定
RESET_START:					
	MOVW	SP,	#LOWW STACKTOP		

この例では、スタック・ポインタにはFFEE0H (= FFEC0H + 20H) のアドレスが指定されます。

上記の記述により、RAM領域の末端の32バイト (FFEC0H- FFEDFH) をスタック領域として確保することが可能となります。

#### 【例 2】 RAM領域のFFE00H- FFE1FH (32バイト) を、スタック領域として使用する場合

XSTK	DSEG	AT	0FFE00H	}	RAM領域に スタック領域確保
STACKEND:	DS	20H			
STACKTOP:					
XMAIN	CSEG	UNIT		}	リセット解除後に スタック・ポインタ設定
RESET_START:					
	MOVW	SP,	#LOWW STACKTOP		

この例では、スタック・ポインタにはFFE20H (= FFE00H + 20H) のアドレスが指定されます。この設定では、saddr領域を避けて、スタック領域を確保できます。



## 4.4 クロックの設定

### (1) クロック動作モードの設定

X1/P121, X2/EXCLK/P122端子およびXT1/P123, XT2/P124端子の動作モードの設定と、発振回路のゲインを選択します。

プログラムに使用するクロックに合わせて設定を行います。

このサンプル・プログラムでは、 $f_{CLK} = f_{IH}/2^2 = 2\text{MHz}$ になるように、クロック動作モード制御レジスタ (CKC) を設定しています。

図4-4-1 クロック動作モード制御レジスタ (CMC) のフォーマット

アドレス：FFFA0H

EXCLK	OSCSEL	0	OSCSEL	0	0	0	AMPH
	L		LS				

AMPH	高速システム・クロック発振周波数の制御	
0	2MHz	$f_{MX} \leq 10\text{MHz}$
1	$10\text{MHz} < f_{MX}$	20MHz

OSCS ELS	サブシステム・クロック端子の動作モード	XT1/P123端子	XT2/P124端子
0	入力ポート・モード	入力ポート	
1	XT1発振モード	水晶振動子接続	

EXCLK	OSC SELS	高速システム・クロック端子の動作モード	X1/P121 端子	X2/EXCLK/P122 端子
0	0	入力ポート・モード	入力ポート	
0	1	X1発振モード	入力ポート	
1	0	入力ポート・モード	入力ポート	
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

- 注意1. ビット5, 3-1には必ず0を設定してください。  
 2. 表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。

備考  $f_{CLK}$  : CPU / 周辺ハードウェア・クロック周波数  
 $f_{IH}$  : 高速内蔵発振クロック周波数

・アセンブリ言語の場合

```
MOV CMC, #00000000B
```

・C言語の場合

```
CMC = 0b00000000;
```

(2) クロック動作ステータスの設定

高速システム・クロック，高速内蔵発振クロック，サブシステム・クロックの動作を制御します。

このサンプル・プログラムでは，高速内蔵発振クロックのみを使用し，X1端子は入力ポートモードですの  
で，特に設定を行う必要がありませんので，リセット時の状態のままです。

図4-4-2 クロック動作ステータス制御レジスタ (CSC) のフォーマット

アドレス：FFFA1H

MSTO P	XTSTO P	0	0	0	0	0	HIOST OP
-----------	------------	---	---	---	---	---	-------------

HIOST P	高速内蔵発振クロックの動作制御		
0	高速内蔵発振回路動作		
1	高速内蔵発振回路停止		

XTSTOP	サブシステム・クロックの動作制御		
	XT1発振モード時	入力ポート・モード時	
0	XT1発振回路動作	-	
1	XT1発振回路停止	-	

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	入力ポート・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. ビット5-1には必ず0を設定してください。  
2. 表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。

(3) 発振安定時間の選択

STOPモード解除時のX1クロックの発振安定時間を選択します。

CPUクロックにX1クロックを選択した場合、STOPモード解除後は、OSTSで設定した時間を自動でウェイトします。

CPUクロックに高速内蔵発振クロックを選択した場合、STOPモード解除後は、OSTCで発振安定時間が経過したかを確認してください。OSTCでは、あらかじめOSTSで設定した時間までの確認ができます。

このサンプル・プログラムでは、STOPモードを使用しませんので、特に設定を行う必要がありません。レジスタの設定は、リセット時の状態のままです。

図4 - 4 - 3 発振安定時間選択レジスタ (OSTS) のフォーマット

アドレス : FFFA3H

0	0	0	0	0	OSTS2	OSTS1	OSTS0
---	---	---	---	---	-------	-------	-------

OSTS2	OSTS1	OSTS0	発振安定時間の選択
0	0	0	$2^9/f_x$ (32.0 $\mu$ s)
0	0	1	$2^9/f_x$ (64.0 $\mu$ s)
0	1	0	$2^{10}/f_x$ (128 $\mu$ s)
0	1	1	$2^{11}/f_x$ (256 $\mu$ s)
1	0	0	$2^{13}/f_x$ (1.024ms)
1	0	1	$2^{15}/f_x$ (4.096ms)
1	1	0	$2^{17}/f_x$ (16.38ms)
1	1	1	$2^{18}/f_x$ (32.76ms)

- 注意1. ビット7-3には必ず0を設定してください。
2. 表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) 動作スピード・モードの設定

フラッシュ・メモリの高速動作昇圧回路を制御します。

システム・クロックを10 MHz以下の低速で動作する際には、初期値00Hで使用することで、消費電力を低減することができます。

このサンプル・プログラムでは、2MHzで動作を行うため、特に設定を行う必要がありません。レジスタの設定は、リセット時の状態のままです。

図4 - 4 - 4 動作スピード・モード制御レジスタ (OSMC) のフォーマット

アドレス : F00F3H

0	0	0	0	0	0	0	FSEL
---	---	---	---	---	---	---	------

FSEL	$f_{CLK}$ の周波数選択
0	10MHz以下の周波数で動作 (デフォルト)
1	10MHzを越える周波数で動作

注意1. ビット7-1には必ず0を設定してください。

2. 表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。本サンプル・プログラムではフラッシュ・メモリの使用はありませんので、リセット時の状態になります。

(5) システム・クロックの選択

CPU / 周辺ハードウェア・クロックの選択，分周比を設定します。

このサンプル・プログラムでは， $f_{CLK} = f_{IH}/2^2 = 2\text{MHz}$ になるように，システム・クロック制御レジスタ（CKC）を設定しています。

図4-4-5 システム・クロック制御レジスタ（CKC）のフォーマット

アドレス：FFFA4H

CLS	CSS	MCS	MCM0	1	MDIV2	MDIV1	MDIV0
-----	-----	-----	------	---	-------	-------	-------

CSS	MCM0	MDIV2	MDIV1	MDIV0	CPU / 周辺ハードウェア・クロック（ $f_{CLK}$ ）の選択
0	0	0	0	0	$f_{IH}$
		0	0	1	$f_{IH}/2$ （デフォルト）
		0	1	0	$f_{IH}/2^2$
		0	1	1	$f_{IH}/2^3$
		1	0	0	$f_{IH}/2^4$
		1	0	1	$f_{IH}/2^5$
0	1	0	0	0	$f_{MX}$
		0	0	1	$f_{MX}/2$
		0	1	0	$f_{MX}/2^2$
		0	1	1	$f_{MX}/2^3$
		1	0	0	$f_{MX}/2^4$
		1	0	1	$f_{MX}/2^5$ ( $f_{MX} < 4\text{MHz}$ のとき，設定禁止)
1	x	x	x	x	$f_{SUB}/2$
上記以外					設定禁止

- 注意1. ビット3には必ず1を設定してください。  
 2. 表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。

備考  $f_{CLK}$ ：CPU / 周辺ハードウェア・クロック周波数  
 $f_{IH}$ ：高速内蔵発振クロック周波数

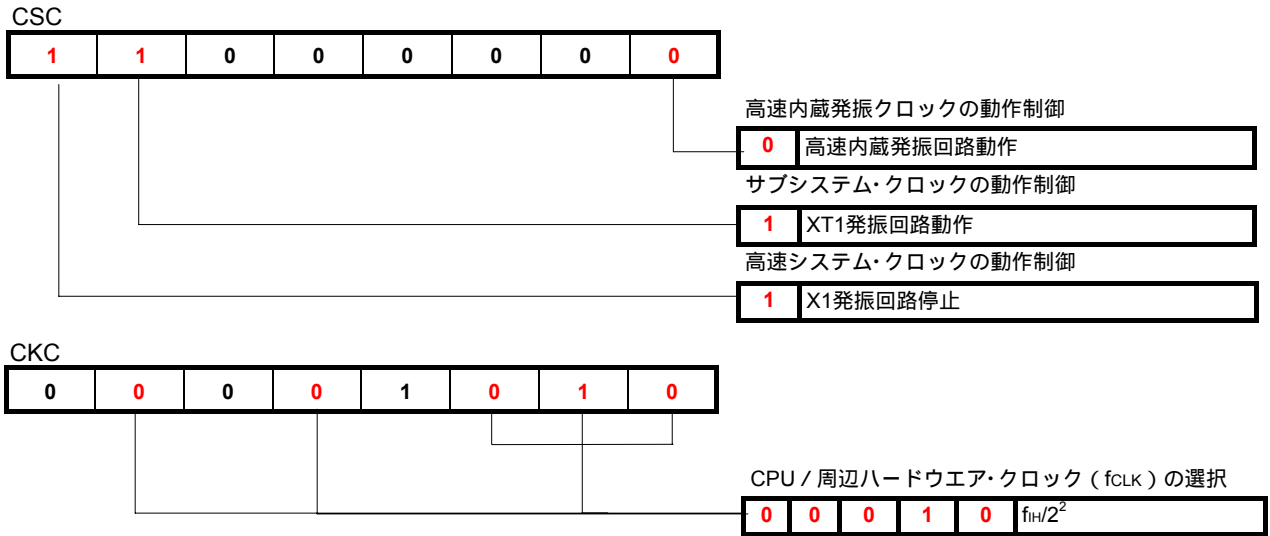
・アセンブリ言語の場合

```
MOV CKC, #00001011B
```

・C言語の場合

```
CKC = 0b00001011;
```

【例1】 CPU/周辺ハードウェア・クロックに高速内蔵発振クロック(2 MHz)を設定する。  
(サンプル・プログラムの設定と同内容)



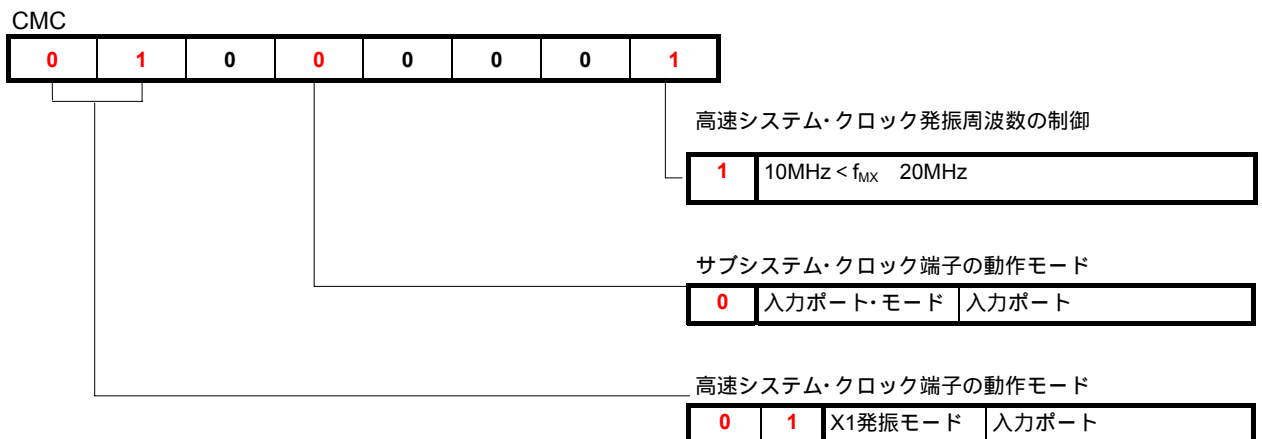
・アセンブリ言語の場合

```
MOV CSC, #11000000B
MOV CKC, #00001010B
```

・C言語の場合

```
CSC = 0b11000000;
CKC = 0b00001010;
```

【例2】 CPU/周辺ハードウェア・クロックに高速システム・クロック(20 MHz)を設定する。



CSC



高速内蔵発振クロックの動作制御

0 高速内蔵発振回路動作

サブシステム・クロックの動作制御

1 XT1発振回路動作

高速システム・クロックの動作制御

0 X1発振回路動作

OSMC



f<sub>CLK</sub>の周波数選択

1 10MHzを越える周波数で動作

CKC



CPU / 周辺ハードウェア・クロック (f<sub>CLK</sub>) の選択

0	1	0	0	0	f <sub>MX</sub>
---	---	---	---	---	-----------------

・アセンブリ言語の場合

```

MOV CMC, #01000001B
MOV CSC, #01000000B
MOV OSMC, #00000001B

_Loop:
    CMP    !OSTC, #0FFH
    BNZ    $_Loop        ; 発振安定時間待ち
MOV CKC, #00011000B
    
```

・C言語の場合

```

CMC = 0b01000001;
CSC = 0b01000000;
OSMC = 0b00000001;
while(OSTC != 0xff);        /* 発振安定時間待ち */
CKC = 0b00011000;
    
```

## 4.5 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。

	78K0R/KE3	78K0R/KF3	78K0R/KG3	78K0R/KH3	78K0R/KJ3
ポート0	P00-P06	P00-P06	P00-P06	P00-P07	P00-P07
ポート1	P10-P17	P10-P17	P10-P17	P10-P17	P10-P17
ポート2	P20-P27	P20-P27	P20-P27	P20-P27	P20-P27
ポート3	P30-P31	P30-P31	P30-P31	P30-P37	P30-P37
ポート4	P40-P43	P40-P47	P40-P47	P40-P47	P40-P47
ポート5	P50-P55	P50-P55	P50-P57	P50-P57	P50-P57
ポート6	P60-P63	P60-P67	P60-P67	P60-P67	P60-P67
ポート7	P70-P77	P70-P77	P70-P77	P70-P77	P70-P77
ポート8	-	-	P80-P87	P80-P87	P80-P87
ポート9	-	P90	P90	P90-P97	P90-P97
ポート10	-	-	-	-	P100-P107
ポート11	-	P110-P111	P110-P111	P110-P117	P110-P117
ポート12	P120-P124	P120-P124	P120-P124	P120-P127	P120-P127
ポート13	P130	P130	P130-P131	P130-P131	P130-P137
ポート14	P140-P141	P140-P145	P140-P145	P140-P145	P140-P147
ポート15	-	-	P150-P157	P150-P157	P150-P157
ポート16	-	-	-	P160-P163	P160-P163



(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、PM7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5を後述の【例 1】、ポート7を【例 2】の内容で設定しています。

図4 - 5 - 1 ポート・モード・レジスタ7 (PM7) のフォーマット

PM7							
PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

P7n (n = 0-7) 端子の入出力モードの選択	
0	出力モード
1	入力モード

(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、P7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5を後述の【例 1】の内容で設定しています。

図4 - 5 - 2 ポート・レジスタ7 (P7) のフォーマット

P7							
P77	P76	P75	P74	P73	P72	P71	P70

P7n (n = 0-7) 端子の出力ラッチのレベル選択	
0	ロウ・レベル出力
1	ハイ・レベル出力

(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、PU7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7を後述の【例 2】の内容で設定しています。

図4 - 5 - 3 プルアップ抵抗オプション・レジスタ7 (PU7) のフォーマット

PU7							
PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70

P7n (n = 0-7) 端子の内蔵プルアップ抵抗の選択	
0	内蔵プルアップ抵抗を接続しない
1	内蔵プルアップ抵抗を接続する

- 【例 1】 ・ P50-P57を出力ポートに設定  
 ・ P50-P52の出力ラッチをハイ・レベル出力に，P53-P57の出力ラッチをロウ・レベル出力に設定  
 (サンプル・プログラムの設定と同内容)

PM5

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



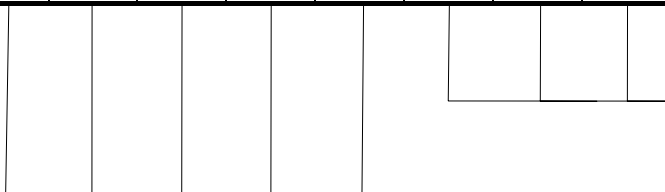
P5n (n = 0-7) 端子の入出力モードの選択

0	出力モード
---	-------

\*このサンプル・プログラムでは，P50-P52をLED点灯用の出力ポートとして使用するため，ポート5を出力ポートに設定します。

P5

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



P50-P52端子の出力ラッチのレベル選択

1	ハイ・レベル出力
---	----------

P53-57端子の出力ラッチのレベル選択

0	ロウ・レベル出力
---	----------

\*このサンプル・プログラムでは，P50-P52をLED点灯用の出力ポートとして使用するため，初期設定でP50-P52の出力ラッチのレベルをハイ・レベル出力にしておきます (LEDは，P50-P52からの出力がロウ・レベルのときに点灯します (2.1 回路図を参照))。

PM5の設定値は「00000000」，P5の設定値は「00000111」となります。

・アセンブリ言語の場合

```
MOV PM5, #00000000B
MOV P5, #00000111B
```

・C言語の場合

```
PM5 = 0b00000000
P5 = 0b00000111
```

- 【例 2】 ・ P70, P71を入力ポートに設定  
 ・ P70, P71に内蔵プルアップ抵抗接続を設定  
 (サンプル・プログラムの設定と同内容)

PM7

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

P70, P71端子の入出力モードの選択

1	入力モード
---	-------

PU7

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

P70, P71端子の入出力モードの選択

1	内蔵プルアップ抵抗を接続する
---	----------------

PM7の設定値は、「00000011」となります。また、PU7の設定値は、「00000011」となります。

・アセンブリ言語の場合

```
MOV PM7, #00000011B
MOV PU7, #00000011B
```

・C言語の場合

```
PM7 = 0b00000011
PU7 = 0b00000011
```

## 4.6 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。

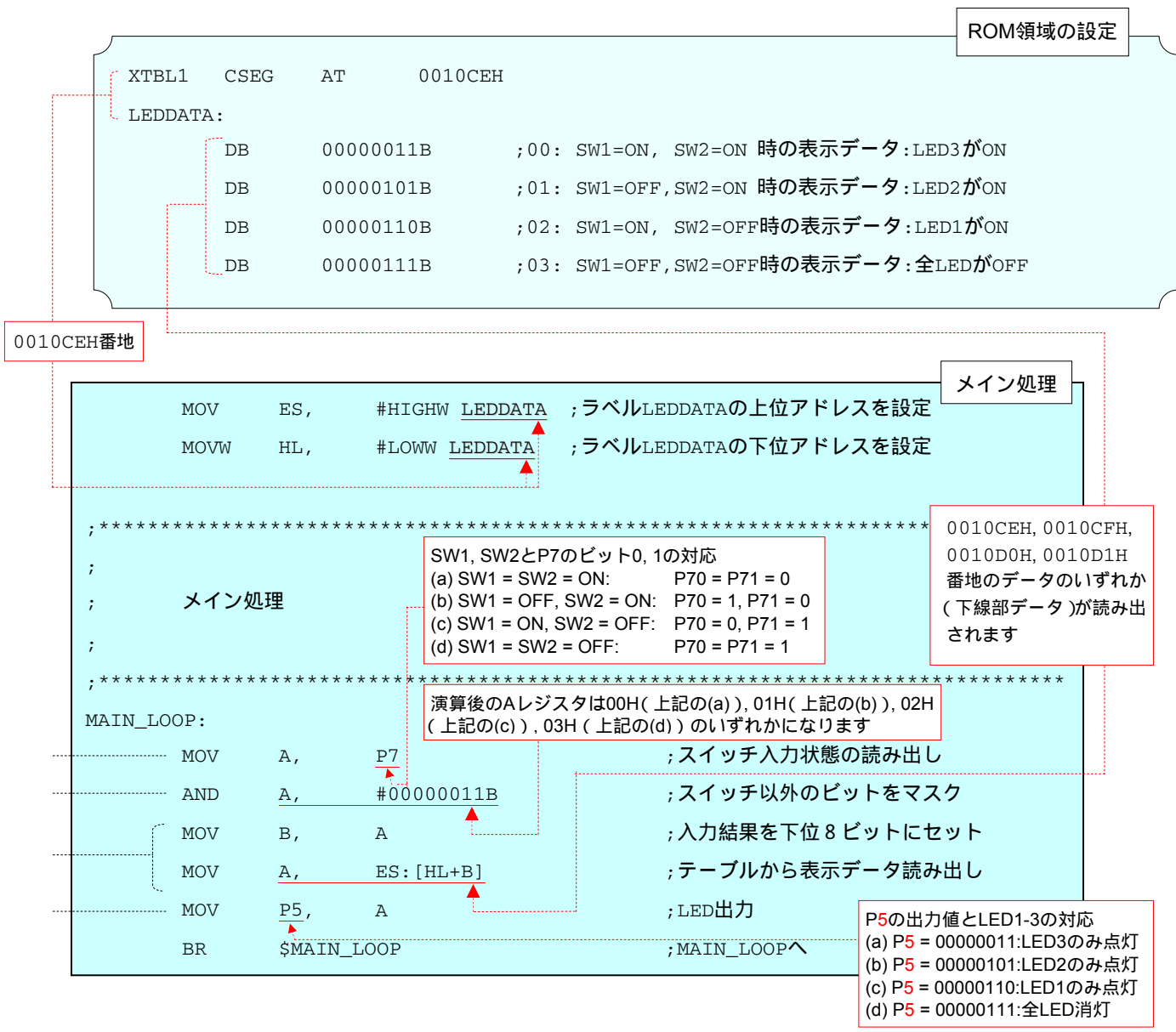
P7のデータを読み込みます。

読み込んだ8ビット・データのうち、入力ポート（P70, P71）以外のビットを0にします。

P70, P71の入力レベルの組み合わせに対応する出力データを、010CEH-010D1H番地（「LEDDATA」テーブル）より取り出します。

取り出したデータをP5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P70, P71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。



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。

```

/*****
*/

メイン処理

*****/
void main(void)
{

    const unsigned char OUTDATA[] =          /* 表示データパターン用配列 */
    {
        0b00000011
        ,0b00000101
        ,0b00000110
        ,0b00000111
    };

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

    while(1){
        INDATA = P7 & 0b00000011; /* 有効スイッチ情報取得 */
        P5 = OUTDATA[INDATA];      /* テーブルから表示データを読み出して表示 */
    }
}

```

{ }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	P70, P71	INDATA	OUTDATA	LED点灯
SW1 = ON, SW2 = ON	P70 = 0, P71 = 0	0b00000000	0b00000100	LED3のみ点灯
SW1 = OFF, SW2 = ON	P70 = 1, P71 = 0	0b00000001	0b00000010	LED2のみ点灯
SW1 = ON, SW2 = OFF	P70 = 0, P71 = 1	0b00000010	0b00000001	LED1のみ点灯
SW1 = OFF, SW2 = OFF	P70 = 1, P71 = 1	0b00000011	0b00000000	全LED消灯

## 第5章 関連資料

資料名		和文 / 英文
78K0R/KE3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0R/KF3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0R/KG3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0R/KH3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0R/KJ3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0Rマイクロコントローラ 命令編 ユーザーズ・マニュアル		<a href="#">PDF</a>
RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
CC78K0R Cコンパイラ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		<a href="#">PDF</a>

## 付録A プログラム・リスト

プログラム・リスト例として、78K0R/KG3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      78K0R/KG3シリーズ
;
;*****
; 78K0R/KG3シリーズ      サンプル・プログラム
;*****
; LED点灯のスイッチ制御
;*****
; 【履歴】
; 2007.12.--      新規作成
;*****
;
; 【概要】
;
; このサンプル・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、
; 78K0R/Kx3マイクロコントローラの基本的な使用する周辺の初期設定を行います。また、
; 使用する周辺の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。
;
;
; <使用する周辺の初期設定の主な設定内容>
;
; ・割り込みの禁止
; ・CPU/周辺ハードウェア・クロックを高速内蔵発振クロック動作の2MHzに設定
; ・ポート・モードの設定（入力ポート/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）
; ・割り込みの許可
;
;
; <メイン処理の主な内容>
;
; ・スイッチ入力を検出
; ・LED点灯
```

```

;
;
; <スイッチ入力とLED点灯>
;
; + +
; | スイッチ1 | スイッチ2 | LED1 | LED2 | LED3 |
; | (P70) | (P71) | (P50) | (P51) | (P52) |
; |
; | 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：P70, P71
; 出力ポート：P50-52, P53
; 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
;
;=====
TVCT1 CSEG AT 000000H
      DW RESET_START ;(00) RESET入力,POC,LVI,WDT,TRAP
TVCT2 CSEG AT 000004H
      DW RESET_START ;(04) INTWDTI
      DW RESET_START ;(06) INTLVI
      DW RESET_START ;(08) INTP0
      DW RESET_START ;(0A) INTP1
      DW RESET_START ;(0C) INTP2
      DW RESET_START ;(0E) INTP3
      DW RESET_START ;(10) INTP4
      DW RESET_START ;(12) INTP5
      DW RESET_START ;(14) INTST3
      DW RESET_START ;(16) INTSR3

```



```

DW    RESET_START                ;(18)  INTSRE3
DW    RESET_START                ;(1A)  INTDMA0
DW    RESET_START                ;(1C)  INTDMA1
DW    RESET_START                ;(1E)  INTST0/INTCS100
DW    RESET_START                ;(20)  INTSR0/INTCS101
DW    RESET_START                ;(22)  INTSRE0
DW    RESET_START                ;(24)  INTST1/INTCS110/INTIIC10
DW    RESET_START                ;(26)  INTSR1
DW    RESET_START                ;(28)  INTSRE1
DW    RESET_START                ;(2A)  INTIIC0
DW    RESET_START                ;(2C)  INTTM00
DW    RESET_START                ;(2E)  INTTM01
DW    RESET_START                ;(30)  INTTM02
DW    RESET_START                ;(32)  INTTM03
DW    RESET_START                ;(34)  INTAD
DW    RESET_START                ;(36)  INTRTC
DW    RESET_START                ;(38)  INTRTCI
DW    RESET_START                ;(3A)  INTKR
DW    RESET_START                ;(3C)  INTST2/INTCS120/INTIIC20
DW    RESET_START                ;(3E)  INTSR2
DW    RESET_START                ;(40)  INTSRE2
DW    RESET_START                ;(42)  INTTM04
DW    RESET_START                ;(44)  INTTM05
DW    RESET_START                ;(46)  INTTM06
DW    RESET_START                ;(48)  INTTM07
DW    RESET_START                ;(4A)  INTP6
DW    RESET_START                ;(4C)  INTP7
DW    RESET_START                ;(4E)  INTP8
DW    RESET_START                ;(50)  INTP9
DW    RESET_START                ;(52)  INTP10
DW    RESET_START                ;(54)  INTP11

TBRK  CSEG    AT      00007EH
      DW    RESET_START                ;(7E)  BRK

;=====
;
;   ROMの定義
;
;=====

XTBL1 CSEG    AT      0010CEH
LEDDATA:
      DB    00000011B                ;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      00000110B      ;02: SW1=ON, SW2=OFF時の表示データ:LED1がON
DB      00000111B      ;03: SW1=OFF,SW2=OFF時の表示データ:全LEDがOFF

;=====
;
;   スタック領域の確保
;
;=====
DSTK DSEG  AT      OFFEC0H
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 = FFEE0H

;*****
;
;   使用する周辺の初期設定
;
;*****
XMAIN CSEG  UNIT
RESET_START:

      DI              ;割り込み禁止
;-----
;   レジスタバンク設定
;-----
      SEL      RBO
;-----
;   スタック・ポインタの設定
;-----
      MOVW     SP,    #LOWW 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      CMC,   #00000000B          ;クロック動作モード
;|||||+----- AMPH: 2MHz fMX 10MHz
;|||+++----- <000>
;||+----- OSCSELS: P123/P124端子を入力ポート・モード
;||+----- <0>
;++----- EXCLK/OSCSEL: 入力ポート・モード

```

```

MOV    CSC,    #11000000B          ;クロック動作ステータス制御
        ;|||||+----- HIOSTOP: 高速内蔵発振回路動作
        ;||++++----- <00000>
        ;|+----- XTSTOP: XT1発振回路停止
        ;+----- MSTOP: X1発振回路停止

MOV    OSMC,   #00000000B          ;動作スピード・モード
        ;|||||+----- FSEL: 10MHz以下の周波数で動作
        ;+++++----- <00000>

MOV    CKC,    #00001010B          ;クロック選択
        ;||||+----- MDIV2-0: CPU/周辺ハードウェア・クロック(fCLK)=fIH/2^2
        ;|||+----- <1>
        ;||+----- MCMO: 高速内蔵発振クロック(fIH)
        ;|+----- <R>
        ;|+----- CSS: メイン・システム・クロック(fMAIN)=fCLK
        ;+----- <R>

;-----
;   ポート0の設定
;-----
MOV    P0,     #00000000B          ;P00-P06の出力ラッチLow
MOV    PM0,    #10000000B          ;P00-P06を出力ポートに設定

;-----
;   ポート1の設定
;-----
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

;-----
;   ポート2の設定
;-----
MOV    P2,     #00000000B          ;P20-P27の出力ラッチLow
MOV    PM2,    #00000000B          ;P20-P27(ANI0-ANI7)を入力ポートに設定

;-----
;   ポート3の設定
;-----
MOV    P3,     #00000000B          ;P30-P31の出力ラッチLow
MOV    PM3,    #11111100B          ;P30-P31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   PM4,   #00000000B           ;P40-P47を出力ポートに設定

;-----
;   ポート5の設定
;-----
MOV   P5,    #00000111B           ;P50-P52の出力ラッチHigh , P53-P57の出力ラッチLow
MOV   PM5,   #00000000B           ;P50-P57を出力ポートに設定

;-----
;   ポート6の設定
;-----
MOV   P6,    #00000000B           ;P60-P67の出力ラッチLow
MOV   PM6,   #00000000B           ;P60-P67を出力ポートに設定

;-----
;   ポート7の設定
;-----
MOV   P7,    #00000000B           ;P70-P77の出力ラッチLow
MOV   PM7,   #00000011B           ;P70-P71を入力 , P72-P77を出力ポートに設定
MOV   PU7,   #00000011B           ;P70, P71に内蔵プルアップ抵抗を使用

;-----
;   ポート8の設定
;-----
MOV   P8,    #00000000B           ;P80-P87の出力ラッチLow
MOV   PM8,   #00000000B           ;P80-P87を出力ポートに設定

;-----
;   ポート11の設定
;-----
MOV   P11,   #00000000B           ;P110-P111の出力ラッチLow
MOV   PM11,  #11111100B           ;P110-P111を出力ポートに設定

;-----
;   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

```

```

;-----
;   ポート13の設定
;-----
MOV   P13,   #00000000B           ;P131の出力ラッチLow
MOV   PM13,  #11111100B           ;P131を出力ポートに設定

;-----
;   ポート14の設定
;-----
MOV   P14,   #00000000B           ;P140-P145の出力ラッチLow
MOV   PM14,  #11000000B           ;P140-P145を出力ポートに設定

;-----
;   ポート15の設定
;-----
MOV   P15,   #00000000B           ;P150-P157の出力ラッチLow
MOV   PM15,  #00000000B           ;P150-P157を出力ポートに設定

EI                                     ;割り込み許可

MOV   ES,    #HIGHW LEDDATA       ;ラベルLEDDATAの上位アドレスを設定
MOVW  HL,    #LOWW LEDDATA        ;ラベルLEDDATAの下位アドレスを設定

;*****
;
;   メイン処理
;
;*****
MAIN_LOOP:
MOV   A,     P7                   ;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   B,     A                   ;入力結果を下位8ビットにセット
MOV   A,     ES:[HL+B]            ;テーブルから表示データ読み出し
MOV   P5,    A                   ;LED出力
BR    $MAIN_LOOP                 ;MAIN_LOOPへ

end

```

main.c (C言語版)

/\*\*\*\*\*\*

NEC Electronics      78K0R/KG3シリーズ

\*\*\*\*\*

78K0R/KG3シリーズ      サンプル・プログラム

\*\*\*\*\*

LED点灯のスイッチ制御

\*\*\*\*\*

**【履歴】**

2007.12.--      新規作成

\*\*\*\*\*

**【概要】**

このサンプル・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、78K0R/Kx3マイクロコントローラの基本的な使用する周辺の初期設定を行います。また、使用する周辺の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。

<使用する周辺の初期設定の主な設定内容>

- ・ 割り込みの禁止
- ・ CPU / 周辺ハードウェア・クロックを高速内蔵発振クロック動作の2MHzに設定
- ・ ポート・モードの設定 (入力ポート / 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)
- ・ 割り込みの許可

<メイン処理の主な内容>

- ・ スイッチ入力を検出
- ・ LED点灯

<スイッチ入力とLED点灯>

+							+			
	スイッチ1		スイッチ2		LED1		LED2		LED3	
	(P70)		(P71)		(P50)		(P51)		(P52)	
	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：P70, P71

出力ポート：P50-52, P53

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 DI /\* DI命令を記述可能にする \*/

\*\*\*\*\*

使用する周辺の初期設定

\*\*\*\*\* /

void hdwinit(void)

{

DI(); /\* 割り込み禁止 \*/

/\*-----

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。

```

-----*/
CMC = 0b00000000;          /* クロック動作モード */
/*|+++++----- AMPH: 2MHz fMX 10MHz */
/*|+++----- <000> */
/*|+----- OSCSELS: P123/P124端子を入力ポート・モード */
/*|+----- <0> */
/*++----- EXCLK/OSCSEL: 入力ポート・モード */

CSC = 0b11000000;          /* クロック動作ステータス制御 */
/*|+++++----- HIOSTOP: 高速内蔵発振回路動作 */
/*|+++++----- <00000> */
/*|+----- XTSTOP: XT1発振回路停止 */
/*+----- MSTOP: X1発振回路停止 */

OSMC = 0b00000000;          /* 動作スピード・モード */
/*|+++++----- FSEL: 10MHz以下の周波数で動作 */
/*+++++----- <00000> */

CKC = 0b00001010;          /* クロック選択 */
/*|+++++----- MDIV2-0: CPU/周辺ハードウェア・クロック(fCLK)=fIH/2^2 */
/*|+++----- <1> */
/*|+----- MCM0: 高速内蔵発振クロック(fIH) */
/*|+----- <R> */
/*|+----- CSS: メイン・システム・クロック(fMAIN)=fCLK */
/*+----- <R> */

/*-----
ポート0の設定
-----*/
P0 = 0b00000000;          /* P00-P06の出力ラッチLow */
PM0 = 0b10000000;          /* P00-P06を出力ポートに設定 */

/*-----
ポート1の設定
-----*/
P1 = 0b00000000;          /* P10-P17の出力ラッチLow */
PM1 = 0b00000000;          /* P10-P17を出力ポートに設定 */

/*-----
ポート2の設定
-----*/
P2 = 0b00000000;          /* P20-P27の出力ラッチLow */
PM2 = 0b00000000;          /* P20-P27(ANI0-ANI7)を入力ポートに設定 */

```



```

/*-----
   ポート3の設定
-----*/
P3 = 0b00000000;      /* P30-P31の出力ラッチLow */
PM3 = 0b11111100;    /* P30-P31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 */
PM4 = 0b00000000;    /* P40-P47を出力ポートに設定 */

/*-----
   ポート5の設定
-----*/
P5 = 0b00000111;      /* P50-P52の出力ラッチHigh, P53-P57の出力ラッチLow */
PM5 = 0b00000000;    /* P50-P57を出力ポートに設定 */

/*-----
   ポート6の設定
-----*/
P6 = 0b00000000;      /* P60-P67の出力ラッチLow */
PM6 = 0b00000000;    /* P60-P67を出力ポートに設定 */

/*-----
   ポート7の設定
-----*/
P7 = 0b00000000;      /* P70-P77の出力ラッチLow */
PM7 = 0b00000011;    /* P70-P71を入力, P72-P77を出力ポートに設定 */
PU7 = 0b00000011;    /* P70, P71に内蔵プルアップ抵抗を使用 */

/*-----
   ポート8の設定
-----*/
P8 = 0b00000000;      /* P80-P87の出力ラッチLow */
PM8 = 0b00000000;    /* P80-P87を出力ポートに設定 */

/*-----
   ポート11の設定
-----*/
P11 = 0b00000000;     /* P110-P111の出力ラッチLow */
PM11 = 0b11111100;   /* P110-P111を出力ポートに設定 */

```

```

/*-----
   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 */

/*-----
   ポート13の設定
-----*/

P13 = 0b00000000;      /* P131の出力ラッチLow */
PM13 = 0b11111110;    /* P131を出力ポートに設定 */

/*-----
   ポート14の設定
-----*/

P14 = 0b00000000;      /* P140-P145の出力ラッチLow */
PM14 = 0b11000000;    /* P140-P145を出力ポートに設定 */

/*-----
   ポート15の設定
-----*/

P15 = 0b00000000;      /* P150-P157の出力ラッチLow */
PM15 = 0b00000000;    /* P150-P157を出力ポートに設定 */

EI();                  /* 割り込み許可 */

}

/*****

   メイン処理

*****/

void main(void)
{

    const unsigned char OUTDATA[]=      /* 表示データパターン用配列 */
    {

        0b00000011
        ,0b00000101
        ,0b00000110
        ,0b00000111
    }
}

```

```
};

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

while(1){
    INDATA = P7 & 0b00000011; /* 有効スイッチ情報取得 */
    P5 = OUTDATA[INDATA];     /* テーブルから表示データを読み出して表示 */
}
}
```

## 付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	December 2008	-	-

## 【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](mailto: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。

---