

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。

お客様各位

ZUD-CC-09-0101-1

2009年7月9日

アプリケーション・ノート

NECエレクトロニクス株式会社
マイクロコンピュータ事業部
製品ソリューショングループ
グループ マネージャー

大場 浩司

(担当：西浦 真平)



78K0R/Kx3-L

サンプル・プログラム キー・リターン編

この資料は、サンプル・プログラムの動作概要や使用方法、およびキー・リターン機能を使用する際の設定方法や活用方法を説明したものです。サンプル・プログラムでは、4×4のキー・マトリクスよりキー・リターン割り込みによる全ポートのスキャンを行い、押下が確定ボタンの番号を7セグLEDに表示します。

対象デバイス

- 78K0R/KC3-L(44pin)
 マイクロコントローラ
- 78K0R/KC3-L(48pin)
 マイクロコントローラ
- 78K0R/KD3-Lマイクロコントローラ
- 78K0R/KE3-Lマイクロコントローラ
- 78K0R/KF3-Lマイクロコントローラ
- 78K0R/KG3-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	・・・5
2.1	回路イメージ	・・・5
2.2	周辺ハードウェア	・・・5
第3章	ソフトウェアについて	・・・6
3.1	ファイル構成	・・・6
3.2	使用する内蔵周辺機能	・・・7
3.3	キー・リターン機能の設定と動作概要	・・・7
3.4	フロー・チャート	・・・8
第4章	設定方法について	・・・11
4.1	キー・リターンの設定	・・・11
4.2	割り込みの設定	・・・11
4.3	キー・リターンの設定レジスタ	・・・12
4.4	キー・リターンの設定概要	・・・15
4.5	割り込みの設定概要	・・・17
第5章	関連資料	・・・18
5.1	ダウンロードファイルの解説	・・・18
5.2	サンプル・プログラムのHEXファイル生成	・・・19
5.3	開発環境のダウンロード、インストール	・・・21
第6章	関連資料	・・・22
付録A	プログラム・リスト	・・・23
・	アセンブリ言語	・・・23
・	C言語	・・・44
付録B	改版履歴	・・・66

・本資料に記載されている内容は2009年1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。

第1章 概 要

このサンプル・プログラムは、キー・リターン機能を用いて4×4のキー・マトリクス・スキャンを行います。

キー・リターン信号により、全ポートをスキャンし押下されたボタンの番号を調査します。INTKRの検出から30ms経過後に同信号がOFFであった場合は、チャタリングと判断します。押下が確定したボタンの番号は、2桁の7セグLEDに出力します。2ボタンの同時押下時はFF表示、キー入力なし時は00表示を行います。

(1) 初期設定の主な内容

< オプション・バイトでの設定 >

ウォッチドッグ・タイマのカウンタ動作制御（動作停止）

高速内蔵発振回路の周波数（8MHz）

LVI動作設定

オンチップ・デバッグ許可

< リセット解除後の初期化処理での設定 >

入出力ポートの設定^注

CPU / 周辺ハードウェア・クロックを8MHzに設定

各周辺ハードウェア・マクロの使用可否の設定（TAUSの初期設定）

- ・ TAUS / チャンネル0をインターバル・タイマ・モードに設定
- ・ キー・リターン機能の設定
- ・ 割り込みの設定（INTKRの割り込みを許可）

注：未使用端子のポート設定は、各デバイスによって設定が異なります。以下の各デバイスのポートをLow出力オープンに設定してください。ポートの設定については、サンプル・プログラム初期設定編アプリケーション・ノートの「4.4 ポートの設定レジスタ」を参照して下さい。

・ KC3-L(44pin)

未使用端子：P20-27，P32，P40-41，P50-52，P74-75，P80-83，P120-124，P150-151

・ KC3-L(48pin)

未使用端子：P20-27，P32，P40-41，P50-51，P60-61，P74-P75，P80-83，P120-124
P140，P150-152

・ KD3-L

未使用端子：P00-01，P20-27，P32，P40-41，P50-52，P60-61，P74-77，P80-83，P120-124
P140，P150-152

・ KE3-L

未使用端子：P00-01，P14-17，P20-27，P32-33，P40-43，P50-53，P60-61，P74-77，P80-83
P120-124，P140-141，P150-153

・KF3-L

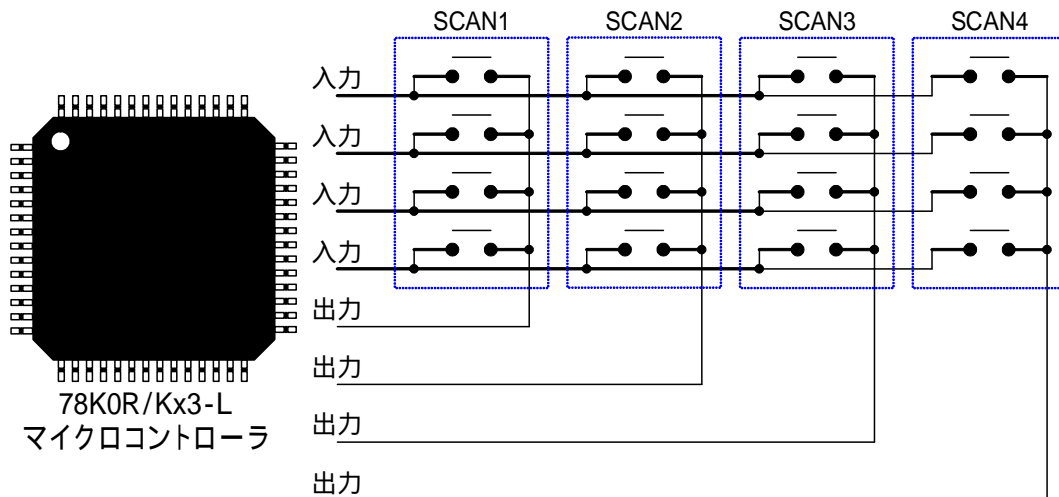
未使用端子：P02-06 , P14-17 , P20-27 , P40-47 , P50-55 , P60-67 , P74-77 , P90-91 , P110-111
P120-124 , P130 , P140 , P142-144 , P150-153

・KG3-L

未使用端子：P00-06 , P14-17 , P20-27 , P40-47 , P50-57 , P60-67 , P74-77 , P80-87 , P91
P110-111 , P120-124 , P130-131 , P140-145 , P150-157

(2) メインループ以降の内容

初期設定完了後は、キー・リターン割り込みによる、4×4キー・マトリクス・スキャン動作を行います。キー・リターン割り込みから30ms経過後に同信号がOFFであった場合は、チャタリングと判断します。押下が確定したボタンの番号は、2桁の7セグLEDに出力します。複数ボタンの同時押下はFF表示を行います。



注意：キー・スキャンを連続して行う場合、プルアップ抵抗による立ち上がり時間、立ち下り時間に遅延が生じ、誤動作する可能性があります。回避策として、スキャン・ラインの出力ラッチに0をセットしておき、ポート・モード・レジスタを制御することで、CMOS出力ポートをN-chオープン・ドレイン出力と同様にすることで、遅延による誤動作を防ぐことができます。

参考：http://www.necel.com/ja/faq/mi_com/_com_ksw.html

(3) キー・リターンの種類

KR0-KR7の8ビット^注で使用する事が可能です。

注意：製品によっては異なります(以下参照)。

デバイス	78K0R/KC3-L	78K0R/KD3-L	78K0R/KE3-L	78K0R/KF3-L	78K0R/KG3-L
キー・リターン	KR0-KR5	KR0-KR5	KR0-K07	KR0-KR7	KR0-K07

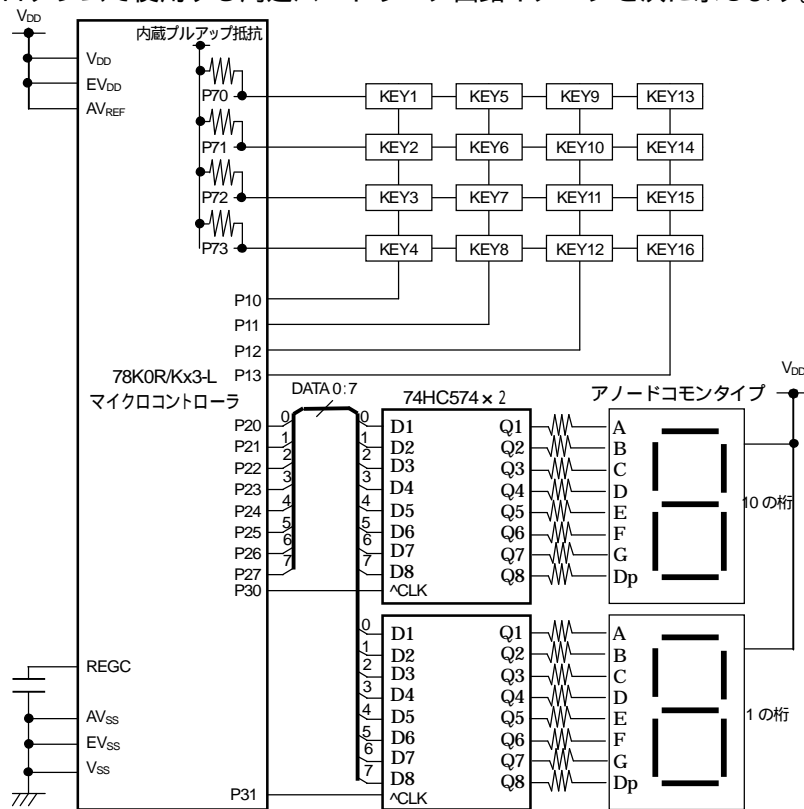
尚、デバイス使用上の注意事項については、[78K0R/Kx3-L](#)ユーザズ・マニュアルを参照してください。

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回路イメージおよび周辺ハードウェアを説明します。

2.1 回路イメージ

サンプル・プログラムで使用する周辺ハードウェア回路イメージを次に示します。



注意：この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電流など電気的特性を満たすように設計してください。（P121～P124は入力専用ポートになりますから個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい）

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。

- 外部入力端子（P70-P73）
キー・リターン入力用として、外部入力端子を4ポート使用します。
- 外部出力端子（P10-P13）
キー・スキャン信号出力用として、外部出力端子を4ポート使用します。
- 外部出力端子（P20-P27）
LEDデータ出力用として、外部出力端子を8ポート使用します。
- 外部出力端子（P30-P31）
LEDデータのラッチ信号として、外部出力端子を2ポート使用します。



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・プログラムの動作概要、およびフロー・チャートを説明します。


3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。



【C言語版】


ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
Kx3-L_KR.c	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		
OP.asm	オプション・バイトの指定ファイル		
78K0RKx3-L_sample_program.prw	統合開発環境 PM+用ワーク・スペース・ファイル	-	
78K0RKx3-L_sample_program.prj	統合開発環境 PM+用設定データ	-	


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

【アセンブリ言語版】

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
Kx3-L_KR.asm	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		
OP.asm	オプション・バイトの指定ファイル		
78K0RKx3-L_sample_program.prw	統合開発環境 PM+用ワーク・スペース・ファイル	-	
78K0RKx3-L_sample_program.prj	統合開発環境 PM+用設定データ	-	

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

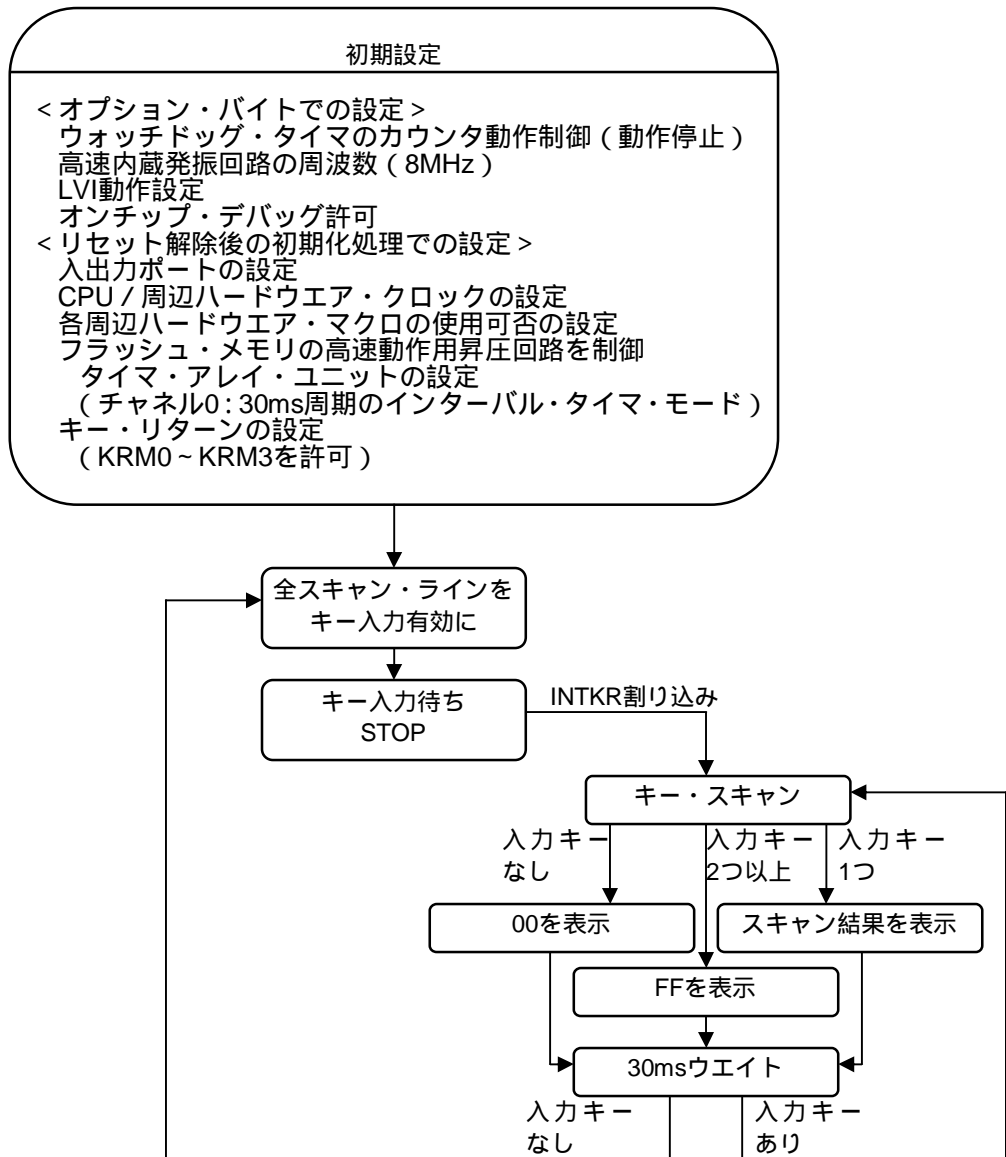
3.2 使用する内蔵周辺機能

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。

- ・キー・リターン : P70, P71, P72, P73
- ・キー・スキャン : P10, P11, P12, P13
- ・タイマ : TAUSチャンネル0
- ・7セグLED : P20-27 (データ), P30 (10の桁ラッチ), P31 (1の桁ラッチ)

3.3 キー・リターン, キー・スキャンの設定と動作概要

このサンプル・プログラムでは、キー・リターンとタイマの二つの割り込みフラグを利用しています。最初はキー・スキャンを行わず、全スキャン・ラインをアクティブとして、キーのどれかが押されるまで、キー・リターン割り込みを用いて待ちます。どれかが押されてキー・リターン割り込みが発生したら、キー・スキャンを開始します。検出から30ms経過後に信号がOFFであった場合は、チャタリングと判断し、以降の処理は行いません。押下が確定したスイッチの番号を、2桁の7セグLEDに出力します。この時、複数スイッチの同時押下状態であった場合は、7セグLEDにFFを表示します。



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。

(1)全体図フロー



注1. オプション・バイトについては、サンプル・プログラム初期設定アプリケーション・ノートの “4.6 オプション・バイトの設定概要” を参照して下さい。

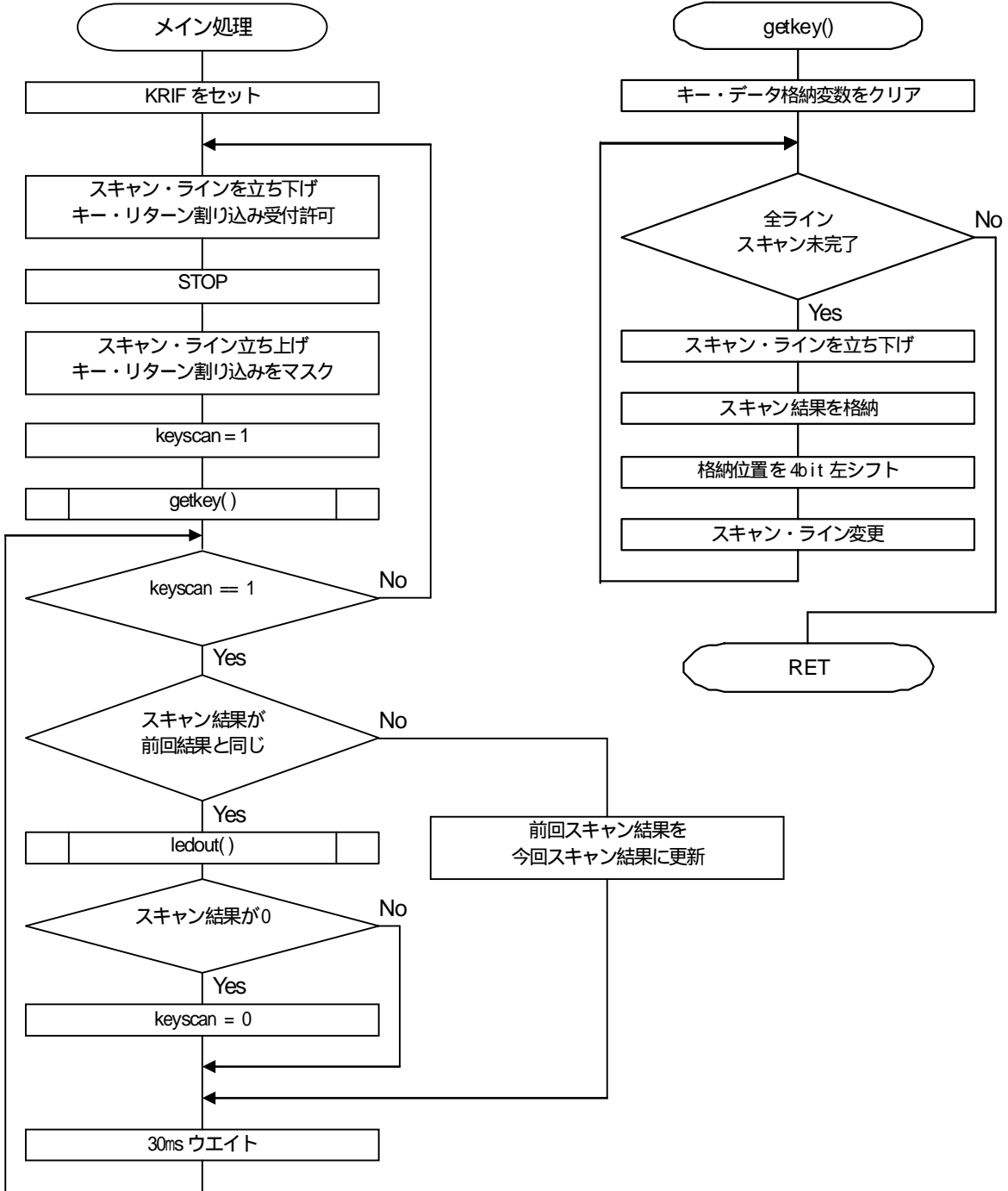
2. 入出力ポートの設定 (f_ini_port()) については、サンプル・プログラム初期設定編アプリケーション・ノートの “3.4 フロー・チャート” を参照して下さい。

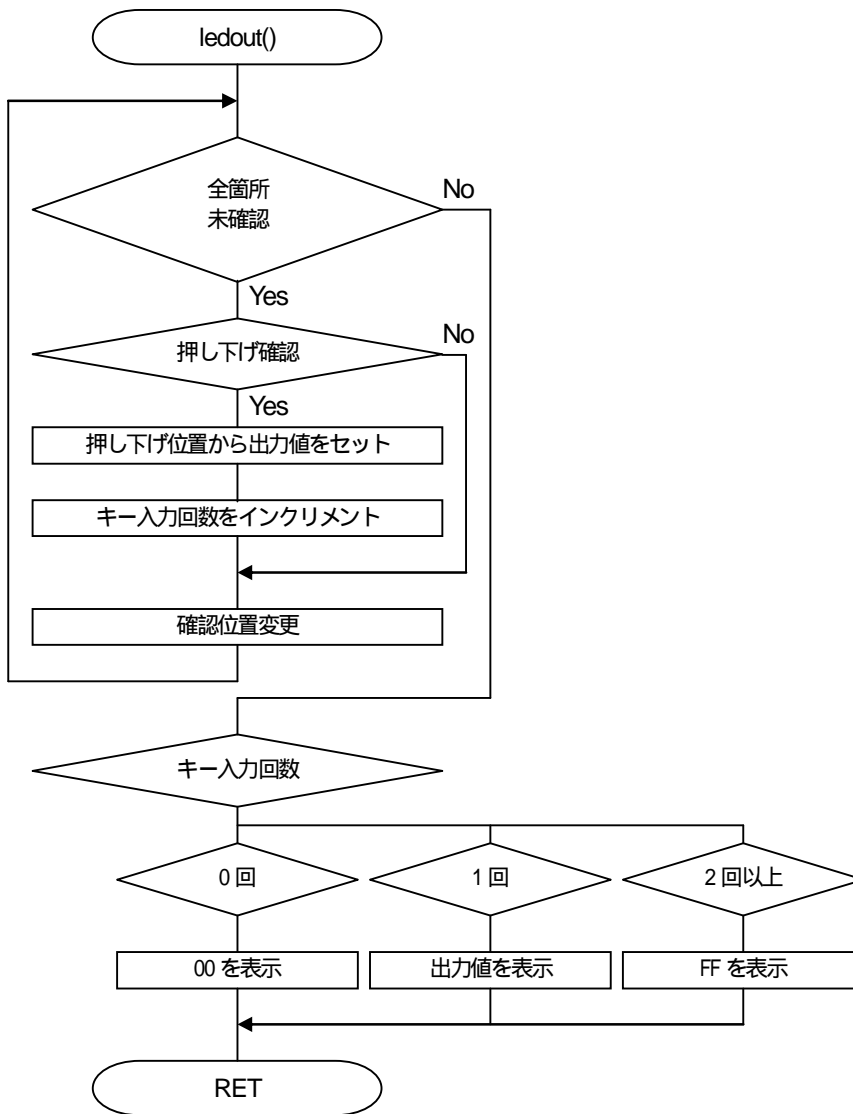
3. 低電圧検出設定 (f_ini_lvi()) については、サンプル・プログラム低電圧検出回路編アプリケーション・ノートの “3.4 フロー・チャート” を参照して下さい。

4. CPU初期設定(f_ini_cpu())については、サンプル・プログラム初期設定編アプリケーション・ノートの “3.4 フロー・チャート” を参照して下さい。

5. タイマ・アレイ・ユニット設定については、サンプル・プログラムインターバル・タイマ編アプリケーション・ノートの “3.4 フロー・チャート” を参照して下さい。

(2)メイン処理





第4章 設定方法について

この章では、キー・リターン割込みを使用する際の設定について説明します。

その他の初期設定については、78K0R/Kx3-L サンプル・プログラム(初期設定編)アプリケーション・ノートを参照してください。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアル([78K0R/Kx3-L](#))を参照してください。

アセンブラ命令については、[78K0R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 キー・リターンの設定

キー・リターン機能を使用する際に設定するレジスタは、主に次の種類があります。

キー・リターン設定で使用するレジスタ

- ・キー・リターン・モード・レジスタ (KRM)

4.2 割り込みの設定

以下にアプリケーションで許可する割り込みを設定するレジスタを示します。

割り込み設定で使用するレジスタ

- ・割り込み要求フラグ・レジスタ (IF1H)
- ・割り込みマスク・フラグ・レジスタ (MK1H)

4.3 キー・リターンの設定レジスタ

(1)キー・リターン・モード・レジスタ (KRM)

キー・リターン割り込み信号の検出有効 / 無効を設定するレジスタです。

略号： KRM

7	6	5	4	3	2	1	0
KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	キー割り込みモードの制御
0	キー割り込み信号を検出しない
1	キー割り込み信号を検出する

注意

KRM0-KRM7のうち使用するビットに1を設定する場合、それに対応するプルアップ抵抗レジスタ7 (PU7) のビット0-7 (PU70-PU77) に1を設定してください。

KRMを変更すると、割り込み要求フラグがセットされる場合があります。したがって、あらかじめ割り込みを禁止してからKRMレジスタを変更し、割り込み要求フラグをクリアしてから、割り込みを許可してください。

キー割り込みモードで使用していないビットは通常ポートとして使用可能です。

(2)ポート・モード・レジスタ7 (PM7)

ポートの入力 / 出力を1ビット単位で設定するレジスタです。

PM7は、それぞれ1ビット・メモリ操作命令または8ビット・メモリ操作命令で設定します。

リセット信号の発生により、FFHになります。

略号： PM7

7	6	5	4	3	2	1	0
PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n端子の入出力モードの選択
0	出力モード (出力バッファ・オン)
1	入力モード (出力バッファ・オフ)

(3)プルアップ抵抗オプション・レジスタ7 (PU7)

内蔵プルアップ抵抗を使用するか、しないかを設定するレジスタです。

PU7は、1ビット・メモリ操作命令または8ビット・メモリ操作命令で設定します。

リセット信号の発生に、00Hになります。

略号： PU7

7	6	5	4	3	2	1	0
PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70

PU7n	P7n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

4.3 割り込みの設定レジスタ

(1)割り込み要求フラグ・レジスタ(IF1)

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

割り込みが受け付けられた場合、まず割り込み要求フラグが自動的にクリアされてから割り込みルーチンに入ります。

IF1Lは、1ビット・メモリ操作命令または8ビット・メモリ操作命令で設定します。

IF1LとIF1Hをあわせて16ビット・レジスタIF1として使用するときは、16ビット・メモリ操作命令で設定します。

リセット信号の発生により、00Hになります。

略号： IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	SREIF1	SRIF1	STIF1 CSIIF10 IICIF10

略号： IF1H

7	6	5	4	3	2	1	0
TMIF04	MDIF	0	0	KRIF	RTCIIF	RTCIF	ADIF

xxIFx	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

(2)割り込みマスク・フラグ・レジスタ(MK1)

割り込みマスク・フラグは、対応するマスカブル割り込み処理の許可/禁止を設定するフラグです。

MK1Lは、1ビット・メモリ操作命令または8ビット・メモリ操作命令で設定します。

MK1LとMK1Hをあわせて16ビット・レジスタMK1として使用する場合は、16ビット・メモリ操作命令で設定します。

リセット信号の発生により、FFHになります。

備考 このレジスタへの書き込み命令を行った場合、命令実行クロック数が2クロック長くなります。

略号： MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	SREMK1	SRMK1	STMK1 CSIMK10 IICMK10

略号： MK1H

7	6	5	4	3	2	1	0
TMMK04	MDMK	0	0	KRMK	RTCIMK	RTCMK	ADMK

xxMKx	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

4.4 キー・リターンの設定概要

キー・リターンを用いる際の各レジスタ設定の流れを以下に示します。

ポートの設定

ポート・モード・レジスタ7 (PM7)
 プルアップ抵抗オプション・レジスタ (PU7)
 キー・リターンとして使用するビットを入力モードとし、内蔵プルアップ抵抗を接続します。

略号 : PU7

7	6	5	4	3	2	1	0
PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70
0	0	0	0	1	1	1	1

PU7n	P7n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

略号 : PM7

7	6	5	4	3	2	1	0
PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
0	0	0	0	1	1	1	1

PM7n	P7n端子の入出力モードの選択
0	出力モード (出力バッファ・オン)
1	入力モード (出力バッファ・オフ)

【使用例】

キー・リターンとして、P70, P71, P72, P73を使用する場合

アセンブリ言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

```
MOV    PU7    ,    #00001111B
MOV    PM7    ,    #00001111B
```

C言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

```
PU7 = 0b00001111;
PM7 = 0b00001111;
```

キー・リターン使用設定

キー・リターン・モード・レジスタ (KRM)
 キー・リターン信号に対するキー・リターン割り込みの発生を関連付け
 します。

略号 : KRM

	7	6	5	4	3	2	1	0
	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
	0	0	0	0	1	1	1	1

KRMx	キー割り込みモードの制御
0	キー割り込み信号を検出しない
1	キー割り込み信号を検出する

【使用例】

キー・リターンとして、P70、P71、P72、P73を使用する場合

アセンブリ言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

MOV KRM , #00001111B

C言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

KRM = 0b00001111;

4.5 割り込みの設定概要

INTKR 割り込み設定

割り込み要求フラグ・レジスタ (IF1H)
INTKR 割り込み要因フラグをクリアします
割り込みマスク・フラグ・レジスタ (MK1H)
INTKR 割り込みマスク・フラグを解除します

略号 : IF1H

7	6	5	4	3	2	1	0
TMIF04	MDIF	0	0	KRIF	RTCIF	RTCIF	ADIF
x	x	0	0	0	x	x	x

KRIF	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

略号 : MK1H

7	6	5	4	3	2	1	0
TMMK04	MDMK	0	0	KRMK	RTCIMK	RTCMK	ADMK
x	x	0	0	0	x	x	x

KRIF	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

【記述例】

キー・リターン割り込みを使用する場合

アセンブリ言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

```
CLR1    KRIF
CLR1    KRMK
```

C言語の場合 (78K0R/KE3-Lマイクロコントローラ使用時)

```
KRIF = 0;
KRMK = 0;
```





第5章 PM+を用いたHEXファイルの生成

この章では、PM+とダウンロードしたC言語用のファイルを用い、サンプル・プログラムからHEXファイルに生成する方法を説明します。





5.1 ダウンロードファイルの解説

ダウンロードした各種ファイルとの説明をします。


(1) C言語版

	ファイル名	内容
	78K0RKx3-L_sample_program.prw	統合開発環境 PM+用ワーク・スペース・ファイル
	78K0RKx3-L_sample_program.prj	統合開発環境 PM+用設定データ
	Kx3-L_KR.c	キー・リターンのC言語ソース・ファイル
	OP.asm	オプション・バイトのアセンブリ言語ソース・ファイル

(2) アセンブラ版

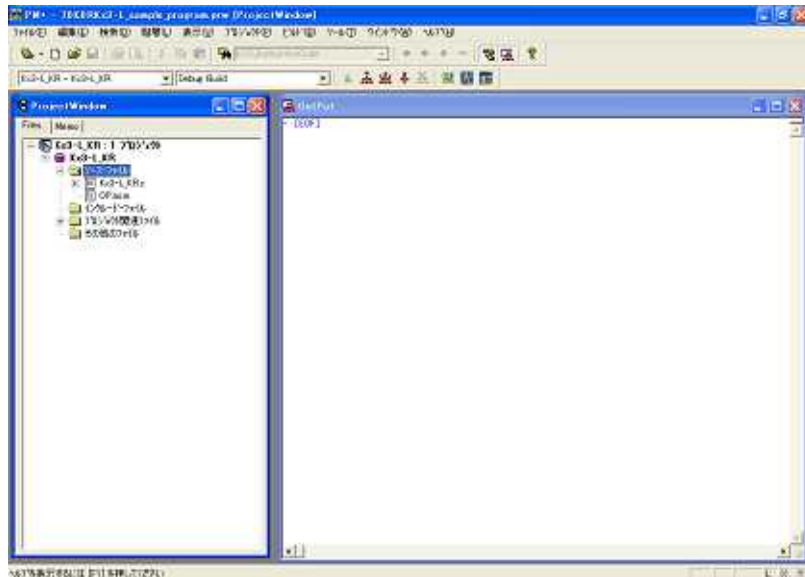
	ファイル名	内容
	78K0RKx3-L_sample_program.prw	統合開発環境 PM+用ワーク・スペース・ファイル
	78K0RKx3-L_sample_program.prj	統合開発環境 PM+用設定データ
	Kx3-L_KR.asm	キー・リターンのアセンブリ言語ソース・ファイル
	OP.asm	オプション・バイトのアセンブリ言語ソース・ファイル

5.2 サンプル・プログラムのHEXファイル生成（ビルド）

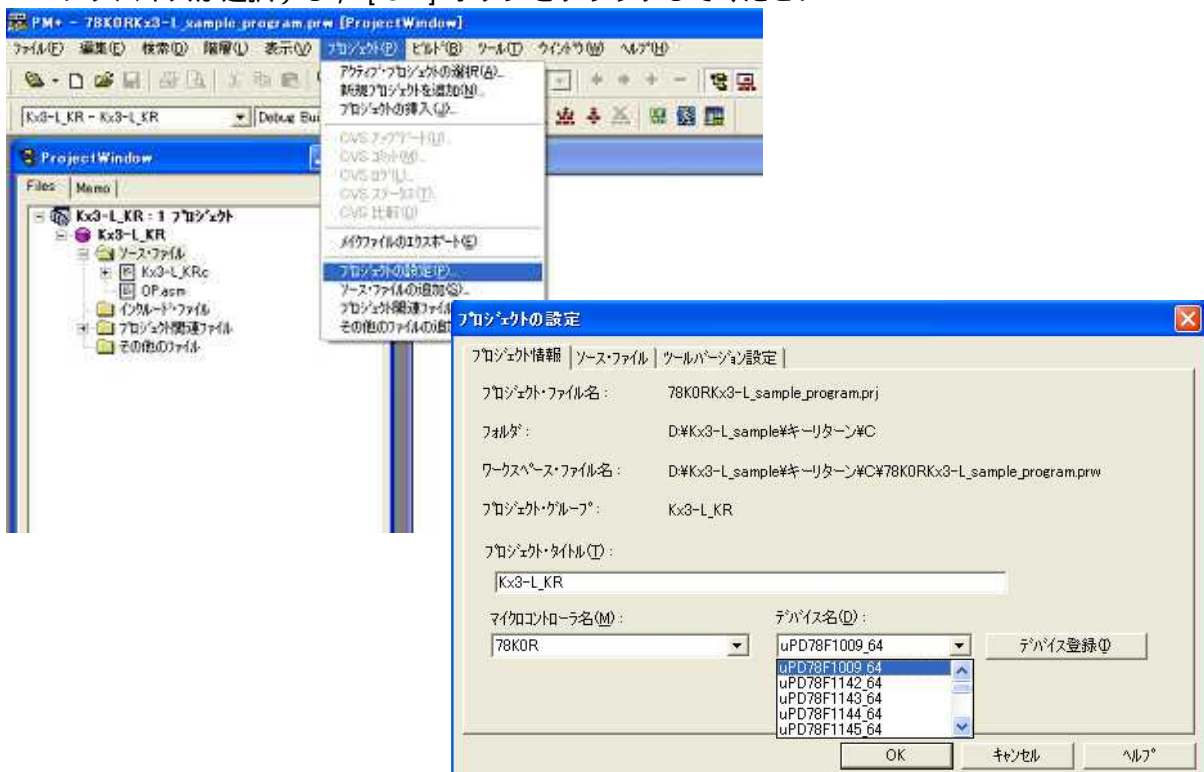
サンプル・プログラムからHEXファイルに生成するために、PM+を用いてサンプル・プログラムをビルドする必要があります。ここでは  でダウンロードしたC言語版のファイルを用いて、統合開発環境PM+にてビルドしてから、HEXファイルを生成するまでの動作の一例を説明します。

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

- (1) ダウンロードしたファイルを解凍し、「78K0RKx3-L_sample_program.prw」をダブルクリックしてください。ワークスペースが開き、その中にソース・ファイルが自動的に読み込まれます。




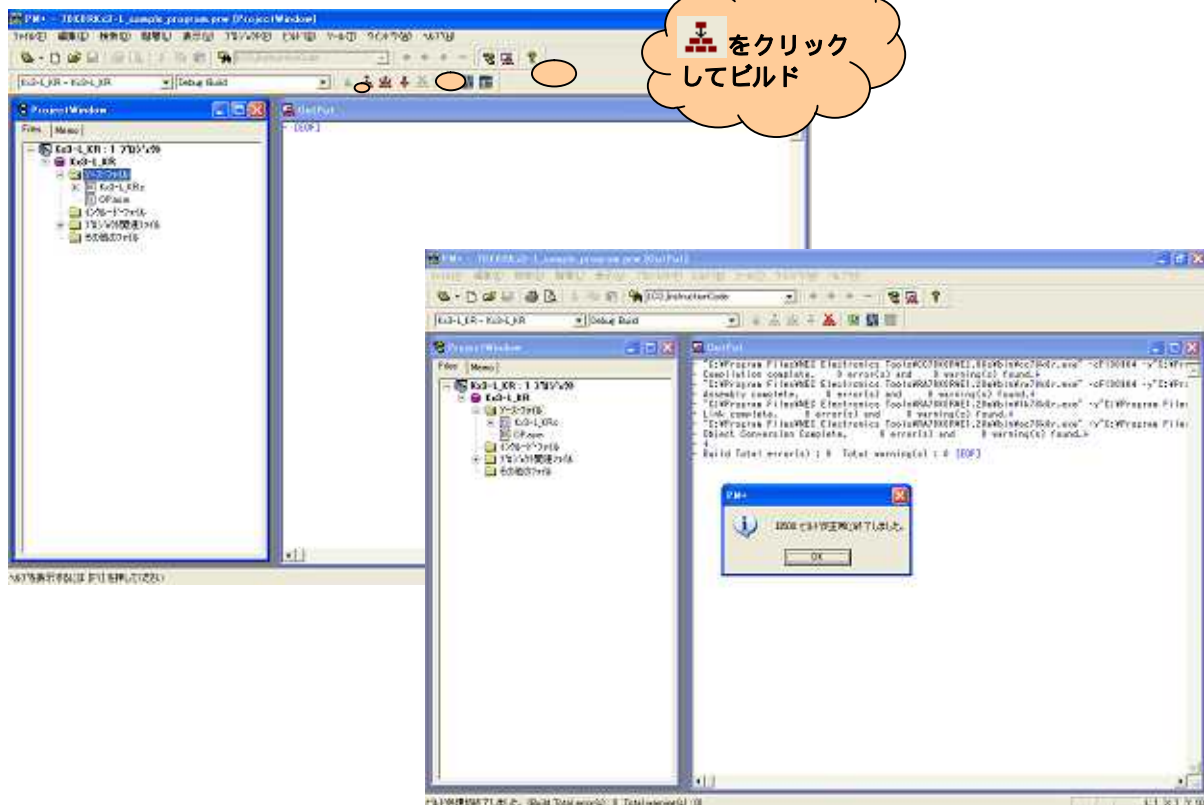
- (2) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が表示されたら、使用するデバイス名を選択（デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択）し、[OK] ボタンをクリックしてください



- (3) [ツール] [オブジェクトコンバータオプションの設定] を選択してください。[オブジェクトコンバータオプションの設定] 画面が表示されたら, [出力1] タグページが表示されているのを確認し, その中の出力ファイル名に任意のファイル名+拡張子 (.HEX) を入力し, [OK] をクリックします。



- (4) PM+画面の中央上付近の  をクリックしてください。自動でビルドが実行され, ソース・ファイルの「Kx3-L_KR.c」と「OP.asm」からHEXファイルが生成され, 「I3500:ビルドが正常に終了しました」というメッセージ画面が表示されます。
[OK] をクリックしてビルドを終了します。



5.3 開発環境のダウンロード，インストール

78K0R/Kx3-Lマイクロコントローラの開発ツールのフリーツールは，次のサイトより入手可能です。

→<http://www.necel.com/micro/ja/freesoft/78k0r/index.html>

「RA78K0R」「CC78K0R」「78K0R/Kx3-L用デバイス・ファイル」の3ファイルをダウンロードし，インストールすることで，サンプル・プログラムの動作確認が可能となります。

ダウンロード，インストールは，上記サイトの画面および説明に従って，行ってください。

- 備考
1. PM+は，RA78K0Rに同封されています。
 2. ダウンロード後，登録したEメール・アドレスに，RA78K0R，CC78K0RのプロダクトIDが送付されます。このプロダクトIDは，各ツールのインストール時に必要となります。

第6章 関連資料

資料名		和文 / 英文
78K0R/Kx3-L ユーザーズ・マニュアル		PDF
78K0Rシリーズ 命令編 ユーザーズ・マニュアル		PDF
RA78K0R 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
CC78K0R Cコンパイラ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		PDF
SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル		PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0R/KE3-Lマイクロコントローラ用のソース・プログラムを次に示します。

```

Kx3-L_KR.asm (アセンブリ言語版)
;*****
;
;
;   NEC Electronics   78K0R/KE3-Lシリーズ
;
;*****
;   78K0R/KE3-Lシリーズ サンプル・プログラム(キー割り込み機能)
;*****
;   キー・リターン
;*****
; 【履歴】
;   2009.01.-- 新規作成
;*****
;
; 【概要】
;
; 本サンプル・プログラムはキー割り込み機能の使用例を示すものです。4×4のキー・マ
; トリクスよりキー割り込みの発生後、キー・スキャンを行い、押下が確定したキーの番
; 号を7セグLEDに表示します。キー割り込みが発生したら、キー・スキャンを開始します。
; キー・スキャンから30ms経過後にキーの押下状態に変化があった場合は、チャタリング
; とし、キー・オフの状態としてキー番号は00として7segLEDに出力します。キーの押下
; が確定した場合は、01から16のキー番号をの7segLEDに出力します。このとき、複数の
; キーの同時押下があった場合は、キー無効の状態としてキー番号はFFとして出力します。
;
;
; <初期設定の主な内容>
; (オプション・バイトでの設定)
;   ・ウォッチドッグ・タイマの動作禁止
;   ・高速内蔵発振回路に8MHz/20MHzを選択
;   ・LVIデフォルト・スタート機能動作
;   ・オンチップ・デバッグを動作許可に設定
; (リセット解除後の初期化処理での設定)
;   ・入出力ポートの設定
;   ・低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
;   ・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8MHzに設定

```

```

;   ・X1/XT1発振回路の停止
;   ・タイマ・アレイ・ユニットの設定
;   ・キー割り込み機能の設定
;
;
; <タイマ・アレイ・ユニットの設定>
;   ・インターバル・タイマ・モードに設定
;   ・動作クロック周波数 = fCLK/2^3 (1MHz)
;   ・インターバル = 30ms (1[us/clock] × 30000[count] = 30[ms])
;   ・INTTMO0割り込み処理禁止
;
;
; <キー割り込み機能の設定>
;   ・KR0, KR1, KR2, KR5のキー割り込み信号の検出設定
;   ・INTKR割り込み処理禁止
;
;
; <キー・マトリクス>
;
; +-----+-----+-----+-----+
; | KEY01 | KEY05 | KEY09 | KEY13 | KR0(P70)
; +-----+-----+-----+-----+
; | KEY02 | KEY06 | KEY10 | KEY14 | KR1(P71)
; +-----+-----+-----+-----+
; | KEY03 | KEY07 | KEY11 | KEY15 | KR2(P72)
; +-----+-----+-----+-----+
; | KEY04 | KEY08 | KEY12 | KEY16 | KR3(P73)
; +-----+-----+-----+-----+
;   KS3     KS2     KS1     KS0
;   (P10)   (P11)   (P12)   (P13)
;
;   KS : キー・スキャン・ライン
;   KR : キー・リターン・ライン
;
;
; <入出力ポートの設定>
;   入力ポート : P70-P73 . . . キー・リターン用
;   出力ポート : P10-P13 . . . キー・スキャン用
;                 P20-P27 . . . 7segLED用
;                 P30, P31 . . . 7segLEDラッチ信号用
;   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
;
;=====
TVECT1          CSEG AT 00000H
                DW  RESET_START          ;00000H   RESET入力,POC,LVI,WDT,TRAP
TVECT2          CSEG AT 00004H
                DW  IINIT                ;00004H   INTWDTI
                DW  IINIT                ;00006H   INTLVI
                DW  IINIT                ;00008H   INTP0
                DW  IINIT                ;0000AH  INTP1
                DW  IINIT                ;0000CH  INTP2
                DW  IINIT                ;0000EH  INTP3
                DW  IINIT                ;00010H  INTP4
                DW  IINIT                ;00012H  INTP5
TVECT3          CSEG AT 00016H
                DW  IINIT                ;00016H  INTSR3
                DW  IINIT                ;00018H  INTSRE3
                DW  IINIT                ;0001AH  INTDMA0
                DW  IINIT                ;0001CH  INTDMA1
                DW  IINIT                ;0001EH  INTST0/INTCSI00
                DW  IINIT                ;00020H  INTSR0/INTCSI01
                DW  IINIT                ;00022H  INTSRE0
                DW  IINIT                ;00024H  INTST1/INTCSI10/INTIIC10
                DW  IINIT                ;00026H  INTSR1
                DW  IINIT                ;00028H  INTSRE1
                DW  IINIT                ;0002AH  INTIICA
                DW  IINIT                ;0002CH  INTTM00
                DW  IINIT                ;0002EH  INTTM01
                DW  IINIT                ;00030H  INTTM02
                DW  IINIT                ;00032H  INTTM03
                DW  IINIT                ;00034H  INTAD
                DW  IINIT                ;00036H  INTRTC
                DW  IINIT                ;00038H  INTRTCI
                DW  IINIT                ;0003AH  INTKR
TVECT4          CSEG AT 00040H
                DW  IINIT                ;00040H  INTSRE2
                DW  IINIT                ;00042H  INTTM04
                DW  IINIT                ;00044H  INTTM05
                DW  IINIT                ;00046H  INTTM06

```

```
DW IINIT          ;00048H   INTTM07
DW IINIT          ;0004AH   INTP6
DW IINIT          ;0004CH   INTP7
```

```
;=====
;
;
;   スタック領域の確保
;
;=====
```

DSTK DSEG BASEP

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
```

```
;=====
;
;
;   ROMの定義
;
;=====
```

TMAIN CSEG UNIT

```
-----
;
;   LED表示用設定データ・テーブル
;
;-----
;
;   LED表示する数値に対応した設定データのテーブルです。
;   ロウ・アクティブ
;   消灯なら出力データを1，点灯なら出力データを0とします。
;-----
```

TLEDNUM:

```
    ;   h ----- h (P27)   aaaaa
    ;   |g ----- g (P26)   f     b
    ;   ||f ----- f (P25)  f     b
    ;   |||e ----- e (P24)  ggggg
    ;   ||||d ----- d (P23) e     c
    ;   |||||c ----- c (P22) e     c
    ;   |||||b ----- b (P21) ddddd  hh
    ;   |||||a ----- a (P20)          hh
    ;   |||||
DB   11000000B ;"0"
DB   11111001B ;"1"
DB   10100100B ;"2"
DB   10110000B ;"3"
```

```

DB 10011001B ;"4"
DB 10010010B ;"5"
DB 10000010B ;"6"
DB 11111000B ;"7"
DB 10000000B ;"8"
DB 10010000B ;"9"
DB 10001000B ;"A"
DB 10000011B ;"B"
DB 11000110B ;"C"
DB 10100001B ;"D"
DB 10000110B ;"E"
DB 10001110B ;"F"

```

XMAIN_CSEG UNIT

```

;*****
;
;
;   不要な割り込み要因による割り込み処理
;
;*****

```

INIT:

```

;   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

```

;-----
;   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

;-----
;   入出力ポートの設定
;-----
CALL !!SINIPOINT             ;出力に設定できるものは全て出力ポートに設定

;-----
;   低電圧検出
;-----
CALL !!SINILVI              ;2.7V以上の電源電圧を確保

;-----
;   クロック周波数の設定
;-----
CALL !!SINICKL              ;高速内蔵発振クロックを8MHzで動作

;-----
;   タイマ・アレイ・ユニットの設定
;-----
;タイマ・アレイ・ユニット初期設定
SET1 TAU0EN                 ;タイマ・アレイ・ユニットの入カクロック供給

MOV  TPSOL, #00000011B ;タイマ・クロック選択レジスタ0
;|||+---- PRS003-PRS000
;++++---- PRS013-PRS010
;           [動作クロック(CK00/CK01)の選択]
;           0000: fCLK
;           0001: fCLK/2
;           0010: fCLK/2^2
;           0011: fCLK/2^3
;           0100: fCLK/2^4
;           0101: fCLK/2^5
;           0110: fCLK/2^6
;           0111: fCLK/2^7
;           1000: fCLK/2^8
;           1001: fCLK/2^9
;           1010: fCLK/2^10
;           1011: fCLK/2^11
;           1100: fCLK/2^12

```

```

;          1101: fCLK/2^13
;          1110: fCLK/2^14
;          1111: fCLK/2^15

```

;チャンネル初期設定

MOVW AX, #0000000000000000B;タイマ・モード・レジスタ00

MOVW TMR00, AX

```

;|||||++++ MD003-MD000
;||||| [チャンネル0の動作モードの設定]
;||||| 0000: インターバル・タイマ・モード
;||||| (カウント開始時にタイマ割り込みを発生しない)
;||||| 0001: インターバル・タイマ・モード
;||||| (カウント開始時にタイマ割り込みを発生する)
;||||| 0100: キャプチャ・モード
;||||| (カウント開始時にタイマ割り込みを発生しない)
;||||| 0101: キャプチャ・モード
;||||| (カウント開始時にタイマ割り込みを発生する)
;||||| 0110: イベント・カウンタ・モード
;||||| 1000: ワンカウント・モード
;||||| (カウント動作中のスタート・トリガは無効とする。
;||||| その際に割り込みも発生しない。)
;||||| 1001: ワンカウント・モード
;||||| (カウント動作中のスタート・トリガを有効とする。
;||||| その際に割り込みも発生する。)
;||||| 1100: キャプチャ&ワンカウント・モード
;||||| 上記以外: 設定禁止
;|||||++----- 必ず0に設定
;|||||++----- CIS001-CIS000
;||||| [TI00端子の有効エッジ選択]
;||||| 00: 立ち下がリエッジ
;||||| 01: 立ち上がりエッジ
;||||| 10: 両エッジ(ロウ・レベル幅測定時)
;||||| 11: 両エッジ(ハイ・レベル幅測定時)
;|||||+++----- STS002-ST000
;||||| [チャンネル0のスタート・トリガ, キャプチャ・
;||||| トリガの設定]
;||||| 000: ソフトウエア・トリガ・スタートのみ有効
;||||| (他のトリガ要因を非選択にする)
;||||| 001: TI00端子入力の有効エッジを, スタート・トリガ,
;||||| キャプチャ・トリガの両方に使用
;||||| 010: TI00端子入力の両エッジを, スタート・トリガと
;||||| キャプチャ・トリガに分けて使用
;||||| 100: マスタ・チャンネルの割り込み信号を使用

```

```

;||||| (連動動作機能のスレーブ・チャンネル時)
;||||| 上記以外:設定禁止
;|||+----- MASER00
;||||| [チャンネル0の単体動作機能, 連動動作機能のスレーブ/
;||||| 連動動作機能のマスタの選択]
;||||| 0:単体動作機能, または連動動作機能でスレーブ・
;||||| チャンネルとして動作
;||||| 1:連動動作機能でマスタ・チャンネルとして動作
;|||+----- CCS00
;||| [チャンネル0のカウント・クロック(TCLK)の選択]
;||| 0:CKS00ビットで指定した動作クロックMCK
;||| 1:T100端子からの入力信号の有効エッジ/サブシステム・
;||| クロックの4分周(fSUB/4)
;|++----- 必ず0に設定
;+----- CKS00
; [チャンネル0の動作クロック(MCK)の選択]
; 0:TPS0レジスタで設定した動作クロックCK00
; 1:TPS0レジスタで設定した動作クロックCK01

MOVW TDR00, #30000-1 ;インターバルを30ms(1[us/cIk] × 30000[count])に設定

CLR1 TMIF00 ;INTTM00割り込み要求クリア
CLR1 TMMK00 ;INTTM00割り込み許可(スタンバイ解除用)

;-----
; キー割り込み機能の設定
;-----
MOV KRM, #00001111B ;キー・リターン・モード・レジスタ
;+++++----- KRM7-KRMO
; [キー割り込みモードの制御]
; 0:キー割り込み信号を検出しない
; 1:キー割り込み信号を検出する

CLR1 KRIF ;INTKR割り込み要求クリア
CLR1 KRMK ;INTKR割り込み許可(スタンバイ解除用)

;-----
; ESの設定
;-----
; ROMテーブルの上位アドレスをESに設定し, 参照できるようにします。
;-----
MOV ES, #HIGHW TLEDNUM ;LED表示用設定データ・テーブル

```



```

;-----
;   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

;*****
;
;   入出力ポートの初期化
;
;*****
SINIPOINT:
;-----
;   デジタル入出力の設定
;-----
   MOV   ADPC, #00010000B             ;A/Dポート・コンフィギュレーション・レジスタ
;|||+++++----- ADPC4-ADPC0
;|||           [アナログ入力(A) / デジタル入出力(D)の切り替え]
;|||           +----- ANI11-ANI18/P153-P150
;|||           |||+++++---- ANI7-ANI10/P27-P20
;|||           0000:AAAAAAAAAAAA
;|||           00001:AAAAAAAAAAD
;|||           00010:AAAAAAAAAAD
;|||           00011:AAAAAAAAADD
;|||           00100:AAAAAADDDDD
;|||           00101:AAAAAADDDDD
;|||           00110:AAAAAADDDDD
;|||           00111:AAAAAADDDDD
;|||           01000:AAAADDDDDDD
;|||           01001:AAADDDDDDDDD
;|||           01010:AADDDDDDDDD
;|||           01011:ADDDDDDDDDDD
;|||           10000:DDDDDDDDDDDD
;|||           ;+++----- 必ず0に設定
;-----
;   ポート0の設定
;-----
   MOV   P0, #00000000B              ;P00-P01の出力ラッチLow
   MOV   PM0, #11111100B            ;P00-P01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 PU1, #00001111B ;P10-P13に内蔵プルアップ抵抗を使用
 MOV PM1, #00000000B ;P10-P17を出力ポートに設定
 ;P10:キー・スキャン用
 ;P11:キー・スキャン用
 ;P12:キー・スキャン用
 ;P13:キー・スキャン用
 ;P14-P17:未使用

 ;
 ; ポート2の設定
 ;

 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-P27:7segLED出力用

 ;
 ; ポート3の設定
 ;

 MOV P3, #00000000B ;P30-P33の出力ラッチLow
 MOV PM3, #11110000B ;P30-P33を出力ポートに設定
 ;P30:7segLEDラッチ信号用
 ;P31:7segLEDラッチ信号用
 ;P33:未使用

 ;
 ; ポート4の設定
 ;

 MOV P4, #00000000B ;P40-P43の出力ラッチLow
 MOV PM4, #11110000B ;P40-P43を出力ポートに設定
 ;P40-P43:未使用

 ;
 ; ポート5の設定
 ;

 MOV P5, #00000000B ;P50-P53の出力ラッチLow
 MOV PM5, #11110000B ;P50-P53を出力ポートに設定
 ;P50-P53:未使用

```

;-----
;   ポート6の設定
;-----
MOV  P6, #00000000B      ;P60-P61の出力ラッチLow
MOV  PM6, #11111100B    ;P60-P61を出力ポートに設定
                               ;P60-P61:未使用

;-----
;   ポート7の設定
;-----
MOV  P7, #00000000B      ;P70-P77の出力ラッチLow
MOV  PU7, #00001111B    ;P70-P73に内蔵プルアップ抵抗を使用
MOV  PM7, #00001111B    ;P70-P73を入力, P74-P77を出力ポートに設定
                               ;P70:キー・リターン用
                               ;P71:キー・リターン用
                               ;P72:キー・リターン用
                               ;P73:キー・リターン用
                               ;P74-P77:未使用

;-----
;   ポート8の設定
;-----
MOV  P8, #00000000B      ;P80-P83の出力ラッチLow
MOV  PM8, #11111000B    ;P80-P83を出力ポートに設定
                               ;P80-P83: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-P124:未使用
                               ; P121-P124は入力ポート

;-----
;   ポート14の設定
;-----
MOV  P14, #00000000B    ;P140-P141の出力ラッチLow
MOV  PM14, #11111100B  ;P140-P141を出力ポートに設定
                               ;P140-P141:未使用

;-----
;   ポート15の設定

```

```

;-----
MOV P15, #00000000B ;P150-P153の出力ラッチLow
MOV PM15, #11110000B ;P150-P153を出力ポートに設定
;P150-P153:未使用

RET

;*****
;
;
; 低電圧検出
;
;-----
; 低電圧検出回路の機能を使用し, 2.7V以上の電源電圧を確保します。
;*****
SINILVI:
;-----
; 低電圧検出
;-----
; 低電圧検出回路の機能を使用し, 2.7V以上の電源電圧を確保します。
;-----
;低電圧検出回路の設定
SET1 LVIMK ;INTLVI割り込み禁止
CLR1 LVISEL ;検出電圧をVDDに設定
MOV LVIS, #00001001B ;低電圧検出レベル選択レジスタ
;||| | ++++----- LVIS3-LVIS0
;||| | [検出レベル]
;||| | 0000:VLVI0 (4.22 ± 0.1V)
;||| | 0001:VLVI1 (4.07 ± 0.1V)
;||| | 0010:VLVI2 (3.92 ± 0.1V)
;||| | 0011:VLVI3 (3.76 ± 0.1V)
;||| | 0100:VLVI4 (3.61 ± 0.1V)
;||| | 0101:VLVI5 (3.45 ± 0.1V)
;||| | 0110:VLVI6 (3.30 ± 0.1V)
;||| | 0111:VLVI7 (3.15 ± 0.1V)
;||| | 1000:VLVI8 (2.99 ± 0.1V)
;||| | 1001:VLVI9 (2.84 ± 0.1V)
;||| | 1010:VLVI10(2.68 ± 0.1V)
;||| | 1011:VLVI11(2.53 ± 0.1V)
;||| | 1100:VLVI12(2.38 ± 0.1V)
;||| | 1101:VLVI13(2.22 ± 0.1V)
;||| | 1110:VLVI14(2.07 ± 0.1V)
;||| | 1111:VLVI15(1.91 ± 0.1V)
;++++----- 必ず0に設定

```

```

CLR1 LVIMD ;低電圧検出時の動作モードを割り込み信号発生に設定
SET1 LVION ;低電圧検出動作許可

;低電圧検出回路の動作安定待ち(約10us)
MOV B, #10 ;カウント回数設定
HRES100:
NOP ; (1clk)
DEC B ; (1clk)
BNZ $HRES100 ;ウェイト完了? No, (2clk/4clk)

;VLVI VDDになるまでのウェイト
HRES300:
NOP
BT LVIF, $HRES300 ;VDD < VLVI? Yes,
CLR1 LVION ;低電圧検出動作停止

RET

;*****
;
;
; 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。
;*****
SINICK:
MOV CMC, #00000000B ;クロック動作モード
;|||||+----- AMPH
;||||| [高速システム・クロック発振周波数の制御]
;||||| 0: 2MHz fMX < 10MHz
;||||| 1: 10MHz < fMX 20MHz
;|||||++----- AMPHS1-AMPHS0
;||||| [XT1発振回路の発振モード選択]
;||||| 00: 低消費発振 (デフォルト)
;||||| 01: 通常発振
;||||| 10: 超低消費発振
;||||| 11: 超低消費発振
;||||+----- 必ず0に設定
;|||+----- OSCSELS
;||| [サブシステム・クロック端子の動作モード]
;||| 0: 入力ポート・モード
;||| 1: XT1発振モード
;||+----- 必ず0に設定

```

```

; ++----- EXCLK/OSCSEL
;           [高速システム・クロック端子の動作モード]
;           00: 入力ポート・モード
;           01: X1発振モード
;           10: 入力ポート・モード
;           11: 外部クロック入力モード

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
; ||++++----- 必ず0に設定
; |+----- XTSTOP
; |             [サブシステム・クロックの動作制御]
; |             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]
;           0: X1発振回路動作
;           1: X1発振回路停止

MOV  OSMC, #10000000B ; 動作スピード・モード
; |||||+----- FSEL/FLPC
; |||||         [fCLKの周波数選択]
; |||||         00: 10MHz以下の周波数で動作 (デフォルト)
; |||||         01: 10MHzを越える周波数で動作
; |||||         10: 1MHzの周波数で動作
; |||||         11: 設定禁止
; ||++++----- 必ず0に設定
; +----- RTCLPC
;           [サブシステム・クロックHALTモード時の設定]
;           0: 周辺機能へのサブシステム・クロック供給許可
;           1: リアルタイム・カウンタ以外の周辺機能への
;               サブシステム・クロック供給停止

MOV  CKC, #00001000B ; クロック選択
; |+|+|+++----- CSS/MCM0/MDIV2-MDIV0
; | | |           [CPU/周辺ハードウェア・クロック (fCLK)の選択]
; | | |           0000: fIH
; | | |           0001: fIH/2 (デフォルト)
; | | |           0010: fIH/2^2
; | | |           0011: fIH/2^3

```

```

;| | |          00100: fIH/2^4
;| | |          00101: fIH/2^5
;| | |          01000: fMX
;| | |          01001: fMX/2
;| | |          01010: fMX/2^2
;| | |          01011: fMX/2^3
;| | |          01100: fMX/2^4
;| | |          01101: fMX/2^5
;| | |          1xxxx: fSUB/2
;| | +----- 必ず1に設定
;| +----- MCS <Read Only>
;|           [メイン・システム・クロック(fMAIN)のステータス]
;|           0: 高速内蔵発振クロック(fIH)
;|           1: 高速システム・クロック(fMX)
;+----- CLS <Read Only>
;           [CPU/周辺ハードウェア・クロック(fCLK)のステータス]
;           0: メイン・システム・クロック(fMAIN)
;           1: サブシステム・クロック(fSUB)

MOV  DSCCTL,    #0000000B    ;20MHz高速内蔵発振制御
;| | | | | | | +----- DSCON
;| | | | | | | [20MHz高速内蔵発振クロック(fIH20)の動作許可/禁止]
;| | | | | | | 0: 動作禁止
;| | | | | | | 1: 動作許可
;| | | | | | | +----- 必ず0に設定
;| | | | | | | +----- SELDSC
;| | | | | | | [CPU/周辺ハードウェア・クロック(fCLK)への
;| | | | | | |           20MHz高速内蔵発振選択]
;| | | | | | | 0: 20MHz高速内蔵発振を選択しない
;| | | | | | | 1: 20MHz高速内蔵発振を選択
;| | | | | | | +----- DSCS <Read Only>
;| | | | | | | [20MHz高速内蔵発振供給状態フラグ]
;| | | | | | | 0: 供給していない
;| | | | | | | 1: 供給している
;+ + + + + + + +----- 必ず0に設定

```

RET

```

;*****
;
;
;   メイン・ループ
;
;*****
;

```

```

MAIN_LOOP:
    BT    KRIF, $LMAIN300          ;INTKR割り込み要求あり? Yes ,
    ;-----
    ;キー・オフ・チェック
    ;-----
    ;1回目
    MOV   A,    P7                ;キー・オフ?
    AND   A,    #00001111B        ;
    CMP   A,    #00001111B        ;
    BNZ   $LMAIN300              ; No,

    ;チャタリング対策用のウエイト
    CALL  !!SWAIT                 ;ウエイト

    ;2回目
    MOV   A,    P7                ;キー・オフ?
    AND   A,    #00001111B        ;
    CMP   A,    #00001111B        ;
    BNZ   $LMAIN300              ; No,

    ;キー・オフ確定
    MOV   A,    #00H              ;キー番号"00"を表示するように設定
    CALL  !!SLEDOUT              ;LED出力

    STOP                                ;スタンバイ遷移

    BR   MAIN_LOOP                ;MAIN_LOOPへ
LMAIN300:
    ;-----
    ;キー・スキャン開始
    ;-----
    MOV   KRM, #00000000B        ;キー割り込み信号検出禁止

    MOV   P1,  #00001111B        ;スキャン・ラインの立ち上がり時間の短縮

    ;1回目
    CALL  !!SGETKEY              ;1回目の押下キー状態の取得
    PUSHBC                          ;1回目の押下キー状態の保存

    ;チャタリング対策用のウエイト
    CALL  !!SWAIT                 ;ウエイト

    ;2回目

```



```

CALL !!SGETKEY          ;2回目の押下キー状態の取得

    POP AX              ;1回目の押下キー状態の復帰
    AND B, A            ;1回目と2回目の押下キーを合成
    XCH A, X            ;
    AND C, A            ;BCreg:押下キー状態

;LED表示設定
CALL !!SKEYNUM          ;表示用キー番号作成
CALL !!SLEDOUT          ;LED出力

    MOV P1, #00000000B  ;スキャン・ライン全開放
    MOV PM1, #00000000B
    MOV KRM, #00001111B ;キー割り込み信号検出許可
    CLR1 KRIF           ;INTKR割り込み要求クリア
LMAIN900:
    BR MAIN_LOOP       ;MAIN_LOOPへ

;*****
;
;
; ウェイト処理
;
;-----
; 約30ms間のウェイトを行います。
;-----
;
; [I N] -
; [OUT] -
; レジスタ破壊なし
;*****
SWAIT:
    SET1 TSOL.0        ;タイマ動作開始
    CLR1 TMIF00        ;INTTM00割り込み要求クリア
SWAIT100:
    HALT
    BF TMIF00, $SWAIT100 ;30ms経過? No,
    MOV TTOL, #00000001B ;タイマ動作停止

    RET

;*****
;
;
; キー・スキャン処理
;
;

```

```

;-----
; [ I N ] -
; [OUT] BCreg : 押下キー状態
;*****
SGETKEY:
    CLRW BC                ; 押下キー状態

; キー・スキャン0
; (KEY16,KEY15,KEY14,KEY13の状態を取得)
    MOV PM1, #00001111B    ; すべてのスキャン・ラインをハイ
    MOV P1, #00000000B     ; スキャン・出力値の設定
    MOV PM1, #00000111B    ; キー・スキャン0の出力
    CALL !!SGETKRET        ; キー・リターン取得
    SHLW BC, 4

; キー・スキャン1
; (KEY12,KEY11,KEY10,KEY09の状態を取得)
    MOV PM1, #00001111B    ; すべてのスキャン・ラインをハイ
    MOV P1, #00000000B     ; スキャン・出力値の設定
    MOV PM1, #00001011B    ; キー・スキャン1の出力
    CALL !!SGETKRET        ; キー・リターン取得
    SHLW BC, 4

; キー・スキャン2
; (KEY08,KEY07,KEY06,KEY05の状態を取得)
    MOV PM1, #00001111B    ; すべてのスキャン・ラインをハイ
    MOV P1, #00000000B     ; スキャン・出力値の設定
    MOV PM1, #00001101B    ; キー・スキャン2の出力
    CALL !!SGETKRET        ; キー・リターン取得
    SHLW BC, 4

; キー・スキャン3
; (KEY04,KEY03,KEY02,KEY01の状態を取得)
    MOV PM1, #00001111B    ; すべてのスキャン・ラインをハイ
    MOV P1, #00000000B     ; スキャン・出力値の設定
    MOV PM1, #00001110B    ; キー・スキャン3の出力
    CALL !!SGETKRET        ; キー・リターン取得

    MOV PM1, #00001111B    ; スキャン終了

    RET

;*****

```

```

;
; キー・リターン取得処理
;
;-----
; [I N] BReg : 押下キー状態
; [OUT] BReg : 押下キー状態
;*****
SGETKRET:
    MOV  A,  P7
    MOV  P1, #00001111B      ;スキャン・ラインの立ち上がり時間の短縮
    XOR  A,  #00001111B
    AND  A,  #00001111B      ;リターン・ポートのみのデータに変換
    OR   C,  A

    RET

;*****
;
; 表示用キー番号作成処理
;
;-----
; [I N] BReg : 押下キー状態
; [OUT] Areg : 表示用キー番号 (00H,01H,02H,03H,04H,05H,06H,07H,08H
;                               09H,10H,11H,12H,13H,14H,15H,16H,FFH)
;
; AX, BC, DEレジスタ破壊
;-----
; 使用するレジスタの説明
; BC : 押下キー状態
; D : キー番号用
; E : キー・オン・カウンタ用
; X : キー番号の保存用
;*****
SKEYNUM:
    MOV  D,  #16      ;キー番号
    MOV  E,  #0      ;キー・オン・カウンタ
; (キーが押下されている数をカウントします。)

SLED0100:
    SHLW BC, 1      ;キー押下あり?
    BNC  $SLED0300  ; No,
    MOV  A,  D      ;
    MOV  X,  A      ; Yes, 押下されているキー番号の保存
    INC  E          ;キー・オン・カウンタを+1

SLED0300:

```

```

DEC D ;キー・スキャン終了?
BNZ $SLED0100 ; No,

MOV A, E ;キー・オン・カウンタ
CMP A, #1 ;キーは単押し?
BZ $SLED0500 ; Yes,
;-----
;多重押し"FF" / キー押下なし"00"の判定
;-----
MOV A, #0FFH
SKNC ;多重押し? Yes, ("FF")
MOV A, #0FFH ; No, ("00")
BR SLED0900
SLED0500:
;-----
;単押し"nn" (nn=01-16)
;-----
MOV A, X
CMP A, #10 ;キー番号>=10?
BC $SLED0900 ; No, ("01"-"09")
;"10"-"16"
ADD A, #6
SLED0900:
RET

;*****
;
;
; LED出力処理
;
;-----
; [I N] Areg :表示用キー番号
; [OUT] -
; AX, Bレジスタ破壊
;*****
SLEDOUT:
;-----
;表示用LEDの設定
;-----
MOV X, A
;下位桁の設定
AND A, #00FH
MOV B, A
MOV A, ES:TLEDNUM[B]

```

```
MOV P2, A
CLR1 P3.1 ;下位桁ラッチ
NOP
SET1 P3.1
;上位桁の設定
MOV A, X
SHR A, 4
MOV B, A
MOV A, ES:TLEDNUM[B]
MOV P2, A
CLR1 P3.0 ;上位桁ラッチ
NOP
SET1 P3.0

RET
```

end

Kx3-L_KR.c (C言語版)

/*****

NEC Electronics 78K0R/Kx3-Lシリーズ

78K0R/Kx-Lシリーズ サンプル・プログラム

キー・リターン

【履歴】

2008.2.-- 新規作成

2009.1.23 全面改定

【概要】

;本サンプル・プログラムはキー割り込み機能の使用例を示すものです。4×4のキー・マ
 ;トリクスよりキー割り込みの発生後，キー・スキャンを行い，押下が確定したキーの番
 ;号を7セグLEDに表示します。キー割り込みが発生したら，キー・スキャンを開始します。
 ;キー・スキャンから30ms経過後にキーの押下状態に変化があった場合は，チャタリング
 ;とし，キー・オフの状態としてキー番号は00として7segLEDに出力します。キーの押下
 ;が確定した場合は，01から16のキー番号をの7segLEDに出力します。このとき，複数の
 ;キーの同時押下があった場合は，キー無効の状態としてキー番号はFFとして出力します。

;

;

;<初期設定の主な内容>

; (オプション・バイトでの設定)

- ; ・ウォッチドッグ・タイマの動作禁止
- ; ・高速内蔵発振回路に8MHz/20MHzを選択
- ; ・LVIデフォルト・スタート機能動作
- ; ・オンチップ・デバッグを動作許可に設定

; (リセット解除後の初期化処理での設定)

- ; ・入出力ポートの設定
- ; ・低電圧検出回路の機能を使用し，2.7V以上の電源電圧を確保
- ; ・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8MHzに設定
- ; ・X1/XT1発振回路の停止
- ; ・タイマ・アレイ・ユニットの設定
- ; ・キー割り込み機能の設定

;

;

;<タイマ・アレイ・ユニットの設定>

```

;   ・ インターバル・タイマ・モードに設定
;   ・ 動作クロック周期 = fCLK/2^3 (1MHz)
;   ・ インターバル = 30ms (1[us/cclk] × 30000[count] = 30[ms])
;   ・ INTTMO0割り込み処理禁止
;
;
;
; <キー割り込み機能の設定>
;   ・ KR0, KR1, KR2, KR3のキー割り込み信号の検出設定
;   ・ INTKR割り込み処理禁止
;
;
; <キー・マトリクス>
;   +-----+-----+-----+-----+
;   | KEY01 | KEY05 | KEY09 | KEY13 | KR0(P70)
;   +-----+-----+-----+-----+
;   | KEY02 | KEY06 | KEY10 | KEY14 | KR1(P71)
;   +-----+-----+-----+-----+
;   | KEY03 | KEY07 | KEY11 | KEY15 | KR2(P72)
;   +-----+-----+-----+-----+
;   | KEY04 | KEY08 | KEY12 | KEY16 | KR3(P73)
;   +-----+-----+-----+-----+
;
;   KS3      KS2      KS1      K03
;   (P10)    (P11)    (P12)    (P13)
;
;
;   KS : キー・スキャン・ライン
;   KR : キー・リターン・ライン
;
;
; <入出力ポートの設定>
;   入力ポート : P70-P73 . . . キー・リターン用
;   出力ポート : P10-P13 . . . キー・スキャン用
;
;                   P20-P27 . . . 7segLED用
;
;                   P30, P31 . . . 7segLEDラッチ信号用
;   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 */
#pragma NOP          /* NOP()を記述可能にする */

```

```

#pragma HALT                /* HALT()を記述可能にする */
#pragma STOP                /* STOP()を記述可能にする */
#pragma DI                  /* DI()を記述可能にする */
#pragma EI                  /* EI()を記述可能にする */
/*=====
    割り込みハンドラ定義
=====*/

/*=====
    関数プロトタイプ宣言
=====*/

    void hdwinit( void );    /* 機能初期化 */
static void f_ini_cpu(void); /* CPU初期設定 */
static void f_ini_lvi(void); /* 電源電圧立ち上がり待ち */
static void f_ini_port(void); /* ポート初期設定 */
static void f_ini_reg(void); /* 周辺レジスタ初期設定 */
static void f_ini_itr(void); /* 割り込み初期設定 */
    void main(void);        /* アプリケーションメイン */
static void wait(void);     /* 30msウエイト */
static void getkey( void ); /* キー押下情報取得 */
static void ledout(unsigned short); /* 7セグLED出力 */

/*=====
    グローバル変数の定義
=====*/

static unsigned char keyscan; /* キースキャンフラグ */
static unsigned short keyon; /* キー・スキャン確定情報 */
static unsigned short keycom; /* キー・スキャン確定情報 */
static unsigned char ledset; /* LED出力フラグ */
static unsigned short bitmsk; /* マスクビット情報 */
static const unsigned char segdata[16] =
/*
    ;      h ----- h (P27)  aaaaa
    ;      |g ----- g (P26)  f    b
    ;      ||f ----- f (P25)  f    b
    ;      |||e ----- e (P24)  ggggg
    ;      ||||d ----- d (P23)  e    c
    ;      |||||c ----- c (P22)  e    c
    ;      |||||b ----- b (P21)  ddddd  hh
    ;      |||||a----- a (P20)           hh
    ;      |||||
*/
{
    0b11000000, /* "0" */
    0b11111001, /* "1" */

```



```

0b10100100,      /* "2" */
0b10110000,      /* "3" */
0b10011001,      /* "4" */
0b10010010,      /* "5" */
0b10000010,      /* "6" */
0b11111000,      /* "7" */
0b10000000,      /* "8" */
0b10010000,      /* "9" */
0b10001000,      /* "A" */
0b10000011,      /* "B" */
0b11000110,      /* "C" */
0b10100001,      /* "D" */
0b10000110,      /* "E" */
0b10001110,      /* "F" */

};

/*****
* Title   : リセット解除後の初期化処理
*****

* Module  : void hdwinit(void)
* Arg     :
* Ret     :
* -----
* Note    :
*****/

void hdwinit(void)
{
    f_ini_port();      /* ポート初期設定 */
    f_ini_lvi();       /* 2.7V以上の電源電圧を確保 */
    f_ini_cpu();       /* CPU初期設定 */
    f_ini_reg();       /* 周辺レジスタ初期設定 */
    f_ini_itr();       /* 割り込み初期設定 */
}

/*****
* Title   : 電源電圧検出
*****

* Module  : static void f_ini_lvi(void)
* Arg     :
* Ret     :
* -----
* Note    : LVI機能を使い電源電圧が2.7V以上になるのを待ちます。
*****/

static void f_ini_lvi(void)

```

```

{
    unsigned char ucCnt10us;          /* LVI起動待ち時間計測用 */
/*-----
;    低電圧検出回路の機能を使用し, 2.7V以上の電源電圧を確保します。
;-----*/

    LVIMK = 1;                        /* INTLVI割り込み禁止 */
    LVIM = 0b00000000;                /* 低電圧検出レジスの設定 */
/*
    |||||+--LVIFフラグ
    |||||
    |||||+---LVIMD: 低電圧検出の動作モード選択
    |||||          0 : 割り込みモード
    |||||          1 : リセット・モード
    |||||
    |||||+----LVISEL: 電圧検出の選択
    |||||          0 : 電源電圧 (VDD) のレベルを検出
    |||||          1 : 外部入力端子からの入力電圧のレベルを検出
    |||||
    |++++----- 必ず0に設定
    |
    +-----LVION: 低電圧検出動作許可
    0 : 動作禁止
    1 : 動作許可
*/

    LVIS = 0b00001001;                /* 検出電圧の設定 */
/*
    ||||+-----LVIS3-0: [検出レベル]
    ||||          0000 : 4.22 ± 0.1V
    ||||          0001 : 4.07 ± 0.1V
    ||||          0010 : 3.92 ± 0.1V
    ||||          0011 : 3.76 ± 0.1V
    ||||          0100 : 3.61 ± 0.1V
    ||||          0101 : 3.45 ± 0.1V
    ||||          0110 : 3.30 ± 0.1V
    ||||          0111 : 3.15 ± 0.1V
    ||||          1000 : 2.99 ± 0.1V
    ||||          1001 : 2.84 ± 0.1V
    ||||          1010 : 2.68 ± 0.1V
    ||||          1011 : 2.53 ± 0.1V
    ||||          1100 : 2.38 ± 0.1V
    ||||          1101 : 2.22 ± 0.1V
    ||||          1110 : 2.07 ± 0.1V
    ||||          1111 : 1.91 ± 0.1V

```

```

        ||||
        +---+----- 必ず0に設定
*/

LV1ON = 1;          /* 低電圧検出動作許可 */

/*
    低電圧検出回路の動作安定待ち(約10us)
*/
for( ucCnt10us = 0; ucCnt10us < 3; ucCnt10us++ ){
    NOP();
    NOP();
}

/*
    VLVI VDDになるまでのウェイト
*/
while(LVIF){
    NOP();
}
LV1ON = 0;          /* 低電圧検出動作停止 */
}

/*****
* Title : CPU初期設定
*****/

* Module : static void f_ini_cpu(void)
* Arg   :
* Ret   :
*-----
* Note :
*****/

static void f_ini_cpu(void)
{

/*-----
;   動作クロックを内蔵高速発振器(8MHz)に設定します。
;-----*/

    CMC = 0b00000000;          /* クロック動作モード制御レジスタ(CMC) */
/*
    |||||+--AMPH: 高速システム・クロック発振周波数の制御
    |||||          0:2MHz<=fMX<=10MHz
    |||||          1:10MHz<=fMX<=20MHz

```

```

|||||
|||||+---AMPHS1,AMPS0: XT1発振回路の発振モード選択
|||||          00:低消費発振(デフォルト)
|||||          01:通常発振
|||||          1x:超低消費発振
|||||
||+|+----- 必ず0に設定
|| |
|| +-----OSCSELS: サブシステム・クロック端子の動作モード
||          0:入力ポート・モード
||          1:XT1発振モード
++-----EXCLK,OSCSEL: 高速システム・クロック端子の動作モード
          00:入力ポート・モード
          01:X1発振モード
          10:入力ポート・モード
          11:外部クロック入力モード

*/

CSC = 0b11000000;          /* クロック動作ステータス制御レジスタ(CSC) */
/*
|||||+---HI0STOP: 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
|||||
||+++++--- 必ず0に設定
||
|+-----XTSTOP: サブシステム・クロックの動作制御
|          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
          0:X1発振回路動作
          1:X1発振回路停止

*/

CKC = 0b00001000;          /* システム・クロック制御レジスタ(CKC) */
/*
|+|+|+++---CSS,MCM0,MDIV2,MDIV1,MDIV0:CPU/周辺ハードウェア・クロックの選択
| | |          00000:f IH
| | |          00001:f IH/2
| | |          00010:f IH/4
| | |          00011:f IH/8
| | |          00100:f IH/16
| | |          00101:f IH/32
| | |          01000:f MX

```

```

| | |          01001: fMX/2
| | |          01010: fMX/4
| | |          01011: fMX/8
| | |          01100: fMX/16
| | |          01101: fMX/32
| | |          1xxxx: fSUB/2
| | |
| | +----- 必ず1に設定
| |
| +-----MCS: メイン・システム・クロック (fMAIN)のステータス
|          0: 高速内蔵発振クロック (fIH)
|          1: 高速システム・クロック (fMX)
|
+-----CLS: CPU/周辺ハードウェア・クロック (fCLK)のステータス
          0: メイン・システム・クロック (fMAIN)
          1: サブシステム・クロック (fSUB)
*/

DSCCTL = 0b00000000;          /* 6) 20 MHz高速内蔵発振制御レジスタ (DSCCTL) */
/*          |||||+--DSCON: 20 MHz高速内蔵発振クロックの動作許可 / 禁止
          |||||          0: 動作禁止
          |||||          1: 動作許可
          |||||
          +---+|+---- 必ず0に設定
          ||
          |+----SELDSC: CPU / 周辺ハードウェア・クロック (fCLK) への
          |          20 MHz高速内蔵発振選択
          |          0: 20 MHz高速内蔵発振を選択しない
          |          1: 20 MHz高速内蔵発振を選択
          |
          +-----20 MHz高速内蔵発振供給状態フラグ
*/

/*-----
;   周辺機能へのクロック供給 (イネーブル・レジスタ)
;-----*/

/*

周辺機能は使用せず、設定はデフォルトのままがいいので設定は省略できる

*/

PERO = 0b00000000;          /* 周辺イネーブル・レジスタ0 (PERO) */

```

```

/*      |+||+|++-- 必ず0に設定
        |  |  |
        |  |  +-----SAU0EN: シリアル・アレイ・ユニットの入カクロックの制御
        |  |                                0: 入カクロック供給停止
        |  |                                1: 入カクロック供給
        |  |
        |  |
        |  |+-----IICAEN: シリアル・インターフェースIICAの入カクロックの制御
        |  |                                0: 入カクロック供給停止
        |  |                                1: 入カクロック供給
        |  |
        |  |
        |  +-----ADCEN: A/Dコンバータの入カクロックの制御
        |                                0: 入カクロック供給停止
        |                                1: 入カクロック供給
        |
        +-----RTCEN: リアルタイム・カウンタ(RTC)の入カクロックの制御
                                0: 入カクロック供給停止
                                1: 入カクロック供給

```

*/

```

PER1 = 0b00000000;      /* 周辺イネーブル・レジスタ1(PER1) */

```

```

/*      +++++|++++-- 必ず0に設定
        +-----OACMPEN: オペアンプの入カクロックの制御
                                0: 入カクロック供給停止
                                1: 入カクロック供給

```

*/

```

PER2 = 0b00000000;      /* 周辺イネーブル・レジスタ2(PER2) */

```

```

/*      ++++++|-- 必ず0に設定
        |
        +----TAU0EN: タイマ・アレイ・ユニットTAUSの入カクロックの制御
                                0: 入カクロック供給停止
                                1: 入カクロック供給

```

*/

```

/*-----
;   動作スピード・モード制御レジスタ(OSMC)
;----- */

```

*/

設定はデフォルトのままがいいので設定は省略

*/

```

OSMC = 0b00000000; /* 動作スピード・モード制御レジスタ(OSMC) */
/*
|+++++|-- 必ず0に設定
|  ||
|   +---FLPC,FSEL: fCLKの周波数選択
|           00:10MHz以下の周波数で動作(デフォルト)
|           01:10MHzを越える周波数で動作
|           10:1MHz以下の周波数で動作
|           11:設定禁止
|
+-----RTCLPC: サブシステム・クロックHLATモード時の設定
           0:周辺機能へのサブシステム・クロック供給許可
           1:RTC以外の周辺機能へのサブシステム・クロック供給停止
*/
}
/*****
* Title : ポート初期設定
*****
* Module : static void f_ini_port(void)
* Arg   :
* Ret   :
*-----
* Note  : 使用しないポートは出力ポートに設定できるものは出力ポート設定する
*****/
static void f_ini_port(void)
{
/*****
;   ポート0の設定(未使用)
;
;
;
;*****/
;

    PO = 0b00000000; /* P01,P00を0に設定 */
    PM0 = 0b11111100; /* P01,P00を出力に設定 */

/*****
;   ポート1の設定(未使用)
;
;
;*****/
;

    P1 = 0b00001111; /* スキャン・ラインは1、他は0に */
    PM1 = 0b00001111; /* P14~P17を出力に設定する */
    PU1 = 0b00001111; /* P10~P13のプルアップ抵抗を有効に */

```

```

/*
    - |||||+----- P1.0:入力ポート/初期値High
    - |||||+----- P1.1:入力ポート/初期値High
    - ||||+----- P1.2:入力ポート/初期値High
    - |||+----- P1.3:入力ポート/初期値High
    - ++++----- x (未使用)
*/

/*****
;   ポート 2 の設定
;
;   LED表示データを外部ラッチに出力ために使用する。
;
;*****/
/*   設定しないが参考として示す
    ADPC =    0b00010000;*/           /* A/Dポート・コンフィギュレーション・
                                       レジスタ (ADPC) : 全てデジタルに */

    P2 = 0b00000000;                 /* 全ての出力ラッチを0に設定 */
    PM2 = 0b00000000;                /* 全てのビットを出力に設定 */
*/

    - |||||+---- P2.0:出力ポート/初期値LOW
    - |||||+---- P2.1:出力ポート/初期値LOW
    - ||||+----- P2.2:出力ポート/初期値LOW
    - |||+----- P2.3:出力ポート/初期値LOW
    - ||+----- P2.4:出力ポート/初期値LOW
    - |+----- P2.5:出力ポート/初期値LOW
    - +----- P2.6:出力ポート/初期値LOW
    - +----- P2.7:出力ポート/初期値LOW

*/

/*****
;   ポート 3 の設定
;
;   P30-P31をLED出力データのラッチ出力 (立ち上がり有効) で使用
;
;*****/
/*   PIM3とPOM3は設定しないが参考として示す
    PIM3 =    0b00000000;*/           /* P32 ~ P33は通常入力 */
/*   POM3 =    0b00000000;*/           /* P32 ~ P30は通常出力モード */
    P3 = 0b00000000;                 /* P33 ~ P30の出力ラッチを0に設定 */
    PM3 = 0b11110000;                /* P30 ~ P33は出力に設定 */
*/

```



```

-      |+--- P3.0:出力ポート/初期値LOW
-      +---- P3.1:出力ポート/初期値LOW
*/
/*****
;   ポート4の設定(未使用)
;
;
;*****/

P4 = 0b00000000;          /* P43~P40の出力ラッチを0に設定 */
PM4 = 0b11110000;        /* 全てのビットを出力に設定 */

/*****
;   ポート5の設定(未使用)
;
;
;*****/

P5 = 0b00000000;          /* P53~P50の出力ラッチを0に設定 */
PM5 = 0b11110000;        /* 全てのビットを出力に設定 */

/*****
;   ポート6の設定(未使用)
;
;
;*****/

P6 = 0b00000000;          /* P61~P60の出力ラッチを0に設定 */
PM6 = 0b11111100;        /* 全てのビットを出力に設定 */

/*****
;   ポート7の設定
;
;
;*****/
/* POM7は設定しないが参考として示す*/
PIM7 = 0b00000000;        /* P77~P70は通常入力 */
/* POM7 = 0b00000000;*/    /* P77~P70は通常出力モード */
P7 = 0b00000000;          /* P77~P70の出力ラッチを0に設定 */

PU7 = 0b00001111;         /* 入力にはプルアップ抵抗を有効に */
PM7 = 0b00001111;         /* P7.3~P7.0を入力に設定 */
/*
-   |||||
-   |||||+---- P7.0:入力ポート/プルアップ使用
-   |||||+---- P7.1:入力ポート/プルアップ使用
-   ||||+----- P7.2:入力ポート/プルアップ使用
-   |||+----- P7.3:入力ポート/プルアップ使用

```

```

- |||+----- X
- ||+----- X
- |+----- X
- +----- X

*/

/*****
;   ポート 8 の設定 (未使用)
;
;
; *****/
/* PIM8は設定しないが参考として示す
   PIM8 = 0b00001111; /* P83~P80はデジタル入力許可 */
   P8 = 0b00000000; /* P83~P80の出力ラッチを0に設定 */
   PM8 = 0b11110000; /* 全てのビットを出力に設定 */

/*****
;   ポート 1 2 の設定 (未使用)
;
;
;   P120を出力ポートとして使用する。そのほかのビットは入力専用ポート
;
; *****/

   P12 = 0b00000000; /* P124~P120の出力ラッチを0に設定 */
   PM12 = 0b11111110; /* P120を出力に設定する */

/*****
;   ポート 1 4 の設定 (未使用)
;
;
; *****/

   P14 = 0b00000000; /* P61~P60の出力ラッチを0に設定 */
   PM14 = 0b11111100; /* 全てのビットを出力に設定 */

/*****
;   ポート 1 5 の設定 (未使用)
;
;
; *****/

   P15 = 0b00000000; /* P153~P150の出力ラッチを0に設定 */
   PM15 = 0b11110000; /* 全てのビットを出力に設定 */

}

```

```

/*****
* Title   : 周辺レジスタ初期設定
*****

* Module : static void f_ini_reg(void)
* Arg    :
* Ret    :
* -----

* Note   :
*****/

static void f_ini_reg(void)
{
    /*-----
    -   TAUSへの入力クロック供給
    -----*/
    TAU0EN = 1;                /* 入力クロック供給 */

    TPS0L = 0b00000011;       /* タイマ・クロック選択レジスタ0L(TPS0L)
    ; ++++++--- 動作クロックの選択
    ; |||++++--- PRS003-PRS000 : CK00の選択
    ; ++++----- PRS013-PRS010 : CK01の選択
    ;                               [動作クロック(CK00/CK01)の選択]
    ;                               0000: fCLK
    ;                               0001: fCLK/2
    ;                               0010: fCLK/2^2
    ;                               0011: fCLK/2^3
    ;                               0100: fCLK/2^4
    ;                               0101: fCLK/2^5
    ;                               0110: fCLK/2^6
    ;                               0111: fCLK/2^7
    ;                               1000: fCLK/2^8
    ;                               1001: fCLK/2^9
    ;                               1010: fCLK/2^10
    ;                               1011: fCLK/2^11
    ;                               1100: fCLK/2^12
    ;                               1101: fCLK/2^13
    ;                               1110: fCLK/2^14
    ;                               1111: fCLK/2^15

    */

    /*
    チャンネルの動作モード確定
    チャンネル0 : インターバル・タイマ・モード
    */

```



```

; | |                                     スレーブ/連動動作機能のマスタの選択
; | |                                     0 : 単体動作機能, または連動動作機能で
; | |                                     スレーブ・チャンネルとして動作
; | |                                     1 : 連動動作機能でマスタ・チャンネルとして動作
; | |
; | +----- CCS0n : チャンネル2のカウンタ・クロック (TCLK) の選択
; |                                     0 : CKS00ビットで指定した動作クロック fMCK
; |                                     1 : TI00端子からの入力信号の有効エッジ
; +----- CKS0n : チャンネル2の動作クロック (fMCK) の選択
;                                     0 : PRSレジスタで設定した動作クロックCK00
;                                     1 : PRSレジスタで設定した動作クロックCK01

*/

/*-----
   インターバル時間 (タイマ・データ・レジスタ00) の設定
-----*/

TDR00 = 30000-1;          /* 周期を30msに設定 */

/*-----
-   キー・リターン割り込みの有効設定
-----*/
KRM = 0b00001111;      /* (KRM: キー・リターン・モード・レジスタ)
- |||||
- |||||+---- KRM0 :使用
- |||||+---- KRM1 :使用
- ||||+----- KRM2 :使用
- |||+----- KRM3 :使用
- ||+----- KRM4 :未使用
- |+----- KRM5 :未使用
- |+----- KRM6 :未使用
- +----- KRM7 :未使用
          KRM0,1,2,5を有効設定

*/
}

/*****
* Title : 割込初期設定
*****
* Module : static void f_ini_itr(void)
* Arg :
* Ret :

```

```

*-----
* Note :
*****/
static void f_ini_itr(void)
{
    /*-----
    - 割り込み要求フラグ・レジスタ
    - 00000000 (IF1L:割り込み要求フラグ・レジスタ)
    - |||
    - |||+---- STIF1 (INTST1 > 0:割り込み要求無し / 1:割り込み要求中)
    - ||| CSIF10 (INTCSI10> 0:割り込み要求無し / 1:割り込み要求中)
    - ||| IICIF10 (INTIIC10> 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- SRIF1 (INTSR 1 > 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- SREIF1 (INTSRE1 > 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- IICAF1 (INTIICA > 0:割り込み要求無し / 1:割り込み要求中)
    - ||+---- TMIF00 (INTTM00 > 0:割り込み要求無し / 1:割り込み要求中)
    - ||+---- TMIF01 (INTTM01 > 0:割り込み要求無し / 1:割り込み要求中)
    - |+---- TMIF02 (INTTM02 > 0:割り込み要求無し / 1:割り込み要求中)
    - +---- TMIF03 (INTTM03 > 0:割り込み要求無し / 1:割り込み要求中)
    -----*/
    TMIF00 = 0; /* TMIF00クリア */

    /*-----
    - 割り込み要求フラグ・レジスタ
    - 00000000 (IF1H:割り込み要求フラグ・レジスタ)
    - |||
    - |||+---- ADIF (INTAD > 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- RTCIF (INTRTC > 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- RTCIIF (INTRTCI> 0:割り込み要求無し / 1:割り込み要求中)
    - |||+---- KRIF (INTKR > 0:割り込み要求無し / 1:割り込み要求中)
    - ||+---- 0
    - ||+---- 0
    - |+---- MDIF (INTMD > 0:割り込み要求無し / 1:割り込み要求中)
    - +---- TMIF04 (INTTM04> 0:割り込み要求無し / 1:割り込み要求中)
    -----*/
    KRIF = 0; /* KRIFクリア */

    /*-----
    - 割り込みマスク・フラグ・レジスタ
    - 00000000 (MK1L:割り込みマスク・フラグ・レジスタ)
    - |||
    - |||+---- STMK1 (INTST1 > 0:割り込み処理許可 / 1:割り込み処理禁止)
    - ||| CSIMK10 (INTCSI10> 0:割り込み処理許可 / 1:割り込み処理禁止)

```

```

-   ||||| IICMK10 (INTIIC10> 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |||||+---- SRMK1 (INTSR1 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   ||||+----- SREMK1 (INTSRE1 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |||+----- IICAMK (INTIICA > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   ||+----- TMMK00 (INTTMO0 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |+----- TMMK01 (INTTMO1 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |----- TMMK02 (INTTMO2 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   +----- TMMK03 (INTTMO3 > 0:割り込み処理許可 / 1:割り込み処理禁止)
-----*/
TMMK00 = 0;          /* INTTMO0割り込み許可 */

/*-----
-   割り込みマスク・フラグ・レジスタ
-   00000000 (MK1H:割り込みマスク・フラグ・レジスタ)
-   |||||
-   |||||+---- ADMK (INTAD > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |||||+---- RTCMK (INTRTC > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   ||||+----- RTCIMK (INTRTCI> 0:割り込み処理許可 / 1:割り込み処理禁止)
-   |||+----- KRMK (INTKR > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   ||+----- 1
-   |+----- 1
-   |----- MDMK (INTMD > 0:割り込み処理許可 / 1:割り込み処理禁止)
-   +----- TMMK04 (INTTMO4> 0:割り込み処理許可 / 1:割り込み処理禁止)
-----*/
KRMK = 0;          /* INTKR割り込み許可 */
}

/*****
* Title : 割込処理
*****
* Module : __interrupt void INTKR_hdr(void)
* Arg :
* Ret :
* -----
* Note :
*****/

/*****
* Title : メイン・ループ
*****
* Module : void main(void)
* Arg :

```

```

* Ret   :
* -----
* Note  :
*****/

void main(void)
{

    KRIF =    1;          /* 表示初期化のため */

    while(1){
        KRM = 0b00001111; /* キー割り込み信号検出オン */
        KRIF =    0;      /* KRIFクリア */
        KRMK =    0;      /* キー割り込み許可 */

        P1 &= 0b11110000; /* スキャン・データをクリア */
        PM1 &=    0b11110000; /* 全ラインをアクティブに */

        while (!KRIF) {
            STOP();
        }

        KRM = 0b00000000; /* キー割り込み信号検出オフ */
        keyon =    0;      /* 初期状態ではキー入力なし */
        keyscan = 1;      /* キー・スキャンを開始 */
        P1 |= 0b00001111; /* スキャン・ライン立ち上げ */

        while ( keyscan ) {
            getkey();      /* キー・スキャン実行 */
            if( keycom == keyon ){ /* 有効キー入力あり */
                ledout(keyon); /* 入力キー・コードの表示 */
                if( keyon == 0 ){ /* 有効キーはない */
                    keyscan = 0; /* キー・スキャンを中止 */
                }
            }else{
                keyon =    keycom; /* 新しいキー情報に更新 */
            }
            wait();
        }
    }
}

/*****
* Title : wait
*****

```



```

* Module : static void wait(void)
* Arg   :
* Ret   :
*-----
* Note  :
*****/
static void wait(void)
{
    TSOL = 0b00000001;    /* チャンネル0の動作許可 */
    TMIF00 = 0;
    while ( !TMIF00 ) {
        HALT();
    }
    TTOL = 0b00000001;    /* チャンネル0の動作禁止 */
}

/*****
* Title : getkey
*****
* Module : static unsigned short getkey( void )
* Arg   : キー・スキャン
* Ret   :
*-----
* Note  :
*****/
void getkey( void )
{
    unsigned char scanline;    /* スキャン・ライン指定用 */
    unsigned char keywork;

    keycom = 0b0000000000000000; /* 押下キー状態を初期化 */

    for( scanline = 0b00001000 ; scanline >= 0b00000001 ; ){
        PM1 |= 0b00001111;    /* スキャン・ラインを入力に */
        P1 &= 0b11110000;    /* スキャン用データを0に設定 */
        PM1 ^= scanline;    /* スキャン・ラインを出力する */
        keycom <<= 4;    /* 結果の保存領域を確保 */
        keywork = P7 & 0b00001111; /* リターン・ラインの読み出し */
        P1 |= 0b00001111;    /* スキャン・ラインを立ち上げる */

        keywork ^= 0b00001111; /* キー・オン・ビットを1にする */
        keycom |= keywork;    /* スキャン結果を追加する */
        scanline >>= 1;
    }
}

```

```

    }

}

/*****
* Title : ledout
*****/
* Module : static void ledout( unsigned shrot onkey)
* Arg    : 押下キー番号出力
* Ret    :
* -----
* Note   :
*****/

    unsigned char lp,keycode;      /* ワーク変数定義 */
    bitmsk = 0b1000000000000000;
    ledset = 0;
    keycode = 0;

/*
   キー・パターンをコード(1~16)に変換する
*/
for (lp=16;lp>0;lp--) {
    if ( onkey & bitmsk ) { /* キーが押下されていたら */
        keycode = lp;      /* コードを設定し */
        ledset += 1;      /* キー数をカウントアップ */
    }
    bitmsk >>= 1;        /* 次のコード確認へ */
}

/*
   コードの判定と結果の7セグLEDへの表示
*/
switch( ledset ){
    case 0 :                /* 押されたがキーない場合 */
        P2 = segdata[0];
        P3 ^= 0b00000011;   /* 00表示 */
        NOP();
        P3 ^= 0b00000011;
        break;

    case 1 :                /* キーが1つの場合 */
        if( keycode > 9 ){ /* コード10以上なら */

```

```

        P2 = segdata[1]; /* 10の桁を1にして */
        keycode -= 10; /* 1の桁を計算する */
    }else{
        P2 = segdata[0]; /* 10の桁が0 のとき */
    }

    P3 ^= 0b00000001; /* 10の桁ラッチ */
    NOP();
    P0 ^= 0b00000001;

    P2 = segdata[keycode]; /* 1の桁を出力する */
    P3 ^= 0b00000010; /* 1の桁ラッチ */
    NOP();
    P3 ^= 0b00000010;
    break;

default : /* 複数キー検出の場合 */
    P2 = segdata[15];
    P3 ^= 0b00000011; /* F F 表示 */
    NOP();
    P3 ^= 0b00000011;
}
}

```

付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	January 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

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，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

【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。
