

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い申し上げます。

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（タッチセンサ）

タッチセンサ・プログラム編

この資料は、サンプル・プログラムの動作概要や使用方法について説明したものです。本サンプル・プログラムは、ポート機能と内蔵タイマを使用して静電容量を測定することによりタッチセンサの状態を検知し、LCDにタッチ位置の表示を行うものです。

対象デバイス

78K0R/LH3マイクロコントローラ

目 次

第1章 概 要 ...	3
第2章 位置検出アルゴリズム...	4
2.1 容量測定...	4
第3章 回路イメージ ...	7
3.1 回路イメージ...	7
3.2 マイコン以外の使用デバイス...	8
第4章 ソフトウェアについて ...	9
4.1 ファイル構成...	9
4.2 使用する内蔵周辺機能...	10
4.3 初期設定と動作概要...	10
4.4 フロー・チャート...	12
第5章 設定方法について ...	14
5.1 使用する周辺の初期設定...	14
5.2 メイン処理...	19
第6章 デバイスでの動作確認例...	24
第7章 関連資料 ...	27
付録A プログラム・リスト...	28
付録B 改版履歴...	72

資料番号 U20227JJ1V0AN00（第1版）

発行年月 February 2010 NS

- ・本資料に記載されている内容は2010年2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。
- 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

本サンプル・プログラムは、ポート機能およびタイマを利用して静電容量測定を行うことにより、タッチセンサ機能を実現するものです。タッチセンサのもつ静電容量に対して充電を行い、その充電にかかる時間を静電容量として処理を行います。タッチセンサへ指先が触れるとタッチセンサの静電容量が増加する現象を利用して、タッチの有無を検知します。

2点のタッチ情報からマトリクス上の位置を算出し、LCD上にタッチ位置を表示します。

第2章 位置検出アルゴリズム

この章では、このサンプル・プログラムで使用する位置検出のアルゴリズムを説明します。位置検出は、個々の容量の充電を行いその充電時間を測定することで容量を測定し容量の変化を検出するという手順で行います。

2.1 容量測定

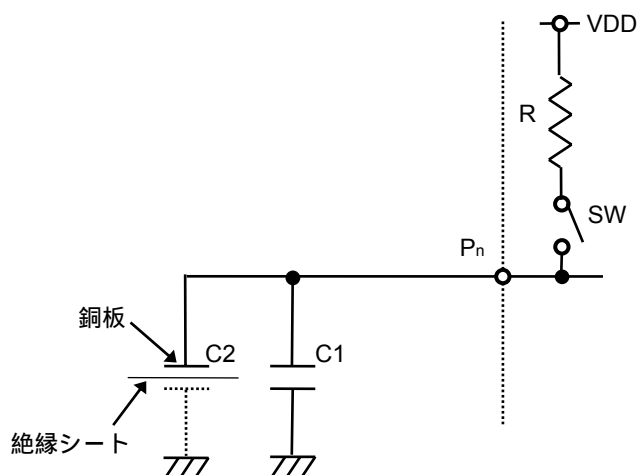


図2 - 1 - 1 タッチ・ポイント回路例

図2 - 1 - 1のような回路を使用します。

コンデンサC1はパッドへの配線等の浮遊容量です。C2は銅板を絶縁シート上から指でタッチした際に発生する容量で、タッチされていない時は容量は殆ど0です。タッチされるとC2の容量が発生しC1、C2の合成容量が増加します。

P_nはマイコンの入出力ポート、Rはマイコンの内蔵プルアップ抵抗、SWは入力設定にするとON、出力設定にするとOFFするマイコン内蔵のスイッチです。

容量の測定はP_nにロウ・レベルを出力し、C1、C2を十分放電した状態でポートP_nを入力設定にすると、SWがONになり、C1、C2の充電が始まります。C2がある（タッチ有）時は充電時間が長くなり、C2が無い（タッチ無）時は充電時間が短くなります。この充電時間の長短によりタッチの有無を判定します。

(1) コンデンサの充電時間を測定

コンデンサの充電時間を測定するために、図2 - 1 - 1の回路を使用します。コンデンサの放電時間はタイマ・アレイ・ユニット0のチャンネル6の入力パルス間隔測定の機能とINTP_nの立ち上がりエッジ検出を使用し、P_nがロウ・レベルの時間を測定します。タイマ・アレイ・ユニット0のチャンネル6の設定の詳細は、5.1 使用する周辺の初期設定 を参照してください。

備考 コンデンサの容量（銅板の面積）は、充電時間を測定する際にタイマ・アレイ・ユニット0のチャンネル6のオーバフローが発生しないよう、また、C2の有無が判定できるように選定しています。

放電時間を測定する手順の概要は以下の通りです。

P_nをロウ・レベル出力にし、コンデンサを放電します。

P_nをINTP_n端子（INTP_nは立ち上がりエッジ検出に設定）として使用し、かつ内蔵プルアップ抵抗を有効にするため入力ポートに設定します。

入力ポートに設定されると同時に内蔵プルアップ抵抗が有効になり、内蔵プルアップ抵抗を介してコンデンサの充電を開始します。

のコンデンサの充電の開始と同時に、タイマ・アレイ・ユニット0のチャンネル6を動作許可し、充電時間の測定を開始します。

コンデンサの充電が終了する（INTP_nの発生）までウェイトします。

タイマ・アレイ・ユニット0のチャンネル6から、測定したコンデンサの充電時間を取得します。

タイマ・アレイ・ユニット0のチャンネル6を動作禁止とします。

(2) タッチされた位置を算出

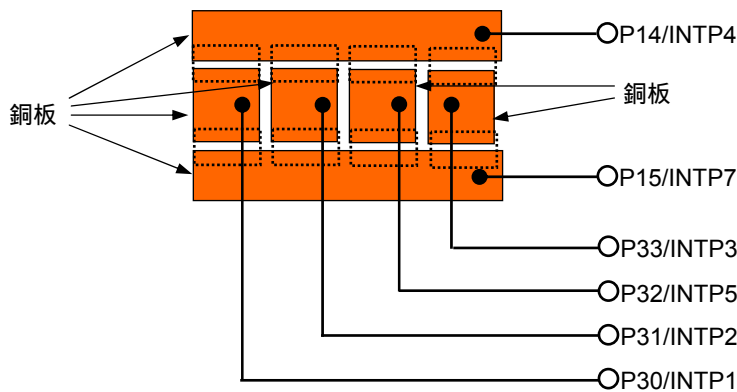


図2 - 1 - 2タッチ・ポイント・マトリクス例

図2 - 1 - 2タッチ・ポイント・マトリクス例のように銅板を各ポート毎に配置します。

この銅板の静電容量をポート毎に充電時間により測定し基準値より一定時間充電時間が長い場合をタッチありと認識します。

基準値はポート毎に変数を用意し、静電容量測定毎に基準値と測定値を比較し、基準値より測定値が小さい場合、基準値に測定値をコピーします。この操作により基準値は測定値中の最低値となります。

尚、本操作だけではノイズ等により一時的にそれまでの基準値より短い測定値があった場合でもその値を基準値としてしまい、以後、基準値が正常な値に戻らなくなりますので、このようなケースへの対策が必要です。

タッチありと認識するための基準値からの充電時間の差は銅板の面積（静電容量）により異なります。本アプリケーションでは $0.25\mu\text{s}$ 以上基準値より長い場合をタッチありと認識しています。

全てのポート測定後、タッチありと認識したポートの組み合わせによりタッチ位置を算出します。

例としてP14とP30にタッチが認識された場合、図2 - 1 - 2タッチ・ポイント・マトリクス例の の位置のタッチを検出します。

本アプリケーションでは、タッチありと認識したポートが2本で、かつ図2 - 1 - 2の ~ のポートの組み合わせの場合をマトリクス上のタッチありとしています。

これ以外の組み合わせの2本以上のタッチありの認識があった場合は多重押しとしています。また、タッチありと認識したポートの数が0と1の場合をマトリクス上のタッチなしとしています。

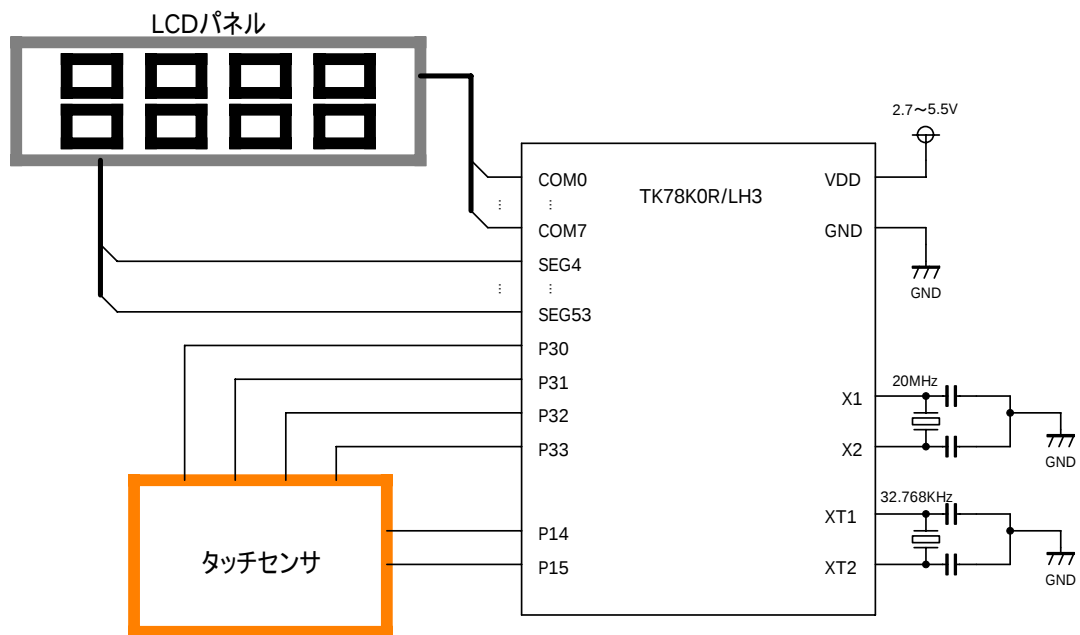
以上のタッチ位置検出を15ms毎に行い、前回の位置検出情報と比較し一致した場合の位置を最終位置情報としています。

第3章 回路イメージ

この章では、このサンプル・プログラムを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について説明します。

3.1 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} , および AV_{DD1} 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. 設定方法の詳細は, TK-78K0R/LH3+LCD ユーザーズ・マニュアルを参照してください。

3.2 マイコン以外の使用デバイス

マイコン以外の使用デバイスを次に示します。

(1) LCD

TK-78K0R/LH3+LCD上のLCDを使用します。

第4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を説明します。

4.1 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.c	マイコンのハードウェア初期化処理，メイン処理のC言語ソース・ファイル	●	●
displayl.c	LCDドライバ初期化，LCD表示処理のC言語ソース・ファイル		
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・タイマの動作設定，低速内蔵発振器の設定などを行います)	●	●
78K0RLx3_sample_program.prw	統合開発環境 PM plus用ワーク・スペース・ファイル		●
78K0RLx3_sample_program.prj	統合開発環境 PM plus用プロジェクト・ファイル		●

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM plusで使用するファイルを同封

4.2 使用する内蔵周辺機能

このサンプル・プログラムで使用するマイコンに内蔵する周辺機能について記述します。

- ・ 高速システム・クロック発振回路
CPUクロック，周辺ハードウェア・クロック用。
- ・ タイマ・アレイ・ユニット0 (TAU0) のチャンネル4
ハードウェア動作安定待ち，浮遊容量への充電待ちのインターバル・タイマとして使用します。
- ・ タイマ・アレイ・ユニット0 (TAU0) のチャンネル6
P14-P15,P30-P33の立ち上がり時間を計測するタイマとしてキャプチャ・モードで使用します。
- ・ P14-P15,P30-P33
タッチセンサを接続します。
- ・ INTP1-INTP5,INTP7
タッチセンサから電荷を入力し、立ち上がりエッジ検出で使用します。
- ・ LCDコントローラ/ドライバ
ボード上のLCDに測定したカウント値を表示します。

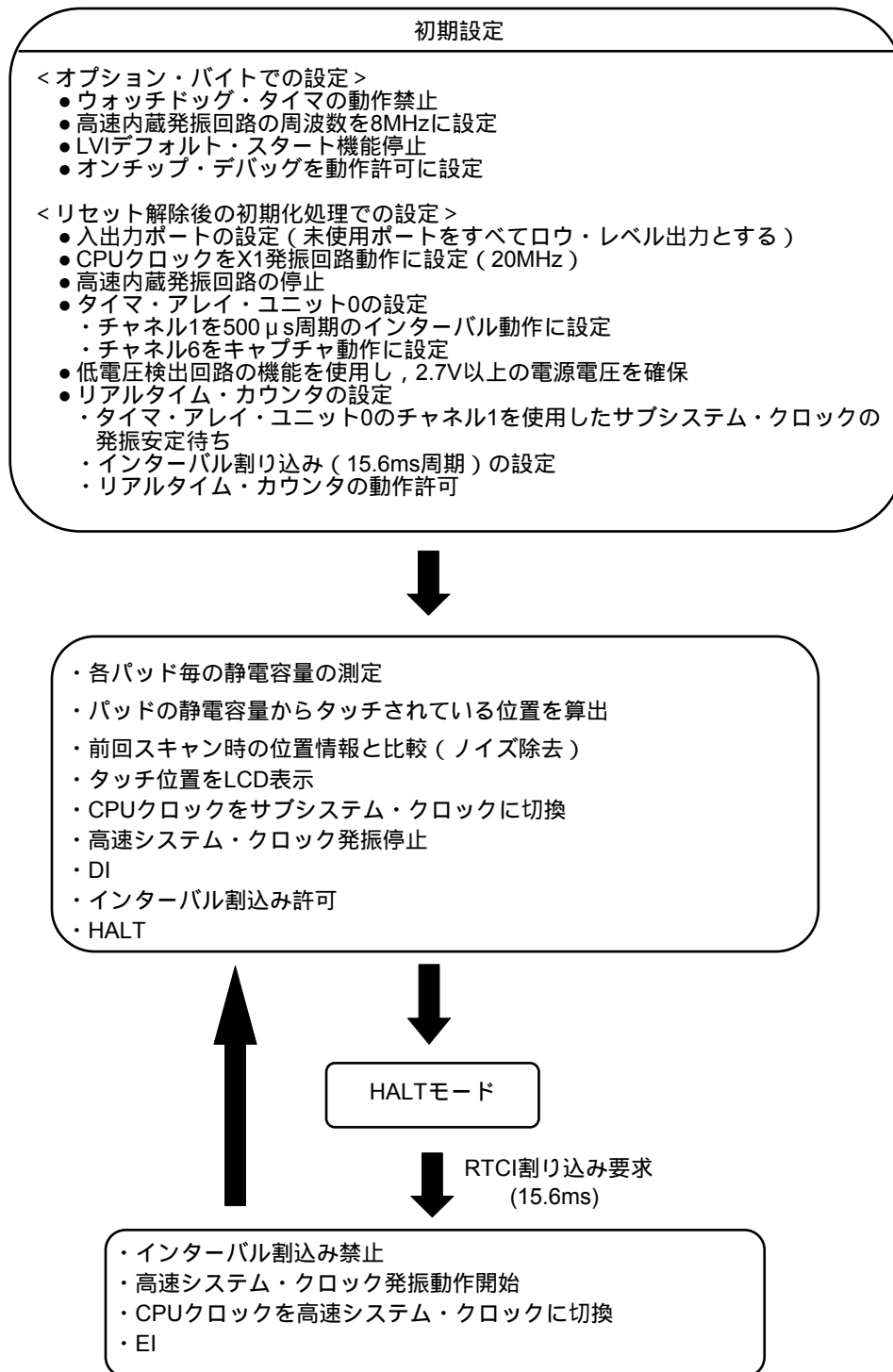
4.3 初期設定と動作概要

このサンプル・プログラムでは，使用する周辺の初期設定にて，ポートの設定や，クロック周波数の選択，タイマ・アレイ・ユニット0，LCDコントローラ/ドライバの設定などを行います。使用する周辺の初期設定完了後，タッチセンサの検出処理を行います。

タッチセンサの検出処理では，対象のポートをロウ出力に切り替えて一定時間ウエイトすることで浮遊容量に充電を行います。その後，対象ポートを入力に切り替えて浮遊容量を放電させるとともにタイマ・アレイ・ユニット0のチャンネル6のカウント動作を開始します。対象ポートの立ち上がりエッジ検出時にタイマ・アレイ・ユニット0のチャンネル6のカウント数を取得します。以上の処理を，P3.0-P3.3,P1.4-P1.5で実行します。

P3.0-P3.3,P1.4-P1.5の検出結果からタッチされた箇所を判定し，LCDに出力します。

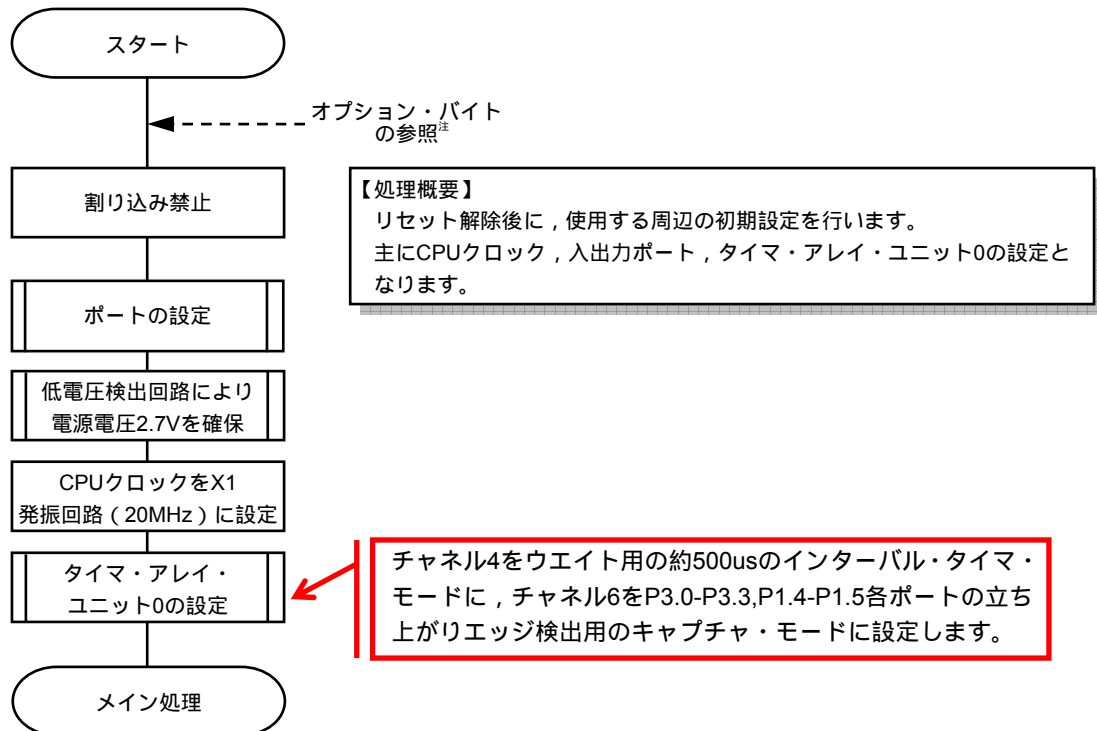
詳細については，次の状態遷移図（ステート・チャート）に示します。



4.4 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。

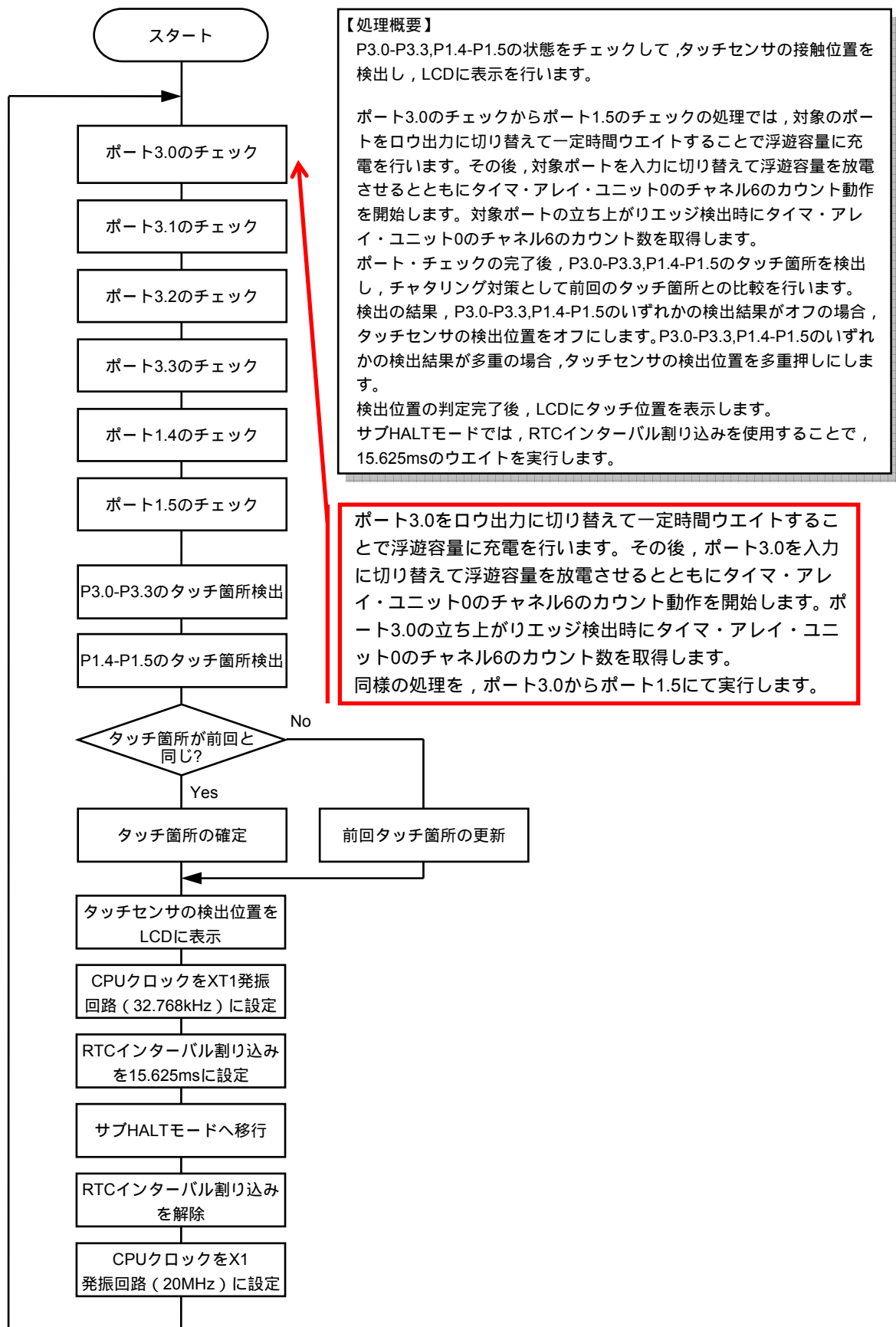
<リセット解除後の初期化処理での設定>



注. 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

<メイン処理>



第5章 設定方法について

この章では、使用する周辺の初期設定、およびサンプル・プログラムの処理内容について説明します。

オプション・バイト、クロック周波数、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/Lx3）とサンプル・プログラムを参照してください。

5.1 使用する周辺の初期設定

(1)リアルタイム・カウンタの初期設定

リアルタイム・カウンタの初期設定処理では、次の動作を行います。

リアルタイム・カウンタへの入力クロックを供給します。

リアルタイム・カウンタを下記の通り設定します。

- ・リアルタイム・カウンタの動作 : カウンタ動作停止
- ・RTC1HZ端子の出力 : RTC1HZ端子の出力（1 Hz）禁止
- ・RTCCL端子の出力 : RTCCL端子の出力（32.768 kHz）禁止
- ・12時間制 / 24時間制 : 24時間制
- ・定周期割り込み（INTRTC） : 0.5秒に1度（秒カウントアップに同期）

リアルタイム・カウンタ割り込みを禁止します。

```

void    fn_InitRtc(void)
{
-----    RTCEN = 1;                /* supplies operational real-time counter (RTC) input clock. */
-----
    RTCC0 = 0b00001010;            /* Real-Time Counter Control Register 0 */
    /*|||||+++--- : Constant-period interrupt (INTRTC) selection */
    /*||||| : 0 0 0 : Does not use constant-period interrupt function. */
    /*||||| : 0 0 1 : Once per 0.5 s */
    /*||||| : 0 1 0 : Once per 1 s */
    /*||||| : 0 1 1 : Once per 1 m */
    /*||||| : 1 0 0 : Once per 1 hour */
    /*||||| : 1 0 1 : Once per 1 day */
    /*||||| : 1 1 x : Once per 1 month */
    /*||||| x = don't care */
    /*||||| */
    /*|||+----- : Selection of 12-/24-hour system */
    /*||| : 0 : 12-hour system */
    /*||| : 1 : 24-hour system */
    /*||| */
    /*|||+----- : RTCCL pin output control */
    /*||| : 0 : Disables output of RTCCL pin (32 kHz). */
    /*||| : 1 : Enables output of RTCCL pin (32 kHz). */
    /*||| */
    /*||+----- : RTC1HZ pin output control */
    /*|| : 0 : Disables output of RTC1HZ pin (1 Hz). */
    /*|| : 1 : Enables output of RTC1HZ pin (1 Hz). */
    /*|| */
    /*|+----- : Be sure to set 0 */
    /*| */
    /*+----- : Real-time counter operation control */
    /* : 0 : Stops counter operation. */
    /* : 1 : Starts counter operation. */

    RTCMK = 1;                    /* disable RTC interrupt */
    RTCIF = 0;                    /* clear RTC interrupt request flag */
}

```

(2) LCDコントローラ/ドライバの初期設定

LCDコントローラ/ドライバの初期設定処理では、次の動作を行います。

LCD駆動電圧生成回路を内部昇圧方式に設定します。

SEG8-SEG26端子をセグメント出力許可に設定します。

P50-P57, P90-P97, P100-P102, P140-P147端子をセグメント出力に設定します。

入力切り替え制御レジスタを下記の通り設定します。

- ・ TI04/SEG50/P53のシュミット・トリガ・バッファの制御 : 入力禁止
- ・ TI02/SEG51/P52のシュミット・トリガ・バッファの制御 : 入力禁止
- ・ RxD3/SEG53/P50のシュミット・トリガ・バッファの制御 : 入力禁止

LCDのRAM領域をクリアします。

LCDクロック制御レジスタ0を下記の通り設定します。

- ・ LCDソース・クロック (f_{LCD}) : f_{SUB}
- ・ LCDクロック (LCDCL) の選択 : $f_{LCD}/2^7$

V_{LCO} 電圧を5.0Vに設定します。

2msウェイトします。

LCD表示モード・レジスタを下記の通り設定します。

- ・ LCD表示の許可 / 禁止 : セグメント端子 / コモン端子にグランド・レベル
を出力
- ・ 昇圧回路と容量分割回路の動作許可 / 停止 : 動作許可
- ・ 表示データ領域 : Aパターン領域のデータを表示
- ・ LCDコントローラ/ドライバの表示モード : 8時分割数, 1/4バイアス法

500msウェイトします。

LCD表示をオンにします。

```

void fn_DisplayInit(void)
{
    /*-----*/
    /* Initialization of LCD controller/driver */
    /*-----*/
    LCDMD = 0b00010000; /* LCD Mode Register */
    /*++|||++--- : Be sure to set 000000 */
    /* || */
    /* ++----- : LCD drive voltage generator selection */
    /* : 0 0 : External resistance division method */
    /* : 0 1 : Internal voltage boosting method */
    /* : 1 0 : Capacitance split method */
    /* : 1 1 : Setting prohibited */
    SEGEN = 0b00011111; /* Segment Enable Register */
    /*|||+--- : Control segment signal output from pins SEG8-SEG11 */
    /*|||+--- : Control segment signal output from pins SEG12-SEG15 */
    /*|||+--- : Control segment signal output from pins SEG16-SEG19 */
    /*|||+--- : Control segment signal output from pins SEG20-SEG23 */
    /*|||+--- : Control segment signal output from pins SEG24-SEG26 */
    /*||| : 0 : segment signal output disable */
    /*||| : 1 : segment signal output enable */
    /*||| */
    /*+++----- : Be sure to set 000 */
    PFALL = 0b01111111; /* Port Function Register ALL */
    /*|||+--- : Pins P50-P53 port/segment output specification */
    /*|||+--- : Pins P54-P57 port/segment output specification */
    /*|||+--- : Pins P90-P93 port/segment output specification */
    /*|||+--- : Pins P94-P97 port/segment output specification */
    /*|||+--- : Pins P100-P102 port/segment output specification */
    /*|||+--- : Pins P140-P143 port/segment output specification */
    /*|||+--- : Pins P144-P147 port/segment output specification */
    /*| : 0 : Used the pins as port (other than segment output) */
    /*| : 1 : Used the pins as segment output */
    /*| */
    /*+----- : Be sure to set 0 */
    ISC = 0b00000000; /* Input Switch Control Register */
    /*|||+--- : Switching external interrupt (INTP0) input */
    /*||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt (normal
operation). */
    /*||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt */
    /*||| (to measure the pulse widths of the sync break field and sync field). */
    /*||| */
    /*|||+--- : ISC1 Switching channel 7 input of timer array unit TAUS */
    /*||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal operation).
*/
    /*||| : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal detection).
*/
    /*||| */
    /*|||+--- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
    /*||| : 0 : Disables input */
    /*||| : 1 : Enables input */
    /*||| */
    /*|||+--- : TI02/SEG51/P52 pin schmitt-triggered buffer control */
    /*||| : 0 : Disables input */
    /*||| : 1 : Enables input */
    /*||| */
    /*|||+--- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
    /*||| : 0 : Disables input */
    /*||| : 1 : Enables input */
    /*||| */
    /*+++----- : Be sure to set 000 */

    fn_DisplayAllClear(); /* clear all LCD String area */
}

```

```

----- LCDCO = 0b00000011;          /* LCD Clock Control Register */
/*|||||+++--- : LCD clock (LCDCL) selection */
/*||||| : 0 0 0 : fLCD/2^4 */
/*||||| : 0 0 1 : fLCD/2^5 */
/*||||| : 0 1 0 : fLCD/2^6 */
/*||||| : 0 1 1 : fLCD/2^7 */
/*||||| : 1 0 0 : fLCD/2^8 */
/*||||| : 1 0 1 : fLCD/2^9 */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*++|+----- : Be sure to set 000 */
/* || */
/* +----- : LCD source clock (fLCD) selection */
/* : 0 0 : fSUB */
/* : 0 1 : fCLK/2^6 */
/* : 1 0 : fCLK/2^7 */
/* : 1 1 : fCLK/2^8 */

----- VLCD = 0x0A;          /* set LCD boost level to 5V */
/* wait for the reference voltage setup time (2ms(min.)) */
fn_Wait500usBase(2000/500);

----- LCDM = 0b00100111;          /* LCD Display Mode Register */
/*|||||+++--- : LCD controller/driver display mode selection */
/*||||| *When the external resistance division method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Two-time-slice mode & 1/2 bias method */
/*||||| : 0 1 1 : Three-time-slice mode & 1/2 bias method */
/*||||| : 1 0 0 : Static */
/*||||| : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
/*||||| */
/*||||| *When the internal voltage boosting method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| : 1 0 0 : Setting prohibited */
/*||||| : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
/*||||| */
/*||||| *When the capacitor split method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| : 1 0 0 : Setting prohibited */
/*||||| : 1 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| */
/*||||| : Other than above : Setting prohibited
/*||||| */
/*|||+----- : LCD display data area control */
/*||| : 0 0 : Display the data of an A pattern area */
/*||| (lower 4 bits of LCD display data memory) */
/*||| : 0 1 : Display the data of an A pattern area */
/*||| (higher 4 bits of LCD display data memory) */
/*||| : 1 0 : Display the data of an A pattern area and the B pattern area in turn. */
/*||| (The on and off light indication which synchronized */
/*||| in a constant-period interrupt timing of RTC) */
/*||| : 1 1 : Display the data of an A pattern area and the B pattern area in turn. */
/*||| (The on and off light indication which synchronized */
/*||| in a constant-period interrupt timing of RTC) */
/*||| */
/*|||+----- : Voltage boost circuit and capacitor split circuit operation
enable/disable */
/*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation */
/*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation */
/*|| */
/*++----- : LCD display enable/disable */
/* : 0 0 : Output ground level to segment/common pin */
/* : 0 1 : Display off (all segment outputs are deselected.) */
/* : 1 0 : Output ground level to segment/common pin */
/* : 1 1 : Display on */

/* software to wait for the operation stabilization time (over 500ms) */
----- fn_Wait500usBase(500000/500);

[----- SCOC = 1;          /* output deselect level to SEG and LCD waveform to COM */
[----- LCDON = 1;        /* display on */
[----- }

```

5.2 メイン処理

メイン処理では、次の動作を行います。

P1.4-P1.5,P3.0-P3.3を立ち上がりエッジ検出に設定します。

LCD表示を初期化します。

P3.0について下記の処理を実行します。また、P3.1-P3.3,P1.4-P1.5についても同様の処理を実行します。

- (a) P3.0をロウ出力に設定して、浮遊容量に電荷をためる。
- (b) P3.0を入力ポートに設定して、P3.0の立ち上がり時間を測定する。
- (c) タッチセンサOFF時のカウント数よりON時のカウント数が小さかった場合、タッチセンサOFF時のカウント数を更新する。

P3.0-P3.3のタッチ箇所を検出します。

P1.4-P1.5のタッチ箇所を検出します。

前回と今回の検出結果が一致した場合、タッチセンサの検出位置を更新します。

P3.0-P3.3,P1.4-P1.5いずれかの検出結果がオフの場合、タッチセンサの検出位置をオフにします。

P3.0-P3.3,P1.4-P1.5いずれかの検出結果が多重押しの場合、タッチセンサの検出位置を多重押しにします。

タッチセンサの検出位置をLCDに表示します。

CPUクロックをサブクロックに切り替えます。

サブHALTモードへ移行し、約15 ms間ウエイトします。

CPUクロックをメイン・クロックに切り替えます。

```

void main(void)
{
    unsigned char    ucCount;          /* work counter */

    for(ucCount = 0; ucCount<4 ; ucCount++){
        ushColumnOff[ucCount] = 0xffff;          /* Column Notouch data */
        ushColumnCurrent[ucCount] = 0xffff;      /* Column current data */
    }

    for(ucCount = 0; ucCount<2 ; ucCount++){
        ushRowOff[ucCount] = 0xffff;             /* Row Notouch data */
        ushRowCurrent[ucCount] = 0xffff;         /* Row current data */
    }

    ucTouchStatusColumnWork = 0;          /* touch column position work */
    ucTouchStatusRowWork = 0;            /* touch row position work */

    ucTouchStatusColumnLast = 0;         /* last touch column position work */
    ucTouchStatusRowLast = 0;           /* last touch row position work */

    ucTouchStatusColumn = 0;             /* touch column position */
    ucTouchStatusRow = 0;               /* touch row position */

    EGPO |= 0b10111110;                 /* set INTP1-INTP5,INTP7 to Rising edge */
    EGNO &= 0b01000001;                 /* reset INTP1-INTP5,INTP7 to Falling edge */

    /*=====*/
    /*-----*/
    /*          Main Loop          */
    /*-----*/
    /*=====*/

    fn_Display(ucTouchStatusRow, ucTouchStatusColumn);          /* initial display */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/

    while (1){
        /*=====*/
        /* check P3.0(INTP1)          */
        /*=====*/

        P3.0 = 0;          /* P3.0(INTP1) Latch Low */
        PM3.0 = 0;         /* set P3.0 output to discharge */
        fn_Wait500usBase(500/500);          /* wait discharge */

        PIF1 = 0;          /* clear INTP1 request */
        TSOL.6 = 1;        /* start TAU0 CH6 */
        TMIF06 = 0;        /* clear TM6 interrupt request */
        PM3.0 = 1;         /* set P3.0 input to start charge */
        while(PIF1 == 0);  /* wait charge end */
        ushColumnCurrent[0] = TCR06; /* get timer count */
        if(ushColumnOff[0] > ushColumnCurrent[0]){
            ushColumnOff[0] = ushColumnCurrent[0];          /* set touch pad off data */
        }
        TTOL.6 = 1;        /* stop TAU0 CH6 */
    }
}

```

```

/*=====*/
/* check P3.1(INTP2) */
/*=====*/

P3.1 = 0; /* P3.1(INTP2) Latch Low */
PM3.1 = 0; /* set P3.1 output to discharge */
fn_Wait500usBase(500/500); /* wait discharge */

PIF2 = 0; /* clear INTP2 request */
TSOL.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request */
PM3.1 = 1; /* set P3.1 input to start charge */
while(PIF2 == 0); /* wait charge end */
ushColumnCurrent[1] = TCR06; /* get timer count */
if(ushColumnOff[1] > ushColumnCurrent[1]){
    ushColumnOff[1] = ushColumnCurrent[1]; /* set touch pad off data */
}
TTOL.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P3.2(INTP5) */
/*=====*/

P3.2 = 0; /* P3.2(INTP5) Latch Low */
PM3.2 = 0; /* set P3.2 output to discharge */
fn_Wait500usBase(500/500); /* wait discharge */

PIF5 = 0; /* clear INTP5 request */
TSOL.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request */
PM3.2 = 1; /* set P3.2 input to start charge */
while(PIF5 == 0); /* wait charge end */
ushColumnCurrent[2] = TCR06; /* get timer count */
if(ushColumnOff[2] > ushColumnCurrent[2]){
    ushColumnOff[2] = ushColumnCurrent[2]; /* set touch pad off data */
}
TTOL.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P3.3(INTP3) */
/*=====*/

P3.3 = 0; /* P3.3(INTP3) Latch Low */
PM3.3 = 0; /* set P3.3 output to discharge */
fn_Wait500usBase(500/500); /* wait discharge */

PIF3 = 0; /* clear INTP3 request */
TSOL.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request */
PM3.3 = 1; /* set P3.3 input to start charge */
while(PIF3 == 0); /* wait charge end */
ushColumnCurrent[3] = TCR06; /* get timer count */
if(ushColumnOff[3] > ushColumnCurrent[3]){
    ushColumnOff[3] = ushColumnCurrent[3]; /* set touch pad off data */
}
TTOL.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P1.4(INTP4) */
/*=====*/

P1.4 = 0; /* P1.4(INTP4) Latch Low */
PM1.4 = 0; /* set P1.4 output to discharge */
fn_Wait500usBase(500/500); /* wait discharge */
PIF4 = 0; /* clear INTP4 request */
TSOL.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request */
PM1.4 = 1; /* set P1.4 input to start charge */
while(PIF4 == 0); /* wait charge end */
ushRowCurrent[0] = TCR06; /* get timer count */
if(ushRowOff[0] > ushRowCurrent[0]){
    ushRowOff[0] = ushRowCurrent[0]; /* set touch pad off data */
}
TTOL.6 = 1; /* stop TAU0 CH6 */

```

```

/*=====*/
/* check P1.5(INTP7) */
/*=====*/

P1.5 = 0; /* P1.5(INTP7) Latch Low */
PM1.5 = 0; /* set P1.5 output to discharge */
fn_Wait500usBase(500/500); /* wait discharge */

PIF7 = 0; /* clear INTP7 request */
TSOL.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request */
PM1.5 = 1; /* set P1.5 input to start charge */
while(PIF7 == 0); /* wait charge end */
ushRowCurrent[1] = TCR06; /* get timer count */
if(ushRowOff[1] > ushRowCurrent[1]){
    ushRowOff[1] = ushRowCurrent[1]; /* set touch pad off data */
}
TTOL.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check touch position */
/*=====*/

ucTouchStatusColumnWork = 0; /* touch column position work */
ucTouchStatusColumn = 0; /* touch column position */

for(ucCount = 0; ucCount < 4; ucCount++){
    if((ushColumnCurrent[ucCount] - ushColumnOff[ucCount]) > 5){
        /* touch detect in column */
        if(ucTouchStatusColumnWork != 0){
            /* multi position touch detect */
            ucTouchStatusColumnWork = 0xff;
            break;
        }
        else{
            ucTouchStatusColumnWork = ucCount; /* set touch position */
        }
    }
}

for(ucCount = 0; ucCount < 2; ucCount++){
    if((ushRowCurrent[ucCount] - ushRowOff[ucCount]) > 5){
        /* touch detect in row */
        if(ucTouchStatusRowWork != 0){
            /* multi position touch detect */
            ucTouchStatusRowWork = 0xff;
            break;
        }
        else{
            ucTouchStatusRowWork = ucCount; /* set touch position */
        }
    }
}

/* compare last detect position to remove noise */
if((ucTouchStatusColumnLast == ucTouchStatusColumnWork) &&
    (ucTouchStatusRowLast == ucTouchStatusRowWork)){
    ucTouchStatusColumn = ucTouchStatusColumnWork;
    ucTouchStatusRow = ucTouchStatusRowWork;
}

/* renew last status */
ucTouchStatusColumnLast = ucTouchStatusColumnWork;
ucTouchStatusRowLast = ucTouchStatusRowWork;

/* notouch point adjust */
if((ucTouchStatusColumn == 0x00) || (ucTouchStatusRow == 0x00)){
    ucTouchStatusColumn = 0x00;
    ucTouchStatusRow = 0x00;
}

/* multi touch point adjust */
if((ucTouchStatusColumn == 0xff) || (ucTouchStatusRow == 0xff)){
    ucTouchStatusColumn = 0xff;
    ucTouchStatusRow = 0xff;
}

```

```

fn_Display(ucTouchStatusColumn, ucTouchStatusRow);    /* LCD display */

/*=====*/
/* Change to Sub-HALT mode (about 15ms) */
/*=====*/

DI();
CKC = 0b01000000;    /* CPU/peripheral hardware clock is subclock */
NOP();
NOP();

/* Confirming the CPU clock status */
while(CLS != 1){
    NOP();
}

MSTOP = 1;    /* X1 oscillator stopped */

RTCIMK = 0;    /* enable RTCI interrupt */
RTCE = 1;    /* Starts counter operation. */
RTCIIF = 0;    /* clear RTCI interrupt request flag */

HALT();    /* Sub-HALT mode */
NOP();
NOP();

RTCIMK = 1;    /* disable RTCI interrupt */

RTCE = 0;    /* Stops counter operation. */
RTCIIF = 0;    /* clear RTCI interrupt request flag */

MSTOP = 0;    /* X1 oscillator operating */
while(OSTC.7 != 1){    /* wait X1 oscillation stabilization */
    NOP();
}
CKC = 0b00010000;    /* CPU/peripheral hardware clock is main clock */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
    NOP();
}
EI();
}

```

第6章 デバイスでの動作確認例

この章では、タッチセンサの動作確認結果を示します。

(1) タッチセンサOFF時

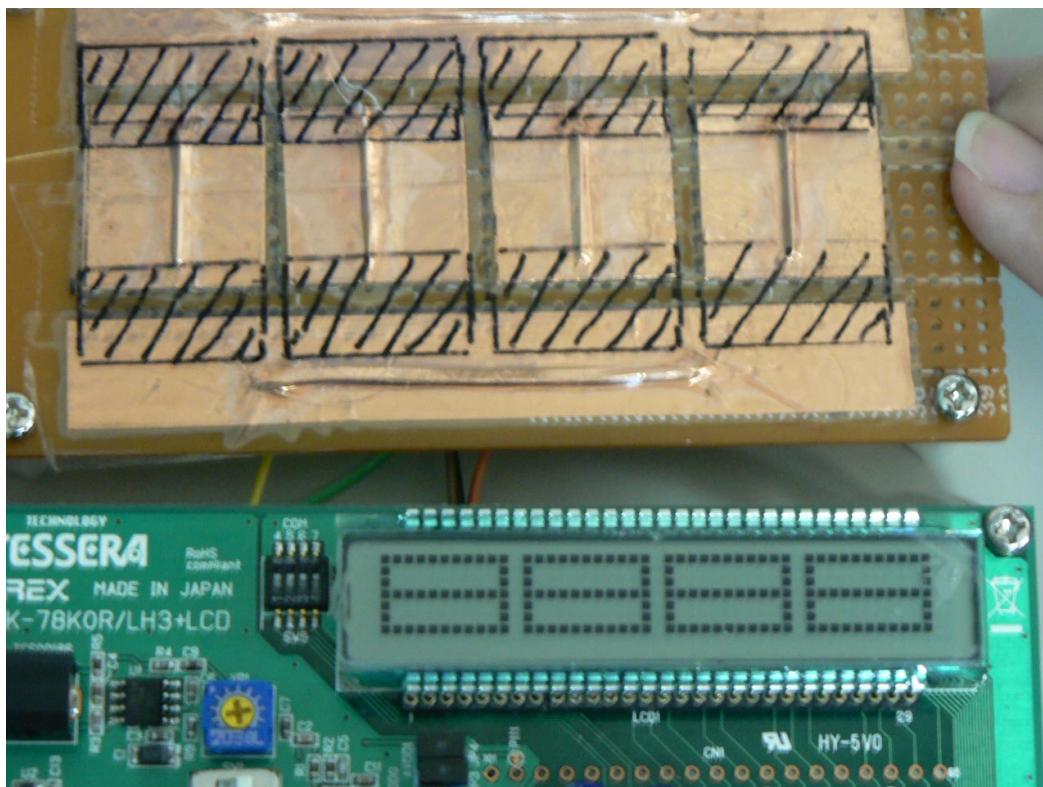


図6-1 タッチセンサOFF時の動作確認結果

(2) タッチセンサON時

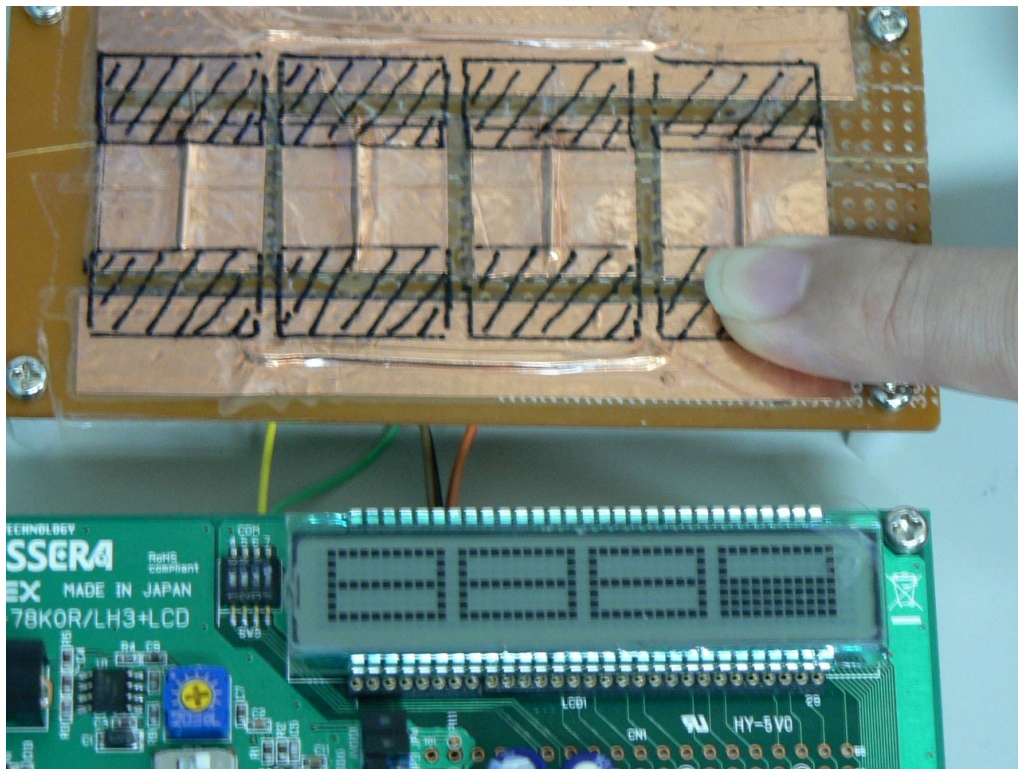


図6-2 タッチセンサOFF時の動作確認結果1

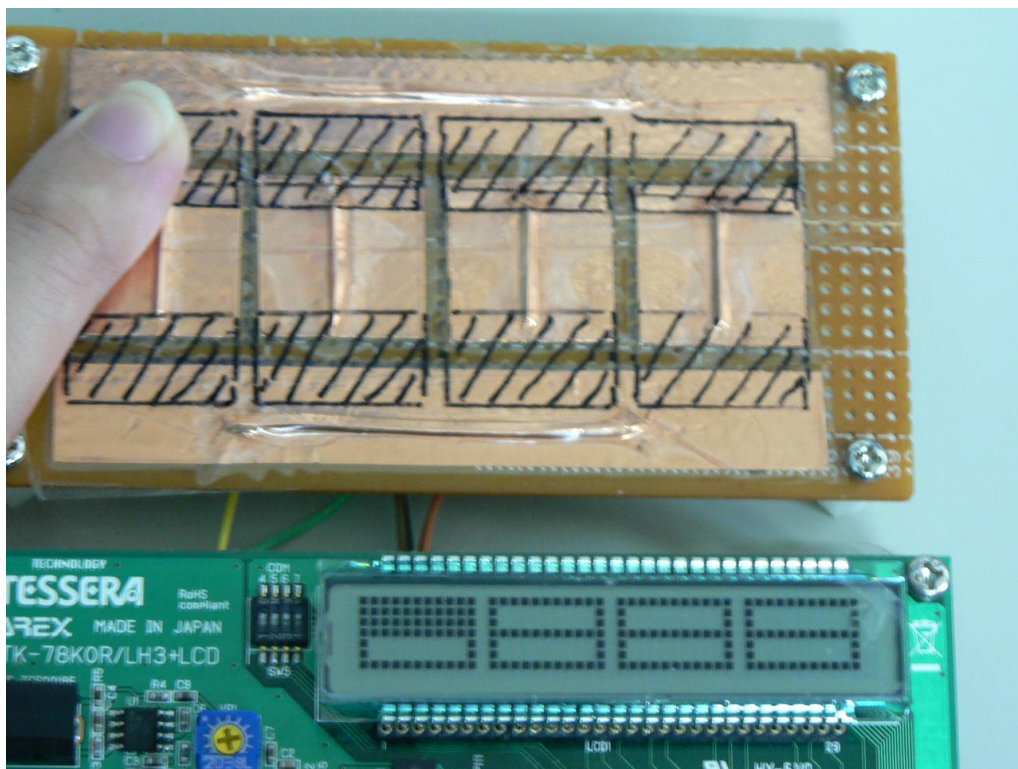


図6-3 タッチセンサOFF時の動作確認結果2

(3) タッチセンサ多重押し時

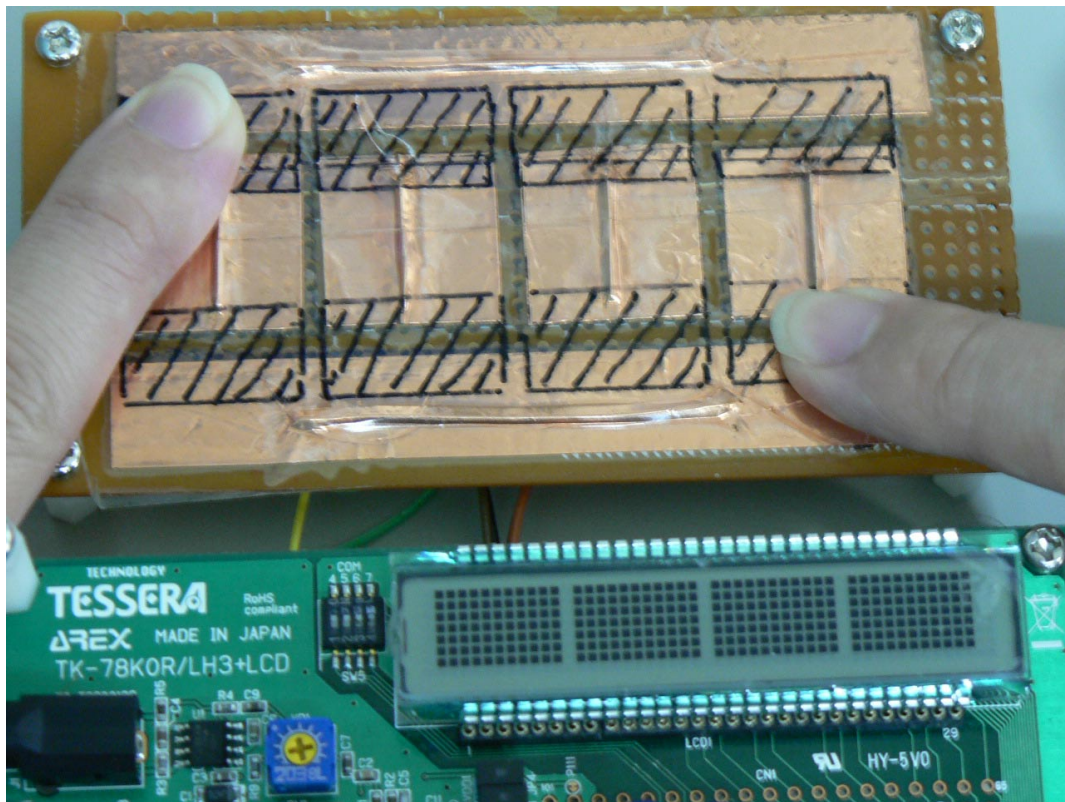


図6-4 タッチセンサ多重押し時の動作確認結果

第7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
TK-78K0R/LH3+LCD ユーザーズ・マニュアル		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

備考 TK-78K0R/LH3+LCDは、テセラ・テクノロジー株式会社の製品です。

問い合わせ先：テセラ・テクノロジー株式会社 (<http://www.tessera.co.jp>)

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.c

```
/*
 * 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      HALT
#pragma      NOP

/*-----*/
/* Include files
                                     */
/*-----*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC  (2500 - 1)      /* 500us (0.2[us/clock] * 2500[count]) */

/*-----*/
/* Function prototyps
```

```

*/
/*-----*/
void    fn_Wait500usBase(unsigned short);    /* Delays the program for (Time * 500us)
*/
void    fn_InitPort(void);                  /* Setting of I/O
ports */
void    fn_InitTau0(void);                  /* Setting of Timer
array unit 0 */
void    fn_InitLvi(void);                  /* Setting of
Low-voltage detector */
void    fn_InitRtc(void);                  /* Setting of
Real-time counter */

extern void    fn_DisplayInit(void);        /* Setting of LCD driver */
extern void    fn_Display(unsigned char ucColumn, unsigned char ucRow); /* LCD display */

/*-----*/
/* Extern variables/constants
*/
/*-----*/

/*-----*/
/* Local constants
*/
/*-----*/

/*-----*/
/* Global variables
*/
/*-----*/

/*-----*/
/* Local variables
*/
/*-----*/

static unsigned short    ushColumnOff[4];    /* Column Notouch data */
static unsigned short    ushRowOff[2];        /* Row Notouch data */

static unsigned short    ushColumnCurrent[4];    /* Column current data */
static unsigned short    ushRowCurrent[2];        /* Row current data */

static unsigned char    ucTouchStatusColumn;    /* touch column position */

```

```

static unsigned char ucTouchStatusRow;          /* touch row position */

static unsigned char ucTouchStatusColumnWork;   /* touch column position work */
static unsigned char ucTouchStatusRowWork;      /* touch row position work */

static unsigned char ucTouchStatusColumnLast;   /* last touch column position work */
static unsigned char ucTouchStatusRowLast;      /* last touch row position work */

/*-----*/
/* Code
*/

/*-----*/

/*-----*/
/* Hardware initialization
*/

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

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

    /*-----*/
    /*      Initialization of clock      */
    /*-----*/
    CMC = 0b01010011;          /* 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 */
XTSTOP = 0;                     /* XT1 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 */

/*-----*/
/*      Initialization of timer      */
/*-----*/
fn_InitTau0();

/* software to wait for the operation stabilization time */
/* (over 200ms from when XT1 enable) */
fn_Wait500usBase(200000/500);

/*-----*/
/* Initialization of low-voltage detector */
/*-----*/
fn_InitLvi();

/*-----*/
/*      Initialization of real-time counter      */
/*-----*/
fn_InitRtc();

/*-----*/
/*      Initialization of LCD                      */
/*-----*/
fn_DisplayInit();

EI();                                /* enable all interrupts */

```

```

}

/*-----*/
/* Module:      fn_InitPort
               */
/* Description:  Setting of I/O ports
               */
/* parameter: --
               */
/* return  : --
               */
/*-----*/
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 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 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 */
/*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
/*||| : 0 0 1 1 0 : A A A A A A A A D D D D D D */
/*||| : 0 0 1 1 1 : A A A A A A A A D D D D D D D */
/*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
/*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
/*||| : 0 1 0 1 0 : A A D D 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 D D D */
/*||| : 1 0 0 0 0 : D D D D D D D D D D D D D D */
/*||| */
/*+++----- : 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-P13:Unused */
        PU1 =    0b00110000;    /* P14,P15 to On-chip pull-up resistor connect */

/*-----*/
/*      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 =     0b00000000;    /* Set P30-P33 Output latch to Low */
                                   /* Set P33,P32 Output latch High */
        PM3 =    0b11110000;    /* Set P30-P33 to output port */
                                   /* Set P34(TI06) to input port */
        PU3 =    0b00001111;    /* P30-P33 to On-chip pull-up resistor connect */

/*-----*/
/*      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 = 0b00000000; /* Set P70-P77 to output port */
                                /* P70-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 = 0b11111111;    /* Set P120 to output port */
                               /* P121-P124:Unused */
                               /* *P120-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_InitTau0
*/

/* Description:   Setting of Timer array unit 0
*/
/*   parameter:  --
*/

/*   return   :  --
*/

/*-----*/
void fn_InitTau0(void)
{
    TAU0EN = 1; /* supplies input clock to timer array unit 0 */
    TPS0L = 0b000000010; /* 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 */

/* CH1:for wait */
    TMR01 = 0b0000000000000000; /* Timer Mode Register 01 */
/*|||||||||++++- : [1]Operation mode of channel 1 */
/*||||||||| [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 TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1
valid edge */

/*||||||| (the timer input used with channel 1 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 1 */
/*||||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||||| : 0 0 1 : Valid edge of TI01 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 TI01 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 1 */
/*||||||| : 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 CKS01 bit */
/*||||||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1
*/

/*||||||| (the timer input used with channel 1 is selected by using TIS0 register).
*/

/*||||||| */

```

```

/*|+----- : Be sure to set 00 */
/*| */
/*+----- : Selection of operation clock (MCK) of channel 1 */
/* : 0 : Operation clock CK00 set by TPS0 register */
/* : 1 : Operation clock CK01 set by TPS0 register */

TDR01 = CCK00_500USEC;          /* set interval time to 500us */
TMMK01 = 1;                      /* disable interrupt */

/* CH6:for charge period measure */
TMR06 = 0b1000000000000100;     /* Timer Mode Register 06 */
/*|+++++ : [1]Operation mode of channel 6 */
/*|      [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 TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1
valid edge */

/*||||||| (the timer input used with channel 4 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 6 */
/*||||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

```

```

/*|||| : 0 0 1 : Valid edge of TI06 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 TI06 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 6 */
/*| : 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 CKS04 bit */
/*| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1
*/

/*| (the timer input used with channel 4 is selected by using TIS0 register).
*/

/*| */
/*|+----- : Be sure to set 00 */
/*| */
/*+----- : Selection of operation clock (MCK) of channel 4 */
/* : 0 : Operation clock CK00 set by TPS0 register */
/* : 1 : Operation clock CK01 set by TPS0 register */

/*      TDR06 = 2000; /* set interval time to 100us(=0.05us * 2000) */
      TMMK06 = 1;      /* disable interrupt */

}

/*-----*/
/* Module:      fn_InitLvi
               */
/* Description:  Setting of Low-voltage detector */
/* parameter: --
               */
/* return  : --
               */
/*-----*/
void fn_InitLvi(void)
{
    unsigned short loop; /* waiting counter */

    LVIMK = 1;           /* disable LVI interrupt */

```

```

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 */

LVIM = 0b10000000; /* Low-Voltage Detection Register */
/*|||+ : LVIF Low-voltage detection flag */
/*||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
/*||| * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
/*||| : 1 : * LVISEL = 0: VDD < VLVI */
/*||| * LVISEL = 1: EXLVI < VEXLVI */
/*||| */
/*|||+ : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD)
*/

/*||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
/*||| when VDD drops lower than VLVI (VDD < VLVI) */
/*||| or when VDD becomes VLVI or higher (VDD >= VLVI).
*/

/*||| * LVISEL = 1: Generates an interrupt signal */
/*||| when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
/*||| or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */

/*||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
/*||| and releases the reset signal when VDD >= VLVI. */
/*||| * LVISEL = 1: Generates an internal reset signal when EXLVI <

```

```

VEXLVI */
                                /*||||| and releases the reset signal when EXLVI >= VEXLVI.
*/
                                /*||||| */
                                /*|||||+----- : Voltage detection selection(LVISEL) */
                                /*||||| : 0 : Detects level of supply voltage (VDD) */
                                /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                                /*||||| */
                                /*|++++----- : Be sure to set 0000 */
                                /*| */
                                /*+----- : Enables low-voltage detection operation */
                                /* : 0 : Disables operation */
                                /* : 1 : Enables operation */

/* software to wait for the operation stabilization time (>210us) */
fn_Wait500usBase(500/500);

/* wait until VLVI VDD */
while( LVIF ){
    NOP();
}

LVIF = 0; /* clear LVI interrupt request flag */
}

/*-----*/
/* Module:      fn_InitRtc
                */
/* Description:   Setting of Real-time counter */
/* parameter: --
                */
/* return : --
                */
/*-----*/
void fn_InitRtc(void)
{
    RTCEN = 1; /* supplies operational real-time counter (RTC) input clock. */

    RTCC0 = 0b00001001; /* Real-Time Counter Control Register 0 */
                        /*|||||++++ : Constant-period interrupt (INTRTC) selection */
                        /*||||| : 0 0 0 : Does not use constant-period interrupt function. */
                        /*||||| : 0 0 1 : Once per 0.5 s */

```

```

/*|||| : 0 1 0 : Once per 1 s */
/*|||| : 0 1 1 : Once per 1 m */
/*|||| : 1 0 0 : Once per 1 hour */
/*|||| : 1 0 1 : Once per 1 day */
/*|||| : 1 1 x : Once per 1 month */
/*|||| x = don't care */
/*|||| */
/*|||+----- : Selection of 12-/24-hour system */
/*|||| : 0 : 12-hour system */
/*|||| : 1 : 24-hour system */
/*|||| */
/*||+----- : RTCCL pin output control */
/*|| : 0 : Disables output of RTCCL pin (32 kHz). */
/*|| : 1 : Enables output of RTCCL pin (32 kHz). */
/*|| */
/*|+----- : RTC1HZ pin output control */
/*| : 0 : Disables output of RTC1HZ pin (1 Hz). */
/*| : 1 : Enables output of RTC1HZ pin (1 Hz). */
/*| */
/*|+----- : Be sure to set 0 */
/*| */
/*+----- : Real-time counter operation control */
/* : 0 : Stops counter operation. */
/* : 1 : Starts counter operation. */

```

RTCC2 = 0b10000011; /* Real-Time Counter Control Register 2 */

```

/*+||||+++- : Constant-period interrupt (INTRTC) selection */
/* |||| : 0 x x x : Interval interrupt is not generated. */
/* |||| : 1 0 0 0 : 2^6/fSUB ( 1.953125 ms) */
/* |||| : 1 0 0 1 : 2^7/fSUB ( 3.90625 ms) */
/* |||| : 1 0 1 0 : 2^8/fSUB ( 7.8125 ms) */
/* |||| : 1 0 1 1 : 2^9/fSUB ( 15.625 ms) */
/* |||| : 1 1 0 0 : 2^10/fSUB ( 31.25 ms) */
/* |||| : 1 1 0 1 : 2^11/fSUB ( 62.5 ms ) */
/* |||| : 1 1 1 x : 2^12/fSUB ( 125 ms ) */
/* |||| x = don't care */
/* |||| */
/* ||++----- : Be sure to set 0 */
/* || */
/*|+----- : Selection of RTCDIV pin output frequency */
/*| : 0 : RTCDIV pin outputs 512 Hz (1.95 ms). */
/*| : 1 : RTCDIV pin outputs 16.384 kHz (0.061 ms). */
/*| */

```

```

/* +----- : RTCDIV pin output control */
/* : 0 : Disables output of RTCDIV pin */
/* : 1 : Enables output of RTCDIV pin */

RTCMK = 1; /* disable RTC interrupt */
RTCIF = 0; /* clear RTC interrupt request flag */

RTCMK = 1; /* disable RTCI interrupt */
RTCIIF = 0; /* clear RTCI interrupt request flag */

}

/*-----*/
/* Module:      fn_Wait500usBase
               */
/* Description:  Delays the program for (Time * 500us) */
/* parameter: wait time(Time)
               */
/* return      : --
               */
/*-----*/
void fn_Wait500usBase(unsigned short Time)
{
    TS0L.1 = 1; /* start TAU0 CH1 */
    TMIF01 = 0;

    for(; Time > 0; Time--){ /* wait for (parameter * 500)us */
        while(!TMIF01){
            NOP();
        }
        TMIF01 = 0;
    }

    TT0L.1 = 1; /* stop TAU0 CH1 */
}

/*-----*/
/* Module:      main
               */
/* Description:  Main process
               */

```

```

/*      parameter: --
                                     */
/*      return   : --
                                     */
/*-----*/
void main(void)
{

    unsigned char    ucCount;          /* work counter */

    for(ucCount = 0; ucCount<4 ; ucCount++){
        ushColumnOff[ucCount] = 0xffff;          /* Column Notouch data */
        ushColumnCurrent[ucCount] = 0xffff;      /* Column current data */
    }

    for(ucCount = 0; ucCount<2 ; ucCount++){
        ushRowOff[ucCount] = 0xffff;             /* Row Notouch data */
        ushRowCurrent[ucCount] = 0xffff;         /* Row current data */
    }

    ucTouchStatusColumnWork = 0;          /* touch column position work */
    ucTouchStatusRowWork = 0;            /* touch row position work */

    ucTouchStatusColumnLast = 0; /* last touch column position work */
    ucTouchStatusRowLast = 0;        /* last touch row position work */

    ucTouchStatusColumn = 0;          /* touch column position */
    ucTouchStatusRow = 0;             /* touch row position */

    EGP0 |= 0b10111110;                /* set  INTP1-INTP5,INTP7  to  Rising
edge */
    EGN0 &= 0b01000001;                /* reset INTP1-INTP5,INTP7 to Falling
edge */

    /*=====*/
    /*-----*/
    /*      Main Loop      */
    /*-----*/
    /*=====*/

```

```

fn_Display(ucTouchStatusRow, ucTouchStatusColumn);          /* initial display */

/*=====*/
/* if system have nothing to do, go to standby for power-saving */
/*=====*/

while (1){

    /*=====*/
    /* check P3.0(INTP1) */
    /*=====*/

    P3.0 = 0;          /* P3.0(INTP1) Latch Low */
    PM3.0 = 0;         /* set P3.0 output to discharge */
*/

    fn_Wait500usBase(500/500); /* wait discharge */

    PIF1 = 0;          /* clear INTP1 request */
    TS0L.6 = 1;        /* start TAU0 CH6 */
    TMIF06 = 0;        /* clear TM6 interrupt request */
*/

    PM3.0 = 1;         /* set P3.0 input to start
charge */

    while(PIF1 == 0);  /* wait charge end */
    ushColumnCurrent[0] = TCR06; /* get timer count */
    if(ushColumnOff[0] > ushColumnCurrent[0]){
        ushColumnOff[0] = ushColumnCurrent[0]; /* set touch pad off data */
    }
    TT0L.6 = 1;        /* stop TAU0 CH6 */

    /*=====*/
    /* check P3.1(INTP2) */
    /*=====*/

    P3.1 = 0;          /* P3.1(INTP2) Latch Low */
    PM3.1 = 0;         /* set P3.1 output to discharge */
*/

    fn_Wait500usBase(500/500); /* wait discharge */

    PIF2 = 0;          /* clear INTP2 request */

```

```

TS0L.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request
*/

PM3.1 = 1; /* set P3.1 input to start
charge */

while(PIF2 == 0); /* wait charge end */
ushColumnCurrent[1] = TCR06; /* get timer count */
if(ushColumnOff[1] > ushColumnCurrent[1]){
    ushColumnOff[1] = ushColumnCurrent[1]; /* set touch pad off data */
}
TT0L.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P3.2(INTP5) */
/*=====*/

P3.2 = 0; /* P3.2(INTP5) Latch Low */
PM3.2 = 0; /* set P3.2 output to discharge
*/

fn_Wait500usBase(500/500); /* wait discharge */

PIF5 = 0; /* clear INTP5 request */
TS0L.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request
*/

PM3.2 = 1; /* set P3.2 input to start
charge */

while(PIF5 == 0); /* wait charge end */
ushColumnCurrent[2] = TCR06; /* get timer count */
if(ushColumnOff[2] > ushColumnCurrent[2]){
    ushColumnOff[2] = ushColumnCurrent[2]; /* set touch pad off data */
}
TT0L.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P3.3(INTP3) */
/*=====*/

P3.3 = 0; /* P3.3(INTP3) Latch Low */
PM3.3 = 0; /* set P3.3 output to discharge
*/

```

```

fn_Wait500usBase(500/500); /* wait discharge */

PIF3 = 0; /* clear INTP3 request */
TS0L.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request
*/

PM3.3 = 1; /* set P3.3 input to start
charge */

while(PIF3 == 0); /* wait charge end */
ushColumnCurrent[3] = TCR06; /* get timer count */
if(ushColumnOff[3] > ushColumnCurrent[3]){
    ushColumnOff[3] = ushColumnCurrent[3]; /* set touch pad off data */
}
TT0L.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P1.4(INTP4) */
/*=====*/

P1.4 = 0; /* P1.4(INTP4) Latch Low */
PM1.4 = 0; /* set P1.4 output to discharge
*/

fn_Wait500usBase(500/500); /* wait discharge */

PIF4 = 0; /* clear INTP4 request */
TS0L.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request
*/

PM1.4 = 1; /* set P1.4 input to start
charge */

while(PIF4 == 0); /* wait charge end */
ushRowCurrent[0] = TCR06; /* get timer count */
if(ushRowOff[0] > ushRowCurrent[0]){
    ushRowOff[0] = ushRowCurrent[0]; /* set touch pad off data */
}
TT0L.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check P1.5(INTP7) */
/*=====*/

```

```

P1.5 = 0; /* P1.5(INTP7) Latch Low */
PM1.5 = 0; /* set P1.5 output to discharge
*/

fn_Wait500usBase(500/500); /* wait discharge */

PIF7 = 0; /* clear INTP7 request */
TS0L.6 = 1; /* start TAU0 CH6 */
TMIF06 = 0; /* clear TM6 interrupt request
*/

PM1.5 = 1; /* set P1.5 input to start
charge */

while(PIF7 == 0); /* wait charge end */
ushRowCurrent[1] = TCR06; /* get timer count */
if(ushRowOff[1] > ushRowCurrent[1]){
    ushRowOff[1] = ushRowCurrent[1]; /* set touch pad off data */
}
TT0L.6 = 1; /* stop TAU0 CH6 */

/*=====*/
/* check touch position */
/*=====*/

ucTouchStatusColumnWork = 0; /* touch column position work */
/*
ucTouchStatusColumn = 0; /* touch column position */

for(ucCount = 0; ucCount < 4; ucCount++){
    if((ushColumnCurrent[ucCount] - ushColumnOff[ucCount]) > 5){
        /* touch detect in column */
        if(ucTouchStatusColumnWork != 0){
            ucTouchStatusColumnWork = 0xff; /* multi position
touch detect */

            break;
        }
        else{
            ucTouchStatusColumnWork = (ucCount + 1);/* set touch
position */

        }
    }
}

ucTouchStatusRowWork = 0; /* touch row position work */
for(ucCount = 0; ucCount < 2; ucCount++){

```

```

        if((ushRowCurrent[ucCount] - ushRowOff[ucCount]) > 5){
            /* touch detect in row */
            if(ucTouchStatusRowWork != 0){
                ucTouchStatusRowWork = 0xff; /* multi position touch detect */
                break;
            }
            else{
                ucTouchStatusRowWork = (ucCount + 1); /* set touch position
*/
            }
        }
    }

    /* compare last detect position to remove noise */
    if((ucTouchStatusColumnLast == ucTouchStatusColumnWork) &&
        (ucTouchStatusRowLast == ucTouchStatusRowWork)
){
        ucTouchStatusColumn = ucTouchStatusColumnWork;
        ucTouchStatusRow = ucTouchStatusRowWork;
    }

    /* renew last status */
    ucTouchStatusColumnLast = ucTouchStatusColumnWork;
    ucTouchStatusRowLast = ucTouchStatusRowWork;

    /* notouch point adjust */
    if((ucTouchStatusColumn == 0x00)|| (ucTouchStatusRow == 0x00)){
        ucTouchStatusColumn = 0x00;
        ucTouchStatusRow = 0x00;
    }

    /* multi touch point adjust */
    if((ucTouchStatusColumn == 0xff)|| (ucTouchStatusRow == 0xff)){
        ucTouchStatusColumn = 0xff;
        ucTouchStatusRow = 0xff;
    }

    fn_Display(ucTouchStatusColumn, ucTouchStatusRow); /* LCD display */

    /*=====*/
    /*  Change to Sub-HALT mode (about 15ms)      */
    /*=====*/

    DI();

```

```

CKC = 0b01000000;          /* CPU/peripheral hardware clock is subclock */
NOP();
NOP();

/* Confirming the CPU clock status */
while(CLS != 1){
    NOP();
}

MSTOP = 1;                  /* X1 oscillator stopped */

RTCE = 1;                   /* Starts counter operation. */
RTCIIF = 0;                 /* clear RTCI interrupt request flag */
RTCIMK = 0;                 /* enable RTCI interrupt */

HALT();                     /* Sub-HALT mode */
NOP();
NOP();

RTCIMK = 1;                 /* disable RTCI interrupt */

RTCE = 0;                   /* Stops counter operation. */
RTCIIF = 0;                 /* clear RTCI interrupt request flag */

MSTOP = 0;                  /* X1 oscillator operating */
while(OSTC.7 != 1){         /* wait X1 oscillation stabilization */
    NOP();
}
CKC = 0b00010000;          /* CPU/peripheral hardware clock is main clock */

/* Confirming the CPU clock status */
while((CLS != 0)|| (MCS != 1)){
    NOP();
}
EI();

}

}

```

- display.c

```

/*
 * 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

/*-----*/
/* Include files
                                     */
/*-----*/
/*#include          "defines.h"*/
#include<string.h>

/*-----*/
/* Function prototyps
                                     */
/*-----*/
          void          fn_DisplayAllClear(void);          /* Display all clear */
extern void          fn_Wait500usBase(unsigned short);    /* Delays the program (500us base) */

/*-----*/
/* Extern variables/constants
                                     */
/*-----*/
/*-----*/
/* Local constants
                                     */
/*-----*/
/* size of area for displaying */

```

```
#define CLCDSIZE_ALL      (unsigned char)(&SEG53 - &SEG4 + 1)  /* all area */
/*#define      CLCDSIZE_NUMBER      4      /* number */
#define CLCDSIZE_NUMBER      5      /* number */
```

/* Display starting position */

```
#define CLCDPOS_START      &SEG4  /* start of all area */
#define CLCDPOS_COLUMN2      &SEG16 /* column 2 start */
#define CLCDPOS_COLUMN3      &SEG28 /* column 3 start */
#define CLCDPOS_COLUMN4      &SEG40 /* column 4 start */
#define CLCDPOS_END_SPACE      &SEG52 /* end space start */
```

/* Display data */

/*=====

< LCD PANEL >

	SEG4	SEG5	SEG6	SEG7	SEG8	= = =	SEG49	SEG50	SEG51	SEG52
SEG53										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM0										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM1										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM2										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM3										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM4										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM5										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM6										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									
COM7										
	+-----+-----+-----+-----+-----+ = = = +-----+-----+-----+-----+-----+									

< example of the data setting >

when you display "A" at area from SEG4 to SEG8

	+-----+-----+-----+-----+-----+									
bit0						<- Blank area is 0.				
	+-----+-----+-----+-----+-----+									
						Not blank area is 1.				

```

bit1 |      |////|      |////|      |
      +---+---+---+---+---+
bit2 |////|      |      |      |////|
      +---+---+---+---+---+
bit3 |////|      |      |      |////|
      +---+---+---+---+---+
bit4 |////|////|////|////|////|
      +---+---+---+---+---+
bit5 |////|      |      |      |////|
      +---+---+---+---+---+
bit6 |////|      |      |      |////|
      +---+---+---+---+---+
bit7 |      |      |      |      |
      +---+---+---+---+---+
      |      |      |      |      |
      0x7C 0x12 0x11 0x12 0x7C
      |      |      |      |      |
SEG4 <---+      |      |      |      |
SEG5 <-----+      |      |      |
SEG6 <-----+      |      |
SEG7 <-----+      |
SEG8 <-----+
Write data to SFR

```

```

=====*/
/*-----*/
/*  Number indication  */
/*-----*/
static const unsigned char aDispNumber[][CLCDSIZE_NUMBER] = {
/* COM 76543210 */
    {0b01111100          /* '0' */
    ,0b10000010
    ,0b10000010
    ,0b01111100
    ,0b00000000}

/* COM 76543210 */
,    {0b00000100          /* '1' */
    ,0b00000100
    ,0b11111110
    ,0b00000000
    ,0b00000000}

```

/* COM 76543210 */

```
,      {0b11000100      /* '2' */
        ,0b10100010
        ,0b10010010
        ,0b10001100
        ,0b00000000}
```

/* COM 76543210 */

```
,      {0b01000100      /* '3' */
        ,0b10010010
        ,0b10010010
        ,0b01101100
        ,0b00000000}
```

/* COM 76543210 */

```
,      {0b01111000      /* '4' */
        ,0b01000100
        ,0b11111110
        ,0b01000000
        ,0b00000000}
```

/* COM 76543210 */

```
,      {0b01001110      /* '5' */
        ,0b10001010
        ,0b10001010
        ,0b01110010
        ,0b00000000}
```

/* COM 76543210 */

```
,      {0b01111100      /* '6' */
        ,0b10010010
        ,0b10010010
        ,0b01100000
        ,0b00000000}
```

/* COM 76543210 */

```
,      {0b00000010      /* '7' */
        ,0b11100010
        ,0b00011010
        ,0b00000110
        ,0b00000000}
```

/* COM 76543210 */

```
/* COM 76543210 */
```

 $\}.$
$$/* \text{-----} */$$

```
/* COM 76543210 */
```

0b00000000

```
,0b01001001          /* | | | */
```

```

,0b01001001      /* | | | */
,0b01111111      /* ----- */
,0b00000000
,0b01111111      /* ----- */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01111111      /* ----- */
,0b00000000
,0b01111111      /* ----- */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01111111      /* ----- */
,0b00000000
,0b00000000
};

static const unsigned char aDispTouchPositionOffFrame[] = {
/* COM 76543210 */
    0b00000000
,0b01111111      /* ----- */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */
,0b01001001      /* | | | */

```

```

        ,0b01111111          /* ----- */
};

static const unsigned char aDispTouchPositionUpperOnFrame[] = {
/* COM 76543210 */
    0b00000000
    ,0b01111111          /* ----- */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01001111          /* | ||| */
    ,0b01111111          /* ----- */
};

```

```

static const unsigned char aDispTouchPositionLowerOnFrame[] = {
/* COM 76543210 */
    0b00000000
    ,0b01111111          /* ----- */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111001          /* ||| | */
    ,0b01111111          /* ----- */
};

```

```

static const unsigned char aDispTouchPositionMultiFrame[] = {
/* COM 76543210 */
    0b00000000
    ,0b01111111          /* ----- */
    ,0b01111111          /* ||||| */
    ,0b01111111          /* ||||| */
    ,0b01111111          /* ||||| */
    ,0b01111111          /* ||||| */

```



```

,0b00000000      /*' '*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000      /*' '*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000      /*' '*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000      /*' '*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000
};

/*-----*/
/* Global variables
*/
/*-----*/

/*-----*/
/* Local variables
*/
/*-----*/

/*-----*/
/* Code
*/

/*-----*/
/*=====*/
/*
*/
/* Common function
*/
/*
*/
/*=====*/
/*-----*/
/* Module:      fn_DisplayInit

```

```

*/
/* Description:      Initialization of Display module */
/*      parameter: -- */
*/
/*      return      : -- */
*/
/*-----*/
void      fn_DisplayInit(void)
{
    /*-----*/
    /* Initialization of LCD controler/driver */
    /*-----*/
    LCDMD = 0b00010000;          /* LCD Mode Register */
        /*++||++++-- : Be sure to set 000000 */
        /*  || */
        /*  ++----- : LCD drive voltage generator selection */
        /*  : 0 0 : External resistance division method */
        /*  : 0 1 : Internal voltage boosting method */
        /*  : 1 0 : Capacitance split method */
        /*  : 1 1 : Setting prohibited */
    SEGEN = 0b00011111;          /* Segment Enable Register */
        /*|||||+--- : Control segment signal output from pins SEG8-SEG11 */
        /*|||||+---- : Control segment signal output from pins SEG12-SEG15 */
        /*|||||+----- : Control segment signal output from pins SEG16-SEG19 */
        /*|||+----- : Control segment signal output from pins SEG20-SEG23 */
        /*|||+----- : Control segment signal output from pins SEG24-SEG26 */
        /*||| : 0 : segment signal output disable */
        /*||| : 1 : segment signal output enable */
        /*||| */
        /*+++----- : Be sure to set 000 */
    PFALL = 0b01111111;          /* Port Function Register ALL */
        /*|||||+--- : Pins P50-P53 port/segment output specification */
        /*|||||+---- : Pins P54-P57 port/segment output specification */
        /*|||||+----- : Pins P90-P93 port/segment output specification */
        /*|||||+----- : Pins P94-P97 port/segment output specification */
        /*|||+----- : Pins P100-P102 port/segment output specification */
        /*||+----- : Pins P140-P143 port/segment output specification */
        /*|+----- : Pins P144-P147 port/segment output specification */
        /*| : 0 : Used the pins as port (other than segment output) */
        /*| : 1 : Used the pins as segment output */
        /*| */
        /*+----- : Be sure to set 0 */
    ISC = 0b00000000;          /* Input Switch Control Register */

```

```

/*|||||+--- : Switching external interrupt (INTP0) input */
/*||||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt (normal
operation). */

/*||||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt */
/*||||| (to measure the pulse widths of the sync break field and sync field). */
/*||||| */
/*|||||+---- : ISC1 Switching channel 7 input of timer array unit TAUS */
/*||||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal operation).
*/

/*||||| : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal detection).
*/

/*||||| */
/*|||||+----- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
/*||||| : 0 : Disables input */
/*||||| : 1 : Enables input */
/*||||| */
/*|||||+----- : TI02/SEG51/P52 pin schmitt-triggered buffer control */
/*||||| : 0 : Disables input */
/*||||| : 1 : Enables input */
/*||||| */
/*|||+----- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
/*||| : 0 : Disables input */
/*||| : 1 : Enables input */
/*||| */
/*+++----- : Be sure to set 000 */

fn_DisplayAllClear();          /* clear all LCD String area */

LCDC0 = 0b00000011;           /* LCD Clock Control Register */
/*|||||+---- : LCD clock (LCDCL) selection */
/*||||| : 0 0 0 : fLCD/2^4 */
/*||||| : 0 0 1 : fLCD/2^5 */
/*||||| : 0 1 0 : fLCD/2^6 */
/*||||| : 0 1 1 : fLCD/2^7 */
/*||||| : 1 0 0 : fLCD/2^8 */
/*||||| : 1 0 1 : fLCD/2^9 */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*++||+----- : Be sure to set 000 */
/* || */
/* ++----- : LCD source clock (fLCD) selection */
/* : 0 0 : fSUB */
/* : 0 1 : fCLK/2^6 */

```

```

/* : 1 0 : fCLK/2^7 */
/* : 1 1 : fCLK/2^8 */

VLCD = 0x0A; /* set LCD boost level to 5V */
/* wait for the reference voltage setup time (2ms(min.)) */
fn_Wait500usBase(2000/500);

LCDM = 0b00100111; /* LCD Display Mode Register */
/*|||||++--- : LCD controller/driver display mode selection */
/*||||| *When the external resistance division method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Two-time-slice mode & 1/2 bias method */
/*||||| : 0 1 1 : Three-time-slice mode & 1/2 bias method */
/*||||| : 1 0 0 : Static */
/*||||| : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
/*||||| */
/*||||| *When the internal voltage boosting method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| : 1 0 0 : Setting prohibited */
/*||||| : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
/*||||| */
/*||||| *When the capacitor split method is used */
/*||||| : 0 0 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 0 1 : Three-time-slice mode & 1/3 bias method */
/*||||| : 0 1 0 : Four-time-slice mode & 1/3 bias method */
/*||||| : 0 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| : 1 0 0 : Setting prohibited */
/*||||| : 1 1 1 : Four-time-slice mode & 1/3 bias method */
/*||||| */
/*||||| : Other than above : Setting prohibited
/*||||| */
/*|||++----- : LCD display data area control */
/*||| : 0 0 : Display the data of an A pattern area */
/*||| (lower 4 bits of LCD display data memory) */
/*||| : 0 1 : Display the data of an A pattern area */
/*||| (higher 4 bits of LCD display data memory) */
/*||| : 1 0 : Display the data of an A pattern area and the B pattern area in turn. */
/*||| (The on and off light indication which synchronized */
/*||| in a constant-period interrupt timing of RTC) */

```

```

/*||| : 1 1 : Display the data of an A pattern area and the B pattern area in turn. */
/*|||      (The on and off light indication which synchronized */
/*|||      in a constant-period interrupt timing of RTC) */
/*||| */
/*||+----- : Voltage boost circuit and capacitor split circuit operation enable/disable
*/

/*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation */
/*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation */
/*|| */
/*++----- : LCD display enable/disable */
/* : 0 0 : Output ground level to segment/common pin */
/* : 0 1 : Display off (all segment outputs are deselected.) */
/* : 1 0 : Output ground level to segment/common pin */
/* : 1 1 : Display on */

/* software to wait for the operation stabilization time (over 500ms) */
fn_Wait500usBase(500000/500);

SCOC = 1;          /* output deselect level to SEG and LCD waveform to COM */
LCDON = 1;         /* display on */

/*-----*/
/*      Initialization of variables      */
/*-----*/
/*      bDispUpdate = 1;          /* initialize display update request flag */
/*      bBlinkOn = 0;             /* initialize blink on flag */
/*      ushOpeningMessage = 0;    /* initialize opening message counter */
}

/*-----*/
/* Module:      fn_LcdWrite
*/
/* Description:  Write data to LCD RAM
*/
/*      parameter: position in which display begins
*/
/*      address of display data
*/
/*      size of display data
*/
/*      return   : --
*/
/*-----*/
static void fn_LcdWrite(unsigned char *Position, unsigned char *DataAddr, unsigned char DataSize)

```

```

{
    memcpy(Position, DataAddr, DataSize);
}

/*-----*/
/* Module:      fn_DisplayAllClear
               */
/* Description:  LCD RAM all clear
               */
/* parameter: --
               */
/* return  : --
               */
/*-----*/
void fn_DisplayAllClear(void)
{
    fn_LcdWrite(CLCDPOS_START, aDispClear, CLCDSIZE_ALL);
}

/*-----*/
/* Module:      fn_Display
               */
/* Description:  Display touch position to LCD
               */
/* parameter: touch column position, touch row position*/
/* return  : --
               */
/*-----*/
void fn_Display(unsigned char ucColumn, unsigned char ucRow)
{
    /*-----*/
    /* column 1 display
    */
    /*-----*/

    if(ucColumn == 1){
        if(ucRow == 1){ /* column 1 ,row 1 position on display */
            fn_LcdWrite(CLCDPOS_START, aDispTouchPositionUpperOnFrame,
sizeof(aDispTouchPositionUpperOnFrame));
        }
        else{
            /* column 1 ,row 2 position on display */
            fn_LcdWrite(CLCDPOS_START, aDispTouchPositionLowerOnFrame,
sizeof(aDispTouchPositionLowerOnFrame));
        }
    }
}

```

```

    }
}
else{
    if(ucColumn == 0xff){/* column 1 multiposition on display */
        fn_LcdWrite(CLCDPOS_START, aDispTouchPositionMultiFrame,
sizeof(aDispTouchPositionMultiFrame));
    }
    else{ /* column 1 position off display */
        fn_LcdWrite(CLCDPOS_START, aDispTouchPositionOffFrame,
sizeof(aDispTouchPositionOffFrame));
    }
}

/*-----*/
/*      column 2 display      */
/*-----*/

if(ucColumn == 2){
    if(ucRow == 1){ /* column 2 ,row 1 position on display */
        fn_LcdWrite(CLCDPOS_COLUMN2, aDispTouchPositionUpperOnFrame,
sizeof(aDispTouchPositionUpperOnFrame));
    }
    else{ /* column 2 ,row 2 position on display */
        fn_LcdWrite(CLCDPOS_COLUMN2, aDispTouchPositionLowerOnFrame,
sizeof(aDispTouchPositionLowerOnFrame));
    }
}
else{ /* column 2 multiposition on display */
    if(ucColumn == 0xff){
        fn_LcdWrite(CLCDPOS_COLUMN2, aDispTouchPositionMultiFrame,
sizeof(aDispTouchPositionMultiFrame));
    }
    else{ /* column 2 position off display */
        fn_LcdWrite(CLCDPOS_COLUMN2, aDispTouchPositionOffFrame,
sizeof(aDispTouchPositionOffFrame));
    }
}

/*-----*/
/*      column 3 display      */
/*-----*/

```

```

        if(ucColumn == 3){
            if(ucRow == 1){          /* column 3 ,row 1 position on display */
                fn_LcdWrite(CLCDPOS_COLUMN3,          aDispTouchPositionUpperOnFrame,
sizeof(aDispTouchPositionUpperOnFrame));
            }
            else{                    /* column 3 ,row 2 position on display */
                fn_LcdWrite(CLCDPOS_COLUMN3,          aDispTouchPositionLowerOnFrame,
sizeof(aDispTouchPositionLowerOnFrame));
            }
        }
        else{                        /* column 3 multiposition on display */
            if(ucColumn == 0xff){
                fn_LcdWrite(CLCDPOS_COLUMN3,          aDispTouchPositionMultiFrame,
sizeof(aDispTouchPositionMultiFrame));
            }
            else{                    /* column 3 position off display */
                fn_LcdWrite(CLCDPOS_COLUMN3,          aDispTouchPositionOffFrame,
sizeof(aDispTouchPositionOffFrame));
            }
        }

        /*-----*/
        /*      column 4 display      */
        /*-----*/

        if(ucColumn == 4){
            if(ucRow == 1){          /* column 4 ,row 1 position on display */
                fn_LcdWrite(CLCDPOS_COLUMN4,          aDispTouchPositionUpperOnFrame,
sizeof(aDispTouchPositionUpperOnFrame));
            }
            else{                    /* column 4 ,row 2 position on display */
                fn_LcdWrite(CLCDPOS_COLUMN4,          aDispTouchPositionLowerOnFrame,
sizeof(aDispTouchPositionLowerOnFrame));
            }
        }
        else{                        /* column 4 multiposition on display */
            if(ucColumn == 0xff){
                fn_LcdWrite(CLCDPOS_COLUMN4,          aDispTouchPositionMultiFrame,
sizeof(aDispTouchPositionMultiFrame));
            }
            else{                    /* column 4 position off display */
                fn_LcdWrite(CLCDPOS_COLUMN4,          aDispTouchPositionOffFrame,
sizeof(aDispTouchPositionOffFrame));
            }
        }
    }
}

```

```

        }
    }

    /*-----*/
    /*      remain area display      */
    /*-----*/

    fn_LcdWrite(CLCDPOS_END_SPACE, aDispClear, 2);          /* remain all space */

}

```

付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	February 2010	-	-

【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/>