

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/LH3

サンプル・プログラム（キー・センス）

ポート・レジスタ読み出しによるキー・センス編

この資料は、5つのキー入力のキー・センスを行うサンプル・プログラムの動作概要や使用方法を説明したものです。キー・センス処理では、キー割り込み機能は使用せず、定期的にキーが接続されたポートを読むことにより、キー・コードの生成を行います。

対象デバイス

μPD78F1506 , 78F1507 , 78F1508

目次

第1章	概要	... 3
第2章	回路イメージ	... 4
2.1	回路イメージ	... 4
2.2	端子機能一覧	... 5
第3章	ソフトウェアについて	... 6
3.1	ファイル構成	... 6
3.2	使用する内蔵周辺機能	... 7
3.3	初期設定と動作概要	... 7
3.4	フロー・チャート	... 8
第4章	設定方法について	... 10
4.1	タイマ・アレイ・ユニット0の設定	... 10
4.2	ソフトウェア記述例	... 17
4.3	キー処理について	... 18
第5章	関連資料	... 21
付録A	プログラム・リスト	... 22
付録B	改版履歴	... 59

- ・本資料に記載されている内容は2009年09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に、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
 - ・当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。
「標準水準」：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
「特別水準」：輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器
「特定水準」：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。
- 注1. 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- 注2. 本事項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、製造製品をいう。
- (M8E0909J)

第1章 概 要

このサンプル・プログラムは、キー・センスの例を示しています。

サンプル・プログラムでは、キー割り込み機能を使用せず、定期的にポートを読むことによりキー・センスを行い、キー・コードを作成します。

(1) 初期設定の主な内容

< オプション・バイトでの設定 >

- ウォッチドッグ・タイマの動作禁止
- 高速内蔵発振回路の周波数を8MHzに設定
- LVIデフォルト・スタート機能停止
- 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
 - ・ P70-P74をキー接続用の入力ポートに設定
- 低電圧検出回路^注の機能を使用し、2.7V以上の電源電圧を確保
- CPUクロックをX1発振クロック動作に設定 (20MHz)
- 高速内蔵発振器の停止
- タイマ・アレイ・ユニット0のチャンネル0設定
 - ・ 約10ms周期のインターバル・タイマに設定

注 低電圧検出回路についての詳細は、ユーザーズ・マニュアルを参照してください。

(2) メイン・ループ以降の内容

初期設定完了後、キー・センス要求の発行と、キー処理の呼び出しを行います。

キー・センス要求は、タイマ・アレイ・ユニット0のタイマ割り込み (INTTM00) をポーリングして約10msのインターバルを検出し、発行します。

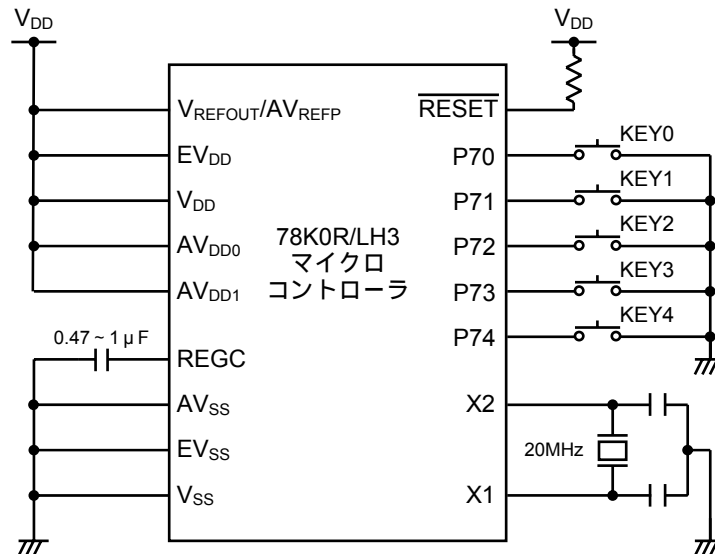
キー処理では、キー・センスとキー・オン・イベントの発行を行います。キー・センスでは、キー・センス要求があると、P70-P74のポート・レジスタを読み出し、チャタリング対策を行い、キー・コードを作成します。また、接続されたキーのいずれかがオフからオンになった場合は、キー・オン・イベントを発行します。

第2章 回路イメージ

この章では、このサンプル・プログラムで使用する場合の回路イメージおよびマイコン以外の使用デバイスを説明します。

2.1 回路イメージ

回路イメージを次に示します。



注意1. 2.7 V V_{DD} 5.5 Vの電圧範囲で使用してください。

2. EV_{DD} 、 AV_{DD0} 、 AV_{DD1} 、および V_{REFOUT}/AV_{REFP} は、 V_{DD} と同電位にしてください。

3. AV_{SS} は EV_{SS} 、 V_{SS} と同電位にし、GNDに直接接続してください。

4. REGCはコンデンサ (0.47 ~ 1 μ F) を介し、 V_{SS} に接続してください。

5. 回路イメージ中に記載のない未使用端子は以下のように処理してください。

出力ポート : 出力モードに設定し、オープン (未接続) にしてください

入力ポート : 個別に抵抗を介して、 V_{DD} または V_{SS} に接続してください

6. このサンプル・プログラムでは、P40/TOOL0端子、およびP41/TOOL1端子をオンチップ・デバッグ用に使用します。

7. P70-P74端子に内蔵プルアップを接続しています。

2.2 端子機能一覧

使用する端子機能を次に示します。



外部デバイス接続時の端子機能		兼用端子
名称	機能	
P70	キー入力 (KEY0)	KR0
P71	キー入力 (KEY1)	KR1
P72	キー入力 (KEY2)	KR2
P73	キー入力 (KEY3)	KR3
P74	キー入力 (KEY4)	KR4

第3章 ソフトウェアについて

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

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
main.asm (アセンブリ言語版)	マイコンのハードウェア初期化処理とメイン処理のソース・ファイル	注	注
main.c (C言語版)			
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル (ウォッチドッグ・タイマの設定、高速内蔵発振回路の周波数選択、LVIデフォルト・スタート機能の設定などを行います)		
78K0R_Lx3_Key.prw	統合開発環境 PM+用ワーク・スペース・ファイル		
78K0R_Lx3_Key.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。

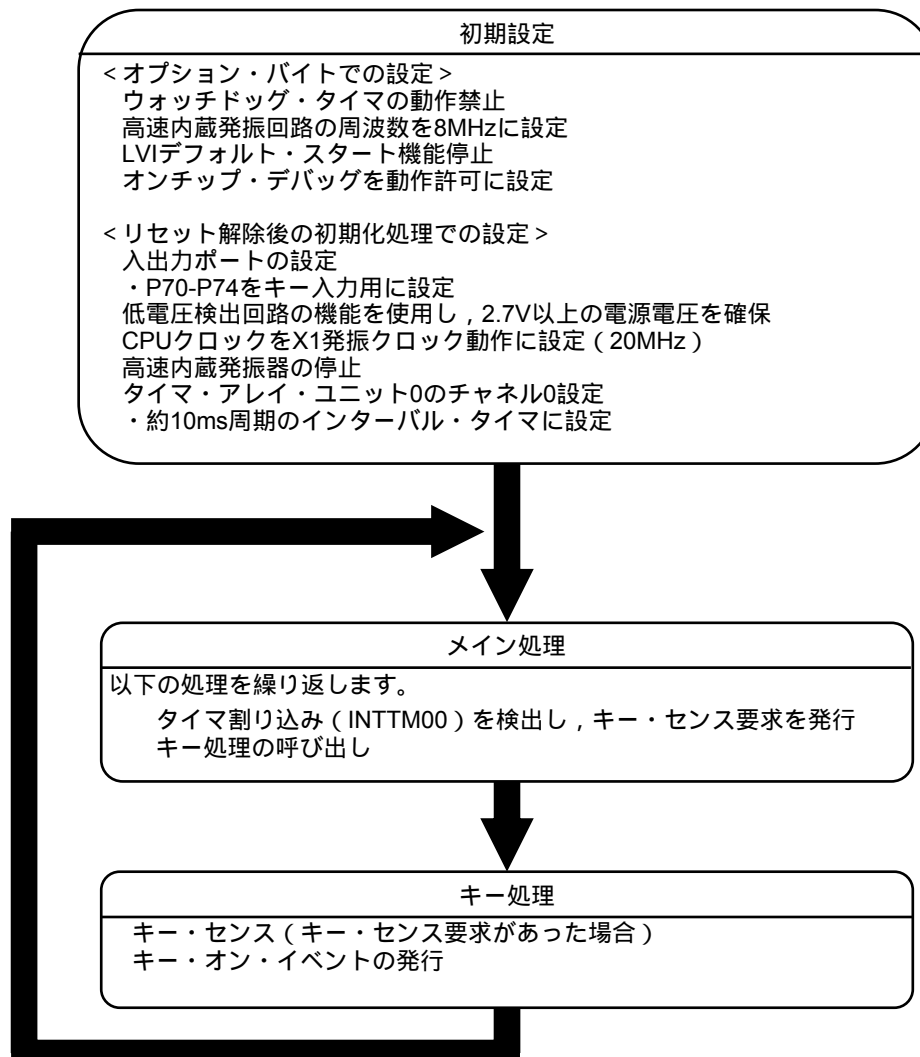
- ・ タイマ・アレイ・ユニット0 : チャネル0をインターバル・タイマとして約10ms周期のキー・センスのタイミグ用に使います。
- ・ P70-P74 : キー (KEY0, KEY1, KEY2, KEY3, KEY4) を接続します。
- ・ 低電圧検出回路 : 2.7V V_{DD} の確認用に使います。

3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、メイン・システム・クロックの選択や、入出力ポートの設定、タイマ・アレイ・ユニット0の設定などを行います。

初期設定完了後は、タイマ・アレイ・ユニット0のタイマ割り込みをポーリングして約10msのインターバルを検出し、キー・センス要求を発行します。キー処理ではキー・センス要求が発行されるとキー・センスを行います。

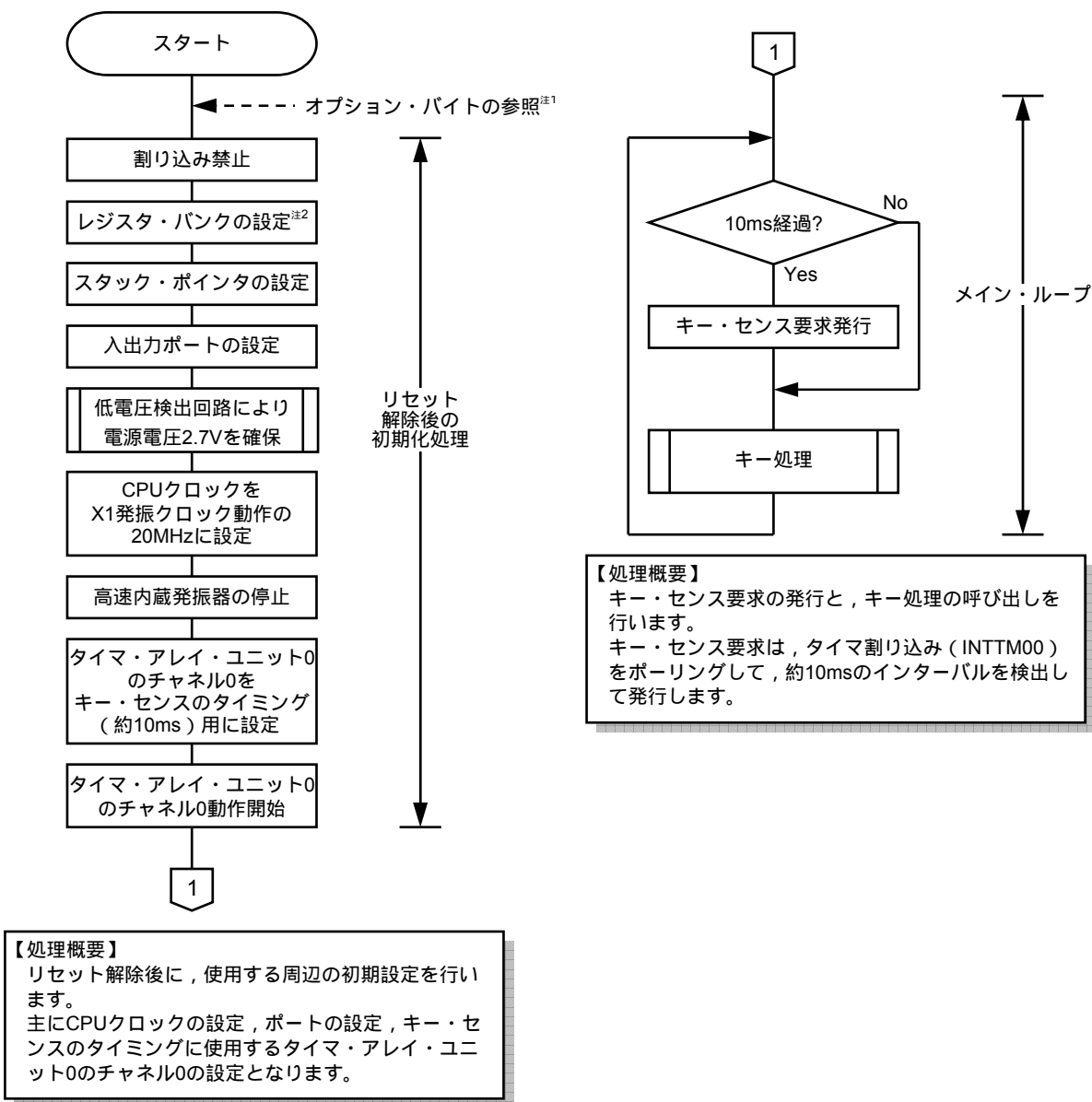
詳細については、次に示します。



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。

<リセット解除後の初期化処理での設定>

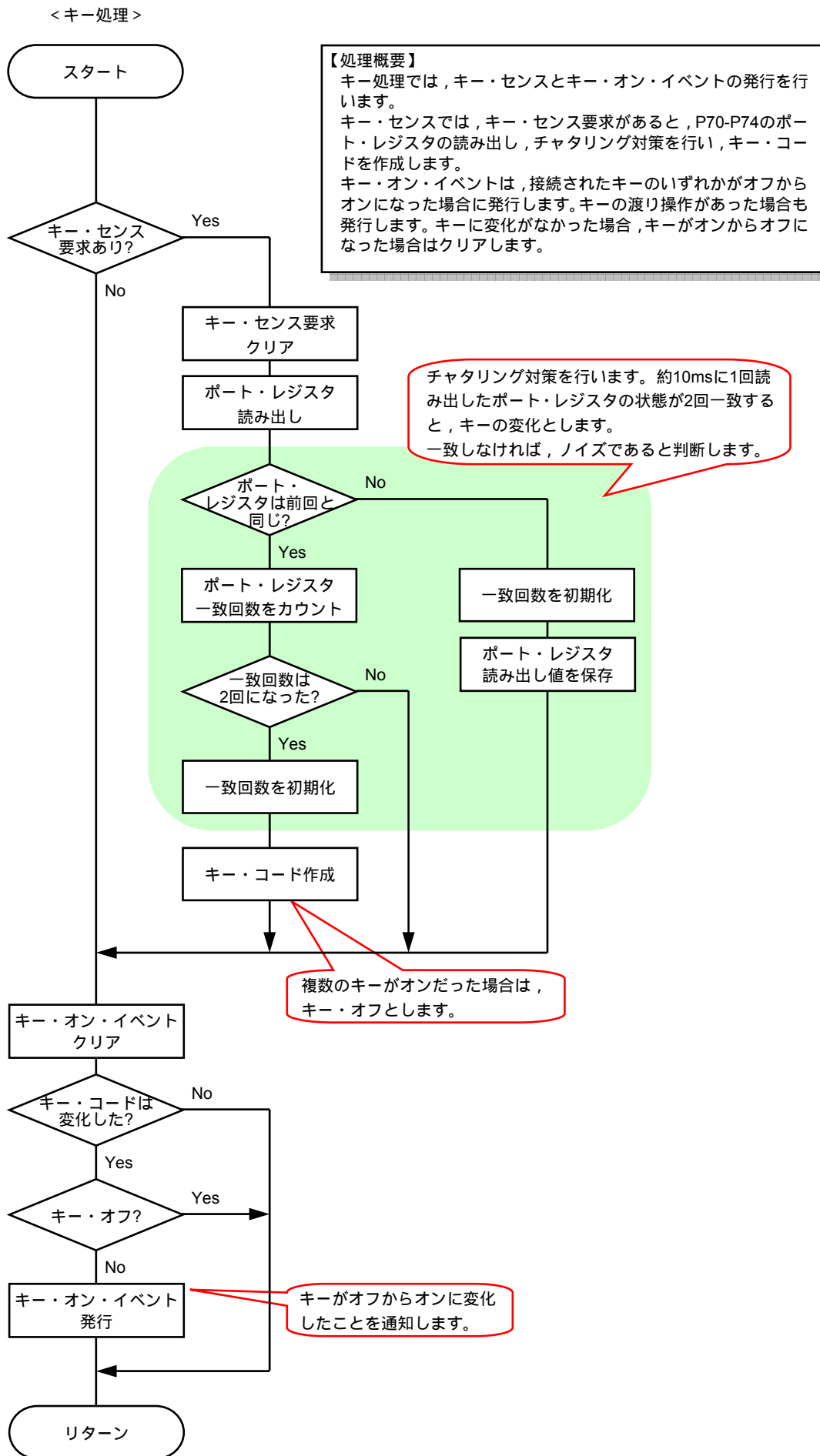


注1. オプション・バイトの参照は、リセット解除後にマイコンが自動的に行います。このサンプル・プログラムでは、オプション・バイトで以下の設定を行います。

- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振回路の周波数を8MHzに設定
- ・LVIデフォルト・スタート機能停止
- ・オンチップ・デバッグを動作許可に設定

2. 78K0R/LH3の汎用レジスタは、4レジスタ・バンク構成になっていますので、通常処理で使用するレジスタと割り込み時で使用するレジスタのバンクを切り替えることにより、効率のよいプログラムを作成できます。なお、このサンプル・プログラムでは、レジスタ・バンク0のみを使用します。

注意 C言語版のサンプル・プログラムの場合、レジスタ・バンクの設定およびスタック・ポインタの設定をスタートアップ・ルーチンで行いますので、ソース・プログラム (main.c) には記述しません。なお、スタートアップ・ルーチンについての詳細はCC78K0Rの操作編のユーザーズ・マニュアルを参照してください。



第4章 設定方法について

この章では、タイマ・アレイ・ユニット0の設定、ソフトウェア記述例、およびキー処理について説明します。その他の初期設定については、「78K0R/Lx3 サンプル・プログラム（初期設定） LED点灯のスイッチ制御編 アプリケーション・ノート」を参照してください。

レジスタ設定方法の詳細については、ユーザーズ・マニュアルを参照してください。

アセンブラ命令については、「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 タイマ・アレイ・ユニット0の設定

タイマ・アレイ・ユニット0をインターバル・タイマとして使用する場合は、主に次の5種類のレジスタで制御します。

- ・周辺イネーブル・レジスタ0 (PER0)
- ・タイマ・クロック選択レジスタ0 (TPS0)
- ・タイマ・モード・レジスタ00 (TMR00)
- ・タイマ・データ・レジスタ00 (TDR00)
- ・タイマ・チャンネル開始レジスタ0 (TS0)

タイマ割り込み発生周期は、次の式で求めることができます。

$$\text{INTTM00 (タイマ割り込み) の発生周期} = \text{カウント・クロックの周期} \times (\text{TDR00の設定値} + 1)$$

【タイマ・アレイ・ユニット0チャンネル0のインターバル割り込みを使用する場合の設定手順例】

(サンプル・プログラムと同内容)

- PER0レジスタのTAU0ENビットに“1” (タイマ・アレイ・ユニット0の入力クロック供給) を設定する
- TPS0レジスタを設定する
- TMR00レジスタを設定する (チャンネルの動作モード確定)
- TDR00レジスタにインターバル (周期) 値を設定する
- TS00ビット^注に“1”を設定する (TE00 = 1になり、カウント動作開始)

注 TS00ビットはトリガ・ビットなので、自動的に“0”に戻ります。

備考 タイマ・アレイ・ユニット0のチャンネル0がインターバル・タイマとして動作中、関連するレジスタへのアクセスは下記のとおりです。

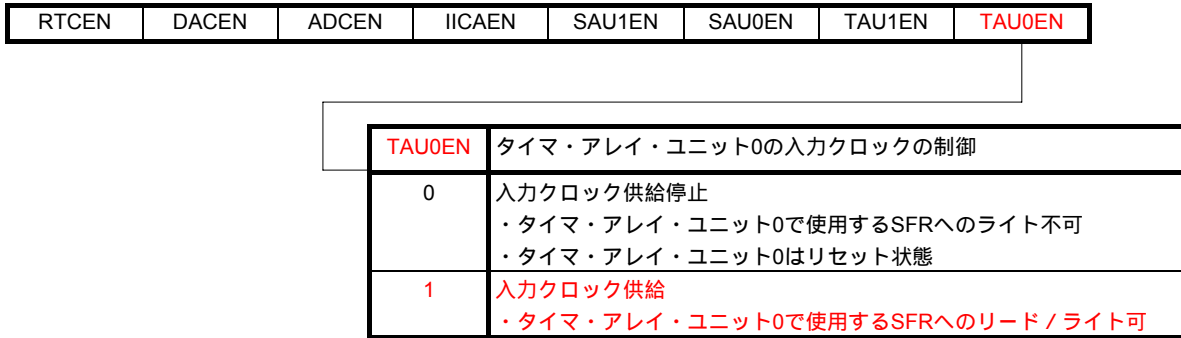
- ・TMR00, TOM0, TOL0レジスタは設定値変更禁止
- ・TDR00レジスタは、任意に設定値変更が可能
- ・TCR00レジスタは、常に読み出し可能
- ・TSR00レジスタは、使用しない
- ・TO0, TOE0レジスタは、設定値変更可能

(1) 周辺イネーブル・レジスタ0 (PER0)

PER0は、各周辺ハードウェア・マクロの使用可否を設定するレジスタです。使用しないハードウェアへはクロック供給も停止させることで、低消費電力化とノイズ低減をはかります。

タイマ・アレイ・ユニット0を使用するときは、必ずビット0 (TAU0EN) を1に設定してください。

図4 - 1 周辺イネーブル・レジスタ0 (PER0) のフォーマット



備考 図の赤字部分がサンプル・プログラムでの設定値となります

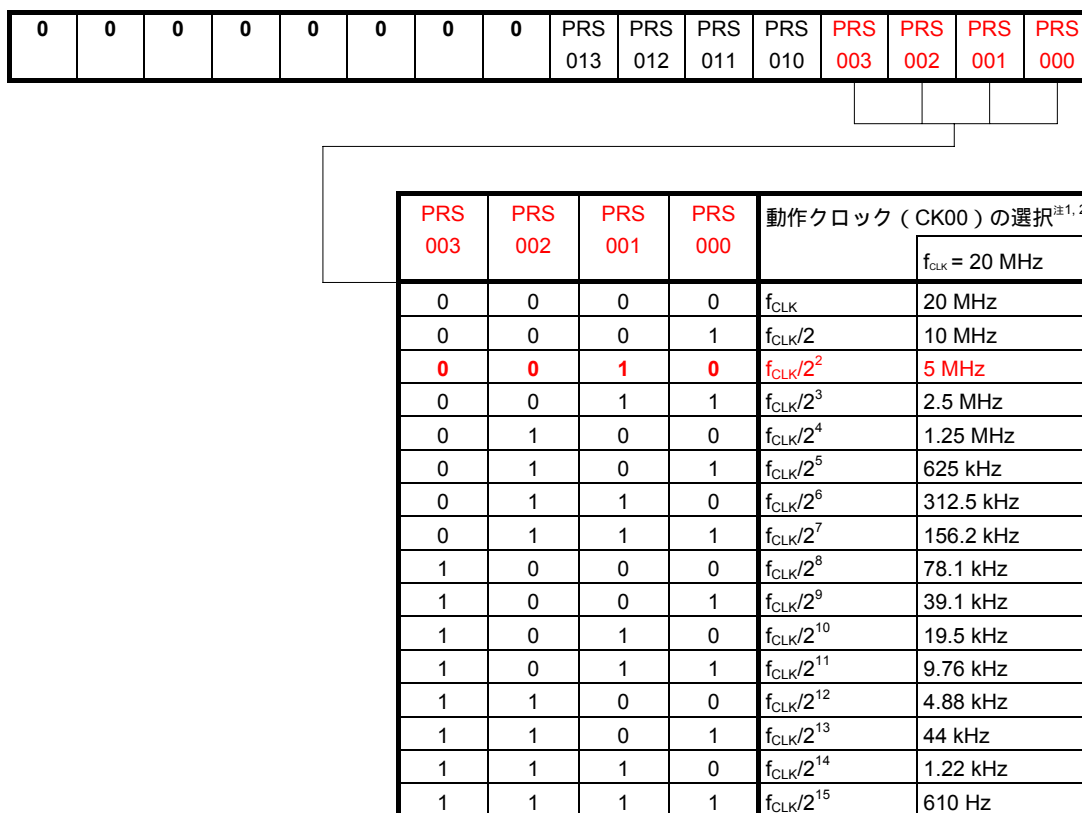
(2) タイマ・クロック選択レジスタ0 (TPS0)

各チャンネルに共通して供給される2種類の動作クロック (CKm0, CKm1) を選択する16ビット・レジスタです。

TPSmは16ビット・メモリ操作命令で設定します。

またTPSmの下位8ビットは、TPSmLで8ビット・メモリ操作命令で設定できます

図4-2 タイマ・クロック選択レジスタ0 (TPS0) のフォーマット



- 注1. f_{CLK} に選択しているクロックを変更 (システム・クロック制御レジスタ (CKC) の値を変更) する場合は、タイマ・アレイ・ユニットを停止 (TT0 = 00FFH, TT1 = 000FH) させてください。
2. SDIV=0, CCS00=1かつTIS00=1の場合にかぎり、CPUクロックを切り替えても、TAU0の使用は継続可能です。ただし、CPUクロックの切り替え時に、次の制限事項があります。
- CPUクロック切り替え時に、次のようにソース・クロックずれが発生します。
 - メイン・システム・クロック サブシステム・クロック(CSS=0 1) : -1クロック
 - サブシステム・クロック メイン・システム・クロック(CSS=1 0) : +1クロック

備考 図の赤字部分がサンプル・プログラムでの設定値となります

(3) タイマ・モード・レジスタ00 (TMR00)

タイマ・アレイ・ユニット0のチャンネル0の動作モード設定レジスタです。

図4-3 タイマ・モード・レジスタ00 (TMR00) フォーマット (1/2)

CKS 00	0	0	CCS 00	MAST ER00	STS 002	STS 001	STS 000	CIS 001	CIS 000	0	0	MD 003	MD 002	MD 001	MD 000
-----------	---	---	-----------	--------------	------------	------------	------------	------------	------------	---	---	-----------	-----------	-----------	-----------

MD 003	MD 002	MD 001	MD 000	チャンネル0の動作モードの設定	TCRのカウンタ動作	単体動作
0	0	0	1/0	インターバル・タイマ・モード	ダウン・カウンタ	可
0	1	0	1/0	キャプチャ・モード	アップ・カウンタ	可
0	1	1	0	イベント・カウンタ・モード	ダウン・カウンタ	可
1	0	0	1/0	ワンカウンタ・モード	ダウン・カウンタ	不可
1	1	0	0	キャプチャ&ワンカウンタ・モード	アップ・カウンタ	可
上記以外				設定禁止		

MD000ビットの動作は、各動作モードによって変わります。

動作モード (MD003-MD001で設定)	MD 000	カウンタ・スタートと割り込みの設定
・インターバル・タイマ・モード (0, 0, 0) ・キャプチャ・モード (0, 1, 0)	0	カウンタ開始時にタイマ割り込みを発生しない (タイマ出力も変化しない)。
	1	カウンタ開始時にタイマ割り込みを発生する (タイマ出力も変化させる)。
・イベント・カウンタ・モード (0, 1, 1)	0	カウンタ開始時にタイマ割り込みを発生しない (タイマ出力も変化しない)。
・ワンカウンタ・モード (1, 0, 0)	0	カウンタ動作中のスタート・トリガは無効とする。 その際に割り込みも発生しない。
	1	カウンタ動作中のスタート・トリガを有効とする ^注 。 その際に割り込みも発生する
・キャプチャ&ワンカウンタ・モード (1, 1, 0)	0	カウンタ開始時にタイマ割り込みを発生しない (タイマ出力も変化しない)。 カウンタ動作中のスタート・トリガは無効とする。 その際に割り込みも発生しない。
上記以外		設定禁止

注 動作中にスタート・トリガ (TS00 = 1) が掛かると、カウンタをクリアし、割り込みを発生して再カウンタ・スタートします。

注意 ビット14, 13, 5, 4には必ず0を設定してください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります

図4 - 3 タイマ・モード・レジスタ00 (TMR00) フォーマット (2/2)

CKS 00	0	0	CCS 00	MAST ER00	STS 002	STS 001	STS 000	CIS 001	CIS 000	0	0	MD 003	MD 002	MD 001	MD 000
-----------	---	---	-----------	--------------	------------	------------	------------	------------	------------	---	---	-----------	-----------	-----------	-----------

CIS 001	CIS 000	TI00端子入力信号, $f_{SUB}/2$, $f_{SUB}/4$, またはINTRTC1の有効エッジ選択 (使用するタイマ入力はTIS0レジスタで選択)
0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	両エッジ (ロウ・レベル幅測定時) スタート・トリガ: 立ち下がりエッジ, キャプチャ・トリガ: 立ち上がりエッジ
1	1	両エッジ (ハイ・レベル幅測定時) スタート・トリガ: 立ち上がりエッジ, キャプチャ・トリガ: 立ち下がりエッジ

STS 002	STS 001	STS 000	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	0	0	ソフトウェア・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)
0	0	1	TI00端子入力信号, $f_{SUB}/2$, $f_{SUB}/4$, またはINTRTC1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
0	1	0	TI00端子入力信号, $f_{SUB}/2$, $f_{SUB}/4$, またはINTRTC1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
1	0	0	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

MAS TER00	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	PCKS00ビットで指定した動作クロックMCK
1	TI00端子からの入力信号, $f_{SUB}/2$, $f_{SUB}/4$, またはINTRTC1の有効エッジ (使用するタイマ入力はTIS0レジスタで選択)

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

注意 ビット14, 13, 5, 4には必ず0を設定してください。

備考 図の赤字部分がサンプル・プログラムでの設定値となります

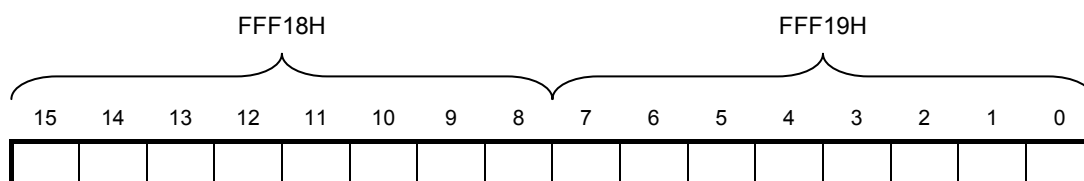
(4) タイマ・データ・レジスタ00 (TDR00)

キャプチャ機能とコンペア機能を切り替えて使用できる16ビットのレジスタです。キャプチャ機能がコンペア機能かは、TMR00のMD003-MD000ビットで動作モードを選択することで切り替わります。

TDR00は任意のタイミングで書き換えることができます。

TDR00をコンペア・レジスタとして使用するとき、TDR00に設定した値からダウン・カウントをスタートして、0000Hになったときに割り込み信号 (INTTM00) を発生します。TDR00は書き換えられるまで値を保持します。キャプチャ・トリガが入力されても、キャプチャ動作を行いません。

図4 - 4 タイマ・データ・レジスタ00 (TDR00) のフォーマット

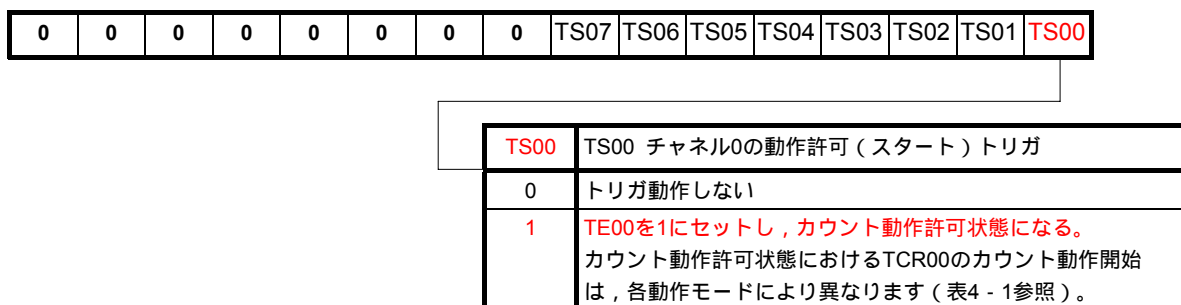


(5) タイマ・チャンネル開始レジスタ0 (TS0)

TS0は、タイマ・カウンタ (TCR00) をクリアし、カウント動作の開始を各チャンネルごとに設定するトリガ・レジスタです。

各ビット (TS00) が1にセットされると、タイマ・チャンネル許可ステータス・レジスタ0 (TE0) の対応ビット (TE00) が1にセットされます。TS00はトリガ・ビットなので、TE00 = 1になるとすぐTS00はクリアされます。

図4 - 5 タイマ・チャンネル開始レジスタ0 (TS0) のフォーマット



注意 ビット15-8には必ず0を設定してください。

備考 TS0レジスタの読み出し値は常に0となります。

表4 - 1 カウント動作許可状態からTCR00のカウント・スタートまでの動作

タイマの動作モード	TS00 = 1セット時の動作
・インターバル・タイマ・モード	スタート・トリガ検出 (TS00 = 1) 後, カウント・クロック発生まで何も動作しません。 最初のカウント・クロックでTDR00の値をTCR00にロードし, 以降のカウント・クロックでダウン・カウント動作を行います。
・イベント・カウンタ・モード	TS00ビットに1を書き込みにより, TDR00の値をTCR00にロードします。以降のカウント・クロックでダウン・カウント動作を行います。TMR00レジスタのSTS002-STS000ビットで選択した外部トリガ検出では, カウント動作を開始しません。
・キャプチャ・モード	スタート・トリガ検出後, カウント・クロック発生まで何も動作しません。 最初のカウント・クロックで0000HをTCR00にロードし, 以降のカウント・クロックでアップ・カウント動作を行います。
・ワンカウント・モード	TE00 = 0の状態では, TS00ビットに1を書き込みむことによりスタート・トリガ待ち状態となります。 スタート・トリガ検出後, カウント・クロック発生まで何も動作しません。 最初のカウント・クロックでTDR00の値をTCR00にロードし, 以降のカウント・クロックでダウン・カウント動作を行います。
・キャプチャ & ワンカウント・モード	TE00 = 0の状態では, TS00ビットに1を書き込みむことによりスタート・トリガ待ち状態となります。 スタート・トリガ検出後, カウント・クロック発生まで何も動作しません。 最初のカウント・クロックで0000HをTCR00にロードし, 以降のカウント・クロックでアップ・カウント動作を行います。

インターバル・タイマ・モード時のスタート・タイミングは, 下記のようになります。

TS00へ1を書き込むことにより, TE00 = 1となります。

TS00への書き込み情報はカウント・クロック発生まで保持されます。

TCR00は, カウント・クロック発生まで初期値を保持しています。

カウント・クロック発生により, 「TDR00値」をTCR00にロードし, カウントを開始します。

4.2 ソフトウェア記述例

ソフトウェアでの記述例として、サンプル・プログラムで行うタイマ・アレイ・ユニットの設定を以下に示します。

(1) アセンブリ言語

```

XMAIN CSEG UNIT
IRESET:

... (略) ...
MOV PER0, #00000001B ; Peripheral Enable Register 0

... (略) ...
MOV TPSOL, #00000010B ; Timer Clock Select Register 0

... (略) ...
MOVW AX, #0000000000000000B ; Timer Mode Register 00
MOVW TMR0, AX ; set CH0 interval timer

... (略) ...
MOVW TDR0, #(50000 - 1) ; set interval time to 10 ms

SET1 TMMK0 ; disable interrupt
SET1 TSOL.0 ; start timer

... (略) ...
    
```

タイマ・アレイ・ユニット0の
入力クロック供給設定

タイマ・アレイ・ユニット0の,
動作クロックを5MHzに設定

チャンネル0の動作クロックを
CK00, 動作モードをインター
バル・タイマ・モードに設定

タイマ割り込み周期を10msに設定
0.2[us/クロック] × 50000[カウント]

INTTM00割り込み
をマスク

タイマ・アレイ・ユニット0
チャンネル0の動作開始

(2) C言語

```

void hdwinit(void)
{
... (略) ...
PER0 = 0b00000001; /* Peripheral Enable Register 0 */

... (略) ...
TPSOL = 0b00000010; /* Timer Clock Select Register 0 */

... (略) ...
TMR0 = 0b0000000000000000; /* Timer Mode Register 00 */

... (略) ...
TDR0 = (50000 - 1); /* set interval time to 10 ms */

TMMK0 = 1; /* disable interrupt */
TSOL.0 = 1; /* start timer */

... (略) ...
}
    
```

タイマ・アレイ・ユニット0の
入力クロック供給設定

タイマ・アレイ・ユニット0の,
動作クロックを5MHzに設定

チャンネル0の動作クロックを
CK00, 動作モードをインター
バル・タイマ・モードに設定

タイマ割り込み周期を
10msに設定

INTTM00割り込み
をマスク

タイマ・アレイ・ユニット0
チャンネル0の動作開始

4.3 キー処理について

キー処理について説明します。

アセンブリ言語のキー処理では、次の動作を行います。

キー・センス要求があった場合は、キー・センスを行います。キー・センス要求は約10msごとに発行されます。

(a) キー入力用のポート・レジスタを読み出します。

(b) チャタリング対策を行います。

20ms以下のポートの変化は、チャタリングによる変化とします。そのため、読みだしたポート・レジスタの値が2回一致すれば、チャタリングの影響はないと判断します。

(c) キー・コードを作成します。

ポート・レジスタから読み出した値にチャタリングの影響がない場合は、ポート・レジスタの状態からキー・コードを作成します。

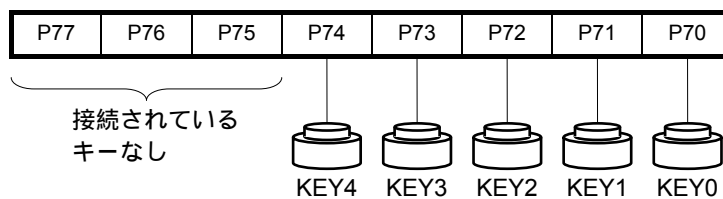
(d) (b) のチャタリング対策により、ポートの変化がチャタリングによるもの判断した場合、チャタリング対策をやり直します。

キー・オン・イベントを発行します。

あるキーがオフからオンに変化した場合は、キー・オン・イベントを発行します。キーの状態が変わらなかった場合は、キー・オン・イベントはクリアされます。キーの渡り操作があった場合も、キー・オン・イベントを発行します。

キー・オン・イベントにより、接続されたキーのいずれかがオフからオンになったことを検知することができます。

キーの多重押しは、全てのキーがオフであると判断しています。キーの多重押しによる処理をする場合は、必要に応じてソースの赤字部分に処理を追加してください。ポート7の各ビットと各キーの対応は下記のとおりです。



```

MKEY:
;*****;
;*      key sense process      *;
;*****;
BF      FKEYSENSEREQ,$LKEY700      ; was key sense request set? , No
; key sense request on
CLR1    FKEYSENSEREQ              ; clear key sense request
(a)     MOV     A,      P7          ; get key return data
(b)     AND     A,      #1FH        ; remove unwanted data
        CMP     A,      RCHATTERKEY ; key return interrupt by chattering ?
        BNZ    $LKEY600           ; Yes

        DEC     RCHATTERCOUNT    ; remove noise
        BNZ    $LKEY700           ; finished remove noise ? , No

; noise removal completion
        MOV     RCHATTERCOUNT,#CCHATTERCOUNT ; noise remove counter initialize

(c)     ; set Key Code
        MOV     RKEYCODE,#CKEY_OFF ; all key off
        CMP     RCHATTERKEY,#00011111B ; all key off ?
        BZ     $LKEY700           ; Yes
LKEY200:
        MOV     RKEYCODE,#CKEY_0   ; key0
        CMP     RCHATTERKEY,#00011110B ; key0 on ?
        BZ     $LKEY700           ; Yes
LKEY250:
        MOV     RKEYCODE,#CKEY_1   ; key1
        CMP     RCHATTERKEY,#00011101B ; key1 on ?
        BZ     $LKEY700           ; Yes
LKEY300:
        MOV     RKEYCODE,#CKEY_2   ; key2
        CMP     RCHATTERKEY,#00011011B ; key2 on ?
        BZ     $LKEY700           ; Yes
LKEY350:
        MOV     RKEYCODE,#CKEY_3   ; key3
        CMP     RCHATTERKEY,#00010111B ; key3 on ?
        BZ     $LKEY700           ; Yes
LKEY400:
        MOV     RKEYCODE,#CKEY_4   ; key4
        CMP     RCHATTERKEY,#00001111B ; key4 on ?
        BZ     $LKEY700           ; Yes
LKEY500:
;-----;
; if you need any combined key process,      ;
; add process with combination of bits here ;
;-----;

        MOV     RKEYCODE,#CKEY_OFF ; all key off
        BR     LKEY700

(d)     LKEY600: ; noise -> restart key sense
        MOV     RCHATTERCOUNT,#CCHATTERCOUNT ; noise remove counter initialize
        MOV     RCHATTERKEY,A ; get KR

LKEY700:
;*****;
;*      place key on event      *;
;*****;
CLR1    FEVENTKEYON              ; clear key on event

        MOV     A,      RKEYCODE    ; lastest key code
        CMP     A,      ROLDCODE    ; key code was changed ?
        BZ     $LKEY900           ; No

        MOV     ROLDCODE,A         ; save key code

        CMP     A,      #CKEY_OFF   ; all key off ?
        SKZ    ; Yes
        SET1   FEVENTKEYON        ; a key turns into ON from OFF, turn on a key ON event flag

LKEY900:
        RET
    
```

C言語の処理も、アセンブリ言語と同様な動作を行います。

```

static void  fn_KeyProcessing(void)
{
    unsigned char    work;

    /*-----*/
    /*      key sense process      */
    /*-----*/
    if(bKeySenseReq){
        /* key sense request on */
        bKeySenseReq = 0;        /* clear key sense request */
        work = (P7 & 0x1f);      /* get key return data */
        if(ucChatterKey == work){
            ucChatterCounter--; /* remove noise */
            if(ucChatterCounter == 0){ /* noise removal completion */
                ucChatterCounter = CCHATTERCOUNT; /* noise remove counter initialize */

                /* set Key Code */
                switch(ucChatterKey){
                    case 0b00011111:
                        ucKeyCode = CKEY_OFF; /* all key off */
                        break;

                    case 0b00011110:
                        ucKeyCode = CKEY_0; /* key0 */
                        break;

                    case 0b00011101:
                        ucKeyCode = CKEY_1; /* key1 */
                        break;

                    case 0b00011011:
                        ucKeyCode = CKEY_2; /* key2 */
                        break;

                    case 0b00010111:
                        ucKeyCode = CKEY_3; /* key3 */
                        break;

                    case 0b00001111:
                        ucKeyCode = CKEY_4; /* key4 */
                        break;

                    /*-----*/
                    /* if you need any combined key process, */
                    /* add "case" with combination of bits here */
                    /*-----*/

                    default:
                        ucKeyCode = CKEY_OFF; /* all key off */
                        break;
                }
            }
        }
        else{
            /* restart key sense */
            ucChatterKey = work; /* get KR */
            ucChatterCounter = CCHATTERCOUNT; /* noise remove counter initialize */
        }
    }

    /*-----*/
    /*      place key on event      */
    /*-----*/
    bEventKeyOn = CEVENT_OFF; /* clear key on event */

    if(ucKeyCode != ucOldKeyCode){
        /* a key turns into ON from OFF, turn on a key ON event flag */
        if(ucKeyCode != CKEY_OFF){
            bEventKeyOn = CEVENT_ON; /* set key on event */
        }
        ucOldKeyCode = ucKeyCode; /* save key code */
    }
}

```

第5章 関連資料

資料名		和文 / 英文
78K0R/Lx3 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

付録A プログラム・リスト

プログラム・リスト例として、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)
;
; Copyright (C) NEC Electronics Corporation 2006
; NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
; All rights reserved by NEC Electronics Corporation.
; This program must be used solely for the purpose for which
; it was furnished by NEC Electronics Corporation. No part of this
; program may be reproduced or disclosed to others, in any
; form, without the prior written permission of NEC Electronics
; Corporation. Use of copyright notice dose not evidence
; publication of the program.
;
;-----;
; Extern function ;
;-----;
;-----;
; Extern variables/constants ;
;-----;
;-----;
; Vector table initialize ;
;-----;
TVCT1 CSEG AT 000000H
      DW IRESET ;(00H) RESET, POC, LVI, WDT, TRAP
TVCT2 CSEG AT 000004H
      DW IRESET ;(04H) INTWDTI
      DW IRESET ;(06H) INTLVI
      DW IRESET ;(08H) INTP0
      DW IRESET ;(0AH) INTP1
      DW IRESET ;(0CH) INTP2
      DW IRESET ;(0EH) INTP3E
      DW IRESET ;(10H) INTP4
      DW IRESET ;(12H) INTP5
      DW IRESET ;(14H) INTST3
      DW IRESET ;(16H) INTSR3
      DW IRESET ;(18H) INTSRE3
```



```

DW      IRESET      ;(1AH) INTDMA0
DW      IRESET      ;(1CH) INTDMA1
DW      IRESET      ;(1EH) INTST0, INTCSI00
DW      IRESET      ;(20H) INTSR0, INTCSI01
DW      IRESET      ;(22H) INTSRE0
DW      IRESET      ;(24H) INTST1, INTCSI10, INTIIC10
DW      IRESET      ;(26H) INTSR1
DW      IRESET      ;(28H) INTSRE1
DW      IRESET      ;(2AH) INTIICA
DW      IRESET      ;(2CH) INTTM00
DW      IRESET      ;(2EH) INTTM01
DW      IRESET      ;(30H) INTTM02
DW      IRESET      ;(32H) INTTM03
DW      IRESET      ;(34H) INTAD
DW      IRESET      ;(36H) INTRTC
DW      IRESET      ;(38H) INTRTCI
DW      IRESET      ;(3AH) INTKR
DW      IRESET      ;(3CH) INTST2, INTCSI20, INTIIC20
DW      IRESET      ;(3EH) INSR2
DW      IRESET      ;(40H) INTSRE2
DW      IRESET      ;(42H) INTTM04
DW      IRESET      ;(44H) INTTM05
DW      IRESET      ;(46H) INTTM06
DW      IRESET      ;(48H) INTTM07
DW      IRESET      ;(4AH) INTP6
DW      IRESET      ;(4CH) INTP7
DW      IRESET      ;(4EH) INTP8
DW      IRESET      ;(50H) INTP9
DW      IRESET      ;(52H) INTP10
DW      IRESET      ;(54H) INTP11
DW      IRESET      ;(56H) INTTM10
DW      IRESET      ;(58H) INTTM11
DW      IRESET      ;(5AH) INTTM12
DW      IRESET      ;(5CH) INTTM13
DW      IRESET      ;(5EH) INTMD
TVCT3   CSEG      AT      00007EH
        DW      IRESET      ;(7EH) BRK

```

-----;

; Stack area definition ;

-----;

DSTK DSEG IHRAM ;Stack Area address

STACKEND:

```

                DS      60H

STACKTOP:

;-----;
; Local constants                                     ;
;-----;
;-----;
; Global variables                                     ;
;-----;
;-----;
; Local variables                                     ;
;-----;

DBMAIN          BSEG      UNIT
                FKEYSENREQ  DBIT          ; Key sense request flag
                FEVENTKEYON DBIT          ; Key on event

DMAIN  DSEG      SADDR
       RKEYCODE: DS      1      ; Key code
       ROLDCODE: DS      1      ; Old key code
               CKEY_OFF EQU      0
               CKEY_0   EQU      1
               CKEY_1   EQU      2
               CKEY_2   EQU      3
               CKEY_3   EQU      4
               CKEY_4   EQU      5
       RCHATTERKEY:   DS      1      ; Chattering key
       RCHATTERCOUNT: DS      1      ; Chattering counter for noise removal
               CCHATTERCOUNT EQU      2      ; Chattering count

;-----;
; Code                                               ;
;-----;

XMAIN  CSEG      UNIT
;-----;
; Hardware initialization                             ;
;-----;

        PUBLIC   IRESET

IRESET:
;-----;
;      Disable all interrupts                         ;
;-----;

DI

```

```

;-----;
;           Set register bank           ;
;-----;
SEL      R0           ; use register bank 0

;-----;
;           Set stack pointer           ;
;-----;
MOVW    SP,          #LOWW STACKTOP

;-----;
;           Initialization of port      ;
;-----;
CALL    !!SINITPORT

;-----;
;           Low-voltage detection       ;
;-----;
CALL    !!SINITLVI           ; ensures 2.7V to VDD

;-----;
;           Initialization of clock     ;
;-----;
MOV     CMC,         #01000011B       ; Clock Operation Mode Control Register
;|||||+--- : Control of high-speed system clock oscillation frequency
;||||| : 0 : 2 MHz <= fMX <= 10 MHz
;||||| : 1 : 10 MHz < fMX <= 20 MHz
;|||||
;|||||+---- : XT1 oscillator oscillation mode selection
;|||| : 0 0 : Low-consumption oscillation
;|||| : 0 1 : Normal oscillation
;|||| : 1 x : Super-low-consumption oscillation
;|||| x = don't care
;||||
;||||+----- : Be sure to set 0
;||||
;||||+----- : [1] Subsystem clock pin operation mode
;|||           [2] XT1/P123 pin and XT2/P124 pin
;||| : 0 : [1] Input port mode
;|||           [2] Input port
;|||
;||| : 1 : [1] XT1 oscillation mode
;|||           [2] Crystal resonator connection

```

```

;|||
;||+----- : Be sure to set 0
;||
;+----- : [1]EXCLK OSCSEL High-speed system clock pin operation mode
;           [2]X1/P121 pin
;           [3]X2/EXCLK/P122 pin
; : 0 0 : [1]Input port mode
;         [2][3]Input port
;
; : 0 1 : [1]X1 oscillation mode
;         [2][3]Crystal/ceramic resonator connection
;
; : 1 0 : [1]Input port mode
;         [2][3]Input port
;
; : 1 1 : [1]External clock input mode
;         [2]Input port
;         [3]External clock input

CLR1    MSTOP                ; X1 oscillator operating

MOV     OSMC, #00000001B      ; Operation Speed Mode Control Register
;|||||+---- : fCLK frequency selection
;||||| : 0 0 : Operates at a frequency of 10 MHz or less.
;||||| : 0 1 : Operates at a frequency higher than 10 MHz.
;||||| : 1 0 : Operates at a frequency of 1 MHz.
;|||||
;+++++----- : Be sure to set 00000
;|
;+----- : Setting in subsystem clock HALT mode
; : 0 : Enables subsystem clock supply to peripheral functions.
;       (See Table 21-1 Operating Statuses in HALT Mode (2/3)
;       for the peripheral functions whose operations are enabled.)
; : 1 : Stops subsystem clock supply to peripheral functions except
;       real-time counter, clock output/buzzer output,
;       and LCD controller/driver.

BF      OSTC.0, $$           ; X1 oscillation stabilization finished ? , No

;*- Caution -----*
;* To increase fCLK to 10 MHz or higher, set FSEL to '1', *
;* then change fCLK after two or more clocks have elapsed. *
;*------*

```

NOP

NOP

```

MOV     CKC,     #00010000B           ; System Clock Control Register
;|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK)
;| | : 0 0 x 0 0 0 : fIH
;| | : 0 0 x 0 0 1 : fIH/2 (default)
;| | : 0 0 x 0 1 0 : fIH/2^2
;| | : 0 0 x 0 1 1 : fIH/2^3
;| | : 0 0 x 1 0 0 : fIH/2^4
;| | : 0 0 x 1 0 1 : fIH/2^5
;| | : 0 1 x 0 0 0 : fMX
;| | : 0 1 x 0 0 1 : fMX/2
;| | : 0 1 x 0 1 0 : fMX/2^2
;| | : 0 1 x 0 1 1 : fMX/2^3
;| | : 0 1 x 1 0 0 : fMX/2^4
;| | : 0 1 x 1 0 1 : fMX/2^5
;| | : 1 x 0 x x x : fSUB
;| | : 1 x 1 x x x : fSUB/2
;| | : Other than above : Setting prohibited
;| | x = don't care
;| |
;| +----- : Status of Main system clock (fMAIN)
;| : 0 : Internal high-speed oscillation clock (fIH)
;| : 1 : High-speed system clock (fMX)
;|
;+----- : Status of CPU/peripheral hardware clock (fCLK)
; : 0 : Main system clock (fMAIN)
; : 1 : Subsystem clock (fSUB)

```

HRST100.; CPU is operating on a High-speed system clock ?

BT CLS, \$HRST100 ; No

BF MCS, \$HRST100 ; No

SET1 HI0STOP ; internal high-speed oscillation stopped

```

MOV     OSTs,    #00000111B           ; Oscillation Stabilization Time Select Register
;|||+|+++++--- : Oscillation stabilization time selection
;||| | : 0 0 0 : 2^8/fX
;||| | : 0 0 1 : 2^9/fX
;||| | : 0 1 0 : 2^10/fX
;||| | : 0 1 1 : 2^11/fX
;||| | : 1 0 0 : 2^13/fX

```

```

;| | | | : 1 0 1 : 2^15/fX
;| | | | : 1 1 0 : 2^17/fX
;| | | | : 1 1 1 : 2^18/fX
;| | | |
;++++----- : Be sure to set 0000

MOV     PER0, #00000001B ; Peripheral Enable Register 0
;| | | | | +--- : Control of timer array unit 0 input clock
;| | | | | : 0 : Stops input clock supply.
;| | | | | * SFR used by timer array unit 0 cannot be written.
;| | | | | * Timer array unit 0 is in the reset status.
;| | | | | : 1 : Supplies input clock.
;| | | | | * SFR used by timer array unit 0 can be read and written.
;| | | | |
;| | | | | +---- : Control of timer array unit 1 input clock
;| | | | | : 0 : Stops input clock supply.
;| | | | | * SFR used by timer array unit 1 cannot be written.
;| | | | | * Timer array unit 1 is in the reset status.
;| | | | | : 1 : Supplies input clock.
;| | | | | * SFR used by timer array unit 1 can be read and written.
;| | | | |
;| | | | +----- : Control of serial array unit 0 input clock
;| | | | : 0 : Stops input clock supply.
;| | | | * SFR used by the serial array unit 0 cannot be written.
;| | | | * The serial array unit 0 is in the reset status.
;| | | | : 1 : Supplies input clock.
;| | | | * SFR used by the serial array unit 0 can be read and written.
;| | | |
;| | | | +----- : Control of serial array unit 1 input clock
;| | | | : 0 : Stops input clock supply.
;| | | | * SFR used by the serial array unit 1 cannot be written.
;| | | | * The serial array unit 1 is in the reset status.
;| | | | : 1 : Supplies input clock.
;| | | | * SFR used by the serial array unit 1 can be read and written.
;| | | |
;| | | +----- : Control of serial interface IICA input clock
;| | | : 0 : Stops input clock supply.
;| | | * SFR used by the serial interface IICA cannot be written.
;| | | * The serial interface IICA is in the reset status.
;| | | : 1 : Supplies input clock.
;| | | * SFR used by the serial interface IICA can be read and written.
;| | |
;| | +----- : Control of A/D converter, operational amplifier, and voltage

```

reference input clock

;|| : 0 : Stops input clock supply.

;|| * SFR used by the A/D converter, operational amplifier, and voltage

reference cannot be written.

;|| * The A/D converter, operational amplifier, and voltage reference

is in the reset status.

;|| : 1 : Supplies input clock.

;|| * SFR used by the A/D converter, operational amplifier, and voltage

reference can be read and written.

;||

;|+----- : Control of D/A converter input clock

;| : 0 : Stops input clock supply.

;| * SFR used by D/A converter cannot be written.

;| * The D/A converter is in the reset status.

;| : 1 : Supplies input clock.

;| * SFR used by the D/A converter can be read and written.

;|

;+----- : Control of real-time counter (RTC) input clock

; : 0 : Stops input clock supply.

; * SFR used by the real-time counter (RTC) cannot be written.

; * The real-time counter (RTC) is in the reset status.

; : 1 : Supplies input clock.

; * SFR used by the real-time counter (RTC) can be read and written.

-----;

; Initialization of timer ;

-----;

MOV TPSOL, #00000010B ; Timer Clock Select Register 0

;|||+---- : Selection of operation clock (CK00)

;++++----- : Selection of operation clock (CK01)

; : 0 0 0 0 : CK0m = fCLK

; : 0 0 0 1 : CK0m = fCLK/2

; : 0 0 1 0 : CK0m = fCLK/2²

; : 0 0 1 1 : CK0m = fCLK/2³

; : 0 1 0 0 : CK0m = fCLK/2⁴

; : 0 1 0 1 : CK0m = fCLK/2⁵

; : 0 1 1 0 : CK0m = fCLK/2⁶

; : 0 1 1 1 : CK0m = fCLK/2⁷

; : 1 0 0 0 : CK0m = fCLK/2⁸

; : 1 0 0 1 : CK0m = fCLK/2⁹

; : 1 0 1 0 : CK0m = fCLK/2¹⁰

; : 1 0 1 1 : CK0m = fCLK/2¹¹

; : 1 1 0 0 : CK0m = fCLK/2¹²

```

; : 1 1 0 1 : CK0m = fCLK/2^13
; : 1 1 1 0 : CK0m = fCLK/2^14
; : 1 1 1 1 : CK0m = fCLK/2^15
; m = 0, 1

; CH0:for timing
MOVW   AX,      #0000000000000000B ; Timer Mode Register 00
MOVW   TMRO0,AX;|||||||++++--- : [1]Operation mode of channel 0
;|||||||           [2]Count operation of TCR
;|||||||           [3]Independent operation
;|||||||           [4]Setting of starting counting and interrupt
;||||||| : 0 0 0 0 : [1]Interval timer mode
;|||||||           [2]Counting down
;|||||||           [3]Possible
;|||||||           [4]Timer interrupt is not generated
;|||||||           when counting is started
;|||||||           (timer output does not change, either).
;|||||||
;||||||| : 0 0 0 1 : [1]Interval timer mode
;|||||||           [2]Counting down
;|||||||           [3]Possible
;|||||||           [4]Timer interrupt is generated
;|||||||           when counting is started
;|||||||           (timer output also changes).
;|||||||
;||||||| : 0 1 0 0 : [1]Capture mode
;|||||||           [2]Counting up
;|||||||           [3]Possible
;|||||||           [4]Timer interrupt is not generated
;|||||||           when counting is started
;|||||||           (timer output does not change, either).
;|||||||
;||||||| : 0 1 0 1 : [1]Capture mode
;|||||||           [2]Counting up
;|||||||           [3]Possible
;|||||||           [4]Timer interrupt is generated
;|||||||           when counting is started
;|||||||           (timer output also changes).
;|||||||
;||||||| : 0 1 1 0 : [1]Event counter mode
;|||||||           [2]Counting down
;|||||||           [3]Possible
;|||||||           [4]Timer interrupt is not generated

```



```

;|||||||||          when counting is started
;|||||||||          (timer output does not change, either).
;|||||||||
;||||||||| : 1 0 0 0 : [1]One-count mode
;|||||||||          [2]Counting down
;|||||||||          [3]Impossible
;|||||||||          [4]Start trigger is invalid
;|||||||||          during counting operation.
;|||||||||          At that time, interrupt
;|||||||||          is not generated, either.
;|||||||||
;||||||||| : 1 0 0 1 : [1]One-count mode
;|||||||||          [2]Counting down
;|||||||||          [3]Impossible
;|||||||||          [4]Start trigger is valid
;|||||||||          during counting operation.
;|||||||||          At that time, interrupt
;|||||||||          is also generated.
;|||||||||
;||||||||| : 1 1 0 0 : [1]Capture & one-count mode
;|||||||||          [2]Counting up
;|||||||||          [3]Possible
;|||||||||          [4]Timer interrupt is not generated
;|||||||||          when counting is started
;|||||||||          (timer output does not change, either).
;|||||||||          Start trigger is invalid
;|||||||||          during counting operation.
;|||||||||          At that time interrupt
;|||||||||          is not generated, either.
;|||||||||
;||||||||| : Other than above : Setting prohibited
;|||||||||
;|||||||||++----- : Be sure to set 00
;|||||||||
;|||||||||++----- : Selection of T100 pin input signal,
;|||||||||          fSUB/2, fSUB/4, or INTRTC1 valid edge
;|||||||||          (the timer input used with channel 0
;|||||||||          is selected by using TIS0 register).
;||||||||| : 0 0 : Falling edge
;||||||||| : 0 1 : Rising edge
;||||||||| : 1 0 : Both edges (when low-level width is measured)
;|||||||||          Start trigger: Falling edge, Capture trigger: Rising edge
;||||||||| : 1 1 : Both edges (when high-level width is measured)

```

```

;|||||      Start trigger: Rising edge, Capture trigger: Falling edge
;|||||
;|||||+++----- : Setting of start trigger or capture trigger of channel
0

;||||| : 0 0 0 : Only software trigger start is valid
;|||||      (other trigger sources are unselected).
;||||| : 0 0 1 : Valid edge of TI00 pin input signal, fSUB/2, fSUB/4, or INTRTC1
;|||||      is used as both the start trigger and capture trigger.
;||||| : 0 1 0 : Both the edges of TI00 pin input signal, fSUB/2, fSUB/4, or
INTRTC1

;|||||      are used as a start trigger and a capture trigger.
;||||| : 1 0 0 : Interrupt signal of the master channel is used
;|||||      (when the channel is used as a slave channel
;|||||      with the combination operation function).
;||||| : Other than above : Setting prohibited
;|||||
;|||||+----- : Selection of slave/master of channel 0
;||||| : 0 : Operates as slave channel with combination operation function.
;||||| : 1 : Operates as master channel with combination operation function.
;|||||
;|||||+----- : Selection of count clock (TCLK) of channel 0
;||||| : 0 : Operation clock MCK specified by CKS00 bit
;||||| : 1 : Valid edge of input signal input from TI00 pin, fSUB/2, fSUB/4, or
INTRTC1

;|||      (the timer input used with channel 0 is selected by using TIS0
register).

;|||
;|||+----- : Be sure to set 00
;|
;+----- : Selection of operation clock (MCK) of channel 0
; : 0 : Operation clock CK00 set by TPS0 register
; : 1 : Operation clock CK01 set by TPS0 register
; CK00 = fCLK/2^3 = 5MHz -> 10ms = 0.2[us/clock] * 50000[count]
MOVW    TDR00,    #(50000 - 1)      ; set interval time to 10 ms

SET1    TMMK00           ; disable interrupt
SET1    TSOL.0          ; start TAU0 CHO

;-----;
;      Initialization of variable      ;
;-----;

CLR1    FEVENTKEYON     ; initialize key on event flag
MOV     RKEYCODE,#CKEY_OFF ; initialize key code

```

```

MOV     ROLDCODE,#CKEY_OFF ; initialize old key code
CLR1    FKEYSENSEREQ       ; initialize key sense request flag
MOV     RCHATTERKEY,#CKEY_OFF ; initialize chattering key code
MOV     RCHATTERCOUNT,#CCHATTERCOUNT ; initialize chattering counter

BR      MMAIN              ; start main loop

;=====;
;-----;
;           Main Loop           ;
;-----;
;=====;

MMAIN:

BF      TMIF00, $LMAIN500 ; 10ms elapsed ? , No
CLR1    TMIF00

;-----;
; every 10msec process ;
;-----;

SET1    FKEYSENSEREQ      ; key sense request ON

LMAIN500:

CALL    !!MKEY            ; key process

;*****.
;*.
;*.
;*.
;*.
;*****.

BR      MMAIN              ; continue main process

;-----;
; Module:      SINITPORT ;
; Description:  Setting of I/O ports ;
; parameter:  -- ;
; return :  -- ;
;-----;

SINITPORT:

;-----;
; Ports configuration for digital input and output

```

```

;-----
MOV     ADPC,    #00010000B      ; A/D Port Configuration Register
;|||+++++--- : Analog input (A)/digital I/O (D) switching
;||| :          +----- ANI15/AVREFM/P157
;||| :          | +-+----- ANI10/P152 - ANI8/AMP2+/P150
;||| :          | | | +-+----- ANI7/AMP20/P27 - ANI0/AMP0-/P20
;||| : 0 0 0 0 0 : A A A A A A A A A A
;||| : 0 0 0 0 1 : A A A A A A A A A A D
;||| : 0 0 0 1 0 : A A A A A A A A A D D
;||| : 0 0 0 1 1 : A A A A A A A A D D D
;||| : 0 0 1 0 0 : A A A A A A A D D D D
;||| : 0 0 1 0 1 : A A A A A A A D D D D
;||| : 0 0 1 1 0 : A A A A A A D D D D D
;||| : 0 0 1 1 1 : A A A A A D D D D D D
;||| : 0 1 0 0 0 : A A A A D D D D D D D
;||| : 0 1 0 0 1 : A A A D D D D D D D D
;||| : 0 1 0 1 0 : A A D D D D D D D D D
;||| : 0 1 1 1 1 : A D D D D D D D D D D
;||| : 1 0 0 0 0 : D D D D D D D D D D
;|||
;+++----- : Be sure to set 000

```

```

;-----
; Setting of Port 0
;-----

```

```

MOV     P0,      #00000000B      ; Set P00-P02 Output latch to Low
MOV     PM0,     #11111000B      ; Set P00-P02 to output port
; P00-P02:Unused

```

```

;-----
; Setting of Port 1
;-----

```

```

MOV     P1,      #00000000B      ; Set P10-P17 Output latch to Low
MOV     PM1,     #00000000B      ; Set P10-P17 to output port
; P10-P15:Unused

```

```

;-----
; Setting of Port 2
;-----

```

```

MOV     P2,      #00000000B      ; Set P20-P27 Output latch to Low
MOV     PM2,     #11111111B      ; Set P20-P27 to input port
; P20-P27:Unused

```

```

;-----
;   Setting of Port 3
;-----
MOV    P3,    #00001000B    ; Set P30-P32,P34 Output latch to Low
                                ; Set P33 Output latch High
MOV    PM3,   #11100000B    ; Set P30-P34 to output port
                                ; P30-P34:Unused

;-----
;   Setting of Port 4
;-----
MOV    P4,    #00000000B    ; Set P40-P41 Output latch to Low
MOV    PM4,   #11111100B    ; Set P40-P41 to output port
                                ; P40-P41:Unused

;-----
;   Setting of Port 5
;-----
MOV    P5,    #00000000B    ; Set P50-P57 Output latch to Low
MOV    PM5,   #11110000B    ; Set P50-P57 to output port
                                ; P50-P57:Unused

;-----
;   Setting of Port 6
;-----
MOV    P6,    #00000000B    ; Set P60-P61 Output latch to Low
MOV    PM6,   #11111100B    ; Set P60-P61 to output port
                                ; P60-P61:Unused

;-----
;   Setting of Port 7
;-----
MOV    P7,    #00000000B    ; Set P70-P77 Output latch to Low
MOV    PM7,   #00011111B    ; et P70-P74 to input port, P75-P77 to output port
MOV    PU7,   #00011111B    ; P74-70 on-chip pull-up resistor connected
                                ; P70-P74:key input port
                                ; P75-P77:Unused

;-----
;   Setting of Port 8
;-----
MOV    P8,    #00000000B    ; Set P80-P88 Output latch to Low
MOV    PM8,   #00000000B    ; Set P80-P88 to output port

```

; P80-P88:Unused

; Setting of Port 9

```
MOV    P9,    #00000000B    ; Set P90-P97 Output latch to Low
MOV    PM9,   #00000000B    ; Set P90-P97 to output port
                                ; P90-P97:Unused
```

; Setting of Port 10

```
MOV    P10,   #00000000B    ; Set P100-P102 Output latch to Low
MOV    PM10,  #11111000B    ; Set P100-P102 to output port
                                ; P100-P102:Unused
```

; Setting of Port 11

```
MOV    P11,   #00000000B    ; Set P110-P111 Output latch to Low
MOV    PM11,  #11111100B    ; Set P110-P111 to output port
                                ; P110-P111:Unused
```

; Setting of Port 12

```
MOV    P12,   #00000000B    ; Set P120 Output latch to Low
MOV    PM12,  #11111110B    ; Set P120 to output port
                                ; P120-P124:Unused
                                ; *P121-P124:Input port
```

; Setting of Port 13

```
MOV    P13,   #00000000B    ; Set P130 Output latch to Low
                                ; P130:Unused
```

; Setting of Port 14

```
MOV    P14,   #00000000B    ; Set P140-P147 Output latch to Low
MOV    PM14,  #00000000B    ; Set P140-P147 to output port
                                ; P140-P147:Unused
```

; Setting of Port 15

```
MOV    P15,    #0000000B    ; Set P150-P152,P157 Output latch to Low
MOV    PM15,   #11111111B   ; Set P150-P152,P157 to input port
                                   ; P150-P152,P157:Unused
```

RET

-----;

; Module: SINITLVI ;

; Description: Ensures 2.7V to the power supply voltage ;

; parameter: -- ;

; return : -- ;

-----;

SINITLVI:

; setting of Low-Voltage Detector

SET1 LVIMK ; disable LVI interrupt

CLR1 LVISEL ; detects level of VDD

MOV LVIS, #00001001B ; Low-Voltage Detection Level Select Register

;|||+--- : Detection level

;||| : 0 0 0 0 : VLV10 (4.22V)

;||| : 0 0 0 1 : VLV11 (4.07V)

;||| : 0 0 1 0 : VLV12 (3.92V)

;||| : 0 0 1 1 : VLV13 (3.76V)

;||| : 0 1 0 0 : VLV14 (3.61V)

;||| : 0 1 0 1 : VLV15 (3.45V)

;||| : 0 1 1 0 : VLV16 (3.30V)

;||| : 0 1 1 1 : VLV17 (3.15V)

;||| : 1 0 0 0 : VLV18 (2.99V)

;||| : 1 0 0 1 : VLV19 (2.84V)

;||| : 1 0 1 0 : VLV110 (2.68V)

;||| : 1 0 1 1 : VLV111 (2.53V)

;||| : 1 1 0 0 : VLV112 (2.38V)

;||| : 1 1 0 1 : VLV113 (2.22V)

;||| : 1 1 1 0 : VLV114 (2.07V)

;||| : 1 1 1 1 : VLV115 (1.91V)

;|||

;++++----- : Be sure to set 0000

CLR1 LVIMD ; generates an internal interrupt signal when detect the

low-voltage

SET1 LVION ; enables low-voltage detection operation

```

; software to wait for the operation stabilization time (over 10us)
MOV     B,      #10          ; set counter
HRES100:
NOP                    ; (1clk)
DEC     B          ; (1clk)
BNZ     $HRES100   ; finished waiting ? , No (2clk/4clk)

; wait for VDD to become VLVI or more
HRES300:
NOP
BT      LVIF,     $HRES300 ; VDD < VLVI? , Yes
CLR1    LVION          ; disables low-voltage detection operation

RET

;-----;
; Module:      MKEY          ;
; Description:  Key sense and place key on event ;
; parameter:  --          ;
; return      : key code(RKEYCODE) ;
;              key on event(FEVENTKEYON) ;
; ; ;
; P7 key conection bit:76543210 ;
; ; ;
;          |||||+---Key0 ;
;          |||||+----Key1 ;
;          |||||+-----Key2 ;
;          ||||+-----Key3 ;
;          |||+-----Key4 ;
;          +++-----no key ;
;-----;
MKEY:
;*****;
;*      key sense process      *;
;*****;
BF      FKEYSENREQ,$LKEY700    ; was key sense request set? , No
; key sense request on
CLR1    FKEYSENREQ            ; clear key sense request
MOV     A,      P7            ; get key return data
AND     A,      #1FH          ; remove unwanted data
CMP     A,      RCHATTERKEY   ; key return interrupt by chattering ?
BNZ     $LKEY600              ; Yes

```



```

DEC      RCHATTERCOUNT          ; remove noise
BNZ      $LKEY700                 ; finished remove noise ? , No

; noise removal completion
MOV      RCHATTERCOUNT,#CCHATTERCOUNT ; noise remove counter initialize

; set Key Code
MOV      RKEYCODE,#CKEY_OFF       ; all key off
CMP      RCHATTERKEY,#00011111B   ; all key off ?
BZ       $LKEY700                 ; Yes
LKEY200:
MOV      RKEYCODE,#CKEY_0         ; key0
CMP      RCHATTERKEY,#00011110B   ; key0 on ?
BZ       $LKEY700                 ; Yes
LKEY250:
MOV      RKEYCODE,#CKEY_1         ; key1
CMP      RCHATTERKEY,#00011101B   ; key1 on ?
BZ       $LKEY700                 ; Yes
LKEY300:
MOV      RKEYCODE,#CKEY_2         ; key2
CMP      RCHATTERKEY,#00011011B   ; key2 on ?
BZ       $LKEY700                 ; Yes
LKEY350:
MOV      RKEYCODE,#CKEY_3         ; key3
CMP      RCHATTERKEY,#00010111B   ; key3 on ?
BZ       $LKEY700                 ; Yes
LKEY400:
MOV      RKEYCODE,#CKEY_4         ; key4
CMP      RCHATTERKEY,#00001111B   ; key4 on ?
BZ       $LKEY700                 ; Yes
LKEY500:
;-----;
; if you need any combined key process, ;
; add process with combination of bits here ;
;-----;

MOV      RKEYCODE,#CKEY_OFF       ; all key off
BR       LKEY700

LKEY600:      ; noise -> restart key sense
MOV      RCHATTERCOUNT,#CCHATTERCOUNT ; noise remove counter initialize
MOV      RCHATTERKEY,A           ; get KR

```

LKEY700:

```

.*****.
;
;*      place key on event      *;
.*****.
CLR1    FEVENTKEYON            ; clear key on event

MOV     A,      RKEYCODE ; latest key code
CMP     A,      ROLDCODE ; key code was changed ?
BZ      $LKEY900          ; No

MOV     ROLDCODE,A           ; save key code

CMP     A,      #CKEY_OFF ; all key off ?
SKZ                                ; Yes
SET1    FEVENTKEYON        ; a key turns into ON from OFF, turn on a key ON event flag

```

LKEY900:

RET

end

```
main.c (C言語版)

/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*-----*/
/* #pragma directive for CC78K0
                                     */
/*-----*/
#pragma      SFR
#pragma      DI
#pragma      EI
#pragma      NOP

/*-----*/
/* Include files
                                     */
/*-----*/
/*-----*/
/* Function prototyps
                                     */
/*-----*/
static void  fn_InitPort(void);    /* Setting of I/O ports */
static void  fn_InitLvi(void);     /* Low-voltage detection */
static void  fn_KeyProcessing(void); /* Key sense and place key on event */

/*-----*/
/* Extern variables/constants
                                     */
/*-----*/
/*-----*/
/* Local constants
                                     */
/*-----*/
/*-----*/
```

```

/* Global variables
*/
/*-----*/
/*-----*/
/* Local variables
*/
/*-----*/
boolean      bKeySenseReq;   /* Key sense request */
boolean      bEventKeyOn;   /* Key on event */
#define      CEVENT_OFF     0
#define      CEVENT_ON      1
static unsigned char    ucKeyCode;      /* Key code */
static unsigned char    ucOldKeyCode;   /* Old key code */
static enum cKeyCode{
    CKEY_OFF
    ,CKEY_0
    ,CKEY_1
    ,CKEY_2
    ,CKEY_3
    ,CKEY_4
};
static unsigned char    ucChatterKey;   /* Chattering key */
static unsigned char    ucChatterCounter; /* Chattering counter for noise removal */
#define      CCHATTERCOUNT 2          /* Chattering count */

/*-----*/
/* Code
*/
/*-----*/
/*-----*/
/* Hardware initialization
*/
/*-----*/
void hdwinit(void)
{
    DI();      /* disable all interrupts */

    /*-----*/
    /*      Initialization of port      */
    /*-----*/
    fn_InitPort();

    /*-----*/

```

```

/*          Low-voltage detection          */
/*-----*/
fn_InitLvi(); /* ensures 2.7V to VDD */

/*-----*/
/*          Initialization of clock        */
/*-----*/
CMC =      0b01000011; /* Clock Operation Mode Control Register */
/*|||||+--- : Control of high-speed system clock oscillation frequency */
/*||||| : 0 : 2 MHz <= fMX <= 10 MHz */
/*||||| : 1 : 10 MHz < fMX <= 20 MHz */
/*||||| */
/*|||||+---- : XT1 oscillator oscillation mode selection */
/*|||| : 0 0 : Low-consumption oscillation */
/*|||| : 0 1 : Normal oscillation */
/*|||| : 1 x : Super-low-consumption oscillation */
/*|||| x = don't care */
/*|||| */
/*||||+----- : Be sure to set 0 */
/*|||| */
/*|||+----- : [1] Subsystem clock pin operation mode */
/*|||          [2] XT1/P123 pin and XT2/P124 pin */
/*||| : 0 : [1]Input port mode */
/*|||          [2]Input port */
/*||| */
/*||| : 1 : [1]XT1 oscillation mode */
/*|||          [2]Crystal resonator connection */
/*||| */
/*||+----- : Be sure to set 0 */
/*|| */
/*++----- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
/*          [2]X1/P121 pin */
/*          [3]X2/EXCLK/P122 pin */
/* : 0 0 : [1]Input port mode */
/*          [2][3]Input port */
/* */
/* : 0 1 : [1]X1 oscillation mode */
/*          [2][3]Crystal/ceramic resonator connection */
/* */
/* : 1 0 : [1]Input port mode */
/*          [2][3]Input port */
/* */
/* : 1 1 : [1]External clock input mode */

```

```

/*      [2]Input port */
/*      [3]External clock input */

MSTOP = 0; /* X1 oscillator operating */

OSMC = 0b00000001; /* Operation Speed Mode Control Register */
/*|||||++--- : fCLK frequency selection */
/*||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
/*||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
/*||||| : 1 0 : Operates at a frequency of 1 MHz. */
/*||||| */
/*|+++++----- : Be sure to set 00000 */
/*| */
/*+----- : Setting in subsystem clock HALT mode */
/* : 0 : Enables subsystem clock supply to peripheral functions. */
/*      (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
/*      for the peripheral functions whose operations are enabled.) */
/* : 1 : Stops subsystem clock supply to peripheral functions */
/*      except real-time counter, clock output/buzzer output, */
/*      and LCD controller/driver. */

while(OSTC.0 != 1){ /* wait X1 oscillation stabilization */
    NOP();
}

/*-- Caution -----*/
/* To increase fCLK to 10 MHz or higher, set FSEL to '1', */
/* then change fCLK after two or more clocks have elapsed. */
/*-----*/
NOP();
NOP();

CKC = 0b00010000; /* System Clock Control Register */
/*|+|+++++----- : Selection of CPU/peripheral hardware clock (fCLK) */
/*| | : 0 0 x 0 0 0 : fIH */
/*| | : 0 0 x 0 0 1 : fIH/2 (default) */
/*| | : 0 0 x 0 1 0 : fIH/2^2 */
/*| | : 0 0 x 0 1 1 : fIH/2^3 */
/*| | : 0 0 x 1 0 0 : fIH/2^4 */
/*| | : 0 0 x 1 0 1 : fIH/2^5 */
/*| | : 0 1 x 0 0 0 : fMX */
/*| | : 0 1 x 0 0 1 : fMX/2 */
/*| | : 0 1 x 0 1 0 : fMX/2^2 */

```

```

/*| | : 0 1 x 0 1 1 : fMX/2^3 */
/*| | : 0 1 x 1 0 0 : fMX/2^4 */
/*| | : 0 1 x 1 0 1 : fMX/2^5 */
/*| | : 1 x 0 x x x : fSUB */
/*| | : 1 x 1 x x x : fSUB/2 */
/*| | : Other than above : Setting prohibited */
/*| | x = don't care */
/*| | */
/*| +----- : Status of Main system clock (fMAIN) */
/*| : 0 : Internal high-speed oscillation clock (fIH) */
/*| : 1 : High-speed system clock (fMX) */
/*| */
/*+----- : Status of CPU/peripheral hardware clock (fCLK) */
/* : 0 : Main system clock (fMAIN) */
/* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)|| (MCS != 1)){
    NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;          /* internal high-speed oscillation stopped */

OSTS =    0b00000111;    /* Oscillation Stabilization Time Select Register */
/*||||+---- : Oscillation stabilization time selection */
/*|||| : 0 0 0 : 2^8/fX */
/*|||| : 0 0 1 : 2^9/fX */
/*|||| : 0 1 0 : 2^10/fX */
/*|||| : 0 1 1 : 2^11/fX */
/*|||| : 1 0 0 : 2^13/fX */
/*|||| : 1 0 1 : 2^15/fX */
/*|||| : 1 1 0 : 2^17/fX */
/*|||| : 1 1 1 : 2^18/fX */
/*|||| */
/*++++----- : Be sure to set 000000 */

PER0 =    0b00000001;    /* Peripheral Enable Register 0 */
/*|||||+---- : Control of timer array unit 0 input clock */
/*||||| : 0 : Stops input clock supply. */
/*|||||      * SFR used by timer array unit 0 cannot be written. */
/*|||||      * Timer array unit 0 is in the reset status. */
/*||||| : 1 : Supplies input clock. */
/*|||||      * SFR used by timer array unit 0 can be read and written. */

```

```

/*||||| */
/*|||||+---- : Control of timer array unit 1 input clock */
/*||||| : 0 : Stops input clock supply. */
/*|||||      * SFR used by timer array unit 1 cannot be written. */
/*|||||      * Timer array unit 1 is in the reset status. */
/*||||| : 1 : Supplies input clock. */
/*|||||      * SFR used by timer array unit 1 can be read and written. */
/*||||| */
/*|||||+---- : Control of serial array unit 0 input clock */
/*||||| : 0 : Stops input clock supply. */
/*|||||      * SFR used by the serial array unit 0 cannot be written. */
/*|||||      * The serial array unit 0 is in the reset status. */
/*||||| : 1 : Supplies input clock. */
/*|||||      * SFR used by the serial array unit 0 can be read and written. */
/*||||| */
/*|||||+----- : Control of serial array unit 1 input clock */
/*||||| : 0 : Stops input clock supply. */
/*|||||      * SFR used by the serial array unit 1 cannot be written. */
/*|||||      * The serial array unit 1 is in the reset status. */
/*||||| : 1 : Supplies input clock. */
/*|||||      * SFR used by the serial array unit 1 can be read and written. */
/*||||| */
/*|||||+----- : Control of serial interface IICA input clock */
/*||||| : 0 : Stops input clock supply. */
/*|||||      * SFR used by the serial interface IICA cannot be written. */
/*|||||      * The serial interface IICA is in the reset status. */
/*||||| : 1 : Supplies input clock. */
/*|||||      * SFR used by the serial interface IICA can be read and written. */
/*||||| */
/*|||+----- : Control of A/D converter, operational amplifier, and voltage reference input clock */
/*||| : 0 : Stops input clock supply. */
/*|||      * SFR used by the A/D converter, operational amplifier, and voltage reference cannot be
written. */
/*|||      * The A/D converter, operational amplifier, and voltage reference is in the reset status. */
/*||| : 1 : Supplies input clock. */
/*|||      * SFR used by the A/D converter, operational amplifier, and voltage reference can be
read and written. */
/*||| */
/*|+----- : Control of D/A converter input clock */
/*| : 0 : Stops input clock supply. */
/*|      * SFR used by D/A converter cannot be written. */
/*|      * The D/A converter is in the reset status. */
/*| : 1 : Supplies input clock. */

```



```

/*|          * SFR used by the D/A converter can be read and written. */
/*| */
/*+----- : Control of real-time counter (RTC) input clock */
/* : 0 : Stops input clock supply. */
/*          * SFR used by the real-time counter (RTC) cannot be written. */
/*          * The real-time counter (RTC) is in the reset status. */
/* : 1 : Supplies input clock. */
/*          * SFR used by the real-time counter (RTC) can be read and written. */

/*-----*/
/*      Initialization of timer          */
/*-----*/
TPS0L =    0b00000010;          /* Timer Clock Select Register 0 */
/*|||++++--- : Selection of operation clock (CK00) */
/*++++----- : Selection of operation clock (CK01) */
/* : 0 0 0 0 : CK0m = fCLK */
/* : 0 0 0 1 : CK0m = fCLK/2 */
/* : 0 0 1 0 : CK0m = fCLK/2^2 */
/* : 0 0 1 1 : CK0m = fCLK/2^3 */
/* : 0 1 0 0 : CK0m = fCLK/2^4 */
/* : 0 1 0 1 : CK0m = fCLK/2^5 */
/* : 0 1 1 0 : CK0m = fCLK/2^6 */
/* : 0 1 1 1 : CK0m = fCLK/2^7 */
/* : 1 0 0 0 : CK0m = fCLK/2^8 */
/* : 1 0 0 1 : CK0m = fCLK/2^9 */
/* : 1 0 1 0 : CK0m = fCLK/2^10 */
/* : 1 0 1 1 : CK0m = fCLK/2^11 */
/* : 1 1 0 0 : CK0m = fCLK/2^12 */
/* : 1 1 0 1 : CK0m = fCLK/2^13 */
/* : 1 1 1 0 : CK0m = fCLK/2^14 */
/* : 1 1 1 1 : CK0m = fCLK/2^15 */
/*   m = 0, 1 */

/* CH0:for timing */
TMR00 =    0b0000000000000000;          /* Timer Mode Register 00 */
/*|||||||||++++--- : [1]Operation mode of channel 0 */
/*|||||||||          [2]Count operation of TCR */
/*|||||||||          [3]Independent operation */
/*|||||||||          [4]Setting of starting counting and interrupt */
/*||||||||| : 0 0 0 0 : [1]Interval timer mode */
/*|||||||||          [2]Counting down */
/*|||||||||          [3]Possible */
/*|||||||||          [4]Timer interrupt is not generated */

```

```

/*|||||||||          when counting is started */
/*|||||||||          (timer output does not change, either). */
/*||||||||| */
/*||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*|||||||||          [2]Counting down */
/*|||||||||          [3]Possible */
/*|||||||||          [4]Timer interrupt is generated */
/*|||||||||          when counting is started */
/*|||||||||          (timer output also changes). */
/*||||||||| */
/*||||||||| : 0 1 0 0 : [1]Capture mode */
/*|||||||||          [2]Counting up */
/*|||||||||          [3]Possible */
/*|||||||||          [4]Timer interrupt is not generated */
/*|||||||||          when counting is started */
/*|||||||||          (timer output does not change, either). */
/*||||||||| */
/*||||||||| : 0 1 0 1 : [1]Capture mode */
/*|||||||||          [2]Counting up */
/*|||||||||          [3]Possible */
/*|||||||||          [4]Timer interrupt is generated */
/*|||||||||          when counting is started */
/*|||||||||          (timer output also changes). */
/*||||||||| */
/*||||||||| : 0 1 1 0 : [1]Event counter mode */
/*|||||||||          [2]Counting down */
/*|||||||||          [3]Possible */
/*|||||||||          [4]Timer interrupt is not generated */
/*|||||||||          when counting is started */
/*|||||||||          (timer output does not change, either). */
/*||||||||| */
/*||||||||| : 1 0 0 0 : [1]One-count mode */
/*|||||||||          [2]Counting down */
/*|||||||||          [3]Impossible */
/*|||||||||          [4]Start trigger is invalid */
/*|||||||||          during counting operation. */
/*|||||||||          At that time, interrupt */
/*|||||||||          is not generated, either. */
/*||||||||| */
/*||||||||| : 1 0 0 1 : [1]One-count mode */
/*|||||||||          [2]Counting down */
/*|||||||||          [3]Impossible */
/*|||||||||          [4]Start trigger is valid */

```

```

/*||||| during counting operation. */
/*||||| At that time, interrupt */
/*||||| is also generated. */
/*||||| */
/*||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||| [2]Counting up */
/*||||| [3]Possible */
/*||||| [4]Timer interrupt is not generated when counting is started */
/*||||| (timer output does not change, either). */
/*||||| Start trigger is invalid */
/*||||| during counting operation. */
/*||||| At that time interrupt */
/*||||| is not generated, either. */
/*||||| */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*|||||++----- : Be sure to set 00 */
/*||||| */
/*|||||++----- : Selection of TI00 pin input signal, */
/*||||| fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||| (the timer input used with channel 0 */
/*||||| is selected by using TIS0 register). */
/*||||| : 0 0 : Falling edge */
/*||||| : 0 1 : Rising edge */
/*||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||| Start trigger: Falling edge, Capture trigger: Rising edge */
/*||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||| Start trigger: Rising edge, Capture trigger: Falling edge */
/*||||| */
/*|||||+++----- : Setting of start trigger or capture trigger of channel 0 */
/*||||| : 0 0 0 : Only software trigger start is valid */
/*||||| (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI00 pin input signal, fSUB/2, fSUB/4, or INTRTC1 */
/*||||| is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI00 pin input signal, fSUB/2, fSUB/4, or INTRTC1 */
/*||||| are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used */
/*||||| (when the channel is used as a slave channel */
/*||||| with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*|||||+----- : Selection of slave/master of channel 0 */
/*||||| : 0 : Operates as slave channel with combination operation function. */

```

```

/*||| : 1 : Operates as master channel with combination operation function. */
/*||| */
/*||+----- : Selection of count clock (TCLK) of channel 0 */
/*|| : 0 : Operation clock MCK specified by CKS00 bit */
/*|| : 1 : Valid edge of input signal input from TI00 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|| (the timer input used with channel 0 is selected by using TIS0 register). */
/*|| */
/*|+----- : Be sure to set 00 */
/*| */
/*+----- : Selection of operation clock (MCK) of channel 0 */
/* : 0 : Operation clock CK00 set by TPS0 register */
/* : 1 : Operation clock CK01 set by TPS0 register */
/* CK00 = fCLK/2^3 = 5MHz -> 10ms = 0.2[us/clk] * 50000[count] */
TDR00 = (50000 - 1); /* set interval time to 10 ms */

TMMK00 = 1; /* disable interrupt */
TSOL.0 = 1; /* start TAU0 CH0 */
}

/*-----*/
/* Module:      fn_InitPort
                */
/* Description:  Setting of I/O ports
                */
/* parameter: --
                */
/* return : --
                */
/*-----*/
static void    fn_InitPort(void)
{
/*-----*/
/*      Ports configuration for digital input and output
                */
/*-----*/
    ADPC =      0b00010000; /* A/D Port Configuration Register */
    /*||+++++ : Analog input (A)/digital I/O (D) switching */
    /*|| :      +----- ANI15/AVREFM/P157 */
    /*|| :      | +-+----- ANI10/P152 - ANI8/AMP2+/P150 */
    /*|| :      ||| +-+----- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
    /*|| : 0 0 0 0 : A A A A A A A A A A */
    /*|| : 0 0 0 1 : A A A A A A A A A D */
    /*|| : 0 0 0 1 0 : A A A A A A A A D D */
    /*|| : 0 0 0 1 1 : A A A A A A A A D D D */
    /*|| : 0 0 1 0 0 : A A A A A A A A D D D D */

```

```

/*||| : 00101 : AAAAAADDDDD */
/*||| : 00110 : AAAAAADDDDD */
/*||| : 00111 : AAAAAADDDDD */
/*||| : 01000 : AAAADDDDDDDDD */
/*||| : 01001 : AAADDDDDDDDD */
/*||| : 01010 : AADDDDDDDDDDD */
/*||| : 01111 : ADDDDDDDDDDDD */
/*||| : 10000 : DDDDDDDDDDDDD */
/*||| */
/*+++----- : Be sure to set 000 */

```

```
/*-----*/
```

```
/*      Setting of Port 0
      */
```

```
/*-----*/
```

```

P0 =      0b00000000;      /* Set P00-P02 Output latch to Low */
PM0 =     0b11111000;      /* Set P00-P02 to output port */
                        /* P00-P02:Unused */

```

```
/*-----*/
```

```
/*      Setting of Port 1
      */
```

```
/*-----*/
```

```

P1 =      0b00000000;      /* Set P10-P17 Output latch to Low */
PM1 =     0b00000000;      /* Set P10-P17 to output port */
                        /* P10-P15:Unused */

```

```
/*-----*/
```

```
/*      Setting of Port 2
      */
```

```
/*-----*/
```

```

P2 =      0b00000000;      /* Set P20-P27 Output latch to Low */
PM2 =     0b11111111;      /* Set P20-P27 to input port */
                        /* P20-P27:Unused */

```

```
/*-----*/
```

```
/*      Setting of Port 3
      */
```

```
/*-----*/
```

```

P3 =      0b00001000;      /* Set P30-P32,P34 Output latch to Low */
                        /* Set P33 Output latch High */
PM3 =     0b11100000;      /* Set P30-P34 to output port */
                        /* P30-P34:Unused */

```

```
/*-----*/
/*      Setting of Port 4
          */
/*-----*/
P4 =      0b00000000;      /* Set P40-P41 Output latch to Low */
PM4 =     0b11111100;      /* Set P40-P41 to output port */
                          /* P40-P41:Unused */

/*-----*/
/*      Setting of Port 5
          */
/*-----*/
P5 =      0b00000000;      /* Set P50-P57 Output latch to Low */
PM5 =     0b11110000;      /* Set P50-P57 to output port */
                          /* P50-P57:Unused */

/*-----*/
/*      Setting of Port 6
          */
/*-----*/
P6 =      0b00000000;      /* Set P60-P61 Output latch to Low */
PM6 =     0b11111100;      /* Set P60-P61 to output port */
                          /* P60-P61:Unused */

/*-----*/
/*      Setting of Port 7
          */
/*-----*/
P7 =      0b00000000;      /* Set P70-P77 Output latch to Low */
PM7 =     0b00011111;      /* Set P70-P74 to input port, P75-P77 to output port */
PU7 =     0b00011111;      /* P74-70 on-chip pull-up resistor connected */
                          /* P70-P74:key input port */
                          /* P75-P77:Unused */

/*-----*/
/*      Setting of Port 8
          */
/*-----*/
P8 =      0b00000000;      /* Set P80-P88 Output latch to Low */
PM8 =     0b00000000;      /* Set P80-P88 to output port */
                          /* P80-P88:Unused */
```

```

/*-----*/
/*      Setting of Port 9
          */
/*-----*/
P9 =      0b00000000;    /* Set P90-P97 Output latch to Low */
PM9 =     0b00000000;    /* Set P90-P97 to output port */
                      /* P90-P97:Unused */

/*-----*/
/*      Setting of Port 10
          */
/*-----*/
P10 =     0b00000000;    /* Set P100-P102 Output latch to Low */
PM10 =    0b11111000;    /* Set P100-P102 to output port */
                      /* P100-P102:Unused */

/*-----*/
/*      Setting of Port 11
          */
/*-----*/
P11 =     0b00000000;    /* Set P110-P111 Output latch to Low */
PM11 =    0b11111100;    /* Set P110-P111 to output port */
                      /* P110-P111:Unused */

/*-----*/
/*      Setting of Port 12
          */
/*-----*/
P12 =     0b00000000;    /* Set P120 Output latch to Low */
PM12 =    0b11111110;    /* Set P120 to output port */
                      /* P120-P124:Unused */
                      /* *P121-P124:Input port */

/*-----*/
/*      Setting of Port 13
          */
/*-----*/
P13 =     0b00000000;    /* Set P130 Output latch to Low */
                      /* P130:Unused */

/*-----*/
/*      Setting of Port 14
          */

```

```

/*-----*/
    P14 =      0b00000000;      /* Set P140-P147 Output latch to Low */
    PM14 =     0b00000000;      /* Set P140-P147 to output port */
                                /* P140-P147:Unused */

/*-----*/
/*      Setting of Port 15
          */
/*-----*/
    P15 =      0b00000000;      /* Set P150-P152,P157 Output latch to Low */
    PM15 =     0b11111111;      /* Set P150-P152,P157 to input port */
                                /* P150-P152,P157:Unused */
}

/*-----*/
/* Module:      fn_InitLvi
                */
/* Description:  Ensures 2.7V to the power supply voltage      */
/* parameter:  --
                */
/* return   :  --
                */
/*-----*/
static void    fn_InitLvi(void)
{
    unsigned char ucCounter;      /* counter */

    /* setting of Low-Voltage Detector */
    LVIMK      = 1;      /* disable LVI interrupt */
    LVISEL     = 0;      /* detects level of VDD */
    LVIS =     0b00001001;      /* Low-Voltage Detection Level Select Register */
    /*|||+---- : Detection level */
    /*||| : 0 0 0 0 : VLVI0 (4.22V) */
    /*||| : 0 0 0 1 : VLVI1 (4.07V) */
    /*||| : 0 0 1 0 : VLVI2 (3.92V) */
    /*||| : 0 0 1 1 : VLVI3 (3.76V) */
    /*||| : 0 1 0 0 : VLVI4 (3.61V) */
    /*||| : 0 1 0 1 : VLVI5 (3.45V) */
    /*||| : 0 1 1 0 : VLVI6 (3.30V) */
    /*||| : 0 1 1 1 : VLVI7 (3.15V) */
    /*||| : 1 0 0 0 : VLVI8 (2.99V) */
    /*||| : 1 0 0 1 : VLVI9 (2.84V) */
    /*||| : 1 0 1 0 : VLVI10 (2.68V) */

```



```

/*|||| : 1 0 1 1 : VLVI11 (2.53V) */
/*|||| : 1 1 0 0 : VLVI12 (2.38V) */
/*|||| : 1 1 0 1 : VLVI13 (2.22V) */
/*|||| : 1 1 1 0 : VLVI14 (2.07V) */
/*|||| : 1 1 1 1 : VLVI15 (1.91V) */
/*|||| */
/*++++----- : Be sure to set 0000 */
LVIMD      = 0;      /* generates an internal interrupt signal when detect the low-voltage */
LVION      = 1;      /* enables low-voltage detection operation */

/* software to wait for the operation stabilization time (over 10us) */
for(ucCounter = 0; ucCounter < 4; ucCounter++){
    NOP();
}

/* wait for VDD to become VLVI or more */
while(LVIF){
    NOP();
}
LVION      = 0;      /* disables low-voltage detection operation */
}

/*-----*/
/* Module:      main
*/
/* Description:  Main process
*/
/* parameter: --
*/
/* return  : --
*/
/*-----*/
void main(void)
{
/*-----*/
/*      Initialization of variables      */
/*-----*/
bEventKeyOn = CEVENT_OFF; /* initialize key on event flag */
ucKeyCode = CKEY_OFF; /* initialize key code */
ucOldKeyCode = CKEY_OFF; /* initialize old key code */
ucChatterKey = CKEY_OFF; /* initialize chattering key code */
ucChatterCounter = CCHATTERCOUNT; /* initialize chattering counter */

```

```

/*=====*/
/*-----*/
/*      Main Loop      */
/*-----*/
/*=====*/
while(1){
    if(TMIF00){
        TMIF00 = 0;
        bKeySenseReq = 1;    /* key sense request ON */
    }

    fn_KeyProcessing(); /* key process */

    /*-----*/
    /*      The main processing writes here      */
    /*      if there is something                */
    /*-----*/

}
}

/*-----*/
/* Module:      fn_KeyProcessing
*/
/* Description:  Key sense and place key on event
*/
/* parameter: --
*/
/* return : key code(ucKeyCode)
*/
/*      key on event(bEventKeyOn)
*/
/*
*/
/* P7 key conection bit:76543210
*/
/*
*/
/*      |||||+---Key0
*/
/*
*/
/*      |||||+---Key1
*/
/*
*/
/*      |||||+-----Key2
*/

```

```

/*                                     |||+-----Key3
                                     */
/*                                     |||+-----Key4
                                     */
/*                                     +++-----no key
                                     */
/*-----*/
static void    fn_KeyProcessing(void)
{
    unsigned char work;

    /*-----*/
    /*      key sense process      */
    /*-----*/
    if(bKeySenseReq){
        /* key sense request on */
        bKeySenseReq = 0; /* clear key sense request */
        work = (P7 & 0x1f); /* get key return data */
        if(ucChatterKey == work){
            ucChatterCounter--; /* remove noise */
            if(ucChatterCounter == 0){ /* noise removal completion */
                ucChatterCounter = CCHATTERCOUNT; /* noise remove counter initialize */

                /* set Key Code */
                switch(ucChatterKey){
                    case 0b00011111:
                        ucKeyCode = CKEY_OFF; /* all key off */
                        break;

                    case 0b00011110:
                        ucKeyCode = CKEY_0; /* key0 */
                        break;

                    case 0b00011101:
                        ucKeyCode = CKEY_1; /* key1 */
                        break;

                    case 0b00011011:
                        ucKeyCode = CKEY_2; /* key2 */
                        break;

                    case 0b00010111:
                        ucKeyCode = CKEY_3; /* key3 */
                }
            }
        }
    }
}

```

```

        break;

    case 0b00001111:
        ucKeyCode = CKEY_4; /* key4 */
        break;

    /*-----*/
    /* if you need any combined key process,      */
    /* add "case" with combination of bits here */
    /*-----*/

    default:
        ucKeyCode = CKEY_OFF; /* all key off */
        break;
    }
}

else{
    /* restart key sense */
    ucChatterKey = work; /* get KR */
    ucChatterCounter = CCHATTERCOUNT; /* noise remove counter initialize */
}

}

/*****
/*      place key on event      */
/*****/
bEventKeyOn = CEVENT_OFF; /* clear key on event */

if(ucKeyCode != ucOldKeyCode){
    /* a key turns into ON from OFF, turn on a key ON event flag */
    if(ucKeyCode != CKEY_OFF){
        bEventKeyOn = CEVENT_ON; /* set key on event */
    }
    ucOldKeyCode = ucKeyCode; /* save key code */
}
}
}

```

付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	September 2009	-	-

【発行】NECエレクトロニクス株式会社 (<http://www.necel.co.jp/>)

【問い合わせ先】 <http://www.necel.com/contact/ja/>