

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

アプリケーション・ノート

78K0R/LH3

サンプル・プログラム（オペアンプ）

オペアンプ応用プログラム編

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、内蔵オペアンプの応用例です。オペアンプに抵抗やキャパシタを接続し、内蔵D/Aコンバータを使用した交流、直流信号を入力した時の動作例を示しています。また、抵抗、キャパシタを接続した発振回路の動作例も示しています。

対象デバイス

78K0R/LH3マイクロコントローラ

目次

<p>第1章 概要 ... 3</p> <p>1.1 非反転増幅回路...3</p> <p>1.2 反転増幅回路...5</p> <p>1.3 反転増幅回路 オフセット調整...7</p> <p>1.4 反転増幅回路 オフセット調整 + ロウパス フィルター...9</p> <p>1.5 非反転増幅回路 交流...11</p> <p>1.6 反転増幅回路 交流...13</p> <p>1.7 インストルメンテーションアンプ...15</p> <p>1.8 比較回路...17</p> <p>1.9 CR発振回路1...19</p> <p>1.10 CR発振回路2...21</p> <p>第2章 回路イメージ ...23</p> <p>2.1 回路イメージ...23</p> <p>2.2 マイコン以外の使用デバイス...33</p> <p>第3章 ソフトウェアについて ...34</p> <p>3.1 ファイル構成...34</p> <p>3.2 使用する内蔵周辺機能...40</p> <p>3.3 初期設定と動作概要...46</p> <p>3.4 フロー・チャート...56</p>	<p>第4章 設定方法について ...76</p> <p>4.1 非反転増幅回路...76</p> <p>4.2 反転増幅回路...82</p> <p>4.3 反転増幅回路 オフセット調整...88</p> <p>4.4 反転増幅回路 オフセット調整 + ロウパスフィルター...95</p> <p>4.5 非反転増幅回路 交流...102</p> <p>4.6 反転増幅回路 交流...109</p> <p>4.7 インストルメンテーションアンプ...116</p> <p>4.8 比較回路...122</p> <p>4.9 CR発振回路1...129</p> <p>4.10 CR発振回路2...135</p> <p>第5章 デバイスでの動作確認例 ...143</p> <p>5.1 非反転増幅回路...143</p> <p>5.2 反転増幅回路...144</p> <p>5.3 反転増幅回路 オフセット調整...145</p> <p>5.4 反転増幅回路 オフセット調整 + ロウパスフィルター...146</p> <p>5.5 非反転増幅回路 交流...147</p> <p>5.6 反転増幅回路 交流...148</p> <p>5.7 インストルメンテーションアンプ...149</p> <p>5.8 比較回路...150</p> <p>5.9 CR発振回路1...151</p> <p>5.10 CR発振回路2...152</p> <p>第6章 関連資料 ...153</p> <p>付録A プログラム・リスト...154</p> <p>付録B 改版履歴...449</p>
--	---

資料番号 U20228JJ1V0AN00（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

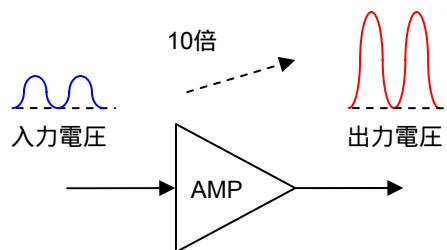
サンプル・プログラムの概要を説明します。

尚、【回路構成】中の **CHx** は“第5章 デバイスでの動作確認事例”の波形のCHを表します。

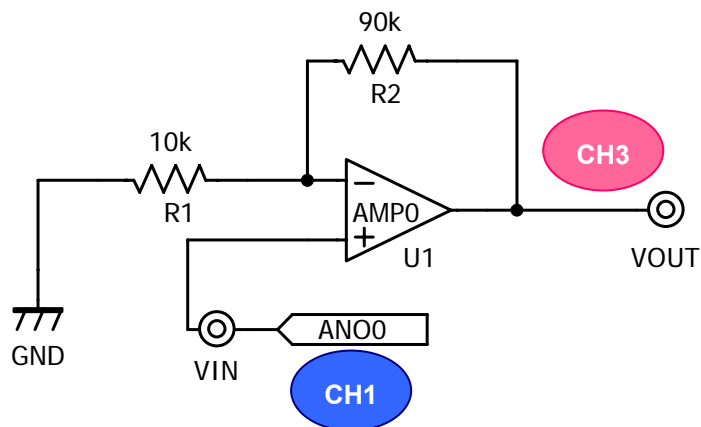
1.1 非反転増幅回路

【概要】

グランドを基準にDAC0出力電圧（ANO0端子）を10倍に増幅します。



【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	(GND)	—	—	—
入力電圧	ANO0	0.1V	±0.1V	1kHz
出力電圧	AMP00	1.0V	±1.0V	1kHz

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を10 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP0O, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0をアナログ出力用に設定
 - ・チャンネル0の動作モードをリアルタイム出力モードに設定
 - ・チャンネル0の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット、転送回数を100回に設定
 - ・転送元をRAM領域、転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

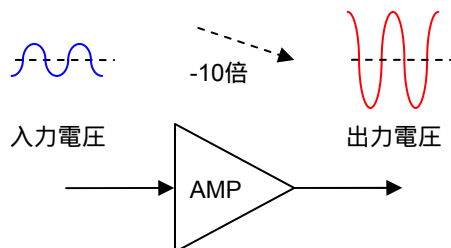
(2) メイン・ループ以降の内容

初期設定完了後、D/Aコンバータのチャンネル0の動作許可、タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可、およびINTDMA0割り込みの許可を行い、HALTモードに移行します。以降は、INTDMA0割り込みが発生するごとにDMAコントローラの再設定を行い、再度HALTモードに移行します。

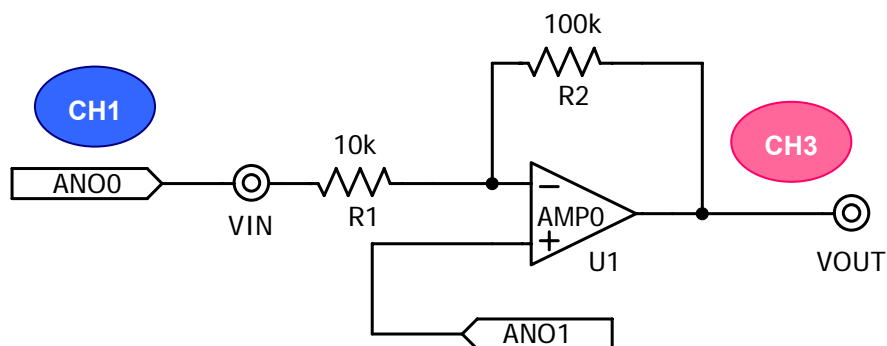
1.2 反転増幅回路

【概要】

DAC1出力電圧（ANO1端子）を基準にDAC0電圧（ANO0端子）を10倍に反転増幅します。



【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	1.0V	—	—
入力電圧	ANO0	1.0V	±0.1V	1kHz
出力電圧	AMP00	1.0V	±1.0V	1kHz

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を10 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP0O, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0, P111/ANO1をアナログ出力用に設定
 - ・チャンネル0, チャンネル1の動作モードを通常モードに設定
 - ・チャンネル0, チャンネル1の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット, 転送回数を100回に設定
 - ・転送元をRAM領域, 転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

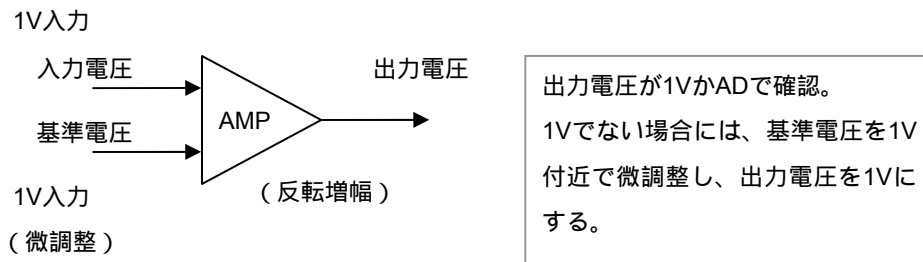
(2) メイン・ループ以降の内容

初期設定完了後、D/Aコンバータのチャンネル0, チャンネル1の動作許可, タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可, およびINTDMA0割り込みの許可を行い, HALTモードに移行します。以降は, INTDMA0割り込みが発生するごとにDMAコントローラの再設定を行い, 再度HALTモードに移行します。

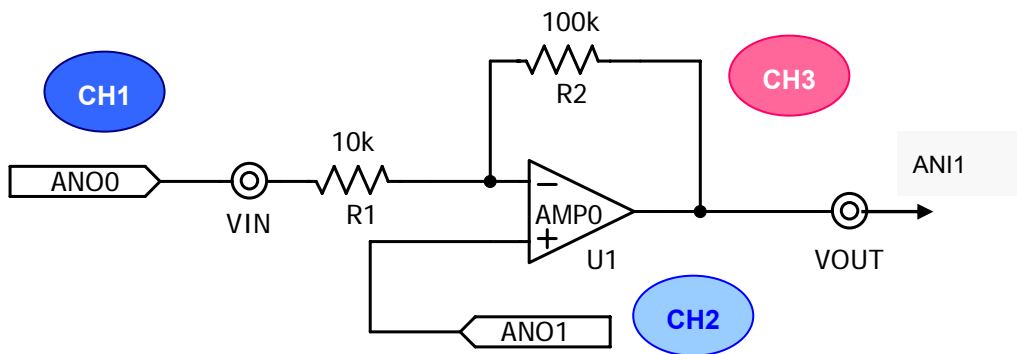
1.3 反転増幅回路 オフセット調整

【概要】

“ 1.2 反転増幅回路 ” のフローの最後に、オフセット調整のフローを追加します。オフセット調整は基準電圧を微調整し、入力電圧 (1V) と出力電圧が一致するところで基準電圧を固定します。



【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	0.98Vから調整	—	—
入力電圧	ANO0	1.0V	±0.1V	1kHz
出力電圧	AMP00	1.0V	±1.0V	1kHz

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を10 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP00, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0, P111/ANO1をアナログ出力用に設定
 - ・チャンネル0の動作モードをリアルタイム出力モードに設定
 - ・チャンネル1の動作モードを通常モードに設定
 - ・チャンネル0, チャンネル1の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット, 転送回数を100回に設定
 - ・転送元をRAM領域, 転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

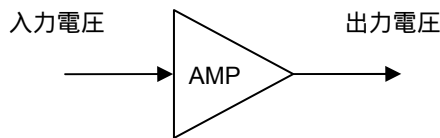
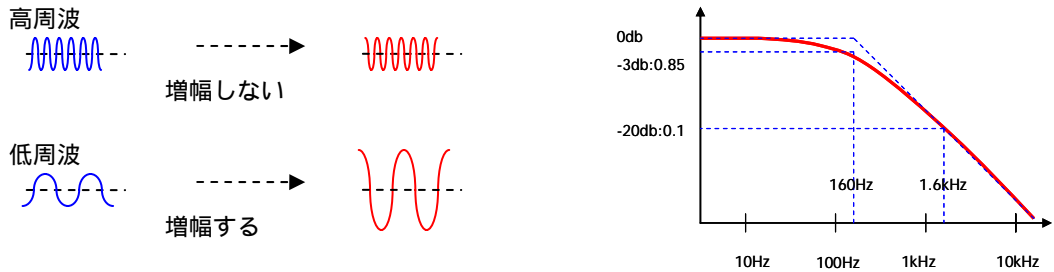
(2) メイン・ループ以降の内容

初期設定完了後、D/Aコンバータのチャンネル0, チャンネル1の動作許可, タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可, およびINTDMA0割り込みの許可を行い, HALTモードに移行します。以降は, INTDMA0割り込みが発生するごとにA/Dコンバータを使用したANO1出力（オペアンプ基準電圧）の調整, およびDMAコントローラの再設定を行い, 再度HALTモードに移行します。

1.4 反転増幅回路 オフセット調整 + ローパスフィルター

【概要】

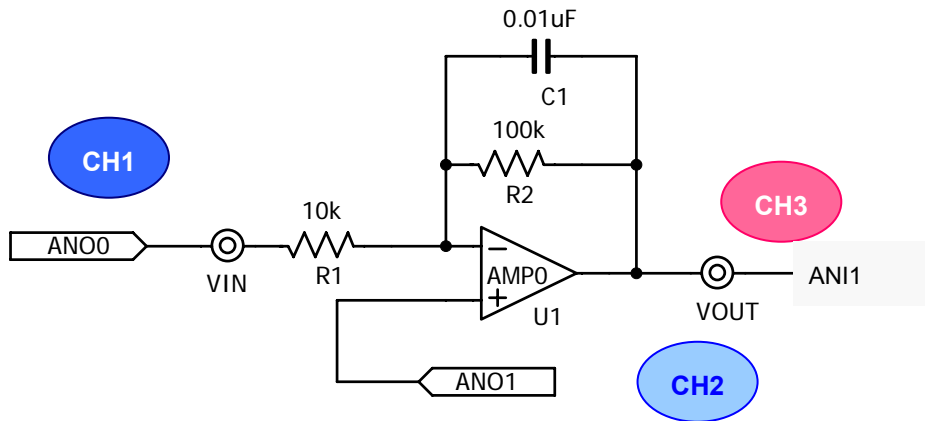
“ 1.3 反転増幅回路 オフセット調整 ” の回路にフィルタ回路 (C 1) を追加します。これにより、1kHzの入力の増幅が低減されることが確認できます。尚、100Hz (低周波) の入力については増幅が確認できます。



カットオフ周波数 : $F_c = 1 / (2 \cdot CR) = 160\text{Hz}$

- ・ 160Hz以上 (カットされる) 1kHz
- ・ 160Hz以下 (カットされない) 100Hz

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	0.98Vから調整	—	—
入力電圧	ANO0	1.0V	±0.1V	1kHz・100Hz
出力電圧	AMP00	1.0V	低減・低減しない	1kHz・100Hz

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を10 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP00, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0, P111/ANO1をアナログ出力用に設定
 - ・チャンネル0の動作モードをリアルタイム出力モードに設定
 - ・チャンネル1の動作モードを通常モードに設定
 - ・チャンネル0, チャンネル1の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット, 転送回数を100回に設定
 - ・転送元をRAM領域, 転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

(2) メイン・ループ以降の内容

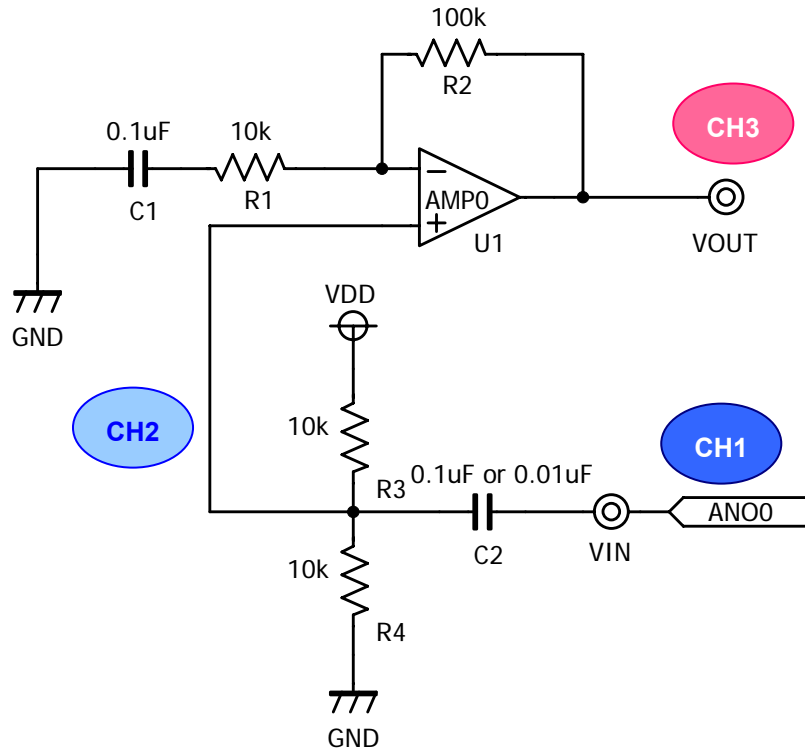
初期設定完了後、D/Aコンバータのチャンネル0, チャンネル1の動作許可, タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可, およびINTDMA0割り込みの許可を行い, HALTモードに移行します。以降は、INTDMA0割り込みが発生するごとにA/Dコンバータを使用したANO1出力（オペアンプ基準電圧）の調整, およびDMAコントローラの再設定を行います。また、一定周期でタイマ・アレイ・ユニット0のチャンネル4のカウント値を変更し、ANO0出力の周波数を切り替えます。

1.5 非反転増幅回路 交流

【概要】

入力電圧の変化（周波数）が大きい場合、増幅する回路です。同じ振幅の高周波と低周波を入力すると、その差分が確認できます。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	(GND)	—	—	—
入力電圧	ANO0	1.0V	±0.1V	高い・低い
出力電圧	AMP00	1.5V	あり・なし	—

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を100 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-、P21/AMP00、P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0をアナログ出力用に設定
 - ・チャンネル0の動作モードをリアルタイム出力モードに設定
 - ・チャンネル0の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット、転送回数を25回に設定
 - ・転送元をRAM領域、転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

(2) メイン・ループ以降の内容

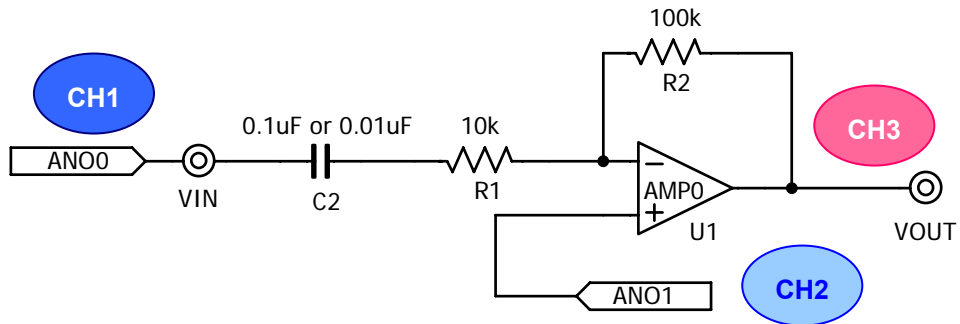
初期設定完了後、D/Aコンバータのチャンネル0の動作許可、タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可、およびINTDMA0割り込みの許可を行い、HALTモードに移行します。以降は、INTDMA0割り込みが発生するごとに、DMAコントローラの転送回数およびタイマ・アレイ・ユニット0のチャンネル4のカウント回数を変更することで、ANO0出力の周波数を切り替えます。

1.6 反転増幅回路 交流

【概要】

入力電圧の変化（周波数）が大きい場合、反転増幅する回路です。同じ振幅の高周波と低周波を入力すると、その差分が確認できます。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	1.0V	—	—
入力電圧	ANO0	1.0V	+0.1V	高い・低い
出力電圧	AMP00	1.0V	あり・なし	—

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を100 μ s周期のインターバル動作に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- ボルテージ・リファレンスの設定
 - ・ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP00, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・P110/ANO0, P111/ANO1をアナログ出力用に設定
 - ・チャンネル0の動作モードをリアルタイム出力モードに設定
 - ・チャンネル1の動作モードを通常モードに設定
 - ・チャンネル0, チャンネル1の分解能を12ビット分解能に設定
- DMAコントローラの設定
 - ・DMA起動要因をINTTM04割り込みに設定
 - ・転送データ・サイズを16ビット, 転送回数を25回に設定
 - ・転送元をRAM領域, 転送先をDACSW0レジスタに設定
 - ・DMAチャンネル0の動作許可

(2) メイン・ループ以降の内容

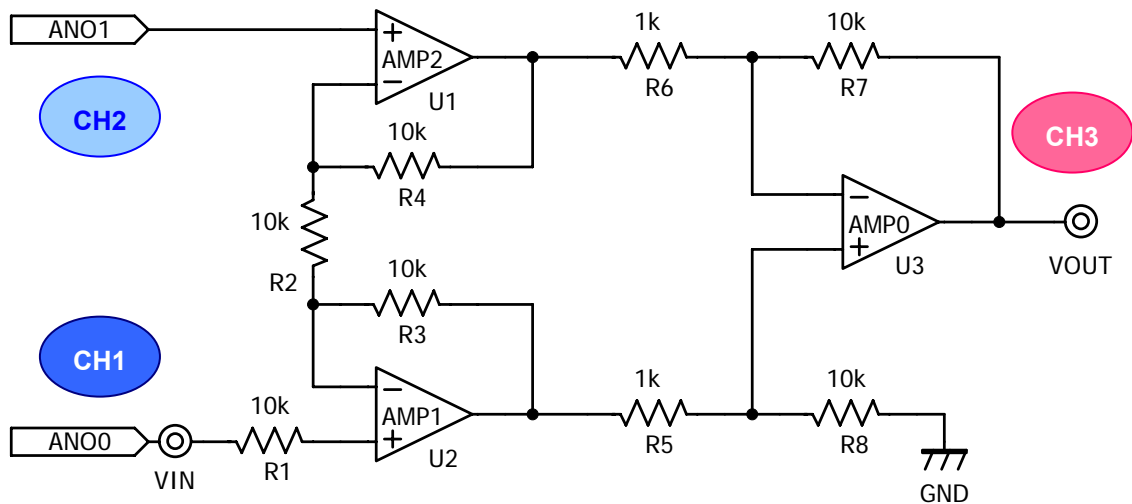
初期設定完了後、D/Aコンバータのチャンネル0, チャンネル1の動作許可, タイマ・アレイ・ユニット0のチャンネル4のカウント動作許可, およびINTDMA0割り込みの許可を行い, HALTモードに移行します。以降は, INTDMA0割り込みが発生するごとに, DMAコントローラの転送回数およびタイマ・アレイ・ユニット0のチャンネル4のカウント回数を変更することで, ANO0出力の周波数を切り替えます。

1.7 インストルメンテーションアンプ

【概要】

DAC1出力（ANO1端子）とDAC0出力（ANO0端子）の電位差を増幅します。DAC1出力（ANO1端子）は0.5Vで固定とし、DAC0出力（ANO0端子）は0.52Vと0.54Vとを交互に切り替えます。以下増幅回路では30倍増幅します。入力電位差としては、 $0.02V \cdot 0.04V$ ですので、出力電圧は $0.6V \cdot 1.2V$ になります。インストルメンテーションアンプは入力のレベルによらず電位差を増幅することが可能なため、ブリッジタイプの2入力センサの増幅用などに利用されます。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	0.5V	—	—
入力電圧	ANO0	0.52V・0.54V	—	—
出力電圧	AMP00	0.6V・1.2V	—	—

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをXT1発振回路動作に設定（32.768KHz）
- 高速内蔵発振回路の停止
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- リアルタイム・カウンタの設定
 - ・ タイマ・アレイ・ユニット0のチャンネル1を使用したサブシステム・クロックの発振安定待ち
 - ・ 定周期割り込み（1秒周期）の設定
- ボルテージ・リファレンスの設定
 - ・ ボルテージ・リファレンス出力電圧を2.0Vに設定
 - ・ タイマ・アレイ・ユニットのチャンネル1を使用したボルテージ・リファレンスの動作安定待ち（約17ms）
- オペアンプの設定
 - ・ P20/AMP0-, P21/AMP0O, P22/AMP0+, P23/AMP1-, P24/AMP1O, P25/AMP1+, P26/AMP2-, P27/AMP2O, P150/AMP2+をアナログ入力端子に設定
 - ・ オペアンプ0, オペアンプ1, オペアンプ2の動作許可
- D/Aコンバータの設定
 - ・ +側の基準電圧をボルテージ・リファレンス出力電圧に設定
 - ・ P110/ANO0, P111/ANO1をアナログ出力用に設定
 - ・ チャンネル0, チャンネル1の動作モードを通常モードに設定
 - ・ チャンネル0, チャンネル1の分解能を12ビット分解能に設定
 - ・ ANO0出力を0.52Vに設定
 - ・ ANO1出力を0.5Vに設定

(2) メイン・ループ以降の内容

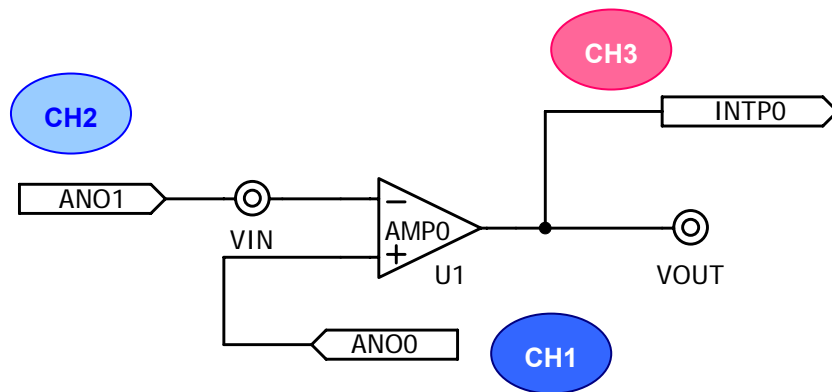
初期設定完了後、D/Aコンバータのチャンネル0, チャンネル1の動作許可, リアルタイム・カウンタの動作許可, INTRTC割り込みの許可を行い, HALTモードに移行します。以降は, INTRTC割り込み（1秒周期）が発生するごとに, ANO0出力を0.52Vと0.54Vに交互に切り替えます。

1.8 比較回路

【概要】

オペアンプをコンパレータとして利用します。DAC0出力（ANO1端子）を基準電圧としAMP0-端子に接続、判定したいレベル、ここではDAC0出力（ANO0端子）をAMP0+端子に接続します。判定したいレベルが基準電圧以上 [以下] になるとAMP0出力（AMP0O端子）はハイ [ロウ] ・レベル出力します。これがINTP0端子に接続しているため、外部割り込みとしての利用が可能となります。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	1.0V	—	—
入力電圧	ANO0	1.0V	±0.1V	1kHz
出力電圧	AMP0O	1.5V	±1.5V	1kHz

(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

- ・ P73を比較結果の出力用に設定

CPUクロックをX1発振回路動作に設定 (20MHz)

高速内蔵発振回路の停止

タイマ・アレイ・ユニット0の設定

- ・ チャネル1を500 μ s周期のインターバル動作に設定
- ・ チャネル4を10 μ s周期のインターバル動作に設定

低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保

ボルテージ・リファレンスの設定

- ・ ボルテージ・リファレンス出力電圧を2.0Vに設定
- ・ タイマ・アレイ・ユニットのチャネル1を使用したボルテージ・リファレンスの動作安定待ち (約17ms)

オペアンプの設定

- ・ P20/AMP0-, P21/AMP00, P22/AMP0+をアナログ入力端子に設定
- ・ オペアンプ0の動作許可

D/Aコンバータの設定

- ・ +側の基準電圧をボルテージ・リファレンス出力電圧に設定
- ・ P110/ANO0, P111/ANO1をアナログ出力用に設定
- ・ チャネル0の動作モードを通常モードに設定
- ・ チャネル1の動作モードを通常モードに設定
- ・ チャネル0, チャネル1の分解能を12ビット分解能に設定

DMAコントローラの設定

- ・ DMA起動要因をINTTM04割り込みに設定
- ・ 転送データ・サイズを16ビット, 転送回数を100回に設定
- ・ 転送元をRAM領域, 転送先をDACSW0レジスタに設定
- ・ DMAチャネル0の動作許可

INTP0端子の有効エッジを立ち上がり, 立ち下がり両エッジに設定

(2) メイン・ループ以降の内容

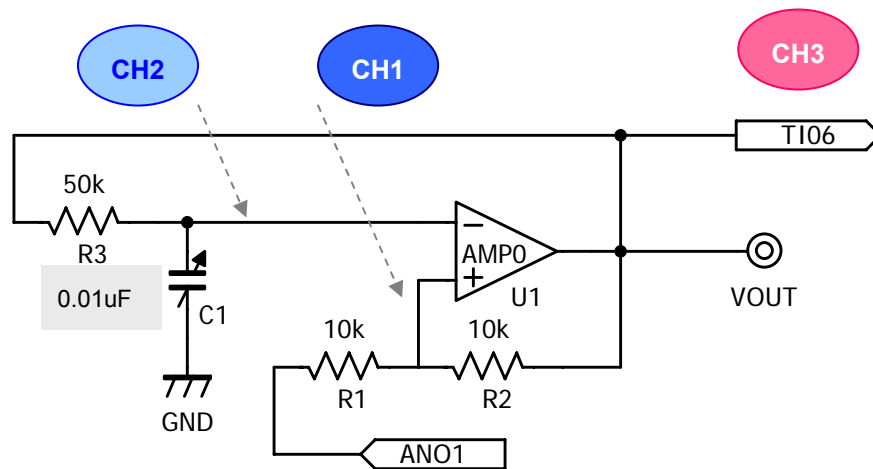
初期設定完了後、D/Aコンバータのチャネル0, チャネル1の動作許可, タイマ・アレイ・ユニット0のチャネル4のカウント動作許可, INTDMA0割り込みの許可, およびINTP0割り込みの許可を行い、HALTモードに移行します。以降は、AMP0出力の立ち上がり・立ち下がり検出によるINTP0割り込みが発生するごとに、INTP0端子の検出レベルをP73に出力します。また、INTDMA0割り込みが発生するごとにDMAコントローラの再設定を行います。

1.9 CR発振回路1

【概要】

D/Aコンバータおよびオペアンプを使用し、CR発振回路を構成します。オペアンプはコンパレータとして利用します。DAC1出力（ANO1端子）を基準電圧とし、AMP0出力（AMP0O端子）との間に抵抗を介し、AMP0+端子へ入力します。AMPの出力状態によりAMP0+端子への入力電圧は変動します。この変動により、AMP0-端子への入力レベルは充放電を繰り返します。周波数測定方法として、タイマ・アレイ・ユニットの入力パルス間隔測定機能によりAMP0出力のパルス幅を測定します。尚、測定結果はLCDに出力されます。このシステムにより、C1の容量変化を周波数として測定することが可能です。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	1.5V	—	—
基準電圧'	(AMP0+)	0.75V・2.25V	—	約0.83kHz
出力電圧	AMP0O	0V・3V	—	約0.83kHz

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル4を1ms周期のインターバル動作に設定
 - ・チャンネル6を入力パルス間隔測定用に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- オペアンプの設定
 - ・P20/AMP0-, P21/AMP0O, P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧源をAV_{DD1}端子に設定
 - ・P111/ANO1をアナログ出力用に設定
 - ・チャンネル1の動作モードを通常モードに設定
 - ・チャンネル1の分解能を12ビット分解能に設定
 - ・ANO1出力を1.5Vに設定
- LCDコントローラ / ドライバの設定
 - ・SEG4-SEG53をセグメント出力用に設定
 - ・LCD駆動電圧生成回路に容量分割方式を選択
 - ・LCD表示モードを8時分割, 1/4バイアスに設定
 - ・Aパターン領域のデータを表示データに設定
 - ・LCDソース・クロック(f_{LCD})を f_{SUB} , LCDクロックを $f_{LCD}/2^7$ に設定
 - ・LCD駆動電圧(V_{LC0})を5.0Vに設定
 - ・LCD表示の許可

(2) メイン・ループ以降の内容

初期設定完了後、D/Aコンバータのチャンネル1の動作許可、およびタイマ・アレイ・ユニット0のチャンネル4のカウント動作とチャンネル6の入力パルス間隔測定動作の開始設定を行います。以降は、INTTM04割り込み（約1ms周期）が発生するごとに、AMP0出力の周波数測定と測定結果のLCD出力を行います。

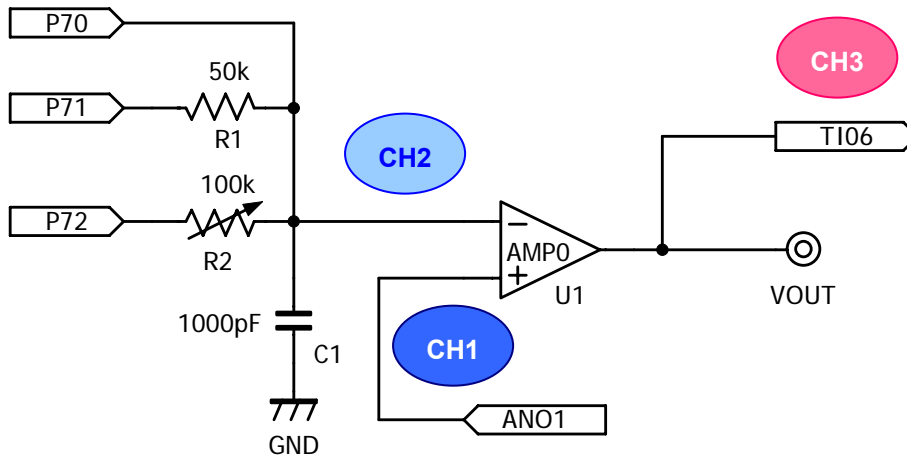
1.10 CR発振回路2

【概要】

D/Aコンバータおよびオペアンプを使用し、CR発振回路を構成します。オペアンプはコンパレータとして利用します。DAC1出力(ANO1端子)を基準電圧とし、AMP0+端子へ入力します。一方AMP0-端子へは、マイコンの命令によりP70ハイ・レベル出力し、C1に電荷を充電します。その後、P71をロウ・レベル出力し既知の抵抗R1を介し、C1の電荷を放電します。放電により、電圧がDAC1出力(ANO1端子)以下となると、AMP00はハイ・レベル出力します。周波数測定方法として、タイマ・アレイ・ユニットの入力パルス間隔測定機能によりAMP0出力のロウ・レベル期間の時間を測定します。同様の内容を未知の抵抗R2が接続されているP72についても実施します。両者の比および既知の抵抗R1より、R2の抵抗を算出します。

尚、基準電圧は1Vと2Vの2回行いその差分の時間を計算します。これにより、スタート処理等の時間による誤差を排除し、より正確な時間測定を実現します。

【回路構成】



	端子	中心電位	振幅	周波数
基準電圧	ANO1	1.0V・2.0V	—	—
出力電圧	AMP00	0V・3V	—	—

	端子		DAC1出力	放電時間
R1測定	P71ロウ・レベル出力	P72入力モード	1.0V	T_REF1
			2.0V	T_REF2
R2測定	P71入力モード	P72ロウ・レベル出力	1.0V	T_TH1
			2.0V	T_TH2

サンプル・プログラムでは、T_REF1とT_REF2のカウント差、およびT_TH1とT_TH2のカウント差をLCDに表示します。

(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（未使用ポートをすべてロウ・レベル出力とする）
- CPUクロックをX1発振回路動作に設定（20MHz）
- 高速内蔵発振回路の停止
- タイマ・アレイ・ユニット0の設定
 - ・チャンネル1を500 μ s周期のインターバル動作に設定
 - ・チャンネル6を入力パルス間隔測定用に設定
- 低電圧検出回路の機能を使用し、2.7V以上の電源電圧を確保
- オペアンプの設定
 - ・P20/AMP0-、P21/AMP00、P22/AMP0+をアナログ入力端子に設定
 - ・オペアンプ0の動作許可
- D/Aコンバータの設定
 - ・+側の基準電圧源をAV_{DD1}端子に設定
 - ・P111/ANO1をアナログ出力用に設定
 - ・チャンネル1の動作モードを通常モードに設定
 - ・チャンネル1の分解能を12ビット分解能に設定
 - ・ANO1出力を1.0Vに設定
- LCDコントローラ/ドライバの設定
 - ・SEG4-SEG53をセグメント出力用に設定
 - ・LCD駆動電圧生成回路に容量分割方式を選択
 - ・LCD表示モードを8時分割、1/4バイアスに設定
 - ・Aパターン領域のデータを表示データに設定
 - ・LCDソース・クロック(f_{LCD})を f_{SUB} 、LCDクロックを $f_{LCD}/2^7$ に設定
 - ・LCD駆動電圧(V_{LCO})を5.0Vに設定
 - ・LCD表示の許可

(2) メイン・ループ以降の内容

初期設定完了後、D/Aコンバータのチャンネル1の動作を許可します。以降は、ポート入出力の切り替えとANO1出力の切り替え（1.0V / 2.0V）によりコンデンサの充電と放電を行います。また、タイマ・アレイ・ユニット0のチャンネル6により放電時間を測定し、測定結果をLCDに出力します。

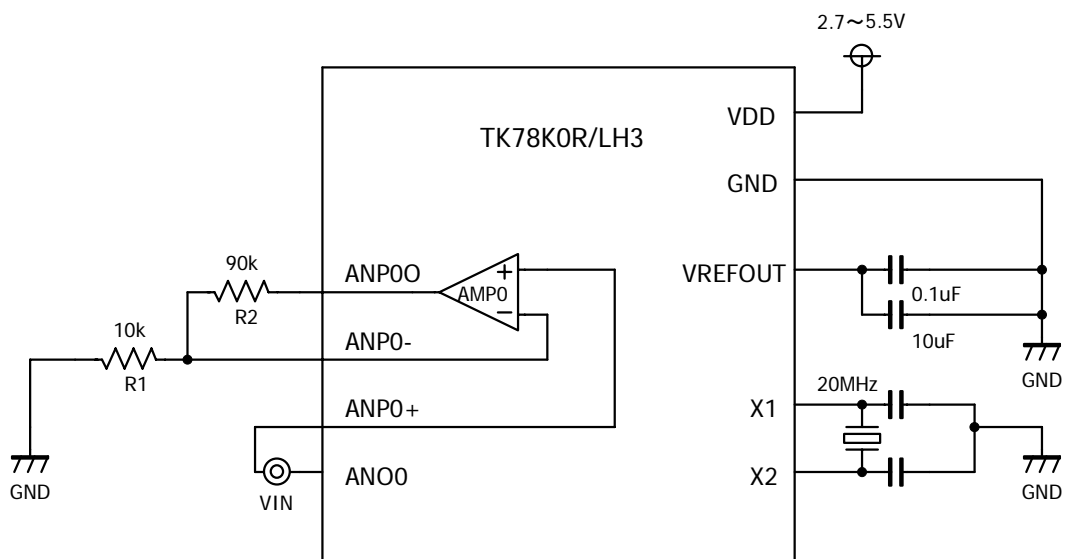
第2章 回路イメージ

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。

2.1 回路イメージ

回路イメージを次に示します。

2.1.1 非反転増幅回路



注意1. 2.7V V_{DD} 5.5Vの電圧範囲で使用してください。

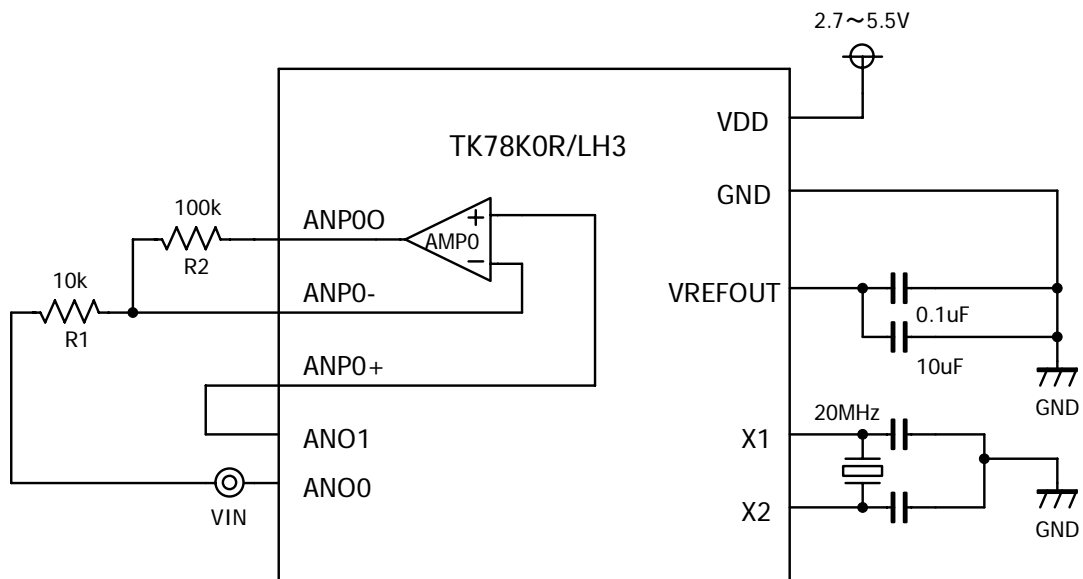
2. EV_{DD} , AV_{DD0} , および AV_{DD1} は、 V_{DD} と同電位にしてください。

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. ボルテージ・リファレンス動作時は、基準電圧安定化用に、 V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ（容量： $10\mu F \pm 30\%$ 、ESR：2（MAX.）、ESL：10nH（MAX.））とセラミック・コンデンサ（容量： $0.1\mu F \pm 30\%$ 、ESR：2（MAX.）、ESL：10nH（MAX.））を接続してください。また、ボルテージ・リファレンス動作時に、 V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。

5. 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。

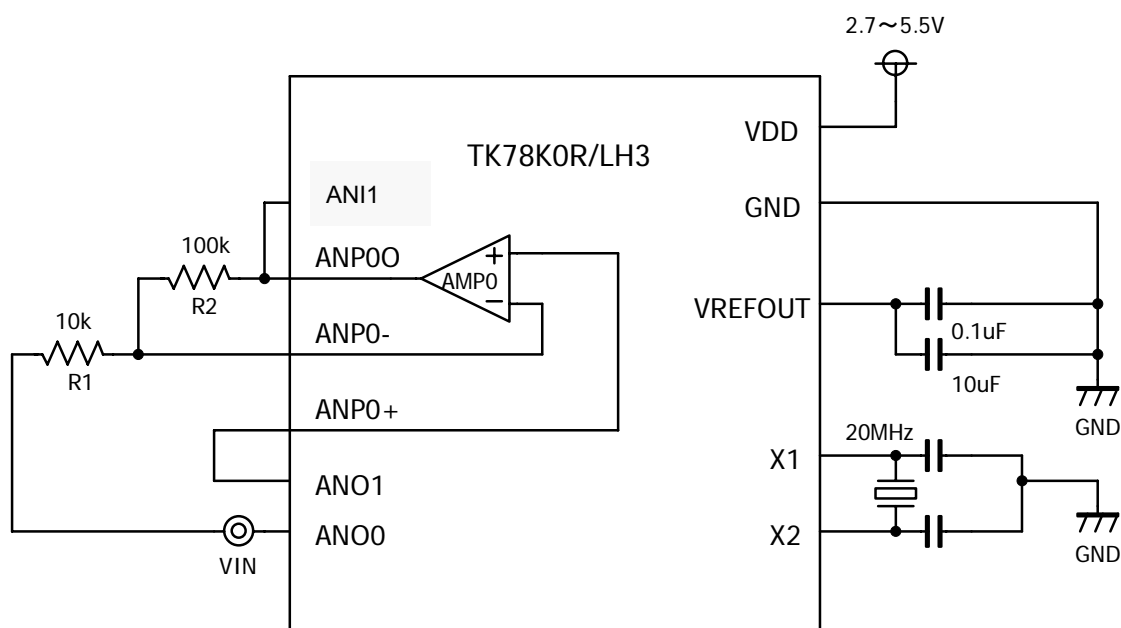
2.1.2 反転増幅回路



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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

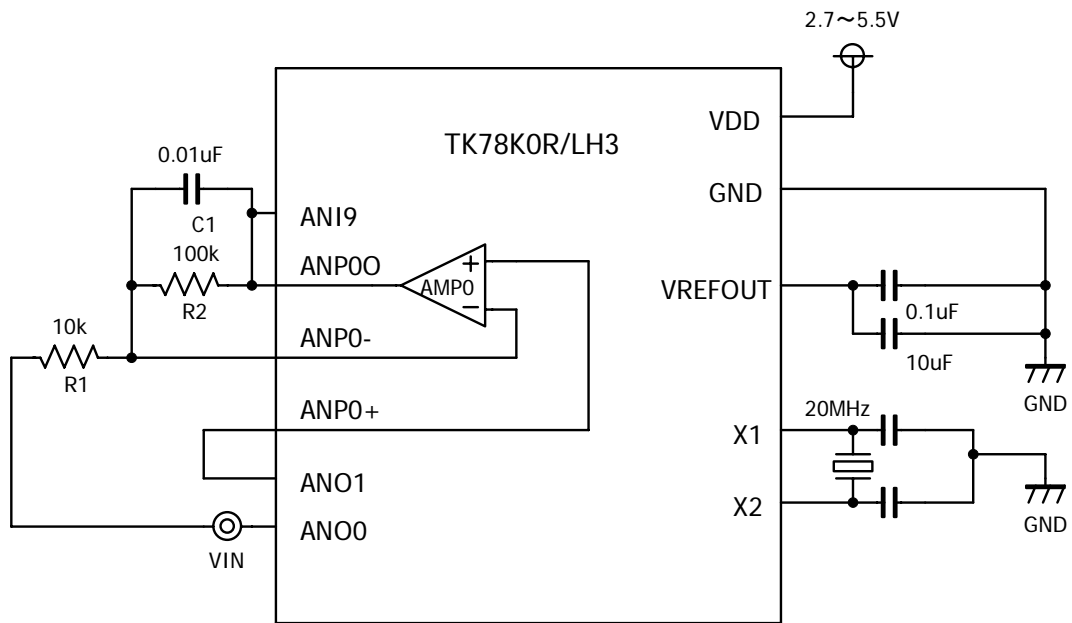
2.1.3 反転増幅回路 オフセット調整



注意1. 2.7V V_{DD} 5.5V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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

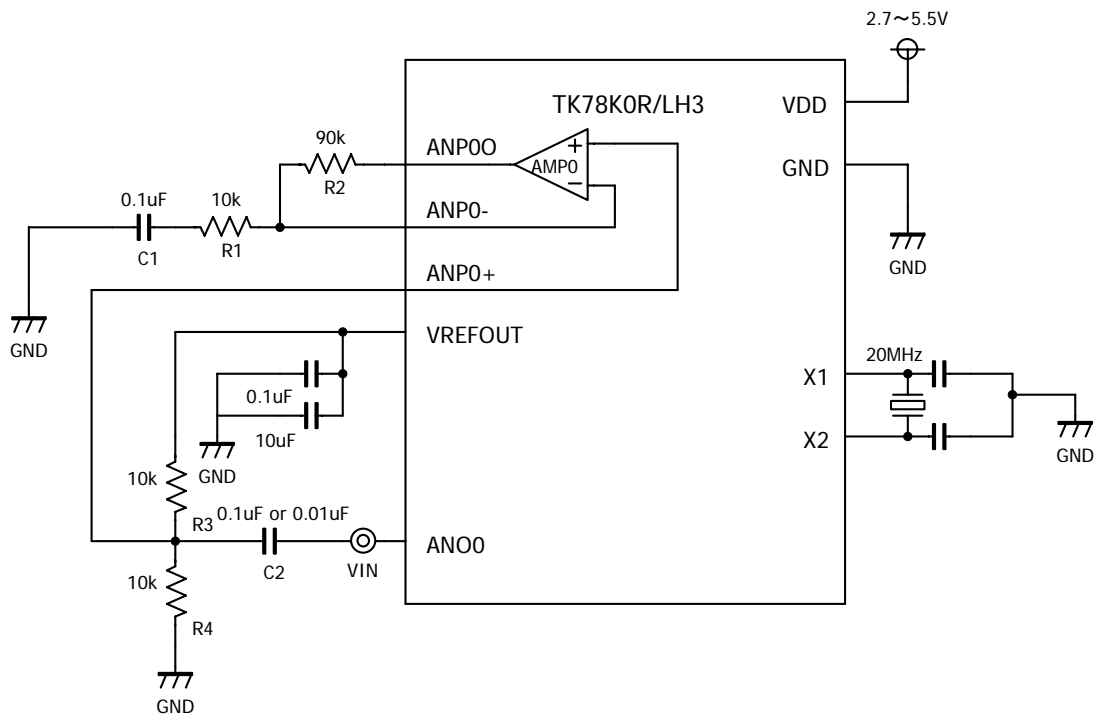
2.1.4 反転増幅回路 オフセット調整+ロウパスフィルター



注意1. 2.7V V_{DD} 5.5V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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

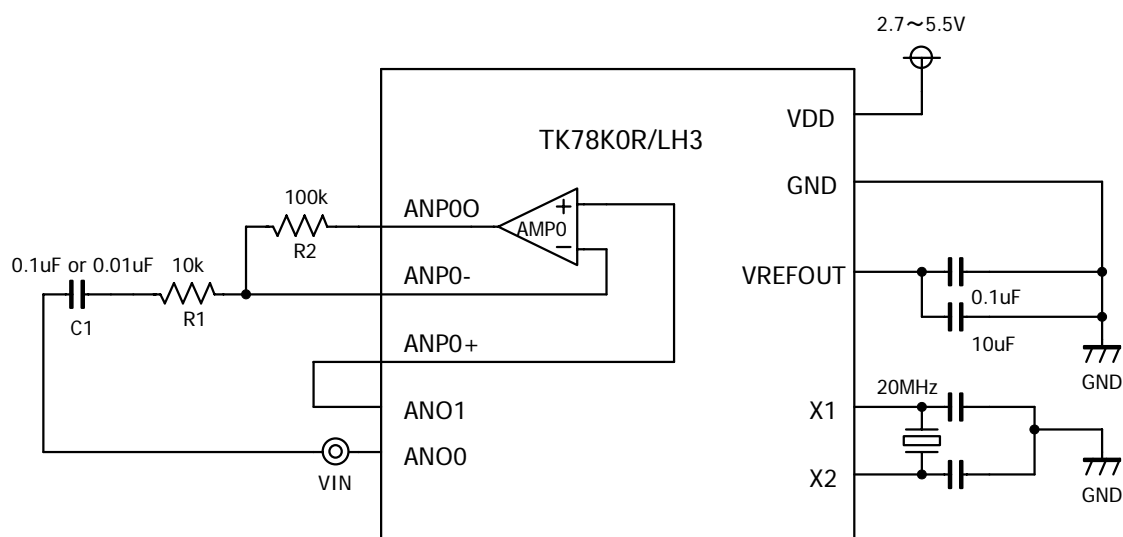
2.1.5 非反転増幅回路 交流



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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

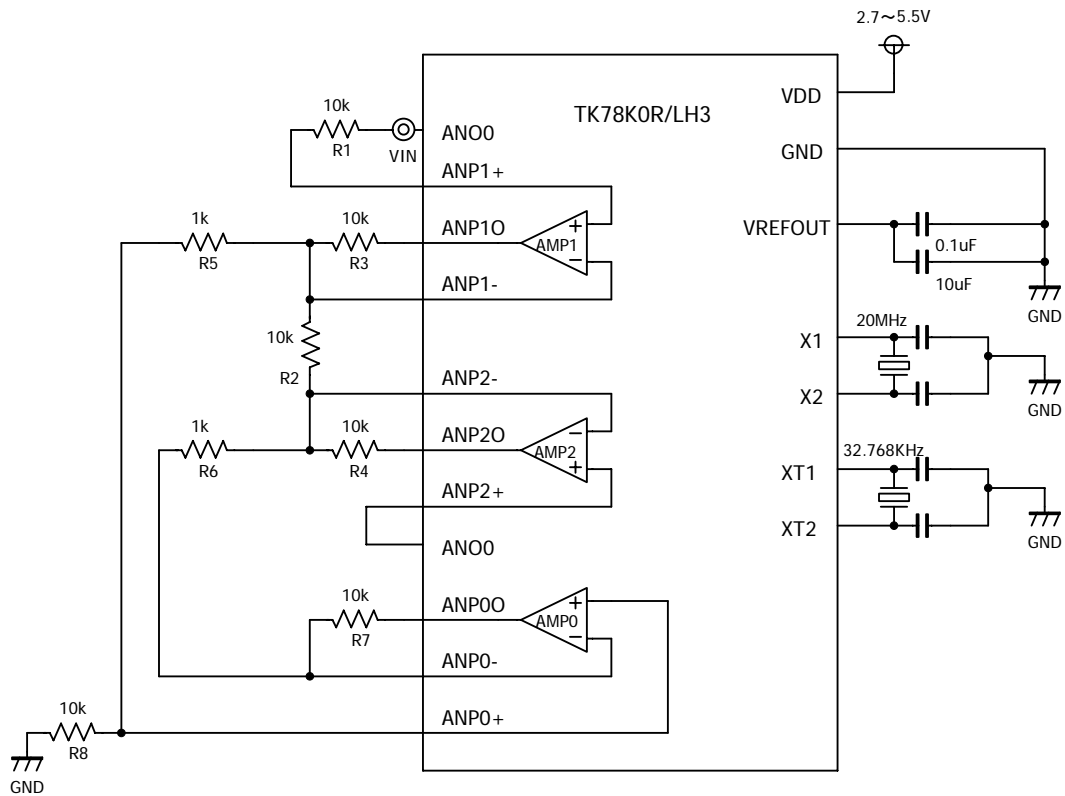
2.1.6 反転増幅回路 交流



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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

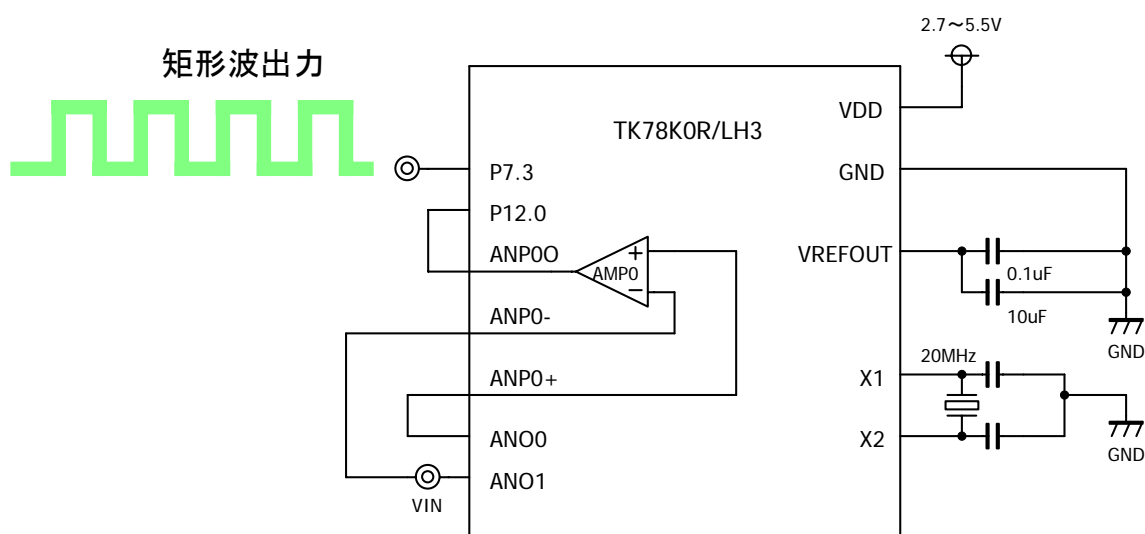
2.1.7 インstrumentation Amp



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. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/AV_{REFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/AV_{REFP} 端子から電圧を印加しないでください。
5. 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。

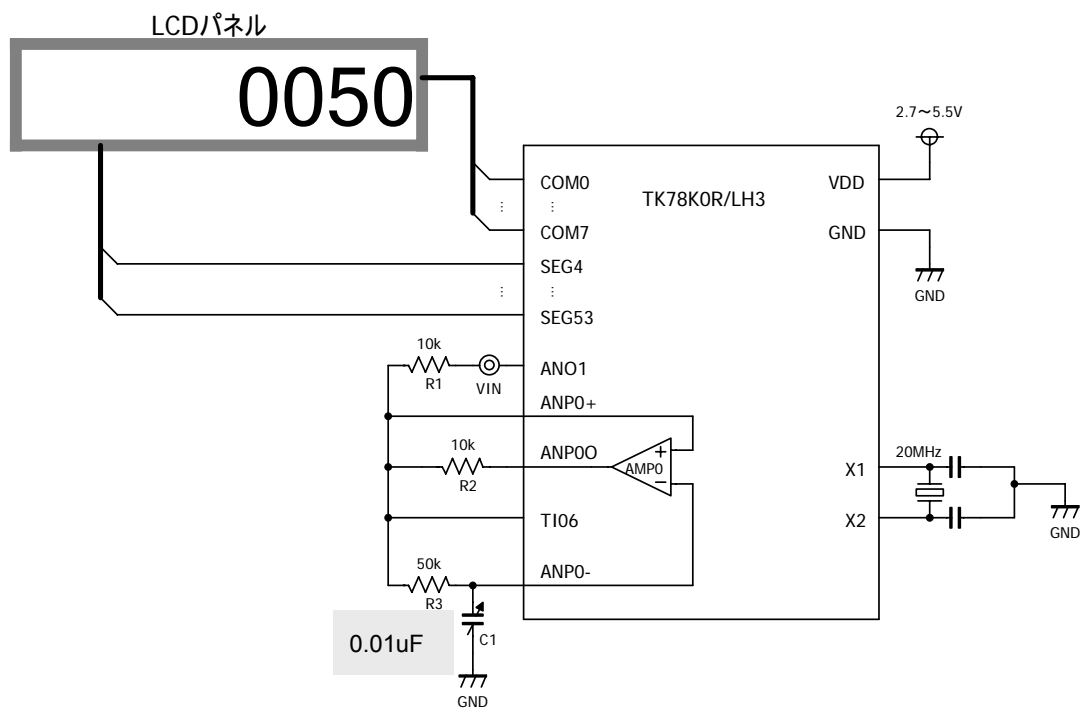
2.1.8 比較回路



注意1. 2.7V V_{DD} 5.5Vの電圧範囲で使用してください。

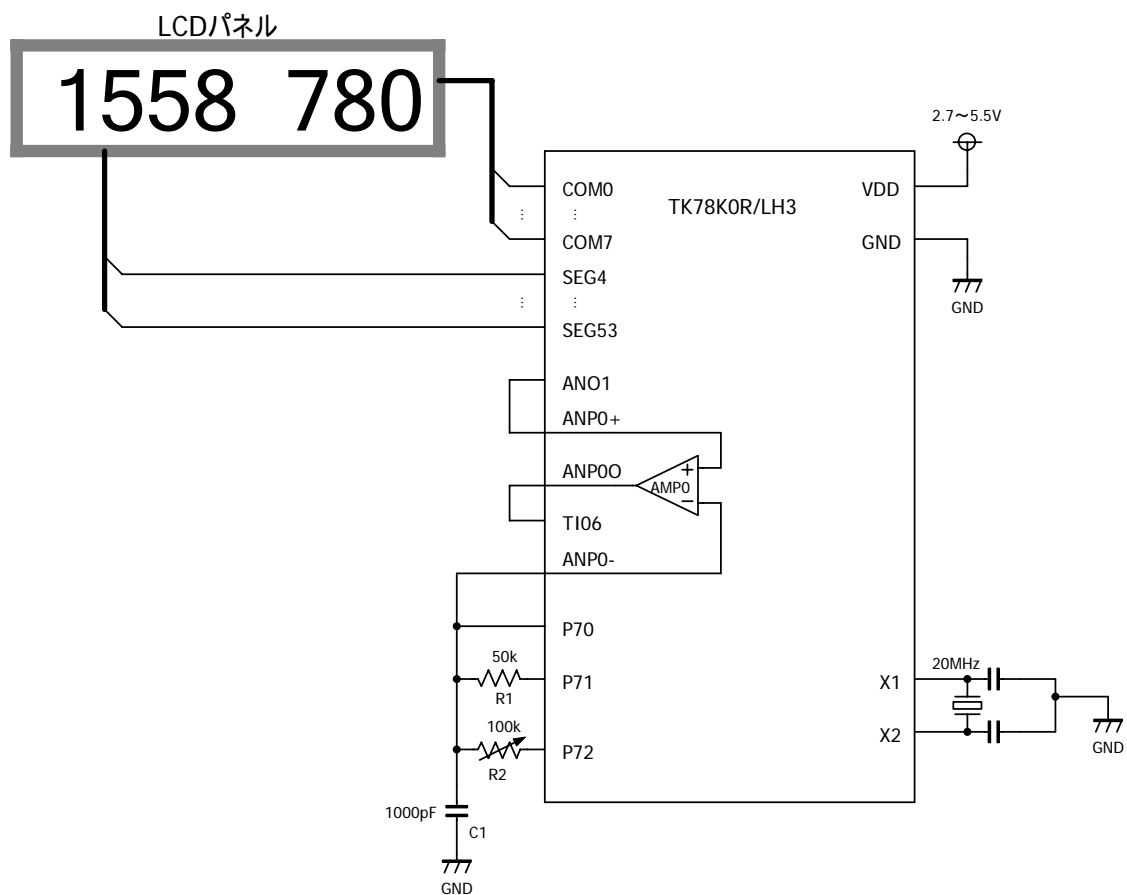
2. E_{VDD} , A_{VDD0} , および A_{VDD1} は, V_{DD} と同電位にしてください。
3. A_{VSS} は E_{VSS} , V_{SS} と同電位にし, GNDに直接接続してください。
4. ボルテージ・リファレンス動作時は, 基準電圧安定化用に, V_{REFOUT}/A_{VREFP} 端子に必ずタンタル・コンデンサ (容量: $10\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) とセラミック・コンデンサ (容量: $0.1\mu F \pm 30\%$, ESR: 2 (MAX.), ESL: 10nH (MAX.)) を接続してください。また, ボルテージ・リファレンス動作時に, V_{REFOUT}/A_{VREFP} 端子から電圧を印加しないでください。
5. 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。

2.1.9 CR発振回路1



- 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. E_{VDD} , A_{VDD0} , および A_{VDD1} は, V_{DD} と同電位にしてください。
 3. A_{VSS} は E_{VSS} , 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。

2.1.10 CR発振回路2



- 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。

2.2 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。

3.1.1 非反転増幅回路

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		
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

3.1.2 反転増幅回路

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		
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

3.1.3 反転増幅回路 オフセット調整

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		
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

3.1.4 反転増幅回路 オフセット調整+ロウパスフィルター

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		
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

3.1.5 非反転増幅回路 交流

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		
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

3.1.6 反転増幅回路 交流

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		
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

3.1.7 インストールメンテーションアンブ

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		
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

3.1.8 比較回路

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		
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

3.1.9 CR発振回路 1

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

3.1.10 CR発振回路 2

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

3.2 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。

3.2.1 非反転増幅回路

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ AMP 0
非反転増幅回路として使用します。

3.2.2 反転増幅回路

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ ANO 1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP 0
反転増幅回路として使用します。

3.2.3 反転増幅回路 オフセット調整

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ ANO 1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP 0
反転増幅回路として使用します。
- ・ ANI1
AMP出力電圧を計測します。

3.2.4 反転増幅回路 オフセット調整+ロウパスフィルター

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ ANO 1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP 0
反転増幅回路として使用します。

3.2.5 非反転増幅回路 交流

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ AMP 0
非反転増幅回路として使用します。

3.2.6 反転増幅回路 交流

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO 0
AMPに入力する信号出力に使用します。
- ・ ANO 1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP 0
反転増幅回路として使用します。

3.2.7 インストルメンテーションアンブ

- ・ 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。
- ・ サブシステム・クロック発振回路
CPUクロック用。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ ANO0
AMP1の非反転入力端子に入力する信号出力に使用します。
- ・ ANO1
AMP2の非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP0
非反転増幅回路として使用します。
- ・ AMP1
非反転増幅回路として使用します。
- ・ AMP2
非反転増幅回路として使用します。

3.2.8 比較回路

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
AMPに入力する信号波形を生成するDMAタイミング用インターバル・タイマとして使用します。
- ・ DMAコントローラ・チャンネル0
AMPに入力する信号波形データをD/Aコンバータに転送します。
- ・ ANO0
AMPに入力する信号出力に使用します。
- ・ ANO1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP0
コンパレータとして使用します。
- ・ INTP0
コンパレータ出力を入力し、両エッジ割り込みで使用します。
- ・ P73
コンパレータ出力のHigh/Lowをソフトウェア出力します。

3.2.9 CR発振回路 1

- ・ 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。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
発振パルスをカウントする間隔のインターバル・タイマとして使用します。
- ・ TAU0・チャンネル6
発振パルスをカウントするイベント・タイマとして使用します。
- ・ ANO1
AMPの非反転入力端子に入力するバイアス電圧出力に使用します。
- ・ AMP0
反転増幅回路として使用します。
- ・ TI06
AMP0出力を入力しイベント入力として使用します。
- ・ 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。

3.2.10 CR発振回路 2

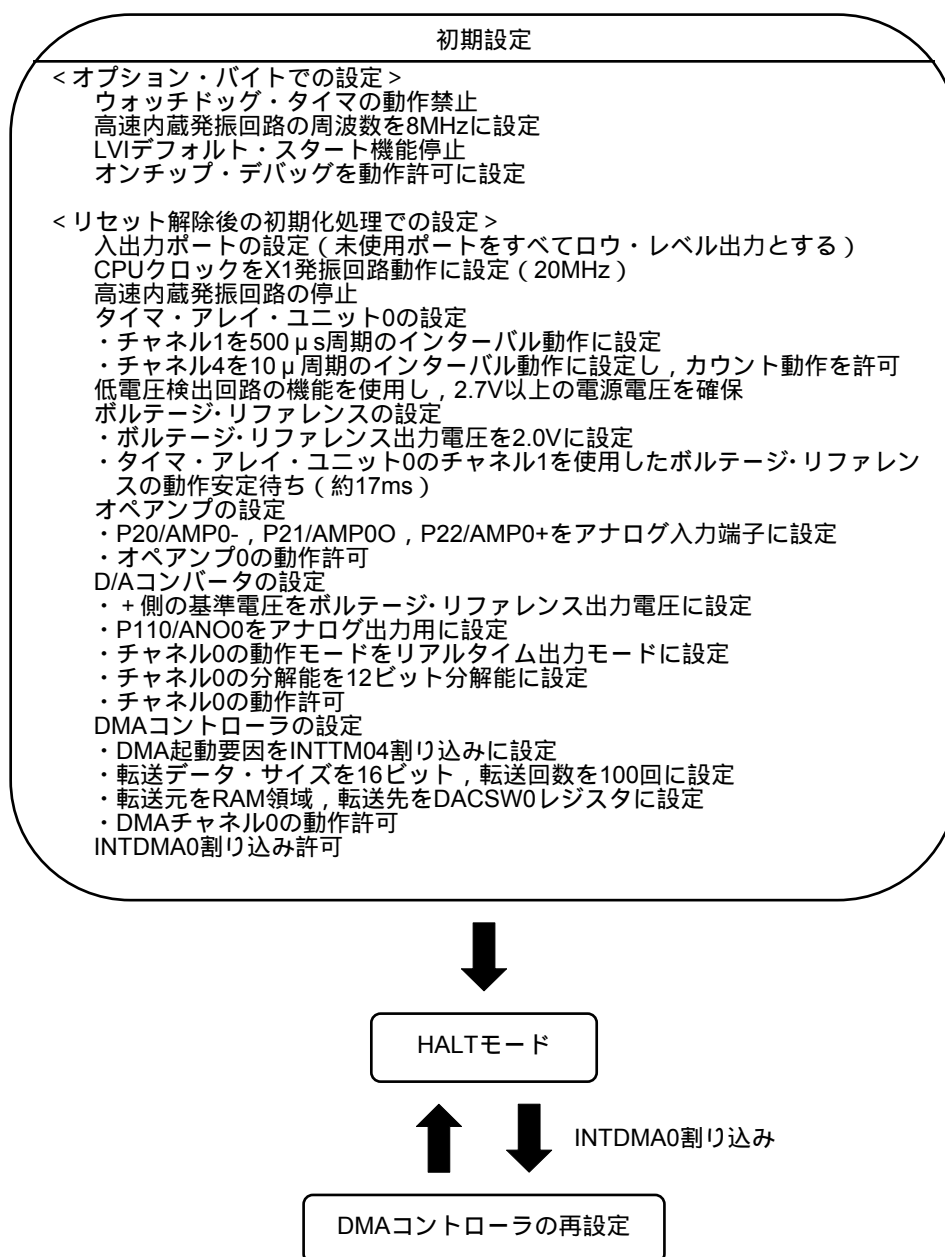
- ・ 高速システム・クロック発振回路
CPUクロック, 周辺ハードウェア・クロック用。
- ・ TAU0・チャンネル1
ハードウェア動作安定待ちのインターバル・タイマとして使用します。
- ・ TAU0・チャンネル4
発振パルスをカウントする間隔のインターバル・タイマとして使用します。
- ・ TAU0・チャンネル6
コンデンサの充電時間を計測するタイマとしてキャプチャ・モードで使用します。
- ・ ANO1
AMPの非反転入力端子に入力する比較基準電圧出力に使用します。
- ・ AMP0
反転増幅回路として使用します。
- ・ TI06
AMP0出力を入力しイベント入力として使用します。
- ・ AMP0
コンパレータとして使用します。
- ・ P70
コンデンサの充電を行います。
- ・ P71
基準抵抗を接続してコンデンサの放電を行います。
- ・ P72
被測定抵抗を接続してコンデンサの放電を行います。
- ・ 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。
- ・ 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。

3.3 初期設定と動作概要

3.3.1 非反転増幅回路

このサンプル・プログラムでは、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、D/Aコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPにサイン波を送り続けます。DMA転送終了割り込みにてDMAコントローラを再設定してサイン波の出力を連続して行います。

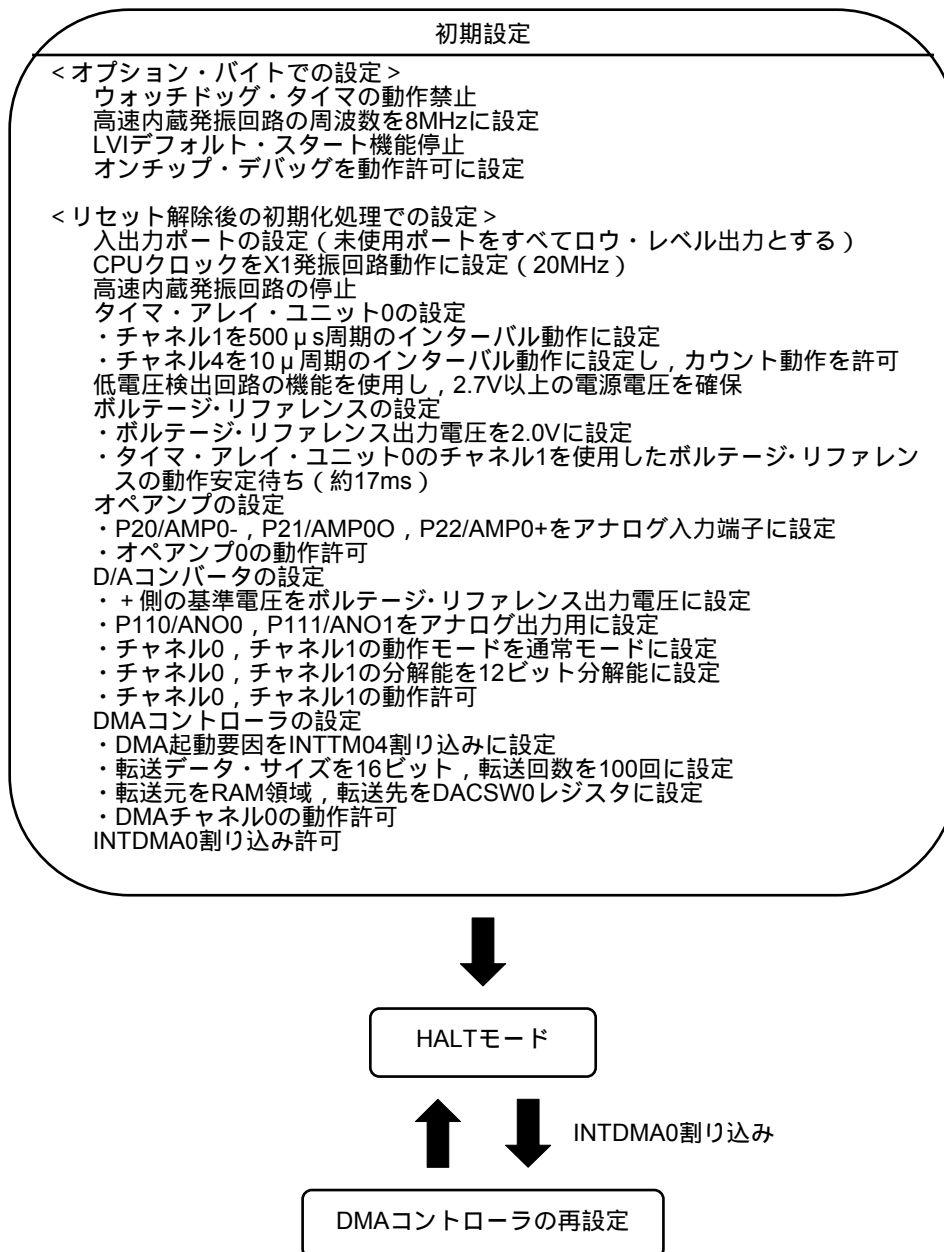
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.2 反転増幅回路

このサンプル・プログラムでは、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、D/Aコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPにサイン波を送り続けます。DMA転送終了割り込みにてDMAコントローラを再設定してサイン波の出力を連続して行います。

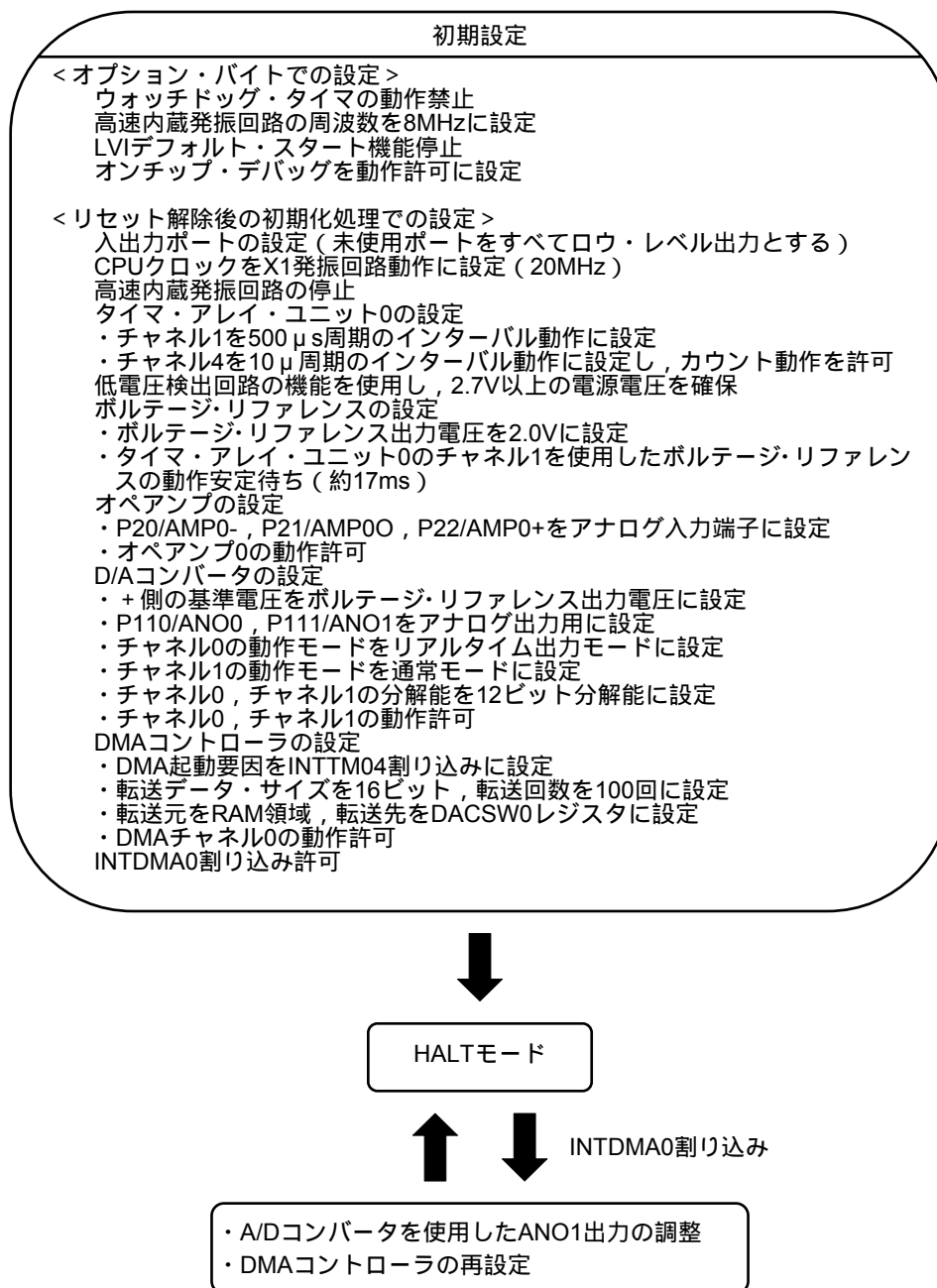
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.3 反転増幅回路 オフセット調整

このサンプル・プログラムでは、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、D/Aコンバータ、A/Dコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPにサイン波を送り続けます。DMA転送終了割り込みにてANI1の測定値が1Vであるか判定し、1Vに近づくようANO1の出力設定を調整します。また、DMAコントローラを再設定してサイン波の出力を連続して行います。

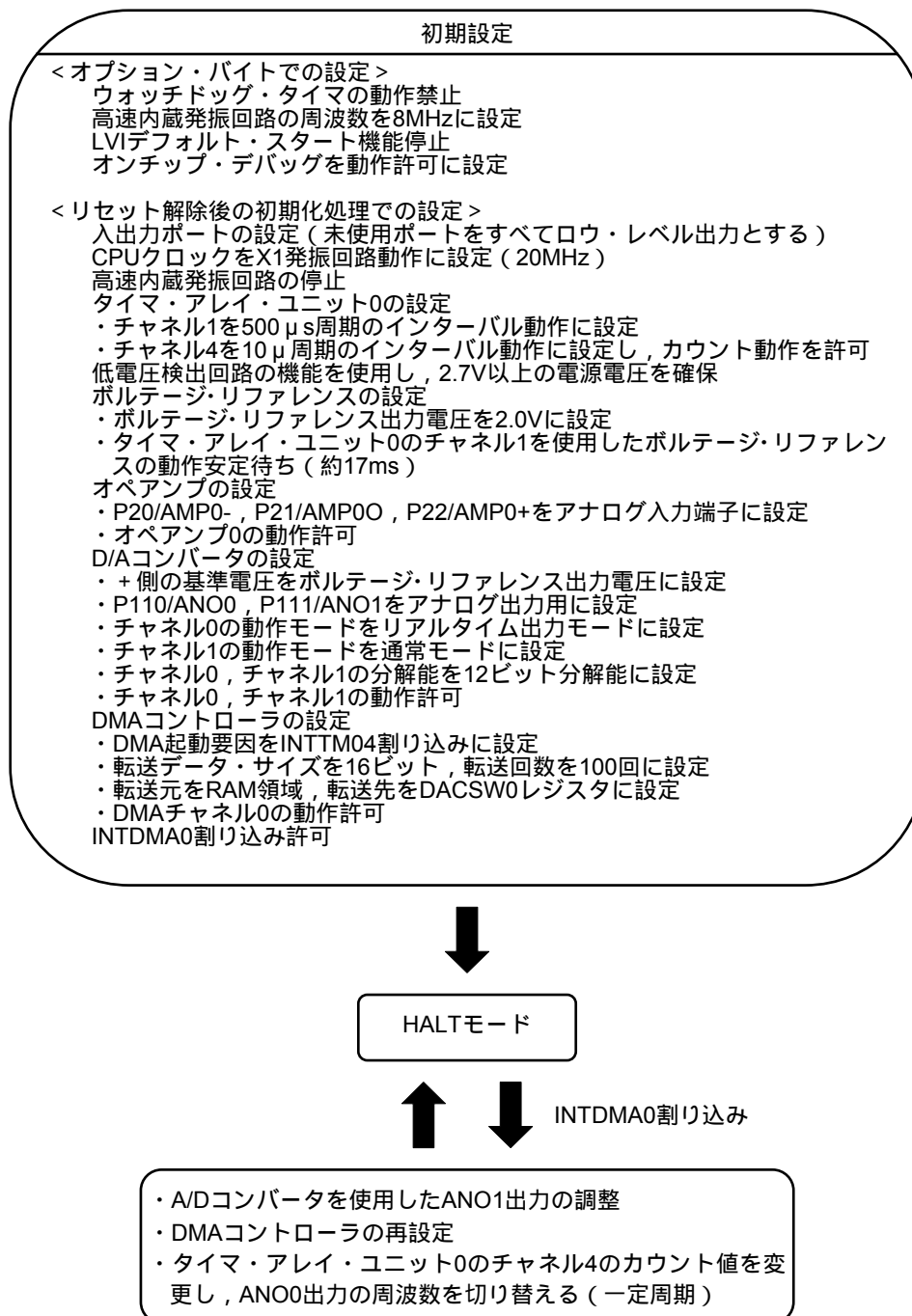
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.4 反転増幅回路 オフセット調整+ロウパスフィルター

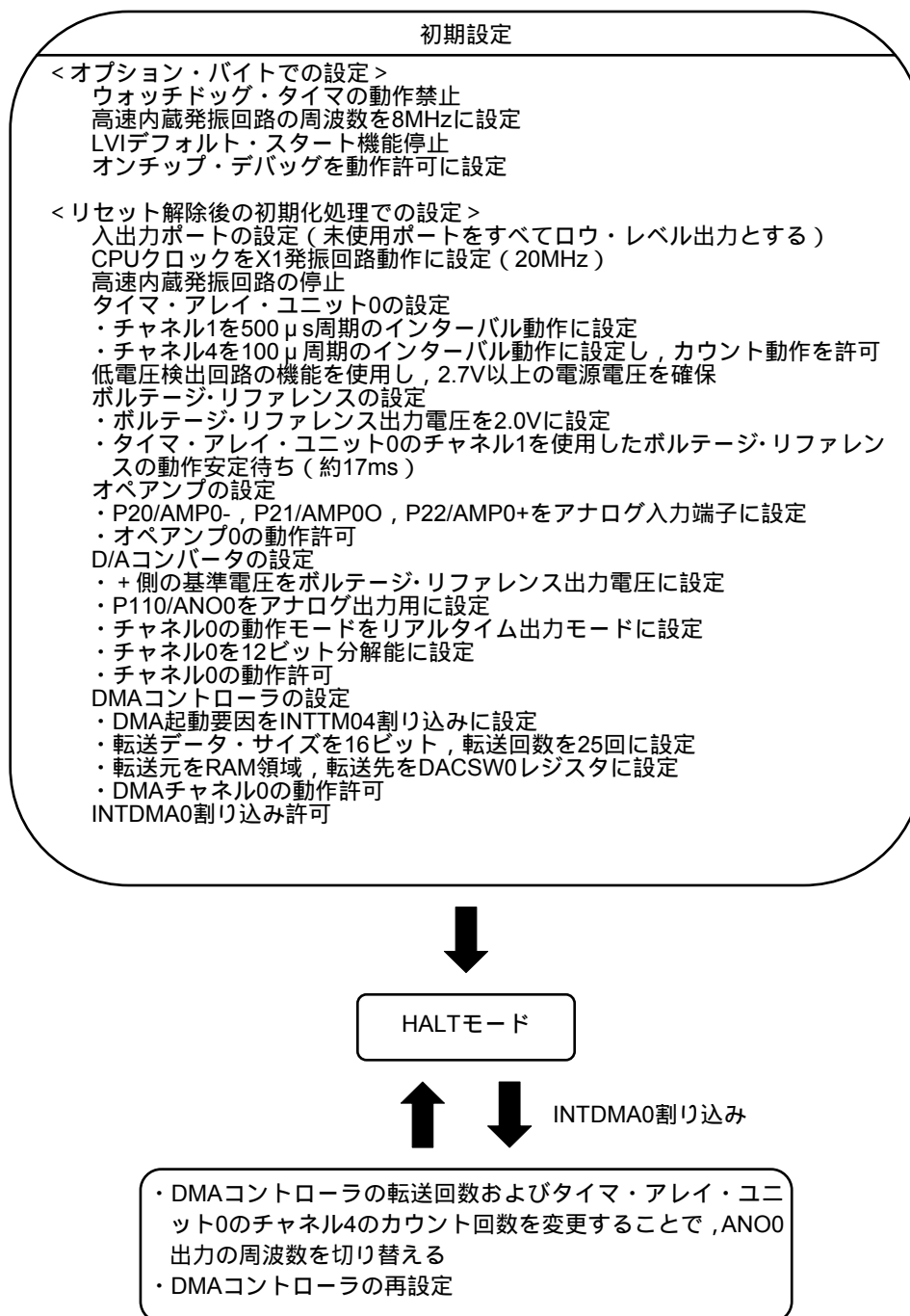
このサンプル・プログラムでは、初期設定にて、ポートの設定や、クロック周波数の選択、タイマ・アレイ・ユニット0、D/Aコンバータ、A/Dコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPにサイン波を送り続けます。DMA転送終了割り込みにてANI1の測定値が1Vであるか判定し、1Vに近づくようANO1の出力設定を調整します。また割り込み回数をカウントし、周期的にタイマ・アレイ・ユニット0・チャンネル4のインターバル時間を書換えることによりD/Aコンバータ出力信号の周波数を切替えます。さらにDMAコントローラを再設定してサイン波の出力を連続して行います。

詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.5 非反転増幅回路 交流

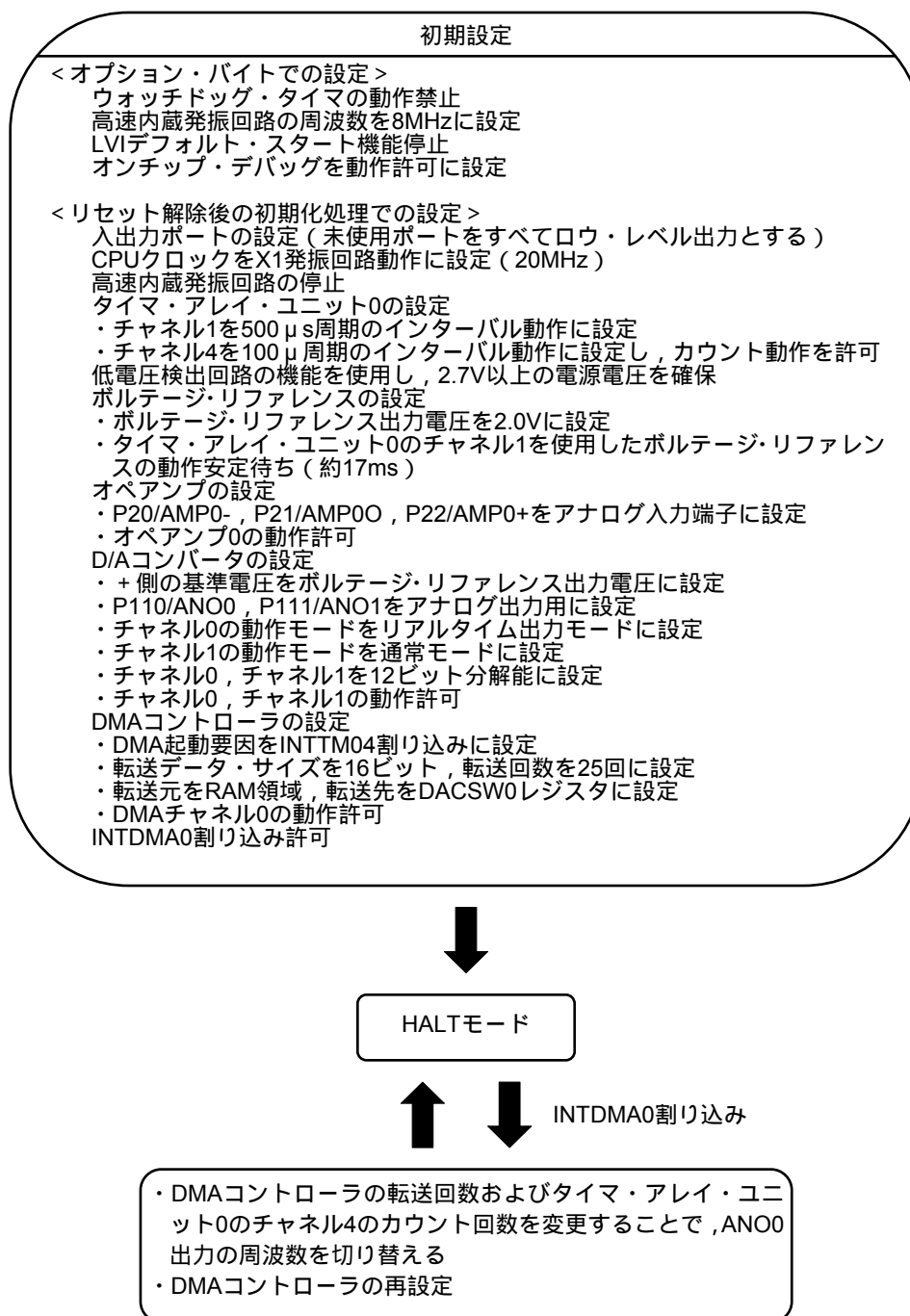
このサンプル・プログラムでは、初期設定にて、ポートの設定や、クロック周波数の選択、タイマ・アレイ・ユニット0、D/Aコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPに交流波形を送り続けます。DMA転送終了割り込みにて設定するDMA転送回数を調整し、出力波形を（sin 0 ~sin1/2・ ）（定電圧）（sin1/2・ ~sin1.5 ）（定電圧）（sin1.5 ~sin2 ）（低速変動電圧）を繰り返すようDMAコントローラを再設定します。詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.6 反転増幅回路 交流

このサンプル・プログラムでは、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、D/Aコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPに交流波形を送り続けます。DMA転送終了割り込みにて設定するDMA転送回数を調整し、出力波形を($\sin 0 \sim \sin 1/2 \cdot$) (定電圧) ($\sin 1/2 \cdot \sim \sin$) (低速変動電圧)を繰り返すようDMAコントローラを再設定します。D/Aコンバータ1はバイアス電圧を出力します。

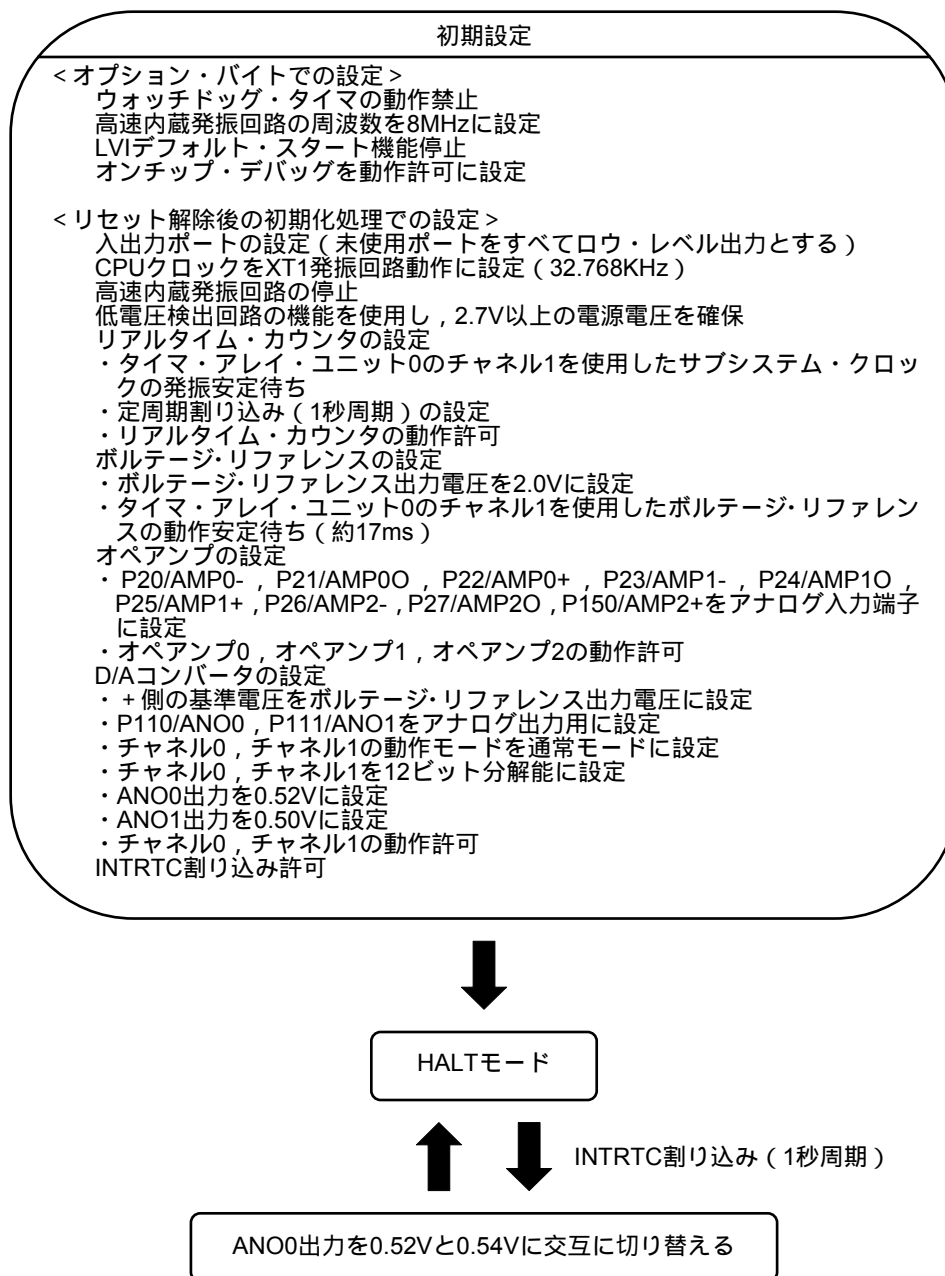
詳細については、次の状態遷移図(ステート・チャート)に示します。



3.3.7 インstrumentationアンブ

このサンプル・プログラムでは、初期設定にて、ポートの設定や、クロック周波数の選択、D/Aコンバータの設定などを行います。初期設定完了後、CPUクロックをサブシステム・クロックに切換えます。D/Aコンバータにより内蔵OPAMPに定電圧を送ります。1秒毎のRTC定周期割込みによりANO0の出力電圧を変更します。ANO1はバイアス電圧を出力します。

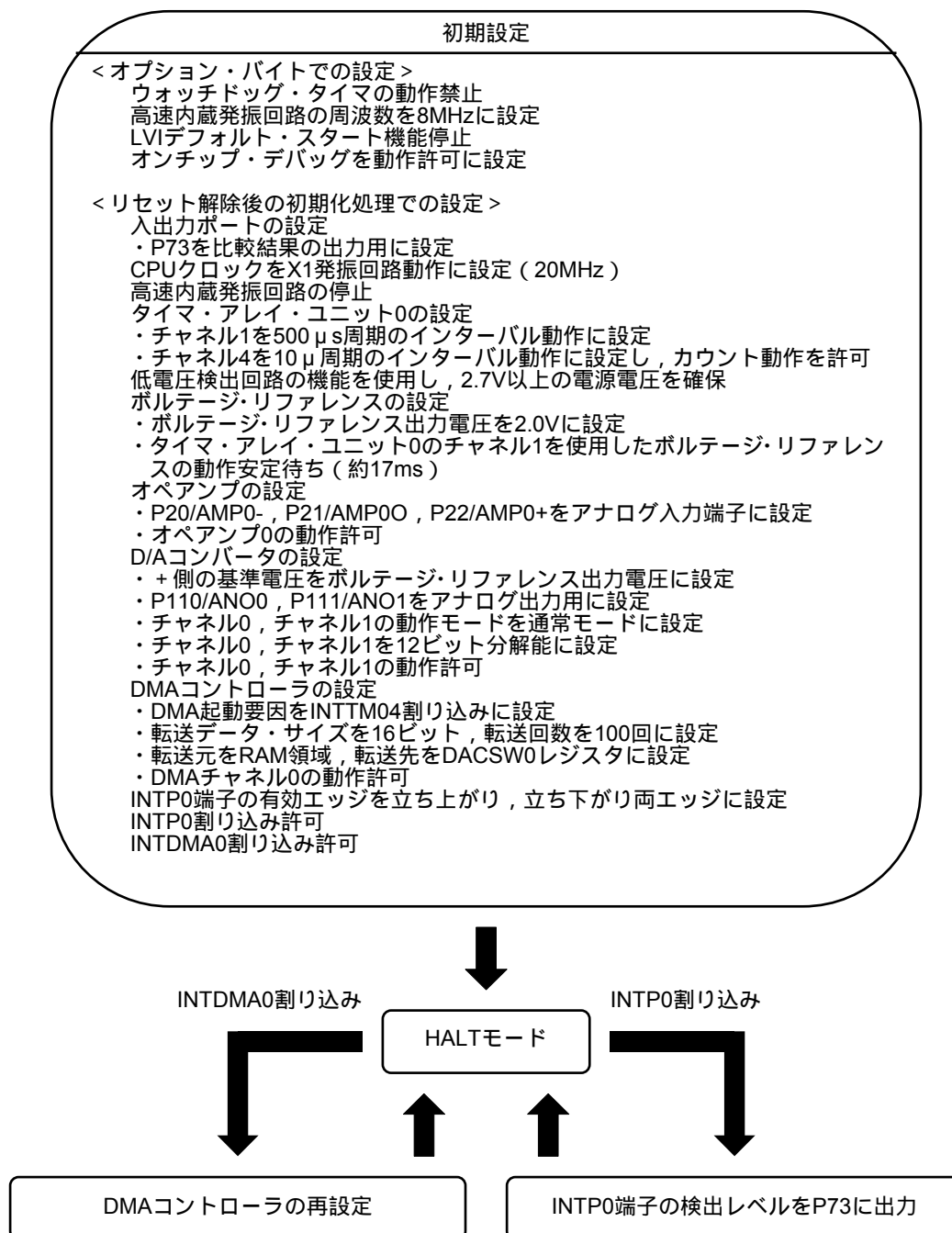
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.8 比較回路

このサンプル・プログラムでは、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、D/Aコンバータ、DMAコントローラの設定などを行います。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4、DMAコントローラ、D/Aコンバータにより内蔵OPAMPの反転入力にサイン波を送り続けます。DMA転送終了割り込みにてDMAコントローラを再設定してサイン波の出力を連続して行います。OPAMPの出力をINTP0に入力します。INTP0は両エッジ割り込みで使用し、INTP0割り込み処理でOPAMPの出力レベルをそのままPORT7.3に出力します。ANO1はOPAMPの非反転入力端子に入力する比較基準電圧を出力します。

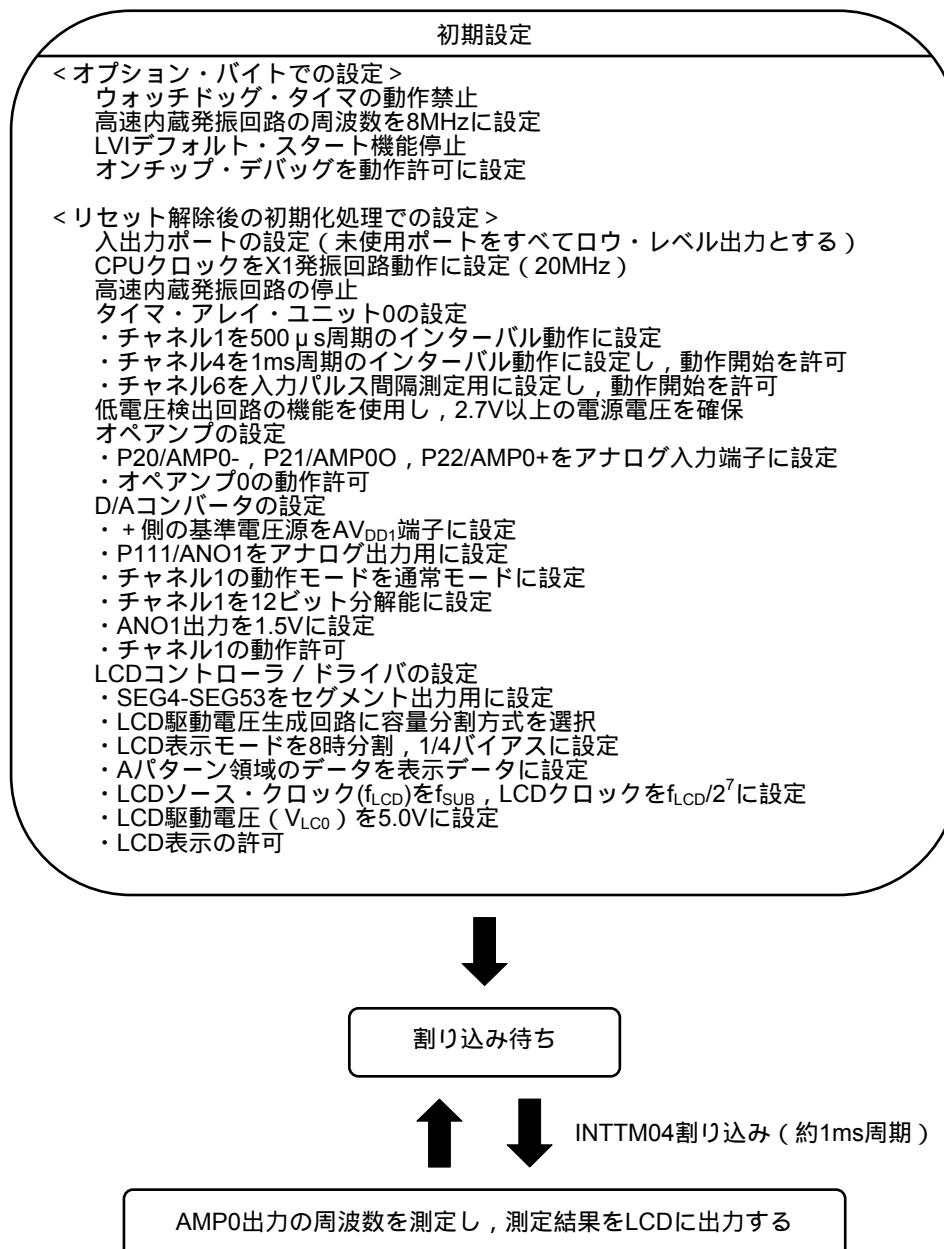
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.3.9 CR発振回路 1

このサンプル・プログラムでは、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。初期設定完了後、タイマ・アレイ・ユニット0・チャンネル4をインターバル・タイマ・モードでTI06へのパルスをカウントする期間(1ms)を計時します。タイマ・アレイ・ユニット0・チャンネル6にてTI06に入力されるパルスをイベント・カウント・モードにてカウントします。ANO1はOPAMPの非反転入力端子に入力するCR発振中心電圧を出力します。タイマ・アレイ・ユニット0・チャンネル4割り込み要求毎にタイマ・アレイ・ユニット0・チャンネル6のカウント値を読み出しLCDに10進数表示を行います。

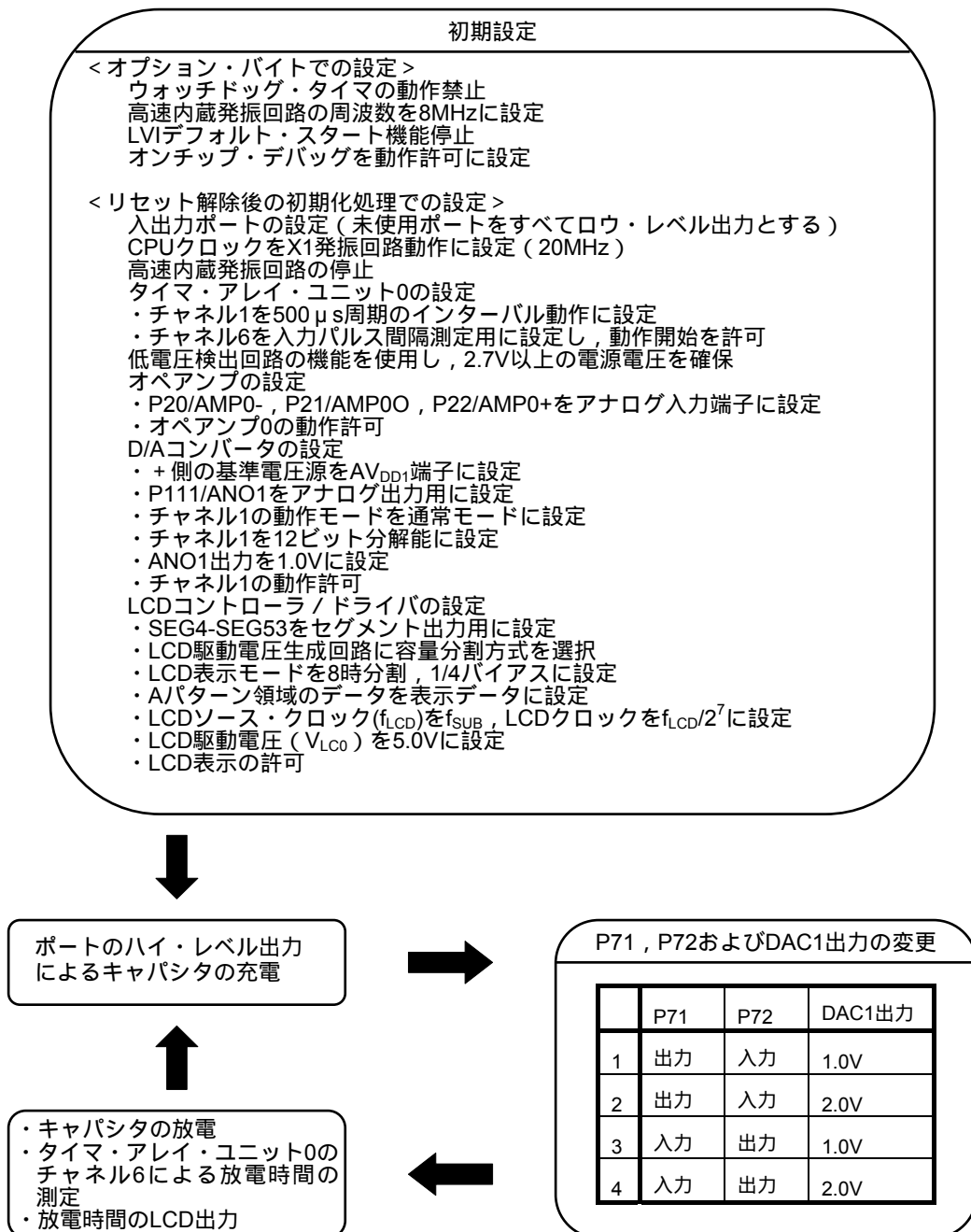
詳細については、次の状態遷移図(ステート・チャート)に示します。



3.3.10 CR発振回路 2

このサンプル・プログラムでは、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。OPAMPはコンパレータとして使用します。初期設定完了後、OPAMPの非反転入力に接続されたDAO1に比較基準電圧1Vを出力します。P70をHigh出力してOPAMPの反転入力に接続したキャパシタを充電します。そのまま一定時間ウェイト（充電完了待ち）後、P70を入力モードに切り替えP70を電氣的に切り放します。基準抵抗の接続されたP71をLow出力（キャパシタの放電開始）すると同時にTI06の立ち下がりエッジ・トリガのキャプチャ・モードに設定したタイマ・アレイ・ユニット0・チャンネル6のカウンタを許可します。タイマ・チャンネル6のキャプチャ割込み（放電完了）でタイマ・アレイ・ユニット0・チャンネル6のカウンタ値（放電時間）を取得します。次にDAO1に比較基準電圧2Vを出力します。DAO1に1V出力した後と同じ手順で放電時間を取得します。P71と同じ手順でP72に接続された被測定抵抗のDAO1に1Vと2Vを出力したときの放電時間を計測します。P71のDAO1に1Vと2V出力時のカウンタ値の差とP72のDAO1に1Vと2V出力時のカウンタ値の差をLCDに10進表示します。

詳細については、次の状態遷移図（ステート・チャート）に示します。

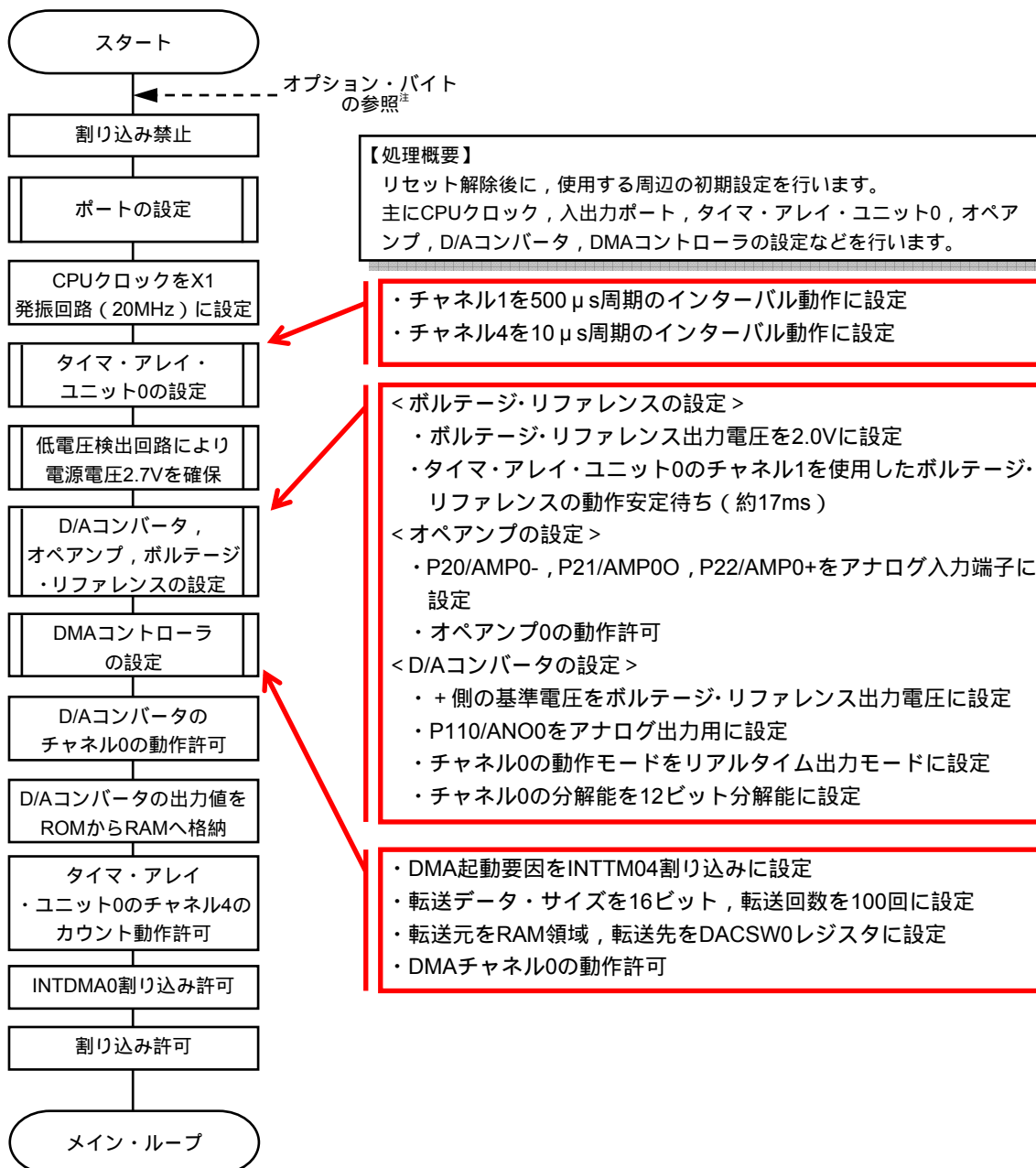


3.4. フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。

3.4.1 非反転増幅回路

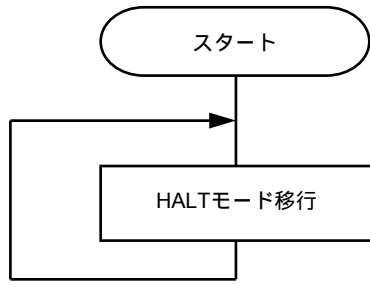
<リセット解除後の初期化処理での設定>



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

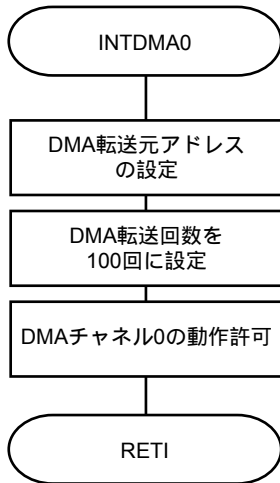
<メイン・ループ>



【処理概要】

HALTモードを使用し、INTDMA0割り込み待ちを行います。

<INTDMA0割り込み処理>

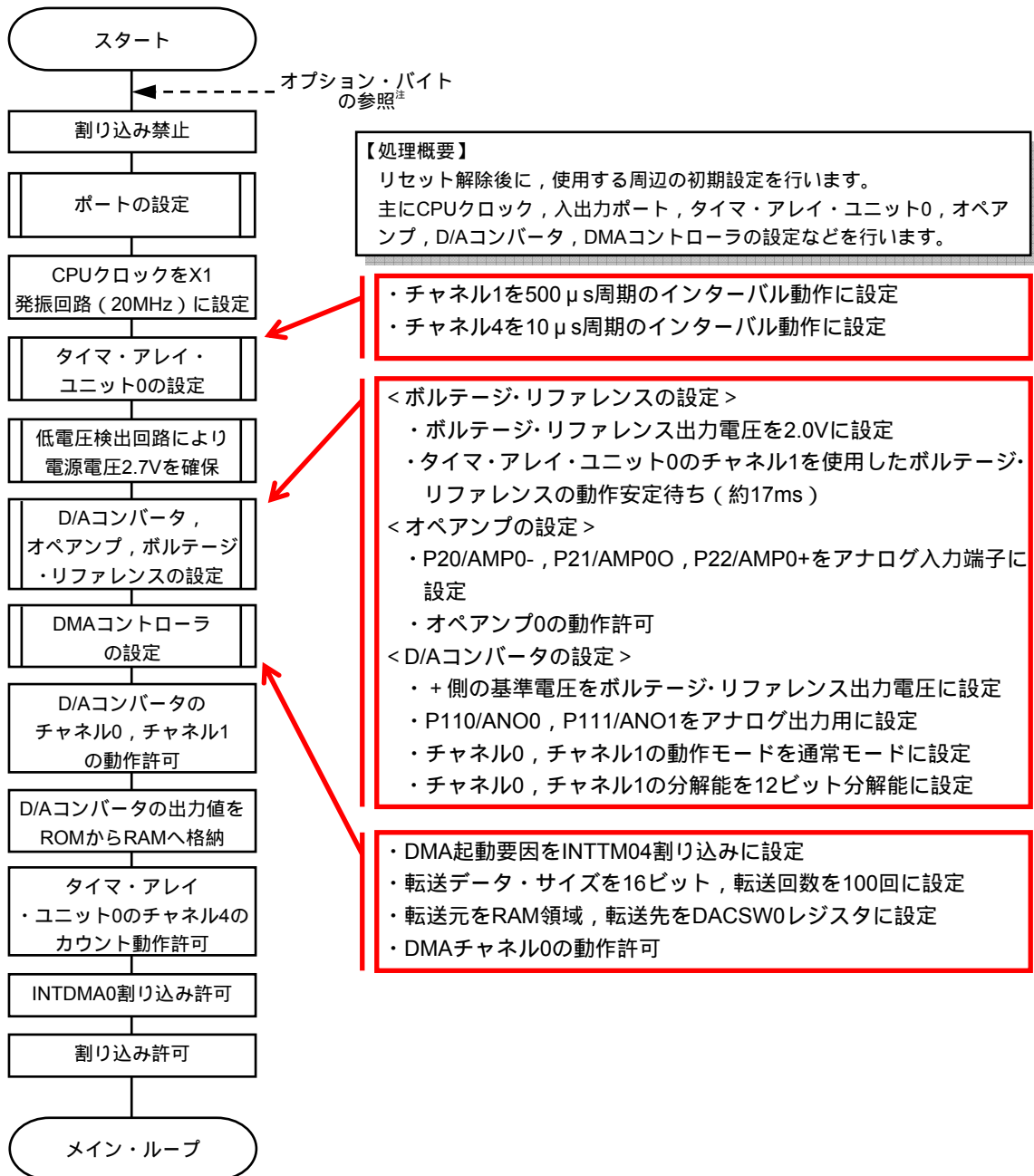


【処理概要】

DMAチャンネル0の再設定を行います。

3.4.2 反転増幅回路

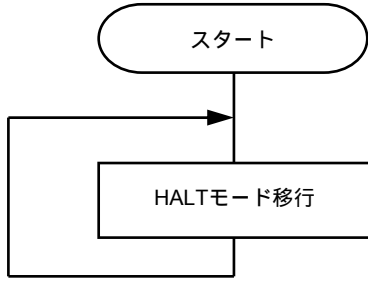
<リセット解除後の初期化処理での設定>



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

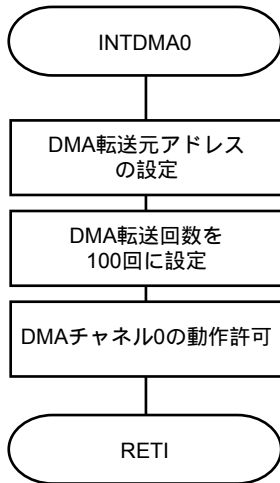
<メイン・ループ>



【処理概要】

HALTモードを使用し、INTDMA0割り込み待ちを行います。

<INTDMA0割り込み処理>

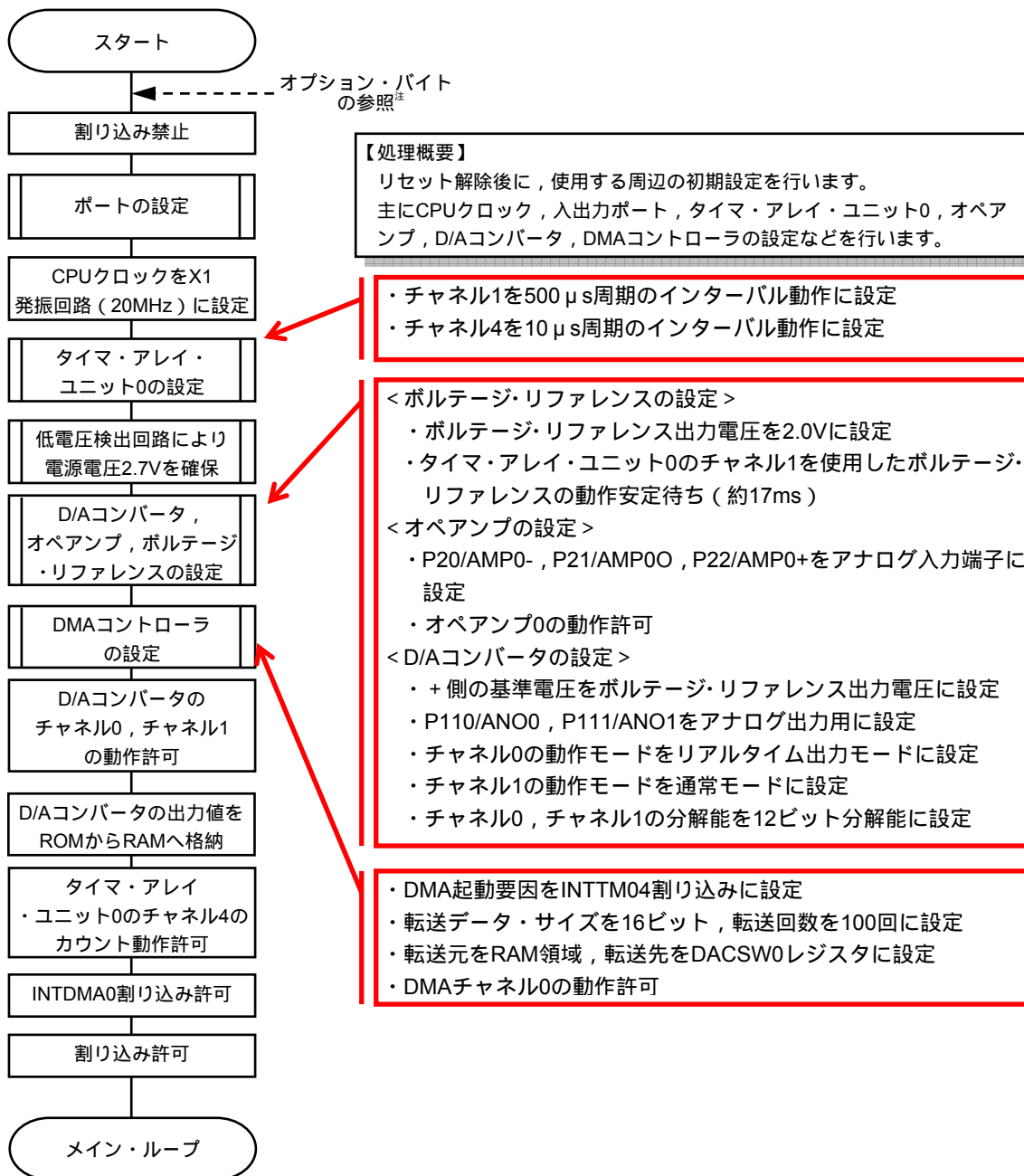


【処理概要】

DMAチャンネル0の再設定を行います。

3.4.3 反転増幅回路 オフセット調整

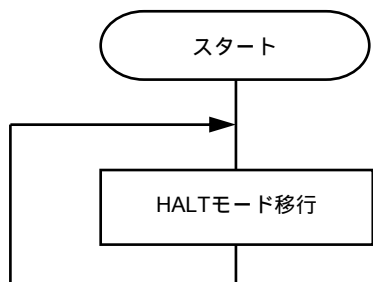
<リセット解除後の初期化処理での設定>



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。

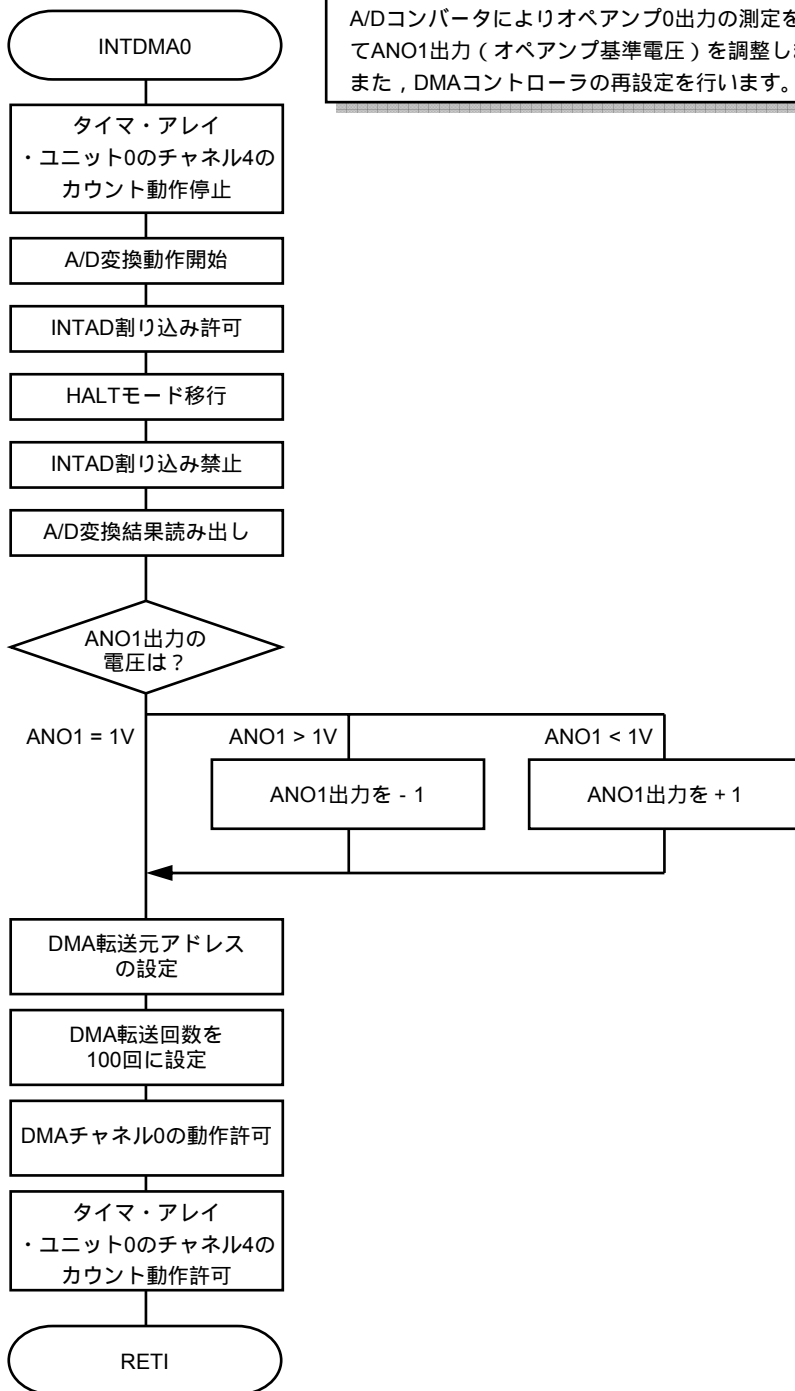
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振回路の周波数を8MHzに設定
- ・LVIデフォルト・スタート機能停止
- ・オンチップ・デバッグを動作許可に設定

<メイン・ループ>



【処理概要】
 HALTモードを使用し、INTDMA0割り込み待ちを行います。

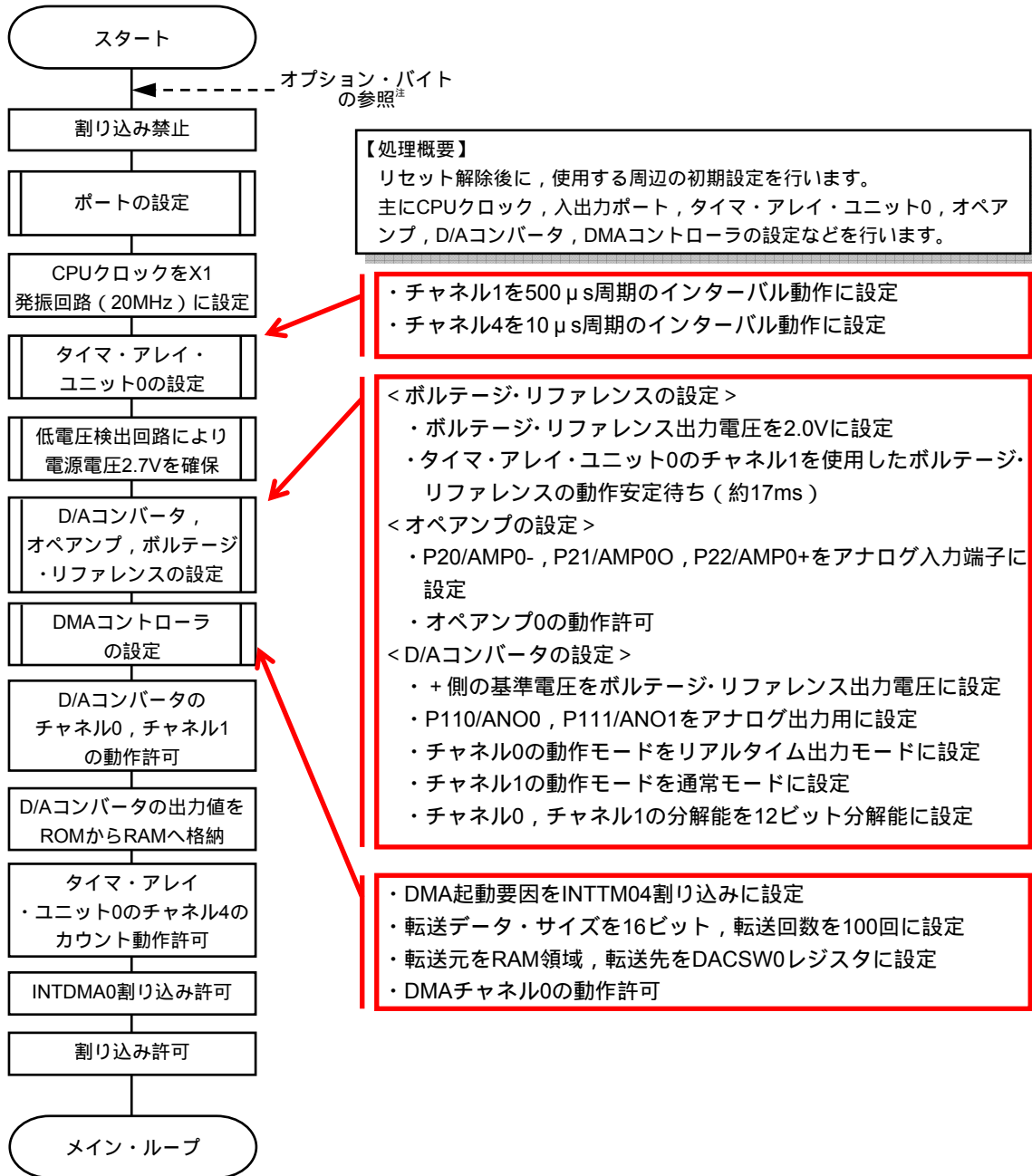
<INTDMA0割り込み処理>



【処理概要】
 A/Dコンバータによりオペアンプ0出力の測定を行い、測定結果に応じてANO1出力（オペアンプ基準電圧）を調整します。
 また、DMAコントローラの再設定を行います。

3.4.4 反転増幅回路 オフセット調整+ロウパスフィルター

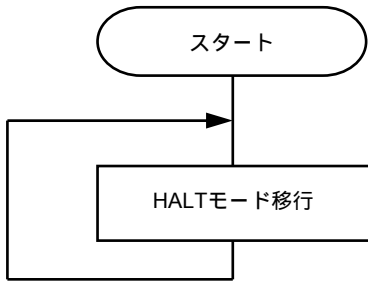
<リセット解除後の初期化処理での設定>



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

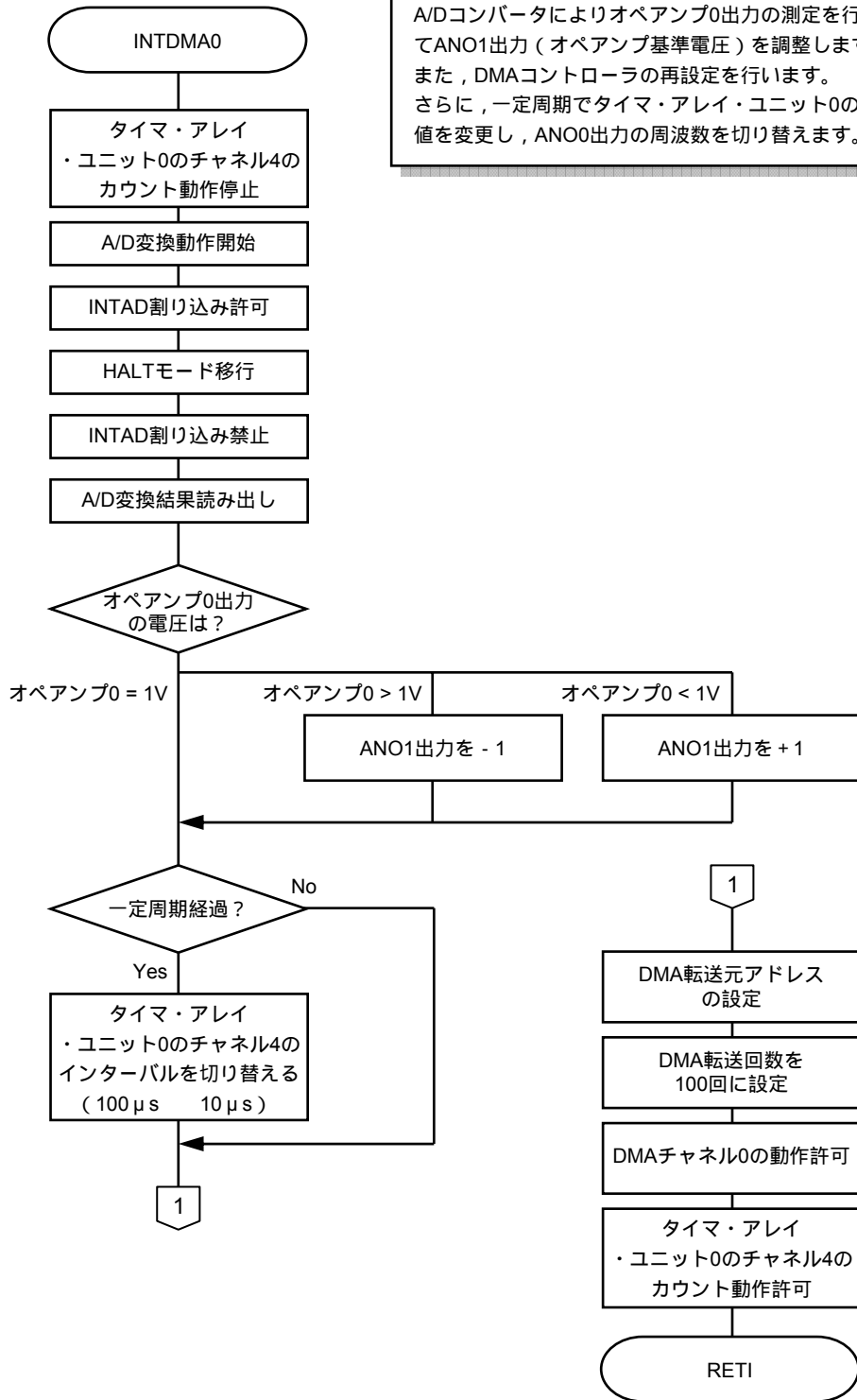
<メイン・ループ>



【処理概要】

HALTモードを使用し、INTDMA0割り込み待ちを行います。

<INTDMA0割り込み処理>

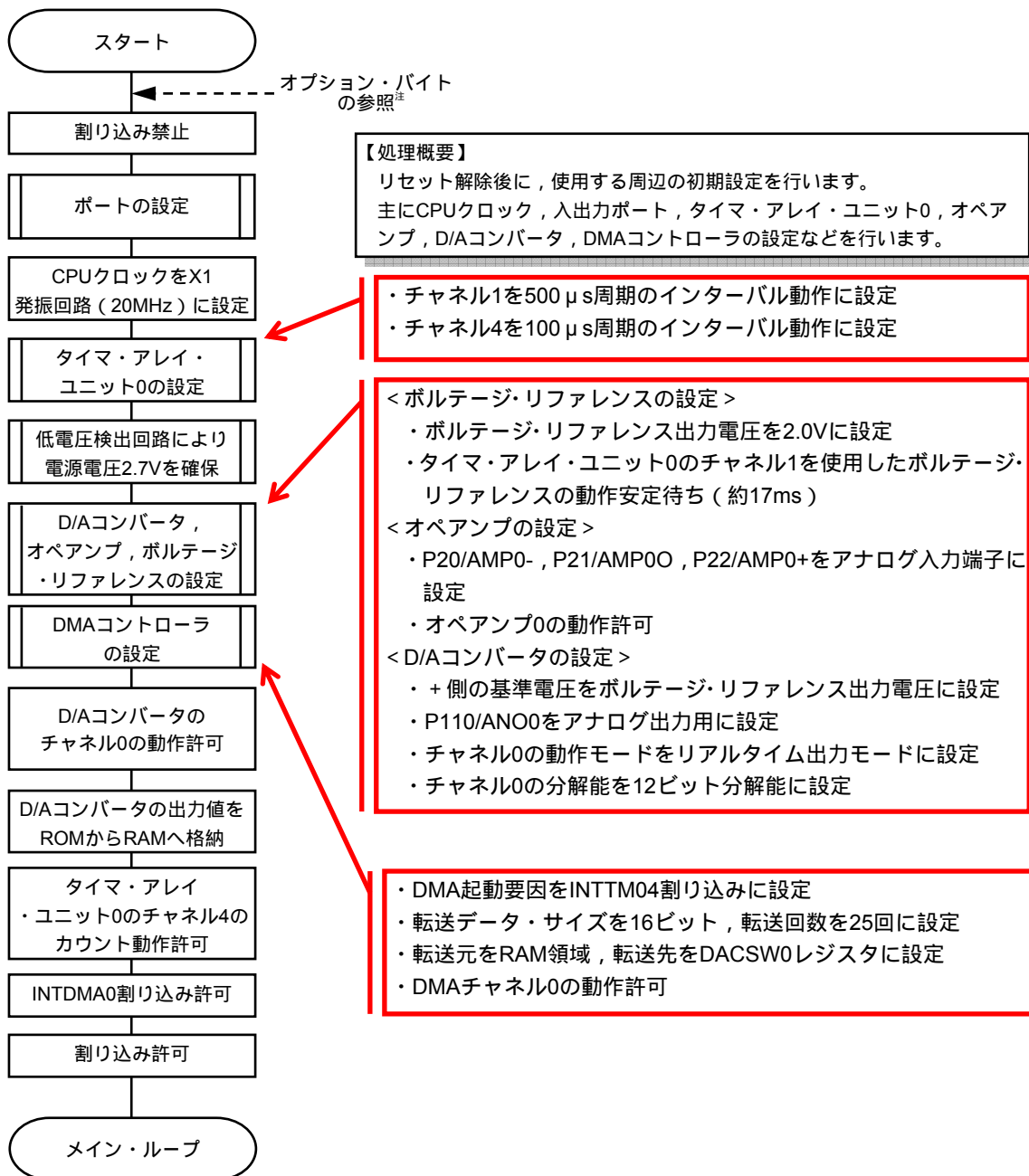


【処理概要】

A/Dコンバータによりオペアンプ0出力の測定を行い、測定結果に応じてANO1出力（オペアンプ基準電圧）を調整します。また、DMAコントローラの再設定を行います。さらに、一定周期でタイマ・アレイ・ユニット0のチャンネル4のカウント値を変更し、ANO0出力の周波数を切り替えます。

3.4.5 非反転増幅回路 交流

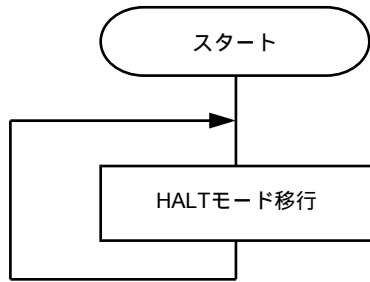
<リセット解除後の初期化処理での設定>



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

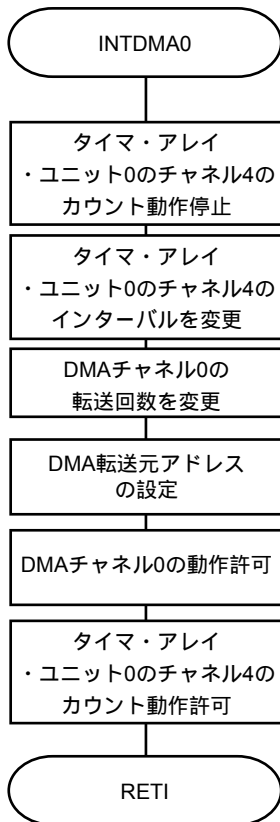
<メイン・ループ>



【処理概要】

HALTモードを使用し、INTDMA0割り込み待ちを行います。

<INTDMA0割り込み処理>

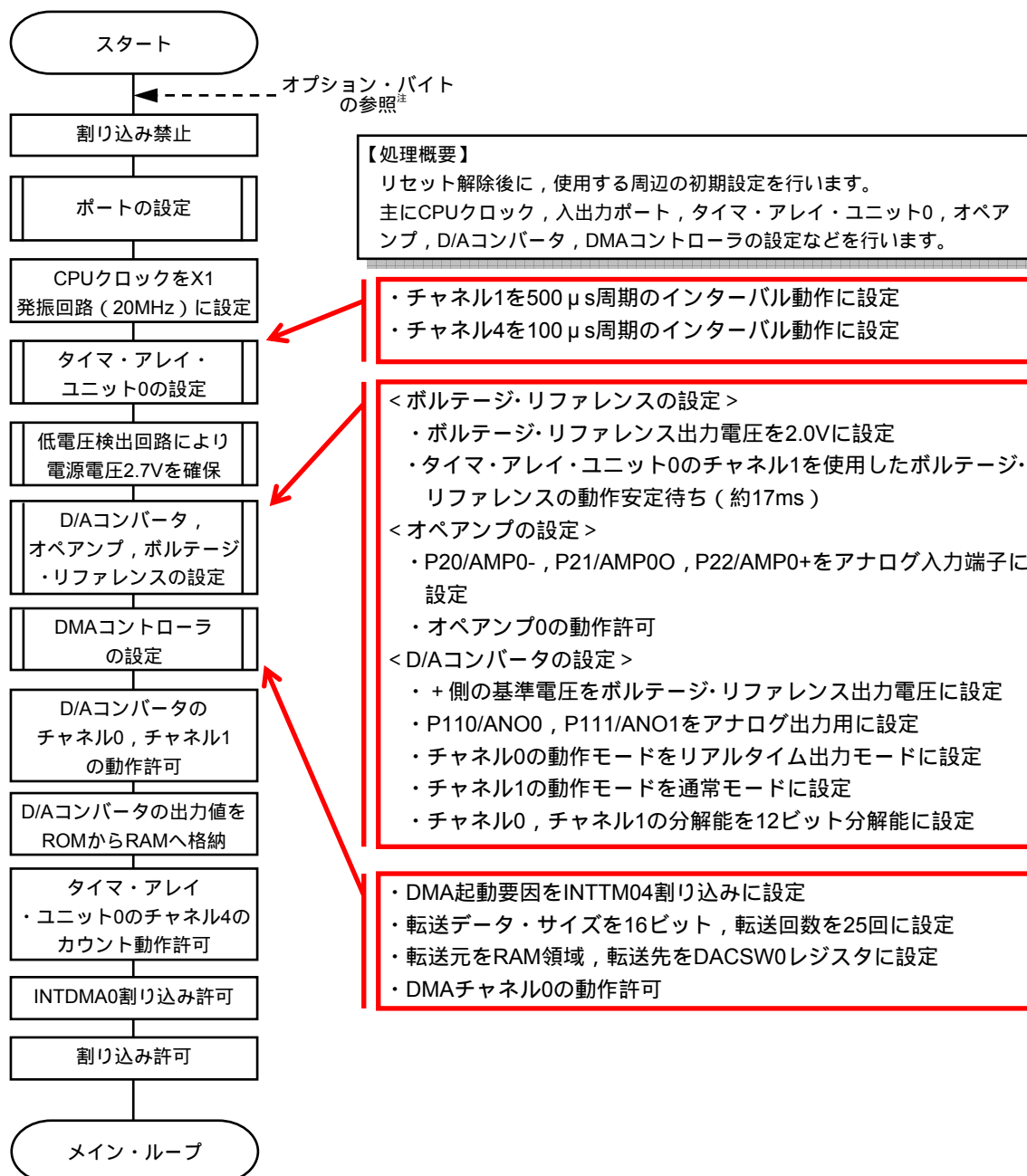


【処理概要】

タイマ・アレイ・ユニット0のチャンネル4のインターバル、およびDMAチャンネル0の転送回数を変更することで、ANO0出力が変化する時間を切り替えます（短い時間での変化と長い時間での変化の切り替え）。また、DMAコントローラの再設定を行います。

3.4.6 反転増幅回路 交流

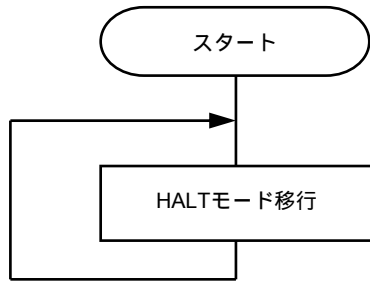
<リセット解除後の初期化処理での設定>



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。

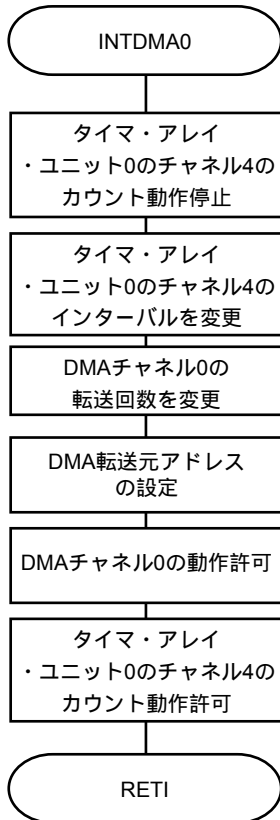
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振回路の周波数を8MHzに設定
- ・LVIデフォルト・スタート機能停止
- ・オンチップ・デバッグを動作許可に設定

<メイン・ループ>



【処理概要】
 HALTモードを使用し、INTDMA0割り込み待ちを行います。

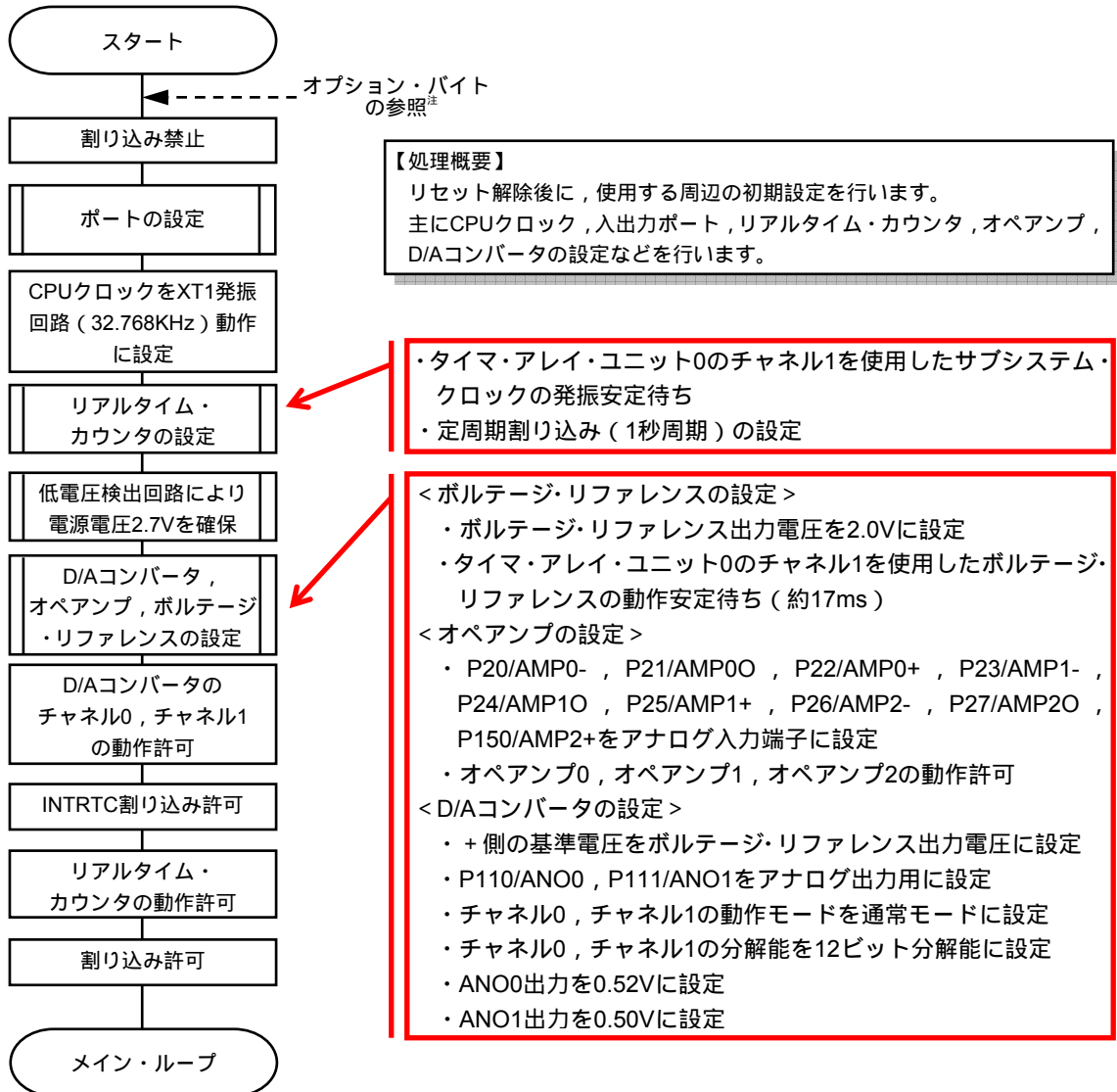
<INTDMA0割り込み処理>



【処理概要】
 タイマ・アレイ・ユニット0のチャンネル4のインターバル、およびDMAチャンネル0の転送回数を変更することで、ANO0出力が変化する時間を切り替えます（短い時間での変化と長い時間での変化の切り替え）。また、DMAコントローラの再設定を行います。

3.4.7 インストールメンテーションアンブ

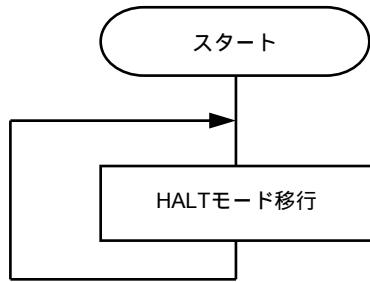
<リセット解除後の初期化処理での設定>



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。

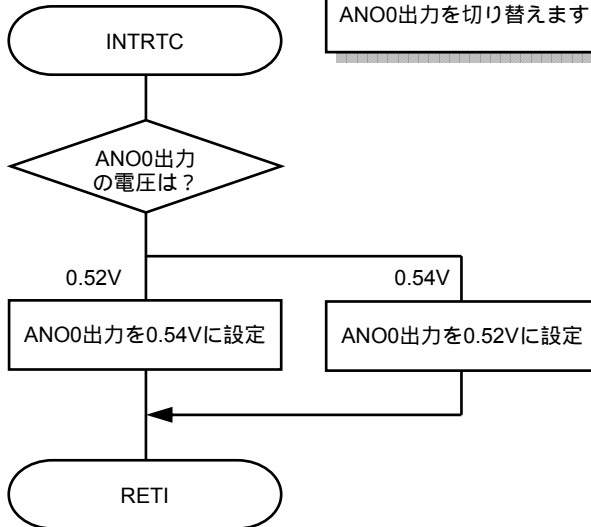
- ・ ウォッチドッグ・タイマの動作禁止
- ・ 高速内蔵発振回路の周波数を8MHzに設定
- ・ LVIデフォルト・スタート機能停止
- ・ オンチップ・デバッグを動作許可に設定

<メイン・ループ>



【処理概要】
 HALTモードを使用し、INTRTC割り込み待ちを行います。

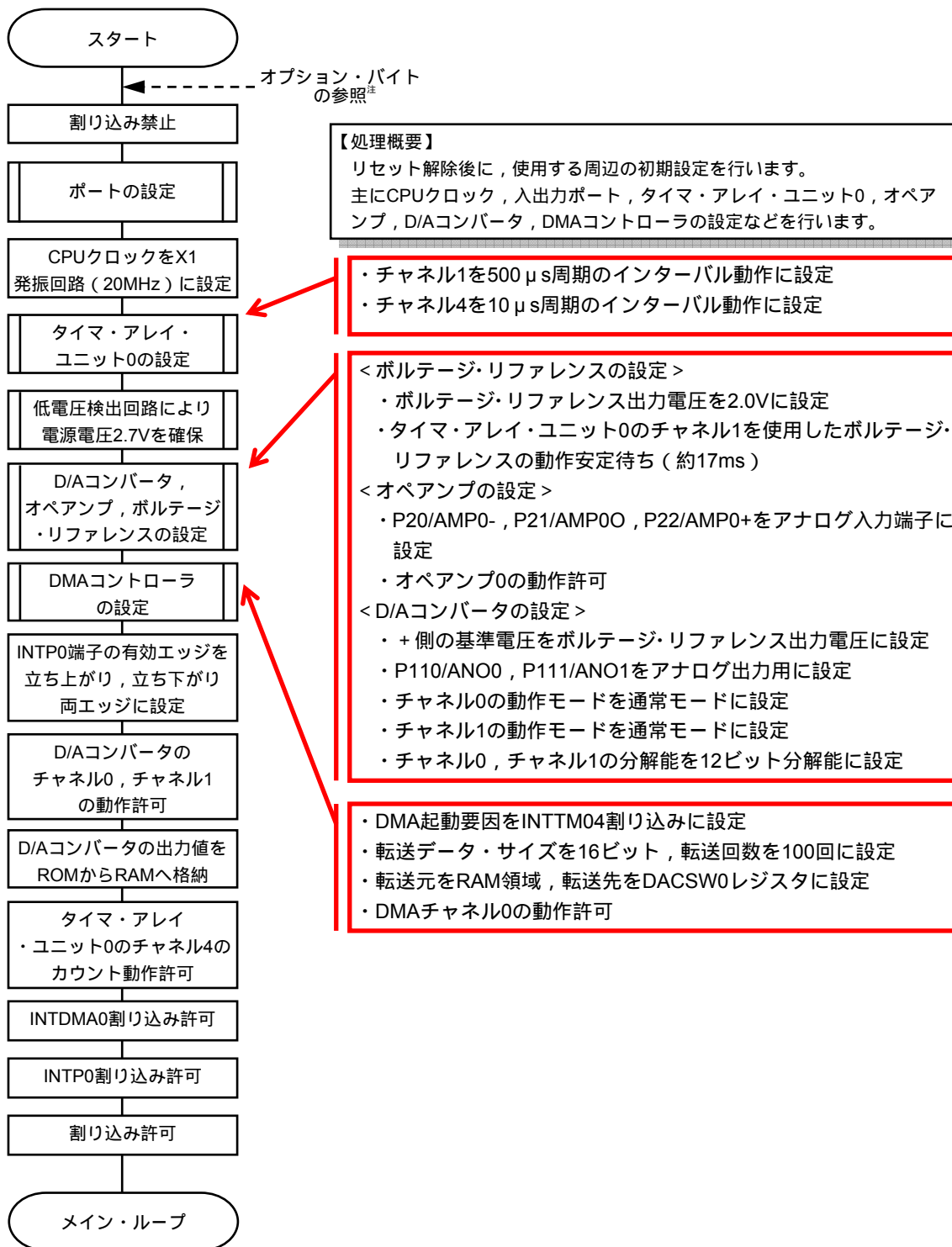
<INTRTC割り込み処理>



【処理概要】
 ANO0出力を切り替えます（0.52V 0.54V）。

3.4.8 比較回路

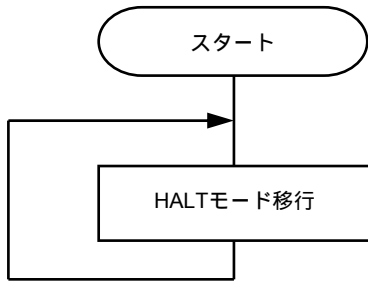
<リセット解除後の初期化処理での設定>



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。

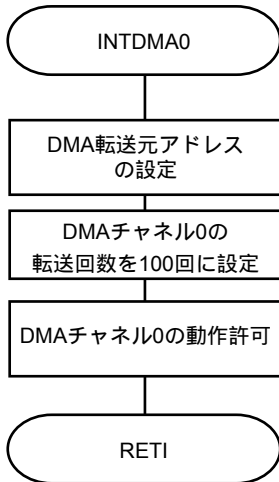
- ・ウォッチドッグ・タイマの動作禁止
- ・高速内蔵発振回路の周波数を8MHzに設定
- ・LVIデフォルト・スタート機能停止
- ・オンチップ・デバッグを動作許可に設定

<メイン・ループ>



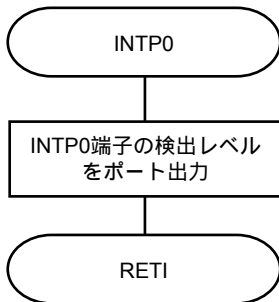
【処理概要】
 HALTモードを使用し、INTDMA0割り込み待ちを行います。

<INTDMA0割り込み処理>



【処理概要】
 DMAコントローラの再設定を行います。

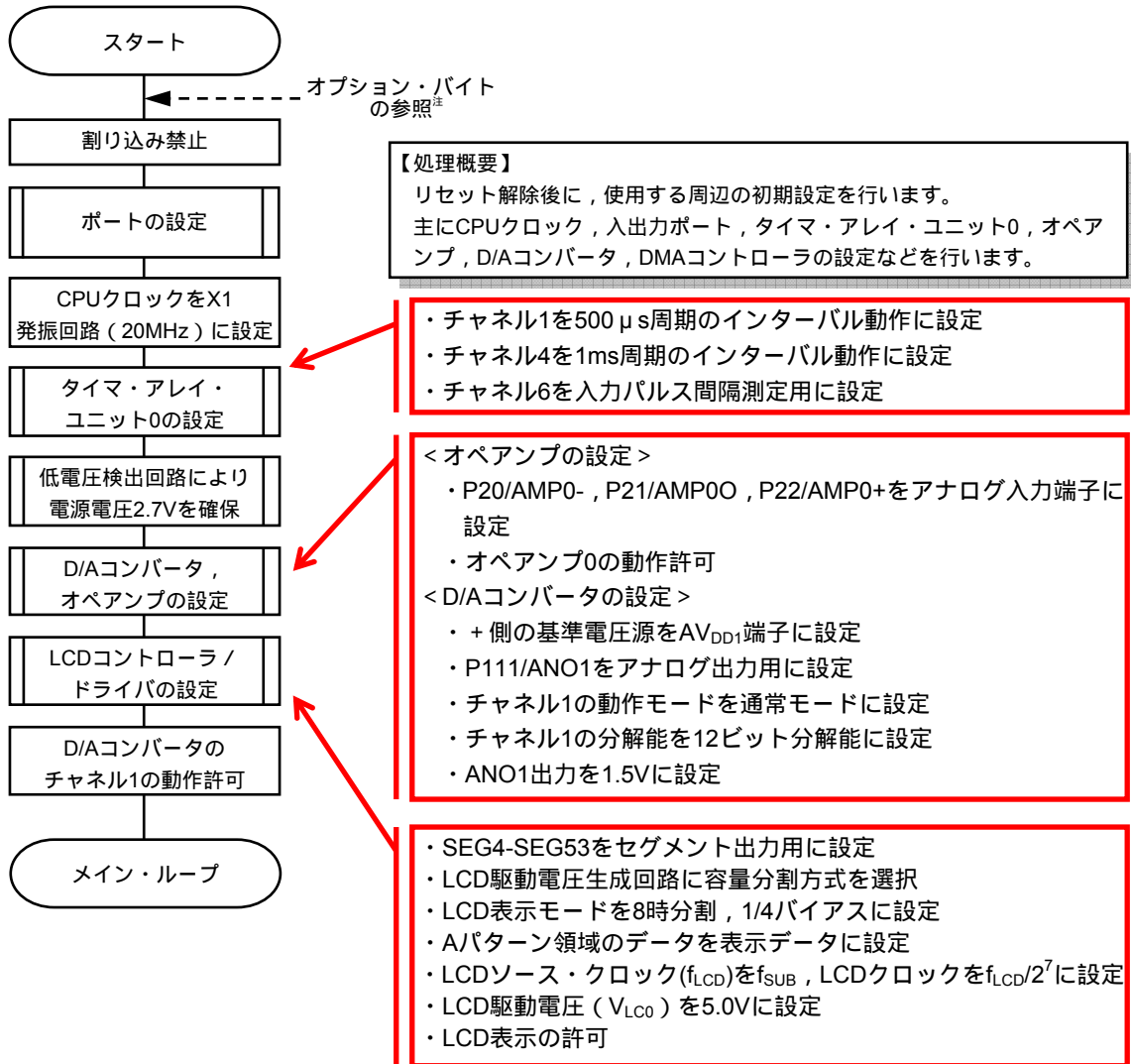
<INTP0割り込み処理>



【処理概要】
 INTP0端子の検出レベルをポート出力します。

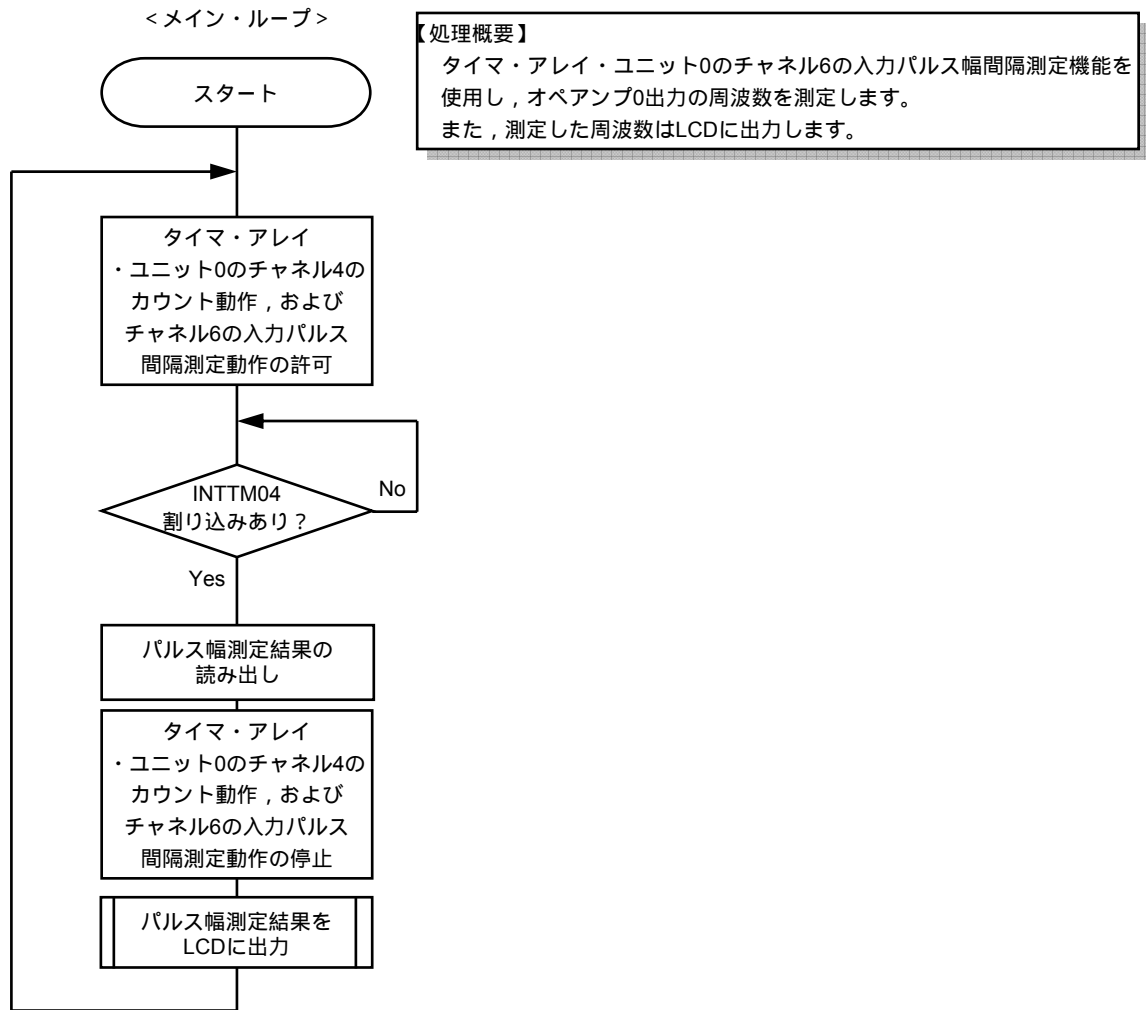
3.4.9 CR発振回路1

<リセット解除後の初期化処理での設定>



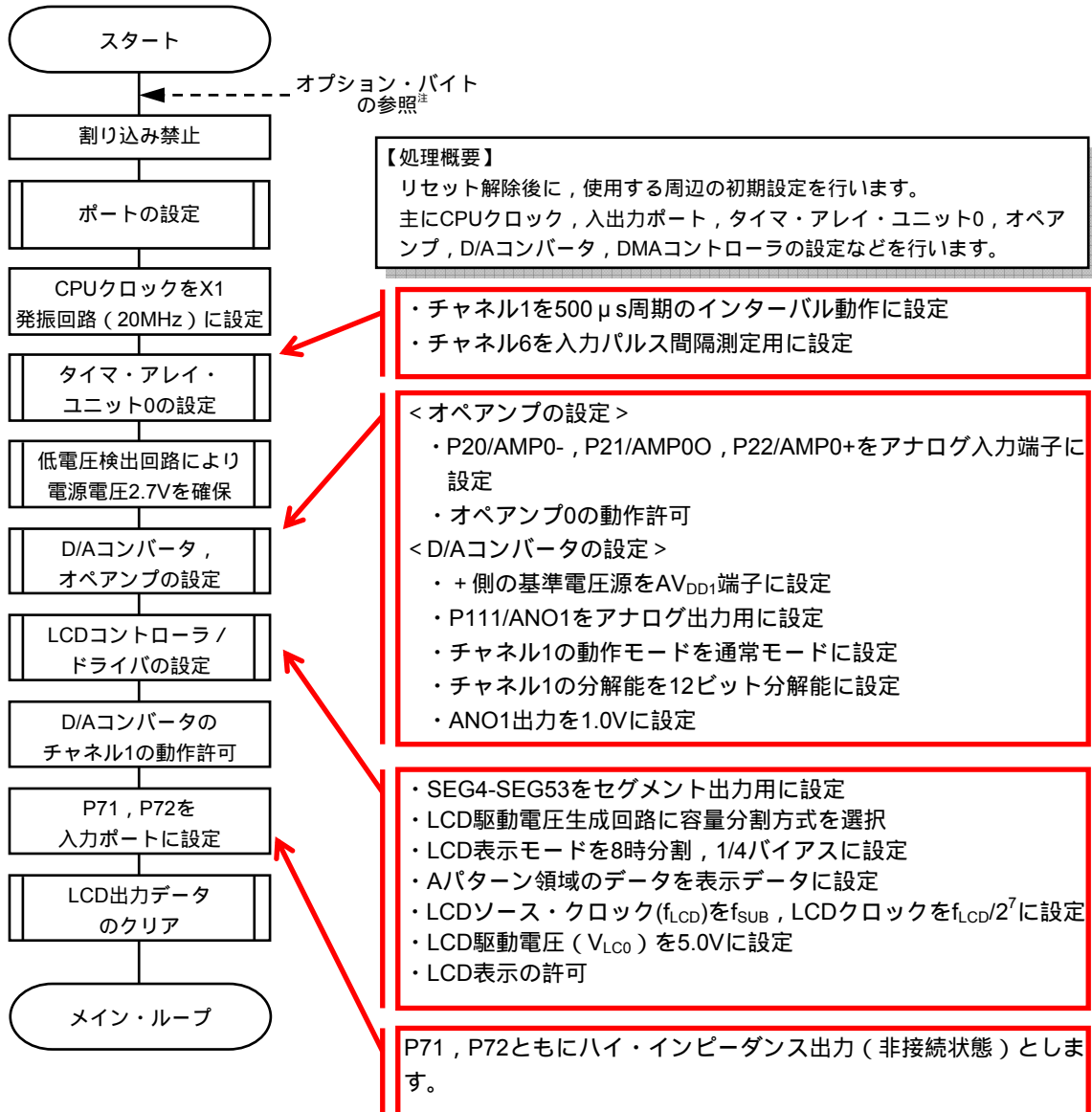
注. オプション・バイトの参照は、リセット解除後にマイコンが自動的に行います。このサンプル・プログラムでは、オプション・バイトで以下の設定を行います。

- ・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



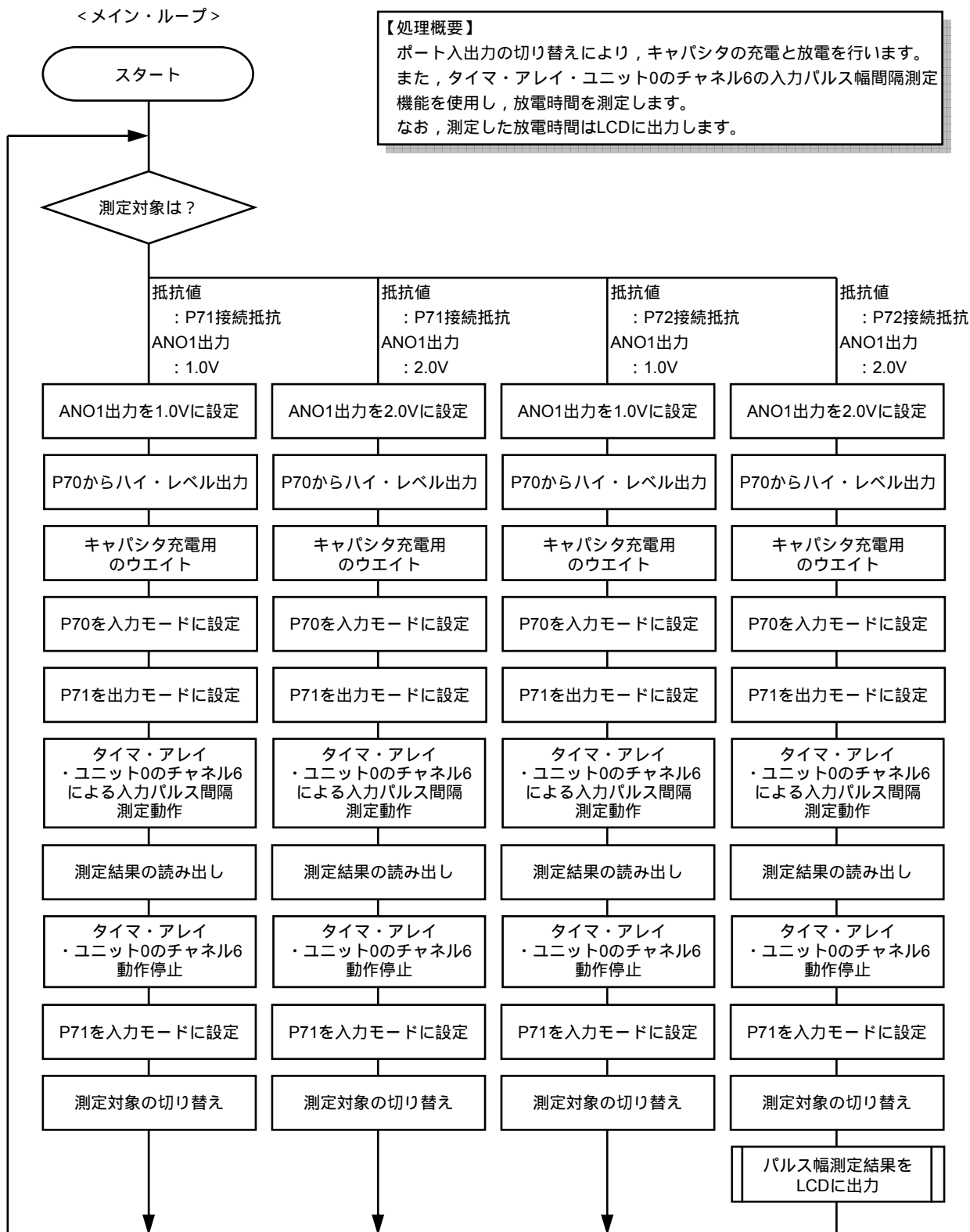
3.4.10 CR発振回路2

<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



第4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。

4.1 非反転増幅回路

4.1.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void fn_DmaInit(void)
{
----- DENO = 1; /* Enables operation of DMA channel 0 */
----- DSAO = (unsigned char)&DACSW0; /* DMA SFR Address Register 0 */
----- DRAO = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
----- DBCO = 100; /* DMA Byte Count Register 0 */
----- DMCO = 0b01100100; /* DMA Mode Control Register 0 */
/* ||||| */
/* ||||| +----- IFC3-IFC0 */
/* ||||| [Selection of DMA start source] */
/* ||||| 0000:Disables DMA transfer by interrupt */
/* ||||| (Only software trigger is enabled.) */
/* ||||| 0010:INTTM00 */
/* ||||| 0011:INTTM01 */
/* ||||| 0100:INTTM04 */
/* ||||| 0101:INTTM05 */
/* ||||| 0110:INTST0/INTCS100 */
/* ||||| 0111:INTSR0/INTCS101 */
/* ||||| 1000:INTST1/INTCS110/INTIIC10 */
/* ||||| 1001:INTSR1 */
/* ||||| 1010:INTST3 */
/* ||||| 1011:INTSR3 */
/* ||||| 1100:INTAD */
/* ||||| +----- DWAIT0 */
/* ||||| [Pending of DMA transfer] */
/* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
/* ||||| 1:Holds DMA start request pending if any. */
/* ||||| +----- DSO */
/* ||||| [Specification of transfer data size for DMA transfer] */
/* ||||| 0:8bits */
/* ||||| 1:16bits */
/* ||||| +----- DRSO */
/* ||||| [Selection of DMA transfer direction] */
/* ||||| 0:SFR to internal RAM */
/* ||||| 1:Internal RAM to SFR */
/* ||||| +----- STGO */
/* ||||| [DMA transfer start software trigger] */
/* ||||| 0:No trigger operation */
/* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
----- DSTO = 1; /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : リアルタイム出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0を動作許可します。

D/A変換の初期値を設定します。


```

void    fn_AdcDacInit(void)
{
    ADCEN = 1;                /* supplies input clock to A/D converter, operational amplifier, and
    voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000;      /* Analog reference voltage control register */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [1]Positive reference voltage supplies selection of A/D and D/A converters
    */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Operationcontrol of voltage reference */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]Output voltage selection of voltage reference */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Operation control of input gate voltage boost circuit for A/D converter
    */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5]Relationship with the conversion mode used */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Stops operation (Hi-Z) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.5 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Stops operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5]Can be set in conversion mode 1 */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Stops operation (Hi-Z) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.0 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5]Can be set in conversion mode 2 or 3 */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Stops operation (pull-down output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.5 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Stops operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5] - */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.5 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5]Can be set in conversion mode 2 or 3 */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Stops operation (pull-down output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.0 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5] - */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [2]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [3]2.0 V */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [4]Enables operation */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : [5]Can be set in conversion mode 2 or 3 */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : Other than the above : Setting prohibited */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : Be sure to set 0000 */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : Reference voltage supply (negative side) of A/D converter selection */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 0 : AVSS */
    /*|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+|+| : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;             /* Enables operation */
    ADVRC.1 = 1;             /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/*||| : 0 0 0 0 0 : A 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 A D */
/*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/*||| : 0 0 0 1 1 : A 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 D D D D D */
/*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*+++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMPO) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 1; /* D/A converter operation is Real-time output mode */
DAM.2 = 1; /* D/A conversion resolution is 12-bit */
DAM.6 = 1; /* D/A converter positive reference voltage is VREFOUT */
DACE0 = 0; /* D/A converter CH0 disable */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

}

```

4.1.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

DMA転送を再設定し、へ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    DACE0 = 1;                       /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL_4 = 1;                       /* start TAU0 CH4 */
    TMIF04 = 0;                       /* reset INTTMO4 interrupt request */

    DMAMK0 = 0;                       /* enable DMA CH0 interrupt */

    EI();                             /* enable all interrupts */

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

    while (1){
        HALT();
        NOP();
    }

    /*-----*/
    /* Module:      fn_intdma0          */
    /* Description: DMA ch0 interrupt process */
    /* parameter:  --                  */
    /* return      : --                  */
    /*-----*/
    __interrupt void fn_intdma0(void)
    {
        DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
        DBC0 = 100;                          /* DMA Byte Count Register 0 */

        DST0 = 1;                            /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */
    }

```

4.2 反転増幅回路

4.2.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・DMA転送方向 : 内蔵RAM SFR
- ・転送データ・サイズ : 16ビット
- ・DMA転送の保留 : 保留しない
- ・DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void    fn_DmaInit(void)
{
-----   DENO = 1;                               /* Enables operation of DMA channel 0 */
-----   DSAO = (unsigned char)&DACSW0;           /* DMA SFR Address Register 0 */
-----   DRAO = (unsigned short)&ushDacData[0];   /* DMA RAM Address Register 0 */
-----   DBCO = 100;                             /* DMA Byte Count Register 0 */
-----   DMC0 = 0b01100100;                     /* DMA Mode Control Register 0 */
          /* ||||| */
          /* ||||| +----- IFC3-IFC0 */
          /* ||||| [Selection of DMA start source] */
          /* ||||| 0000:Disables DMA transfer by interrupt */
          /* ||||| (Only software trigger is enabled.) */
          /* ||||| 0010:INTTM00 */
          /* ||||| 0011:INTTM01 */
          /* ||||| 0100:INTTM04 */
          /* ||||| 0101:INTTM05 */
          /* ||||| 0110:INTST0/INTCSI00 */
          /* ||||| 0111:INTSR0/INTCSI01 */
          /* ||||| 1000:INTST1/INTCSI10/INTIIC10 */
          /* ||||| 1001:INTSR1 */
          /* ||||| 1010:INTST3 */
          /* ||||| 1011:INTSR3 */
          /* ||||| 1100:INTAD */
          /* ||||| +----- DWAIT0 */
          /* ||||| [Pending of DMA transfer] */
          /* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
          /* ||||| 1:Holds DMA start request pending if any. */
          /* ||||| +----- DSO */
          /* ||||| [Specification of transfer data size for DMA transfer] */
          /* ||||| 0:8bits */
          /* ||||| 1:16bits */
          /* ||||| +----- DRS0 */
          /* ||||| [Selection of DMA transfer direction] */
          /* ||||| 0:SFR to internal RAM */
          /* ||||| 1:Internal RAM to SFR */
          /* ||||| +----- STGO */
          /* ||||| [DMA transfer start software trigger] */
          /* ||||| 0:No trigger operation */
          /* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
-----   DSTO = 1;                               /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0の初期値を設定します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void fn_AdcDacInit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /*| | | | + | + + - - - : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /*| | | | | : [2]Operation control of voltage reference */
    /*| | | | | : [3]Output voltage selection of voltage reference */
    /*| | | | | : [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /*| | | | | : [5]Relationship with the conversion mode used */
    /*| | | | | */
    /*| | | | | : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /*| | | | | : [2]Stops operation (Hi-Z) */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Stops operation */
    /*| | | | | : [5]Can be set in conversion mode 1 */
    /*| | | | | */
    /*| | | | | : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /*| | | | | : [2]Stops operation (Hi-Z) */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Stops operation (pull-down output) */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Stops operation */
    /*| | | | | : [5] - */
    /*| | | | | */
    /*| | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Enables operation */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Stops operation (pull-down output) */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5] - */
    /*| | | | | */
    /*| | | | | : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Enables operation */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : Other than the above : Setting prohibited */
    /*| | | | | */
    /*| + + + - - - - - : Be sure to set 0000 */
    /*| */
    /*+----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;      /* Enables operation */
    ADVRC.1 = 1;      /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/* ||| : 0 0 0 0 0 : A 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 A D */
/* ||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/* ||| : 0 0 0 1 1 : A A 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 D D D D D D */
/* ||| : 0 0 1 1 1 : A A A A A D 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 */
/* ||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
/* ||| */
/* +++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMPO) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

/* set D/A channel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting register 1 */

}

```


4.2.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0,1を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

DMA転送を再設定し、へ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    DACE1 = 1;                      /* D/A converter CH1 enable */
    DACE0 = 1;

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                      /* start TAU0 CH4 */
    TMIF04 = 0;                      /* reset INTTMO4 interrupt request */

    DMAMK0 = 0;                      /* enable DMA CH0 interrupt */

    EI();                            /* enable all interrupts */

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

    while (1){
        HALT();
        NOP();
    }

    /*-----*/
    /* Module:          fn_intdma0          */
    /* Description:    DMA ch0 interrupt process          */
    /* parameter:    --          */
    /* return       : --          */
    /*-----*/
    __interrupt void fn_intdma0(void)
    {
        DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
        DBC0 = 100;                              /* DMA Byte Count Register 0 */

        DSTO = 1;                                /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */
    }

```

4.3 反転増幅回路 オフセット調整

4.3.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void    fn_DmaInit(void)
{
-----   DENO = 1;                               /* Enables operation of DMA channel 0 */
-----   DSAO = (unsigned char)&DACSW0;           /* DMA SFR Address Register 0 */
-----   DRAO = (unsigned short)&ushDacData[0];   /* DMA RAM Address Register 0 */
-----   DBCO = 100;                             /* DMA Byte Count Register 0 */
-----   DMC0 = 0b01100100;                       /* DMA Mode Control Register 0 */
          /* ||||| */
          /* ||||| +----- IFC3-IFC0 */
          /* ||||| [Selection of DMA start source] */
          /* ||||| 0000:Disables DMA transfer by interrupt */
          /* ||||| (Only software trigger is enabled.) */
          /* ||||| 0010:INTTM00 */
          /* ||||| 0011:INTTM01 */
          /* ||||| 0100:INTTM04 */
          /* ||||| 0101:INTTM05 */
          /* ||||| 0110:INTST0/INTCS100 */
          /* ||||| 0111:INTSR0/INTCS101 */
          /* ||||| 1000:INTST1/INTCS110/INTIIC10 */
          /* ||||| 1001:INTSR1 */
          /* ||||| 1010:INTST3 */
          /* ||||| 1011:INTSR3 */
          /* ||||| 1100:INTAD */
          /* ||||| +----- DWAIT0 */
          /* ||||| [Pending of DMA transfer] */
          /* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
          /* ||||| 1:Holds DMA start request pending if any. */
          /* ||||| +----- DSO */
          /* ||||| [Specification of transfer data size for DMA transfer] */
          /* ||||| 0:8bits */
          /* ||||| 1:16bits */
          /* ||||| +----- DRS0 */
          /* ||||| [Selection of DMA transfer direction] */
          /* ||||| 0:SFR to internal RAM */
          /* ||||| 1:Internal RAM to SFR */
          /* ||||| +----- STGO */
          /* ||||| [DMA transfer start software trigger] */
          /* ||||| 0:No trigger operation */
          /* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
-----   DSTO = 1;                               /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0の初期値を設定します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void  fn_AdcDacInit(void)
{
    ADCEN = 1;           /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /*| | | | + | + + - - : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /*| | | | | : [2]Operationcontrol of voltage reference */
    /*| | | | | : [3]Output voltage selection of voltage reference */
    /*| | | | | : [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /*| | | | | : [5]Relationship with the conversion mode used */
    /*| | | | | */
    /*| | | | | : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /*| | | | | : [2]Stops operation (Hi-Z) */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Stops operation */
    /*| | | | | : [5]Can be set in conversion mode 1 */
    /*| | | | | */
    /*| | | | | : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /*| | | | | : [2]Stops operation (Hi-Z) */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Stops operation (pull-down output) */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Stops operation */
    /*| | | | | : [5] - */
    /*| | | | | */
    /*| | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Enables operation */
    /*| | | | | : [3]2.5 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Stops operation (pull-down output) */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5] - */
    /*| | | | | */
    /*| | | | | : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /*| | | | | : [2]Enables operation */
    /*| | | | | : [3]2.0 V */
    /*| | | | | : [4]Enables operation */
    /*| | | | | : [5]Can be set in conversion mode 2 or 3 */
    /*| | | | | */
    /*| | | | | : Other than the above : Setting prohibited */
    /*| | | | | */
    /*| + + + + - - - - : Be sure to set 0000 */
    /*| */
    /*+----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;         /* Enables operation */
    ADVRC.1 = 1;         /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/*||| : 0 0 0 0 0 : A A 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 A D */
/*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/*||| : 0 0 0 1 1 : A 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 D D D D D */
/*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*+++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMPO) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

/* set D/A channel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[80]; /* D/A conversion value setting register 1 */

}

```

4.3.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0,1を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

TM04を停止します。

1周期完了時点での出力電圧をA/D変換します。

出力電圧のA/D変換値と理想状態での基準電圧のA/D変換値を比較して、オフセット調整します。

オフセット調整完了後、基準電圧にずれを発生させます（動作確認のため）。

DMA転送を再設定し、TM04の動作を再開してへ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    ushCycleCount = 0;                /* Cycle counter */

    DACE1 = 1;                        /* D/A converter CH1 enable */
    DACE0 = 1;

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                       /* start TAU0 CH4 */
    TMIF04 = 0;                        /* reset INTTM04 interrupt request */

    DMAMKO = 0;                       /* enable DMA CH0 interrupt */

    EI();                              /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }
}

/*-----*/
/* Module:      fn_intdma0          */
/* Description:  DMA ch0 interrupt process */
/* parameter:  --                  */
/* return      :  --                  */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TTOL.4 = 1;                       /* stop TAU0 CH4 */

```

```
/* start A/D conversion */
ADCS = 1;          /* A/D converter enable */
ADIF = 0;          /* clear A/D interrupt request flag */
ADMK = 0;          /* enable A/D interrupt */
HALT();           /* set HALT mode and wait conversion finished */
ADMK = 1;          /* disable A/D interrupt */
ADIF = 0;          /* clear A/D interrupt request flag */

shVoltage = (short)(ADCR & 0x0fff); /* get A/D result */
shVoltage -= (2048 - 1);
if(shVoltage < 0){
    DACSW1 += 1;          /* D/A conversion value setting register 1 */
}
else if(shVoltage > 2){
    DACSW1 -= 1;          /* D/A conversion value setting register 1 */
}

ushCycleCount++;      /* Cycle counter */
if(ushCycleCount > 100){ /* Cycle counter */
    DACSW1 = ushDacData[95]; /* D/A conversion value setting register 1 */
    ushCycleCount = 0;      /* reset Cycle counter */
}

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100; /* DMA Byte Count Register 0 */

DST0 = 1; /* DMA transfer is started when DMA operation is enabled (DENO = 1). */

TSOL_4 = 1; /* start TAU0 CH4 */
TMIF04 = 0; /* reset INTTMO4 interrupt request */
}
```


4.4 反転増幅回路 オフセット調整 + ローパスフィルター

4.4.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void    fn_DmaInit(void)
{
-----   DENO = 1;                               /* Enables operation of DMA channel 0 */
-----   DSAO = (unsigned char)&DACSW0;           /* DMA SFR Address Register 0 */
-----   DRAO = (unsigned short)&ushDacData[0];   /* DMA RAM Address Register 0 */
-----   DBCO = 100;                             /* DMA Byte Count Register 0 */
-----   DMCO = 0b01100100;                       /* DMA Mode Control Register 0 */
          /* ||||| */
          /* ||||| +----- IFC3-IFC0 */
          /* ||||| [Selection of DMA start source] */
          /* ||||| 0000:Disables DMA transfer by interrupt */
          /* ||||| (Only software trigger is enabled.) */
          /* ||||| 0010:INTTM00 */
          /* ||||| 0011:INTTM01 */
          /* ||||| 0100:INTTM04 */
          /* ||||| 0101:INTTM05 */
          /* ||||| 0110:INTST0/INTCS100 */
          /* ||||| 0111:INTSR0/INTCS101 */
          /* ||||| 1000:INTST1/INTCS110/INTIIC10 */
          /* ||||| 1001:INTSR1 */
          /* ||||| 1010:INTST3 */
          /* ||||| 1011:INTSR3 */
          /* ||||| 1100:INTAD */
          /* ||||| +----- DWAIT0 */
          /* ||||| [Pending of DMA transfer] */
          /* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
          /* ||||| 1:Holds DMA start request pending if any. */
          /* ||||| +----- DSO */
          /* ||||| [Specification of transfer data size for DMA transfer] */
          /* ||||| 0:8bits */
          /* ||||| 1:16bits */
          /* ||||| +----- DRSO */
          /* ||||| [Selection of DMA transfer direction] */
          /* ||||| 0:SFR to internal RAM */
          /* ||||| 1:Internal RAM to SFR */
          /* ||||| +----- STGO */
          /* ||||| [DMA transfer start software trigger] */
          /* ||||| 0:No trigger operation */
          /* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
-----   DSTO = 1;                               /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0の初期値を設定します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void    fn_AdcDacInit(void)
{
-----
    ADCEN = 1;                /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

/*-----*/
/* Initialization of Voltage reference */
/*-----*/
    ADVRC = 0b00001000;      /* Analog reference voltage control register */
/* | | | | + | + + --- : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
/* | | | | |          [2]Operationcontrol of voltage reference */
/* | | | | |          [3]Output voltage selection of voltage reference */
/* | | | | |          [4]Operation control of input gate voltage boost circuit for A/D converter
*/
/* | | | | |          [5]Relationship with the conversion mode used */
/* | | | | | */
/* | | | | | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/* | | | | |          [2]Stops operation (Hi-Z) */
/* | | | | |          [3]2.5 V */
/* | | | | |          [4]Stops operation */
/* | | | | |          [5]Can be set in conversion mode 1 */
/* | | | | | */
/* | | | | | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/* | | | | |          [2]Stops operation (Hi-Z) */
/* | | | | |          [3]2.0 V */
/* | | | | |          [4]Enables operation */
/* | | | | |          [5]Can be set in conversion mode 2 or 3 */
/* | | | | | */
/* | | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/* | | | | |          [2]Stops operation (pull-down output) */
/* | | | | |          [3]2.5 V */
/* | | | | |          [4]Stops operation */
/* | | | | |          [5] - */
/* | | | | | */
/* | | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/* | | | | |          [2]Enables operation */
/* | | | | |          [3]2.5 V */
/* | | | | |          [4]Enables operation */
/* | | | | |          [5]Can be set in conversion mode 2 or 3 */
/* | | | | | */
/* | | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/* | | | | |          [2]Stops operation (pull-down output) */
/* | | | | |          [3]2.0 V */
/* | | | | |          [4]Enables operation */
/* | | | | |          [5] - */
/* | | | | | */
/* | | | | | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/* | | | | |          [2]Enables operation */
/* | | | | |          [3]2.0 V */
/* | | | | |          [4]Enables operation */
/* | | | | |          [5]Can be set in conversion mode 2 or 3 */
/* | | | | | */
/* | | | | | : Other than the above : Setting prohibited */
/* | | | | | */
/* | + + + + - - - - : Be sure to set 0000 */
/* | */
/* + - - - - - - - - : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;            /* Enables operation */
    ADVRC.1 = 1;            /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/* ||| : 0 0 0 0 0 : A 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 A D */
/* ||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/* ||| : 0 0 0 1 1 : A A 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 D D D D D D */
/* ||| : 0 0 1 1 1 : A A A A A D 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 */
/* ||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
/* ||| */
/* +++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

/* set D/A channel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[90]; /* D/A conversion value setting register 1 */

}

```

4.4.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0,1を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

TM04を停止します。

1周期完了時点での出力電圧をA/D変換します。

出力電圧のA/D変換値と理想状態での基準電圧のA/D変換値を比較して、オフセット調整します。

オフセット調整完了後、基準電圧にずれを発生させます（動作確認のため）。また、DMA転送間隔を変化させて、入力信号を100Hzと1kHz交互に切り替わるように設定します。

DMA転送を再設定し、TM04の動作を再開して へ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    ushCycleCount = 0;                /* Cycle counter */
    ucCycleStatus = 0;                /* Cycle status */

    DACE1 = 1;                        /* D/A converter CH1 enable */
    DACE0 = 1;

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                        /* start TAU0 CH4 */
    TMIF04 = 0;                        /* reset INTTMO4 interrupt request */

    DMAMK0 = 0;                        /* enable DMA CH0 interrupt */

    EI();                              /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }
}

/*-----*/
/* Module:      fn_intdma0          */
/* Description:  DMA ch0 interrupt process */
/* parameter:  --                  */
/* return      :  --                  */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TTOL.4 = 1;                        /* stop TAU0 CH4 */

```

```

/* start A/D conversion */
ADCS = 1;      /* A/D converter enable */
ADIF = 0;      /* clear A/D interrupt request flag */
ADMK = 0;      /* enable A/D interrupt */
HALT();        /* set HALT mode and wait conversion finished */
ADMK = 1;      /* disable A/D interrupt */
ADIF = 0;      /* clear A/D interrupt request flag */

shVoltage = (short)(ADCR & 0x0fff); /* get A/D result */
shVoltage -= (2048 - 1);
if(shVoltage < 0){
    DACSW1 += 1;      /* D/A conversion value setting register 1 */
}
else if(shVoltage > 2){
    DACSW1 -= 1;      /* D/A conversion value setting register 1 */
}

ushCycleCount++;      /* Cycle counter */
if(ushCycleCount > 50){ /* Cycle counter */
    switch(ucCycleStatus){ /* Cycle status */
        case 0:
            TDR04 = 2000; /* set interval time to 100us(=0.05us *
2000) */
            ucCycleStatus = 1; /* Cycle status (100Hz) */
            ushCycleCount = 20; /* reset Cycle counter */
            break;
        case 1:
            TDR04 = 200; /* set interval time to 10us(=0.05us * 200)
*/
            ucCycleStatus = 0; /* Cycle status (1kHz) */
            ushCycleCount = 0; /* reset Cycle counter */
            break;
    }
    DACSW1 = ushDacData[90];

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100; /* DMA Byte Count Register 0 */

DST0 = 1; /* DMA transfer is started when DMA operation is enabled (DENO = 1). */

TSOL.4 = 1; /* start TAU0 CH4 */
TMIF04 = 0; /* reset INTTMO4 interrupt request */
}

```

4.5 非反転増幅回路 交流

4.5.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。


```

void    fn_DmaInit(void)
{
-----   DENO = 1;                               /* Enables operation of DMA channel 0 */
-----   DSAO = (unsigned char)&DACSW0;           /* DMA SFR Address Register 0 */
-----   DRAO = (unsigned short)&ushDacData[0];   /* DMA RAM Address Register 0 */
-----   DBCO = 25;                               /* DMA Byte Count Register 0 */
-----   DMC0 = 0b01100100;                       /* DMA Mode Control Register 0 */
          /* ||||| */
          /* ||||| +----- IFC3-IFC0 */
          /* ||||| [Selection of DMA start source] */
          /* ||||| 0000:Disables DMA transfer by interrupt */
          /* ||||| (Only software trigger is enabled.) */
          /* ||||| 0010:INTTM00 */
          /* ||||| 0011:INTTM01 */
          /* ||||| 0100:INTTM04 */
          /* ||||| 0101:INTTM05 */
          /* ||||| 0110:INTST0/INTCS100 */
          /* ||||| 0111:INTSR0/INTCS101 */
          /* ||||| 1000:INTST1/INTCS110/INTIIC10 */
          /* ||||| 1001:INTSR1 */
          /* ||||| 1010:INTST3 */
          /* ||||| 1011:INTSR3 */
          /* ||||| 1100:INTAD */
          /* ||||| +----- DWAIT0 */
          /* ||||| [Pending of DMA transfer] */
          /* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
          /* ||||| 1:Holds DMA start request pending if any. */
          /* ||||| +----- DSO */
          /* ||||| [Specification of transfer data size for DMA transfer] */
          /* ||||| 0:8bits */
          /* ||||| 1:16bits */
          /* ||||| +----- DRS0 */
          /* ||||| [Selection of DMA transfer direction] */
          /* ||||| 0:SFR to internal RAM */
          /* ||||| 1:Internal RAM to SFR */
          /* ||||| +----- STGO */
          /* ||||| [DMA transfer start software trigger] */
          /* ||||| 0:No trigger operation */
          /* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
-----   DSTO = 1;                               /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : リアルタイム出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

```

void fn_AdcDacInit(void)
{
    ADCEN = 1;           /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /* |||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /* ||||| |      [2]Operationcontrol of voltage reference */
    /* ||||| |      [3]Output voltage selection of voltage reference */
    /* ||||| |      [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /* ||||| |      [5]Relationship with the conversion mode used */
    /* ||||| | */
    /* ||||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /* ||||| |      [2]Stops operation (Hi-Z) */
    /* ||||| |      [3]2.5 V */
    /* ||||| |      [4]Stops operation */
    /* ||||| |      [5]Can be set in conversion mode 1 */
    /* ||||| | */
    /* ||||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /* ||||| |      [2]Stops operation (Hi-Z) */
    /* ||||| |      [3]2.0 V */
    /* ||||| |      [4]Enables operation */
    /* ||||| |      [5]Can be set in conversion mode 2 or 3 */
    /* ||||| | */
    /* ||||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /* ||||| |      [2]Stops operation (pull-down output) */
    /* ||||| |      [3]2.5 V */
    /* ||||| |      [4]Stops operation */
    /* ||||| |      [5] - */
    /* ||||| | */
    /* ||||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /* ||||| |      [2]Enables operation */
    /* ||||| |      [3]2.5 V */
    /* ||||| |      [4]Enables operation */
    /* ||||| |      [5]Can be set in conversion mode 2 or 3 */
    /* ||||| | */
    /* ||||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /* ||||| |      [2]Stops operation (pull-down output) */
    /* ||||| |      [3]2.0 V */
    /* ||||| |      [4]Enables operation */
    /* ||||| |      [5] - */
    /* ||||| | */
    /* ||||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /* ||||| |      [2]Enables operation */
    /* ||||| |      [3]2.0 V */
    /* ||||| |      [4]Enables operation */
    /* ||||| |      [5]Can be set in conversion mode 2 or 3 */
    /* ||||| | */
    /* ||||| | : Other than the above : Setting prohibited */
    /* ||||| | */
    /* ||||| | : Be sure to set 0000 */
    /* ||||| | */
    /* +----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;           /* Enables operation */
    ADVRC.1 = 1;           /* Output 2.0 V */
}
    
```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/* ||| : 0 0 0 0 0 : A 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 A D */
/* ||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/* ||| : 0 0 0 1 1 : A 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 D D D D D */
/* ||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/* ||| : 0 0 1 1 1 : 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 */
/* ||| : 0 1 0 0 1 : A A A 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 */
/* ||| : 0 1 1 1 1 : A 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 */
/* ||| */
/* +----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMPO) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */

DAM.0 = 1; /* [DAMD0] D/A converter operation is Real-time output mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is VREFOUT */

}

```

4.5.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

TM04を停止します。

DMA転送回数を調整し、D/Aコンバータ出力波形が (sin0 ~ sin(1/2 ·)) (定電圧) (sin(1/2 ·)) ~ sin(3/2 ·)) (定電圧) (sin(3/2 ·)) ~ sin(2 ·)) (低速変動電圧) を繰り返すようDMAコントローラを再設定します。

TM04の動作を再開してへ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    ucCycleStatus = 0;              /* Cycle status */

    DACE0 = 1;                      /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                      /* start TAU0 CH4 */
    TMIF04 = 0;                      /* reset INTTMO4 interrupt request */

    DMAMK0 = 0;                      /* enable DMA CH0 interrupt */

    EI();                             /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }
}

/*-----*/
/* Module:      fn_intdma0          */
/* Description:  DMA ch0 interrupt process */
/* parameter:  --                    */
/* return      :  --                    */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TTOL.4 = 1;                      /* stop TAU0 CH4 */
}

```

```

switch(ucCycleStatus){          /* Cycle status */
  case 0:
    TDR04 = 20000;              /* set interval time to 1000us(=0.05us * 20000) */
    ucCycleStatus = 1;          /* Cycle status (Flat) */
    DBC0 = 1;                   /* DMA Byte Count Register 0 */
    break;

  case 1:
    TDR04 = 200;                /* set interval time to 10us(=0.05us * 200) */
    ucCycleStatus = 2;          /* Cycle status (2/4,3/4 sign) */
    DBC0 = 50 - 1;             /* DMA Byte Count Register 0 */
    break;

  case 2:
    TDR04 = 20000;              /* set interval time to 1000us(=0.05us * 20000) */
    ucCycleStatus = 3;          /* Cycle status (Flat) */
    DBC0 = 1;                   /* DMA Byte Count Register 0 */
    break;

  case 3:
    TDR04 = 200;                /* set interval time to 10us(=0.05us * 200) */
    ucCycleStatus = 4;          /* Cycle status (1kHz) */
    DBC0 = 25 - 1;             /* DMA Byte Count Register 0 */
    break;

  case 4:
    DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
    DACSW0 = ushDacData[99]; /* D/A conversion value setting register 0 */
    TMR04 = 0b0000000000000000; /* Timer Mode Register 04 */
    TDR04 = 20000;              /* set interval time to 1000us(=0.05us * 20000) */
    ucCycleStatus = 5;          /* Cycle status (10Hz) */
    DBC0 = 50;                  /* DMA Byte Count Register 0 */
    break;

  case 5:
    DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
    DACSW0 = ushDacData[99]; /* D/A conversion value setting register 0 */
    TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
    TDR04 = 200;                /* set interval time to 10us(=0.05us * 200) */
    ucCycleStatus = 0;          /* Cycle status (1kHz) */
    DBC0 = 25;                  /* DMA Byte Count Register 0 */
    break;
}

*/
DSTO = 1;                      /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).

TSOL.4 = 1;                    /* start TAU0 CH4 */
TMIF04 = 0;                    /* reset INTTMO4 interrupt request */
}

```

4.6 反転増幅回路 交流

4.6.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void fn_DmaInit(void)
{
----- DENO = 1; /* Enables operation of DMA channel 0 */
----- DSAO = (unsigned char)&DACSW0; /* DMA SFR Address Register 0 */
----- DRAO = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
----- DBCO = 25; /* DMA Byte Count Register 0 */
----- DMCO = 0b01100100; /* DMA Mode Control Register 0 */
/* ||||| */
/* ||||| +----- IFC3-IFC0 */
/* ||||| [Selection of DMA start source] */
/* ||||| 0000:Disables DMA transfer by interrupt */
/* ||||| (Only software trigger is enabled.) */
/* ||||| 0010:INTTM00 */
/* ||||| 0011:INTTM01 */
/* ||||| 0100:INTTM04 */
/* ||||| 0101:INTTM05 */
/* ||||| 0110:INTST0/INTCSI00 */
/* ||||| 0111:INTSR0/INTCSI01 */
/* ||||| 1000:INTST1/INTCSI10/INTIIC10 */
/* ||||| 1001:INTSR1 */
/* ||||| 1010:INTST3 */
/* ||||| 1011:INTSR3 */
/* ||||| 1100:INTAD */
/* ||||| +----- DWAIT0 */
/* ||||| [Pending of DMA transfer] */
/* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
/* ||||| 1:Holds DMA start request pending if any. */
/* ||||| +----- DSO */
/* ||||| [Specification of transfer data size for DMA transfer] */
/* ||||| 0:8bits */
/* ||||| 1:16bits */
/* ||||| +----- DRSO */
/* ||||| [Selection of DMA transfer direction] */
/* ||||| 0:SFR to internal RAM */
/* ||||| 1:Internal RAM to SFR */
/* ||||| +----- STGO */
/* ||||| [DMA transfer start software trigger] */
/* ||||| 0:No trigger operation */
/* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
----- DSTO = 1; /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```


(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : リアルタイム出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void fn_AdcDacInit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /* ||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /* |||| | [2]Operationcontrol of voltage reference */
    /* |||| | [3]Output voltage selection of voltage reference */
    /* |||| | [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /* |||| | [5]Relationship with the conversion mode used */
    /* |||| | */
    /* |||| : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /* |||| : [2]Stops operation (Hi-Z) */
    /* |||| : [3]2.5 V */
    /* |||| : [4]Stops operation */
    /* |||| : [5]Can be set in conversion mode 1 */
    /* |||| */
    /* |||| : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /* |||| : [2]Stops operation (Hi-Z) */
    /* |||| : [3]2.0 V */
    /* |||| : [4]Enables operation */
    /* |||| : [5]Can be set in conversion mode 2 or 3 */
    /* |||| */
    /* |||| : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /* |||| : [2]Stops operation (pull-down output) */
    /* |||| : [3]2.5 V */
    /* |||| : [4]Stops operation */
    /* |||| : [5] - */
    /* |||| */
    /* |||| : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /* |||| : [2]Enables operation */
    /* |||| : [3]2.5 V */
    /* |||| : [4]Enables operation */
    /* |||| : [5]Can be set in conversion mode 2 or 3 */
    /* |||| */
    /* |||| : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /* |||| : [2]Stops operation (pull-down output) */
    /* |||| : [3]2.0 V */
    /* |||| : [4]Enables operation */
    /* |||| : [5] - */
    /* |||| */
    /* |||| : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /* |||| : [2]Enables operation */
    /* |||| : [3]2.0 V */
    /* |||| : [4]Enables operation */
    /* |||| : [5]Can be set in conversion mode 2 or 3 */
    /* |||| */
    /* |||| : Other than the above : Setting prohibited */
    /* |||| */
    /* +++----- : Be sure to set 0000 */
    /* | */
    /* +----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;          /* Enables operation */
    ADVRC.1 = 1;          /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/*||| : 0 0 0 0 0 : A 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 A D */
/*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/*||| : 0 0 0 1 1 : A 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 D D D D D */
/*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*+++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPAmP A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */

DAM.0 = 1; /* [DAMD0] D/A converter operation is Real-time output mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is VREFOUT */

/* set D/A chanel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting reigster 1 */
}

```

4.6.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0,1を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

TM04を停止します。

DMA転送回数を調整し、D/Aコンバータ出力波形が ($\sin 0 \sim \sin(1/2 \cdot)$) (定電圧) ($\sin(1/2 \cdot) \sim \sin$) (低速変動電圧) を繰り返すようDMAコントローラを再設定します。

TM04の動作を再開してへ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    ucCycleStatus = 0;              /* Cycle status */

    DACE0 = 1;                      /* D/A converter CH0 enable */
    DACE1 = 1;                      /* D/A converter CH1 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                      /* start TAU0 CH4 */
    TMIF04 = 0;                      /* reset INTTMO4 interrupt request */

    DMAMKO = 0;                      /* enable DMA CH0 interrupt */

    EI();                             /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }

}

/*-----*/
/* Module:      fn_intdma0          */
/* Description: DMA ch0 interrupt process */
/* parameter:  --                    */
/* return      : --                    */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TTOL.4 = 1;                      /* stop TAU0 CH4 */
}

```

```

        case 0:
            TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000) */
            ucCycleStatus = 1;     /* Cycle status (Flat) */
            DBC0 = 1;             /* DMA Byte Count Register 0 */
            break;

        case 1:
            TDR04 = 200;          /* set interval time to 10us(=0.05us * 200) */
            ucCycleStatus = 2;     /* Cycle status (2/4 sign) */
            DBC0 = 25;           /* DMA Byte Count Register 0 */
            break;

        case 2:
            DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
            DACSW0 = ushDacData[99]; /* D/A conversion value setting register 0 */
            TMR04 = 0b0000000000000000; /* Timer Mode Register 04 */
            TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000) */
            ucCycleStatus = 3;     /* Cycle status (slow) */
            DBC0 = 50;           /* DMA Byte Count Register 0 */
            break;

        case 3:
            DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
            DACSW0 = ushDacData[99]; /* D/A conversion value setting register 0 */
            TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
            TDR04 = 200;          /* set interval time to 10us(=0.05us * 200) */
            ucCycleStatus = 0;     /* Cycle status (1kHz) */
            DBC0 = 25;           /* DMA Byte Count Register 0 */
            break;
    }

    /*
    DSTO = 1; /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).

    TSOL.4 = 1; /* start TAU0 CH4 */
    TMIF04 = 0; /* reset INTTM04 interrupt request */
    */
}

```

4.7 インストゥルメンテーションアンブ

4.7.1 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) : 1秒に1度 (秒カウントアップと同時)

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. */
-----
RTCEN = 1;          /* supplies operational real-time counter (RTC) input clock. */
----- RTCCO = 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) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0-2を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0の初期値を設定します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。


```

void    fn_AdcDacInit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /* | | | | + | ++--- : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /* | | | | |          [2]Operationcontrol of voltage reference */
    /* | | | | |          [3]Output voltage selection of voltage reference */
    /* | | | | |          [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /* | | | | |          [5]Relationship with the conversion mode used */
    /* | | | | | */
    /* | | | | | : 0 0 0 : [1]AVREFFP (external voltage reference input) */
    /* | | | | |          [2]Stops operation (Hi-Z) */
    /* | | | | |          [3]2.5 V */
    /* | | | | |          [4]Stops operation */
    /* | | | | |          [5]Can be set in conversion mode 1 */
    /* | | | | | */
    /* | | | | | : 0 1 0 : [1]AVREFFP (external voltage reference input) */
    /* | | | | |          [2]Stops operation (Hi-Z) */
    /* | | | | |          [3]2.0 V */
    /* | | | | |          [4]Enables operation */
    /* | | | | |          [5]Can be set in conversion mode 2 or 3 */
    /* | | | | | */
    /* | | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /* | | | | |          [2]Stops operation (pull-down output) */
    /* | | | | |          [3]2.5 V */
    /* | | | | |          [4]Stops operation */
    /* | | | | |          [5] - */
    /* | | | | | */
    /* | | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /* | | | | |          [2]Enables operation */
    /* | | | | |          [3]2.5 V */
    /* | | | | |          [4]Enables operation */
    /* | | | | |          [5]Can be set in conversion mode 2 or 3 */
    /* | | | | | */
    /* | | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /* | | | | |          [2]Stops operation (pull-down output) */
    /* | | | | |          [3]2.0 V */
    /* | | | | |          [4]Enables operation */
    /* | | | | |          [5] - */
    /* | | | | | */
    /* | | | | | : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /* | | | | |          [2]Enables operation */
    /* | | | | |          [3]2.0 V */
    /* | | | | |          [4]Enables operation */
    /* | | | | |          [5]Can be set in conversion mode 2 or 3 */
    /* | | | | | */
    /* | | | | | : Other than the above : Setting prohibited */
    /* | | | | | */
    /* | +++----- : Be sure to set 0000 */
    /* | */
    /* +----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;        /* Enables operation */
    ADVRC.1 = 1;        /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/* ||| : 0 0 0 0 0 : A 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 A D */
/* ||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/* ||| : 0 0 0 1 1 : A 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 D D D D D */
/* ||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/* ||| : 0 0 1 1 1 : 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 */
/* ||| : 0 1 0 0 1 : A A A 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 */
/* ||| : 0 1 1 1 1 : A 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 */
/* ||| */
/* +++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPAmP A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 1; /* operational amplifier (AMP1) enable */
OAC.2 = 1; /* operational amplifier (AMP2) enable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
DACSW0 = CDACO_52V; /* D/A conversion value setting register 0 */

/* set D/A channel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = CDACO_50V; /* D/A conversion value setting register 1 */

}

```

4.7.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル0,1を動作許可します。

リアルタイム・カウンタの割り込みを許可します。また、同時にリアルタイム・カウンタの動作を開始します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

D/Aコンバータの出力電圧を再設定して、へ分岐します。

```

void main(void)
{
    DACE1 = 1;          /* D/A converter CH1 enable */
    DACE0 = 1;          /* D/A converter CH0 enable */

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

    RTCIF = 0;          /* reset RCT interrupt request */
    RTCMK = 0;          /* enable RTC interrupt */
    RTCE = 1;           /* starts counter operation */

    EI();               /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }
}

/*-----*/
/* Module:      fn_intrtc          */
/* Description:  RTC interrupt process */
/* parameter:  --                  */
/* return      :  --                  */
/*-----*/
__interrupt void fn_intrtc(void)
{
    switch(DACSW0)
    {
        case CDACO_52V:             /* DAC0 output voltage : 0.52V */
            DACSW0 = CDACO_54V;    /* DAC0 output voltage : 0.52V -> 0.54V */
            break;
        case CDACO_54V:             /* DAC0 output voltage : 0.54V */
            default:
                DACSW0 = CDACO_52V; /* DAC0 output voltage : 0.54V -> 0.52V */
                break;
    }
}

```

4.8 比較回路

4.8.1 使用する周辺の初期設定

(1)DMAの初期設定

DMAの初期設定処理では、次の動作を行います。

DMAチャンネル0を動作許可にします。

DMA転送のSFRアドレス設定を行います。

DMA転送のRAMアドレス設定を行います。

DMA転送回数の設定を行います。

DMA転送モードを下記の通り設定します。

- ・ DMA転送開始ソフトウェア・トリガ : ソフトウェア・トリガ動作しない
- ・ DMA転送方向 : 内蔵RAM SFR
- ・ 転送データ・サイズ : 16ビット
- ・ DMA転送の保留 : 保留しない
- ・ DMA起動要因 : INTTM04

DMAトリガ待ち状態にします。

```

void    fn_DmaInit(void)
{
-----  DENO = 1;                               /* Enables operation of DMA channel 0 */
-----  DSAO = (unsigned char)&DACSW0;           /* DMA SFR Address Register 0 */
-----  DRAO = (unsigned short)&ushDacData[0];   /* DMA RAM Address Register 0 */
-----  DBCO = 100;                             /* DMA Byte Count Register 0 */
-----  DMCO = 0b01100100;                     /* DMA Mode Control Register 0 */
        /* ||||| */
        /* ||||| +----- IFC3-IFC0 */
        /* ||||| [Selection of DMA start source] */
        /* ||||| 0000:Disables DMA transfer by interrupt */
        /* ||||| (Only software trigger is enabled.) */
        /* ||||| 0010:INTTM00 */
        /* ||||| 0011:INTTM01 */
        /* ||||| 0100:INTTM04 */
        /* ||||| 0101:INTTM05 */
        /* ||||| 0110:INTST0/INTCSI00 */
        /* ||||| 0111:INTSR0/INTCSI01 */
        /* ||||| 1000:INTST1/INTCSI10/INTIIC10 */
        /* ||||| 1001:INTSR1 */
        /* ||||| 1010:INTST3 */
        /* ||||| 1011:INTSR3 */
        /* ||||| 1100:INTAD */
        /* ||||| +----- DWAIT0 */
        /* ||||| [Pending of DMA transfer] */
        /* ||||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
        /* ||||| 1:Holds DMA start request pending if any. */
        /* ||||| +----- DSO */
        /* ||||| [Specification of transfer data size for DMA transfer] */
        /* ||||| 0:8bits */
        /* ||||| 1:16bits */
        /* ||||| +----- DRSO */
        /* ||||| [Selection of DMA transfer direction] */
        /* ||||| 0:SFR to internal RAM */
        /* ||||| 1:Internal RAM to SFR */
        /* ||||| +----- STGO */
        /* ||||| [DMA transfer start software trigger] */
        /* ||||| 0:No trigger operation */
        /* ||||| 1:DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
-----  DSTO = 1;                               /* DMA transfer is started when DMA operation is enabled (DENO = 1).
*/
}

```

(2) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ボルテージ・リファレンスを下記の通り設定します。

- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}
- ・ ボルテージ・リファレンスの動作制御 : 動作許可
- ・ ボルテージ・リファレンスの出力電圧 : 2.0V

ポート2, ポート15をアナログ入力に設定します。

設定完了までの間ウエイトします。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル0を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット
- ・ D/Aコンバータの + 側の基準電圧源 : V_{REFOUT}

D/Aコンバータのチャンネル0の初期値を設定します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void fn_AdcDacInit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier, and
voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
    /* |||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A converters
*/
    /* ||||| | [2]Operation control of voltage reference */
    /* ||||| | [3]Output voltage selection of voltage reference */
    /* ||||| | [4]Operation control of input gate voltage boost circuit for A/D converter
*/
    /* ||||| | [5]Relationship with the conversion mode used */
    /* ||||| | */
    /* ||||| : 0 0 0 : [1]AVREFP (external voltage reference input) */
    /* ||||| [2]Stops operation (Hi-Z) */
    /* ||||| [3]2.5 V */
    /* ||||| [4]Stops operation */
    /* ||||| [5]Can be set in conversion mode 1 */
    /* ||||| */
    /* ||||| : 0 1 0 : [1]AVREFP (external voltage reference input) */
    /* ||||| [2]Stops operation (Hi-Z) */
    /* ||||| [3]2.0 V */
    /* ||||| [4]Enables operation */
    /* ||||| [5]Can be set in conversion mode 2 or 3 */
    /* ||||| */
    /* ||||| : 1 0 0 : [1]VREFOUT (voltage reference output) */
    /* ||||| [2]Stops operation (pull-down output) */
    /* ||||| [3]2.5 V */
    /* ||||| [4]Stops operation */
    /* ||||| [5] - */
    /* ||||| */
    /* ||||| : 1 0 1 : [1]VREFOUT (voltage reference output) */
    /* ||||| [2]Enables operation */
    /* ||||| [3]2.5 V */
    /* ||||| [4]Enables operation */
    /* ||||| [5]Can be set in conversion mode 2 or 3 */
    /* ||||| */
    /* ||||| : 1 1 0 : [1]VREFOUT (voltage reference output) */
    /* ||||| [2]Stops operation (pull-down output) */
    /* ||||| [3]2.0 V */
    /* ||||| [4]Enables operation */
    /* ||||| [5] - */
    /* ||||| */
    /* ||||| : 1 1 1 : [1]VREFOUT (voltage reference output) */
    /* ||||| [2]Enables operation */
    /* ||||| [3]2.0 V */
    /* ||||| [4]Enables operation */
    /* ||||| [5]Can be set in conversion mode 2 or 3 */
    /* ||||| */
    /* ||||| : Other than the above : Setting prohibited */
    /* ||||| */
    /* |||||+--- : Be sure to set 0000 */
    /* ||||| */
    /* +----- : Reference voltage supply (negative side) of A/D converter selection */
    /* : 0 : AVSS */
    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;        /* Enables operation */
    ADVRC.1 = 1;        /* Output 2.0 V */

```

```

----- ADPC = 0b00000000; /* 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/AMPO-/P20 */
/* ||| : 0 0 0 0 0 : A A 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 A D */
/* ||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/* ||| : 0 0 0 1 1 : A 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 D D D D D */
/* ||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/* ||| : 0 0 1 1 1 : 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 */
/* ||| : 0 1 0 0 1 : A A A 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 */
/* ||| : 0 1 1 1 1 : A 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 */
/* ||| */
/* +++----- : Be sure to set 000 */

/* wait for setting time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP00) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP10) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP20) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (AN00) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal mode */
DAM.2 = 1; /* [DARES0] D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

/* set D/A channel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal mode */
DAM.3 = 1; /* [DARES1] D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting register 1 */

}

```


4.8.2 メイン処理

メイン処理では、次の動作を行います。

P7.3を出力ポート、P12.0を入力ポートに設定します。

P12.0を両エッジ検出に設定します。

D/Aコンバータのチャンネル0,1を動作許可します。

正弦波出力テーブルをROMからRAMにコピーします。

TM04の割り込みを許可します。

DMA転送チャンネル0の割り込みを許可します。

INTP0の割り込みを許可します。

割り込みを許可します。

割り込みが発生するまで、待機状態に移行します。

DMA転送を再設定し、へ分岐します。

P7.3の出力レベルをP12.0と同じにして、へ分岐します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    /* Port setting */
    P12.0 = 0;
    PM12.0 = 1;
    P7.3 = 0;
    PM7.3 = 0;

    /* INTP0 pin valid edge selection */
    EGPO.0 = 1;                      /* Both rising and falling edges */
    EGN0.0 = 1;

    DACE1 = 1;                       /* D/A converter CH1 enable */
    DACE0 = 1;

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TSOL.4 = 1;                      /* start TAU0 CH4 */
    TMIF04 = 0;                      /* reset INTTMO4 interrupt request */

    DMAMK0 = 0;                      /* enable DMA CH0 interrupt */

    PIF0 = 0;                        /* reset INTP0 interrupt request */
    PMK0 = 0;                        /* enable INTP0 interrupt */

    EI();                            /* enable all interrupts */

    /*=====*/
    /* if system have nothing to do, go to standby for power-saving */
    /*=====*/
    while (1){
        HALT();
        NOP();
    }
}

```

```
/*-----*/
/* Module:      fn_intdma0 */
/* Description: DMA ch0 interrupt process */
/* parameter:  -- */
/* return   :  -- */
/*-----*/
__interrupt void fn_intdma0(void)
{
    DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
    DBC0 = 100;                               /* DMA Byte Count Register 0 */

    DST0 = 1;    /* DMA transfer is started when DMA operation is enabled (DENO = 1). */
}

/*-----*/
/* Module:      fn_intp0 */
/* Description:  INTPO interrupt process */
/* parameter:  -- */
/* return   :  -- */
/*-----*/
__interrupt void fn_intp0(void)
{
    P7.3 = P12.0;
}
```

4.9 CR発振回路1

4.9.1 使用する周辺の初期設定

(1) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ポート2, ポート15をアナログ入力に設定します。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void fn_AdcDacInit(void)
{
----- ADCEN = 1;          /* supplies input clock to A/D converter and operational */

----- ADPC = 0b00000000;    /* 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/AMPO-/P20 */
/*||| : 0 0 0 0 0 : A 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 A D */
/*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
/*||| : 0 0 0 1 1 : A 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 D */
/*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
/*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*+++----- : Be sure to set 000 */

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1;          /* input port mode (AMP00) */
PM2.0 = 1;          /* input port mode (AMPO-) */
PM2.2 = 1;          /* input port mode (AMP0+) */

PM2.4 = 1;          /* input port mode (AMP10) */
PM2.3 = 1;          /* input port mode (AMP1-) */
PM2.5 = 1;          /* input port mode (AMP1+) */

PM2.7 = 1;          /* input port mode (AMP20) */
PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;         /* input port mode (AMP2+) */

PM11.0 = 1;         /* input port mode (AN00) */

ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;          /* operational amplifier (AMPO) enable */
OAC.1 = 0;          /* operational amplifier (AMP1) disable */
OAC.2 = 0;          /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
----- DACEN = 1;          /* supplies input clock to D/A converter */

/* set D/A channel1 */
----- DAM.1 = 0;          /* [DAMD1] D/A converter operation is Normal mode */
----- DAM.3 = 1;          /* [DARES1]D/A conversion resolution is 12-bit */
----- DACSW1 = 2048;     /* D/A conversion value setting register 1 */

}

```

(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}) : $f_{CLK}/2^8$
- ・ LCDクロック (LCDCL) の選択 : $f_{LCD}/2^7$

V_{LCD} 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 (wake-up 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 = 0b00110011;          /* 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 */
}

```

4.9.2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル1を動作許可します。

TM04, TM06を動作開始します。

1ms経過するまでウエイトします。

1ms間のカウント数を取得します。

TM04, TM06を動作停止します。

カウント数をLCDに表示します。

```

void main(void)
{
    unsigned short    ushCount;        /* Pulse counter */

    DACE1 = 1;                    /* D/A converter CH1 enable */
    ushCount = 0;

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

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

    while (1){
        TSOL.6 = 1;                /* start TAU0 CH6 (pulse counter) */
        TSOL.4 = 1;                /* start TAU0 CH4 (interval timer) */
        TMIF04 = 0;                /* clear interval timer request */

        while(TMIF04 == 0){        /* wait interval */
            NOP();
        }

        ushCount = TCR06;          /* get pulse count */

        TTOL.6 = 1;                /* stop TAU0 CH6 (pulse counter)*/
        TTOL.4 = 1;                /* stop TAU0 CH4 (interval timer) */

        ushCount = 0xffff - ushCount; /* get pulse count */
        fn_Display(ushCount);      /* display pulse count */
    }
}

```


4. 10 CR発振回路2

4. 10. 1 使用する周辺の初期設定

(1) オペアンプ, D/Aコンバータの初期設定

オペアンプ, D/Aコンバータの初期設定処理では, 次の動作を行います。

A/Dコンバータへの入力クロックを供給します。

ポート2, ポート15をアナログ入力に設定します。

アナログ入力として使用するポートを入力モードに設定します。

オペアンプ0を動作許可します。

D/Aコンバータへの入力クロックを供給します。

D/Aコンバータのチャンネル1を下記の通り設定します。

- ・ D/Aコンバータの動作モード : 通常出力モード
- ・ D/Aコンバータの分解能 : 12ビット

D/Aコンバータのチャンネル1の初期値を設定します。

```

void fn_AdcDacInit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter and operational */

    ADPC = 0b00000000; /* 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/AMPO-/P20 */
    /*||| : 0 0 0 0 0 : A 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 A D */
    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
    /*||| : 0 0 0 1 1 : A 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 D D D D D */
    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
    /*||| : 0 0 1 1 1 : 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 */
    /*||| : 0 1 0 0 1 : A A A 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 */
    /*||| : 0 1 1 1 1 : A 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 */
    /*||| */
    /*+++----- : Be sure to set 000 */

    /*-----*/
    /* Initialization of operational amplifier */
    /*-----*/
    PM2.1 = 1;          /* input port mode (AMP00) */
    PM2.0 = 1;          /* input port mode (AMPO-) */
    PM2.2 = 1;          /* input port mode (AMP0+) */

    PM2.4 = 1;          /* input port mode (AMP10) */
    PM2.3 = 1;          /* input port mode (AMP1-) */
    PM2.5 = 1;          /* input port mode (AMP1+) */

    PM2.7 = 1;          /* input port mode (AMP20) */
    PM2.6 = 1;          /* input port mode (AMP2-) */
    PM15.0 = 1;         /* input port mode (AMP2+) */

    PM11.0 = 1;         /* input port mode (AN00) */

    ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
    OAC.0 = 1;          /* operational amplifier (AMPO) enable */
    OAC.1 = 0;          /* operational amplifier (AMP1) disable */
    OAC.2 = 0;          /* operational amplifier (AMP2) disable */

    /*-----*/
    /* Initialization of D/A converter */
    /*-----*/
    DACEN = 1;          /* supplies input clock to D/A converter */

    /* set D/A channel1 */
    DAM.1 = 0;          /* [DAMD1] D/A converter operation is Normal mode */
    DAM.3 = 1;          /* [DARES1]D/A conversion resolution is 12-bit */
    DACSW1 = 1365;     /* D/A conversion value setting register 1 */
    /* AN01 = 1V (1/3 * 2^12) */
}

```

(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}) : $f_{CLK}/2^8$
- ・ LCDクロック (LCDCL) の選択 : $f_{LCD}/2^7$

V_{LCD} 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 = 0b00110011;          /* 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) */
/*||| */
enable/disable */
/*|||+----- : Voltage boost circuit and capacitor split circuit operation
/*||| : 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 */
}

```

4. 10. 2 メイン処理

メイン処理では、次の動作を行います。

D/Aコンバータのチャンネル1を動作許可します。

P7.1,P7.2を入力ポートに設定します。

割り込みを許可します。

LCD表示を初期化します。

D/Aコンバータのチャンネル1の出力電圧を1Vに設定します。

6クロック分、P7.1をハイレベルの出力ポートに設定し、再度入力ポートに設定します。

P7.1をロウレベルの出力ポートに設定し、その間のカウント数を取得します。その後、再度P7.1を入力ポートに設定します。

D/Aコンバータのチャンネル1の出力電圧を2Vに設定します。

6クロック分、P7.1をハイレベルの出力ポートに設定し、再度入力ポートに設定します。

P7.1をロウレベルの出力ポートに設定し、その間のカウント数を取得します。その後、再度P7.1を入力ポートに設定します。

D/Aコンバータのチャンネル1の出力電圧を1Vに設定します。

6クロック分、P7.2をハイレベルの出力ポートに設定し、再度入力ポートに設定します。

P7.2をロウレベルの出力ポートに設定し、その間のカウント数を取得します。その後、再度P7.2を入力ポートに設定します。

D/Aコンバータのチャンネル1の出力電圧を2Vに設定します。

6クロック分、P7.2をハイレベルの出力ポートに設定し、再度入力ポートに設定します。

P7.2をロウレベルの出力ポートに設定し、その間のカウント数を取得します。その後、再度P7.2を入力ポートに設定します。

P7.1,P7.2それぞれのポートの、2.0V時のカウント数から1.0V時のカウント数を引いた値をLCDに表示します。

```

void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    unsigned short    *pMeasure;     /* pointer for measuring data */

    DACE1 = 1;                       /* D/A converter CH1 enable */

    ucMeasureStatus = 0;             /* measuring status */

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

    PM7.1 = 1;                       /* set R_REF HiZ(disconnect) */
    PM7.2 = 1;                       /* set R_REF HiZ(disconnect) */

    EI();                            /* enable all interrupts */

    fn_Display(0, 0);                /* initial display */

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

    while (1){
        switch(ucMeasureStatus){     /* measuring status */
            case 0:                   /* R_REF1 */
                DACSW1 = 1365;       /* D/A conversion value setting register 1 */
                /* AN01 = 1V (1/3 * 2^12) */

                P7.0 = 1;
                PM7.0 = 0;          /* charge CAPACITOR */
                NOP();
                NOP();
                NOP();
                NOP();
                NOP();
                NOP();
                /* wait charge complete */
                PM7.0 = 1;          /* end charge */

                P7.1 = 0;
                PM7.1 = 0;          /* set R_REF low(start discharge) */
                TSOL.6 = 1;         /* start TIMER06 */
                TMIF06 = 0;         /* clear measure end */
                while(TMIF06 == 0); /* wait measure end */
                ushR_REF1 = TDR06;  /* get measuring data */
                TTOL.6 = 1;         /* stop TAU0 CH6 */
                PM7.1 = 1;          /* set R_REF HiZ(stop discharge) */

                ucMeasureStatus = 1; /* set R_REF2 measure */
                break;
        }
    }
}

```

```

case 1:          /* R_REF2 */
DACSW1 = 2730; /* D/A conversion value setting register 1 */
                /* AN01 = 2V (2/3 * 2^12) */

P7.0 = 1;
PM7.0 = 0;     /* charge CAPACITOR */
NOP();
NOP();
NOP();
NOP();
NOP();
NOP();        /* wait charge complete */
PM7.0 = 1;     /* end charge */

P7.1 = 0;
PM7.1 = 0;     /* set R_REF low(start discharge) */
TSOL.6 = 1;   /* start TIMER06 */
TMIF06 = 0;   /* clear measure end */
while(TMIF06 == 0); /* wait measure end */
ushR_REF2 = TDR06; /* get measuring data */
TTOL.6 = 1;   /* stop TAU0 CH6 */
PM7.1 = 1;   /* set R_REF HiZ(stop discharge) */

ucMeasureStatus = 2; /* set R_TH1 measure */
break;

case 2:          /* R_TH1 */
DACSW1 = 1365; /* D/A conversion value setting register 1 */
                /* AN01 = 1V (1/3 * 2^12) */

P7.0 = 1;
PM7.0 = 0;     /* charge CAPACITOR */
NOP();
NOP();
NOP();
NOP();
NOP();
NOP();        /* wait charge complete */
PM7.0 = 1;     /* end charge */

P7.2 = 0;
PM7.2 = 0;     /* set R_TH low(start discharge) */
TSOL.6 = 1;   /* start TIMER06 */
TMIF06 = 0;   /* clear measure end */
while(TMIF06 == 0); /* wait measure end */
ushR_TH1 = TDR06; /* get measuring data */
TTOL.6 = 1;   /* stop TAU0 CH6 */
PM7.2 = 1;   /* set R_TH HiZ(stop discharge) */

ucMeasureStatus = 3; /* set R_TH2 measure */
break;

case 3:          /* R_TH2 */
DACSW1 = 2730; /* D/A conversion value setting register 1 */
                /* AN01 = 2V (2/3 * 2^12) */

P7.0 = 1;
PM7.0 = 0;     /* charge CAPACITOR */
NOP();
NOP();
NOP();
NOP();
NOP();
NOP();        /* wait charge complete */
PM7.0 = 1;     /* end charge */

P7.2 = 0;
PM7.2 = 0;     /* set R_TH low(start discharge) */
TSOL.6 = 1;   /* start TIMER06 */
TMIF06 = 0;   /* clear measure end */
while(TMIF06 == 0); /* wait measure end */
ushR_TH2 = TDR06; /* get measuring data */
TTOL.6 = 1;   /* stop TAU0 CH6 */
PM7.2 = 1;   /* set R_TH HiZ(stop discharge) */

/* display R_TH,R_REF data to LCD */
fn_Display((ushR_REF1 - ushR_REF2), (ushR_TH1 - ushR_TH2));

ucMeasureStatus = 0; /* set R_REF1 measure */
}
}
}

```


第5章 デバイスでの動作確認例

この章では、それぞれの回路における信号測定結果を示します。

5.1 非反転増幅回路

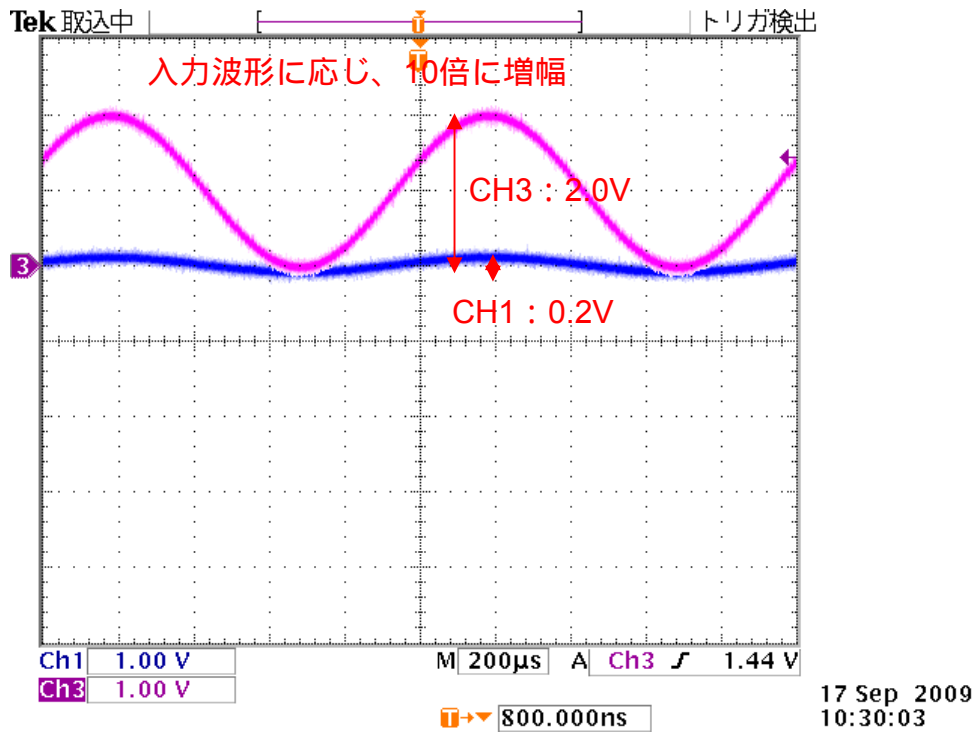


図5-1 非反転増幅回路の入出力信号波形

5.2 反転増幅回路

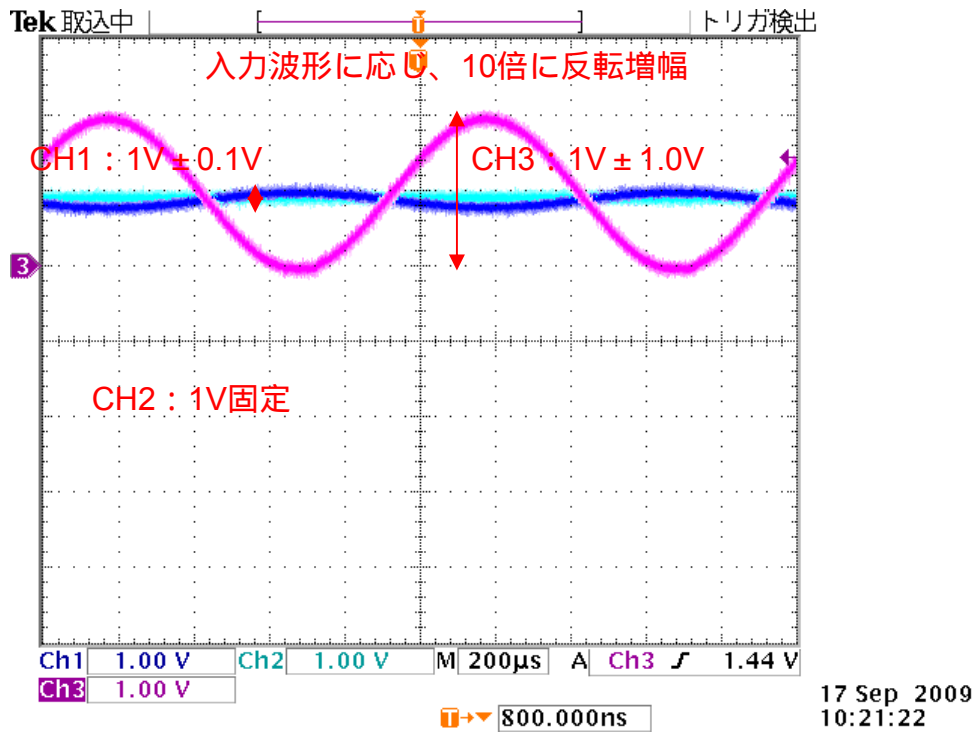


図5-2 反転増幅回路の入出力信号波形

5.3 反転増幅回路 オフセット調整

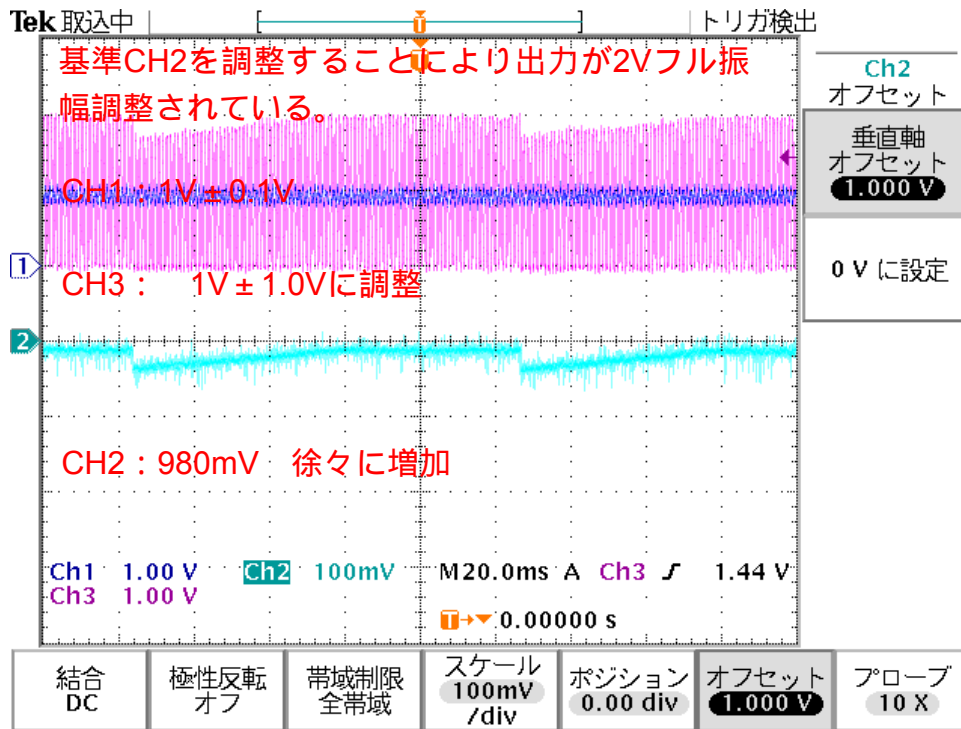


図5-3 反転増幅回路 オフセット調整の入出力信号波形

5.4 反転増幅回路 オフセット調整+ロウパスフィルター

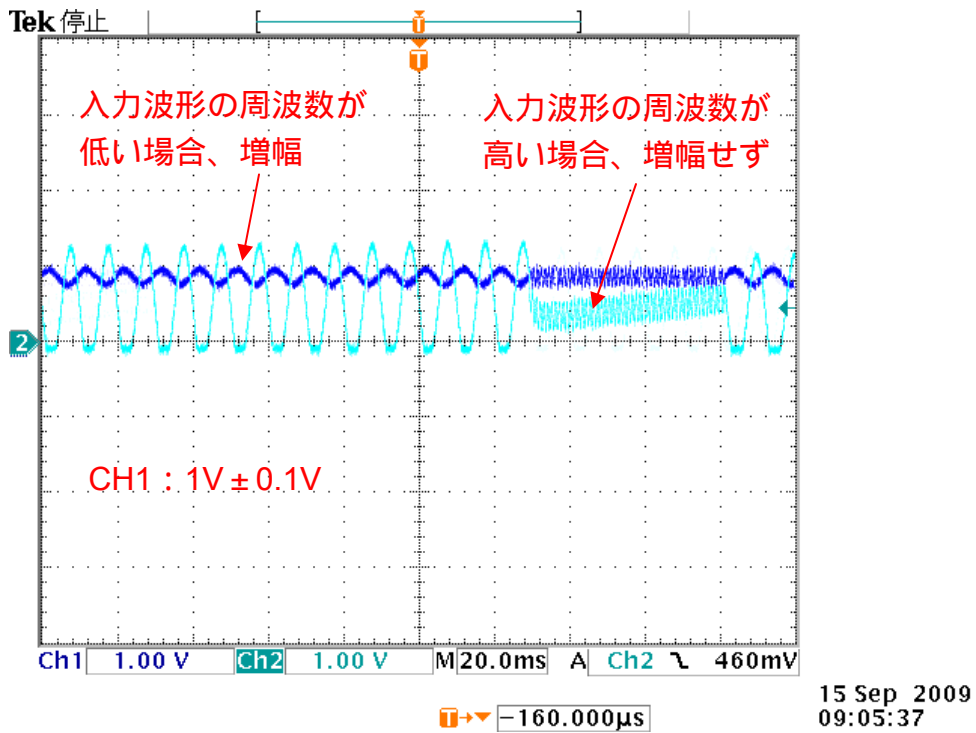
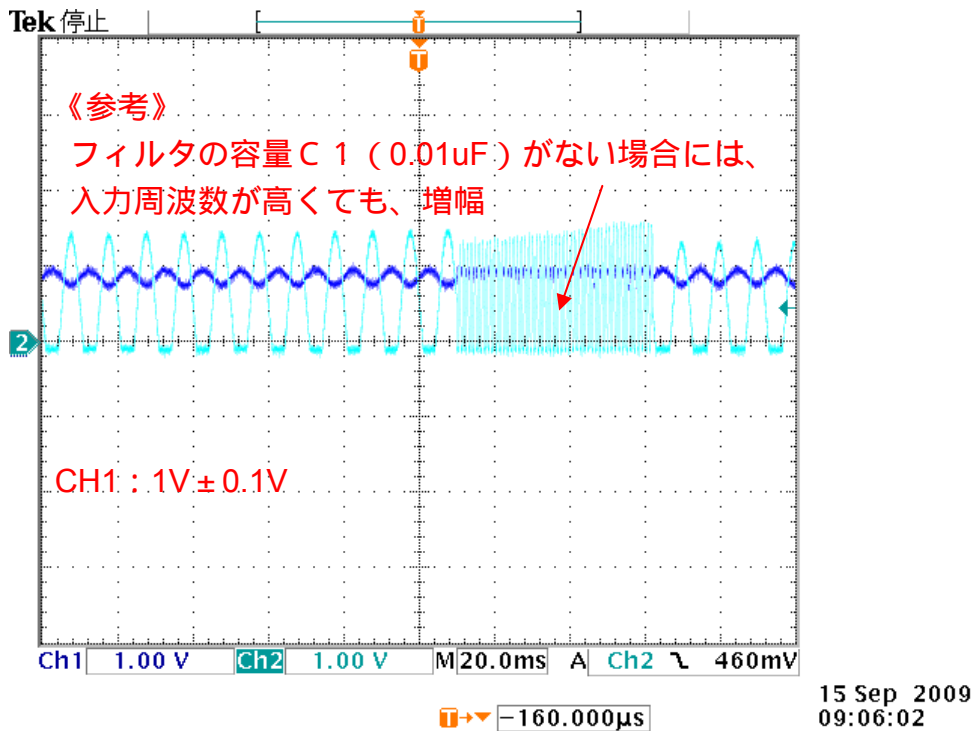


図5-4 反転増幅回路 オフセット調整+ロウパスフィルターの入出力信号波形



《参考》 図5-4-2 反転増幅回路 オフセット調整+ロウパスフィルターの入出力信号波形 (フィルタCなし)

5.5 非反転増幅回路 交流

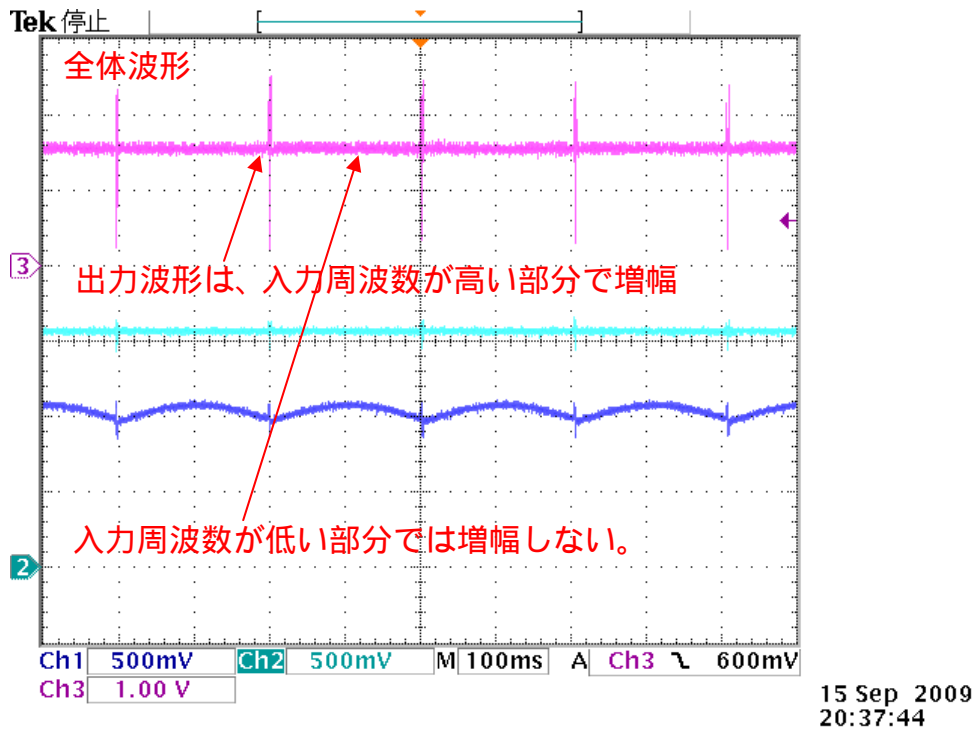


図5-5-1 非反転増幅回路 交流の入出力信号波形

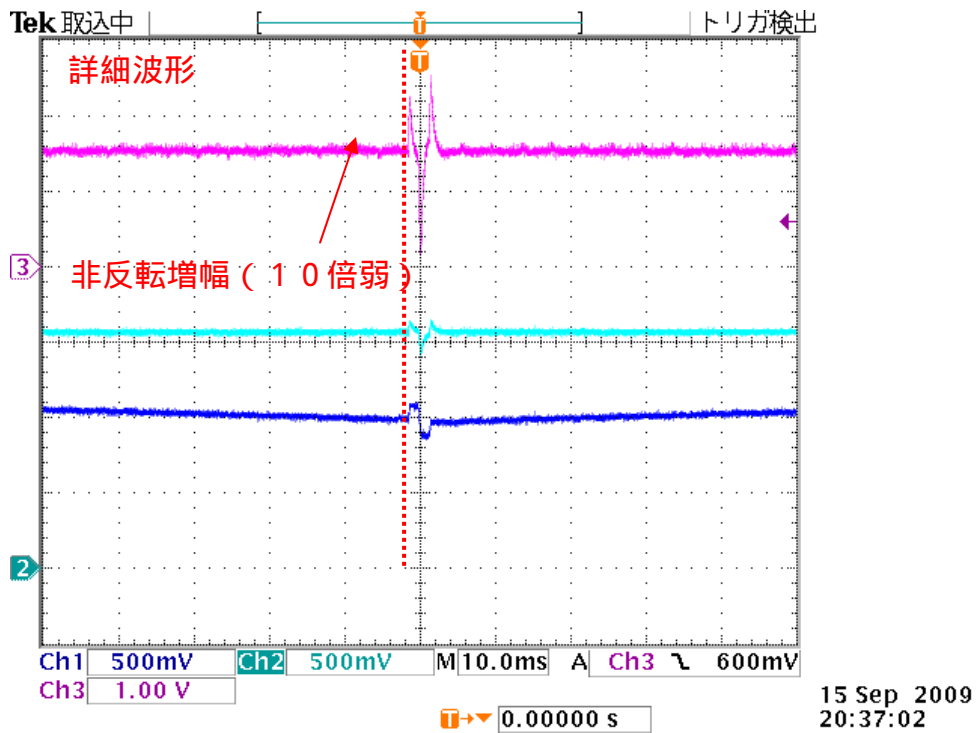
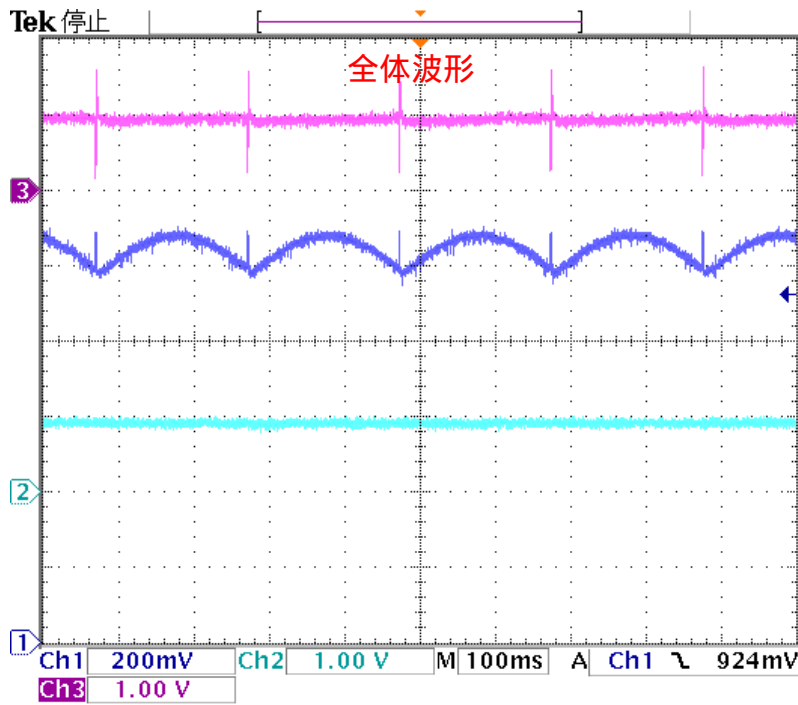


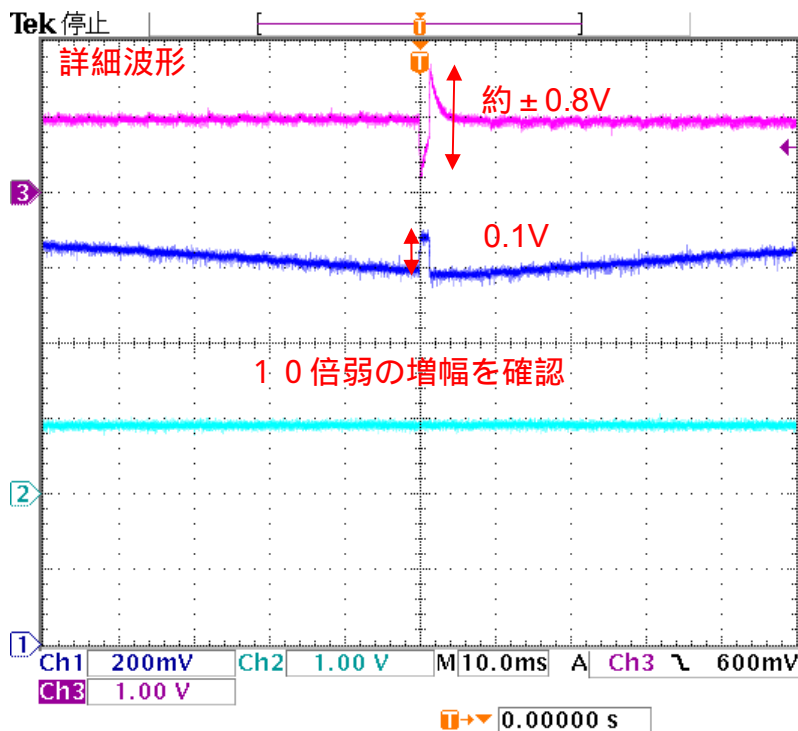
図5-5-2 非反転増幅回路 交流の入出力信号波形

5.6 反転増幅回路 交流



15 Sep 2009
20:20:03

図5-6-1 反転増幅回路 交流の入出力信号波形



15 Sep 2009
20:20:57

図5-6-2 反転増幅回路 交流の入出力信号波形

5.7 インストルメンテーションアンプ

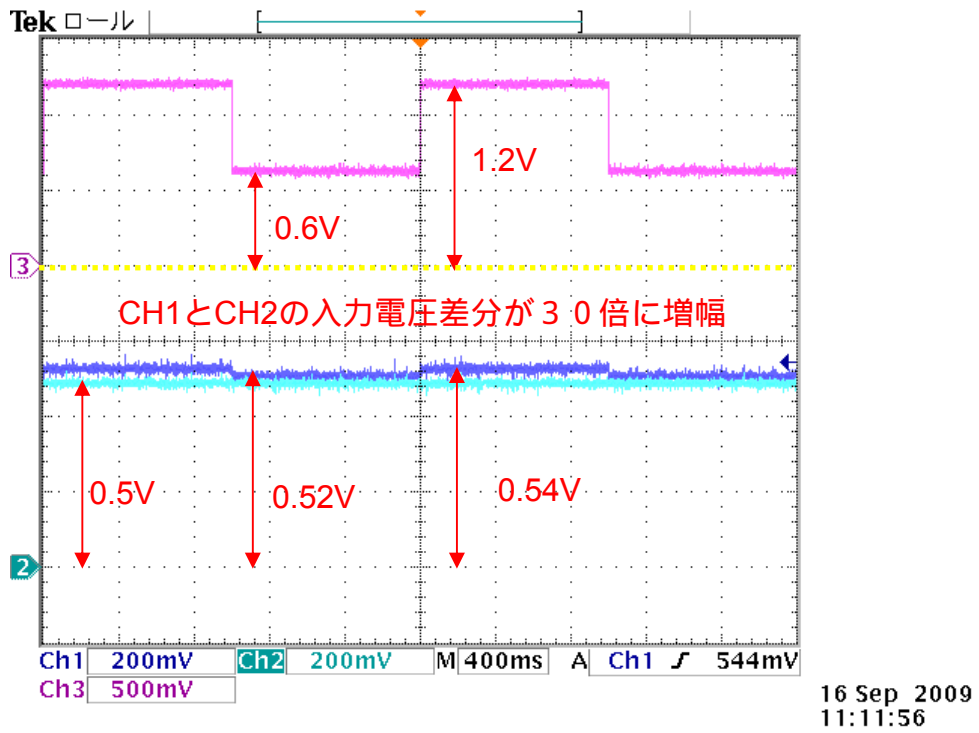


図5-7 インストルメンテーションアンプの入出力信号波形

5.8 比較回路

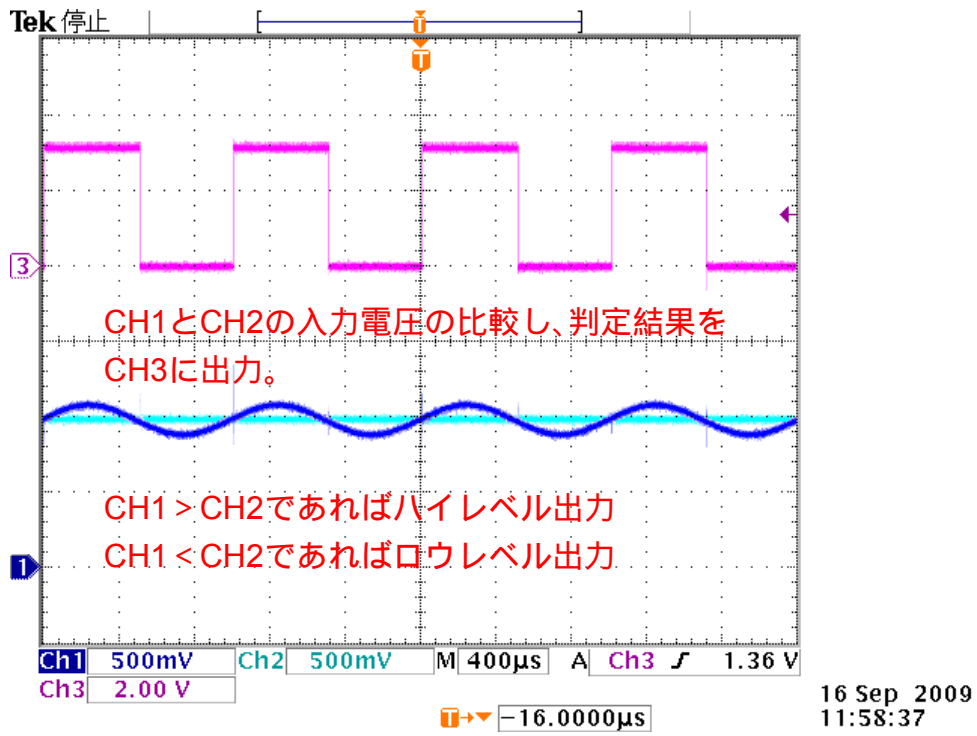


図5-8 比較回路の入出力信号波形

5.9 CR発振回路1

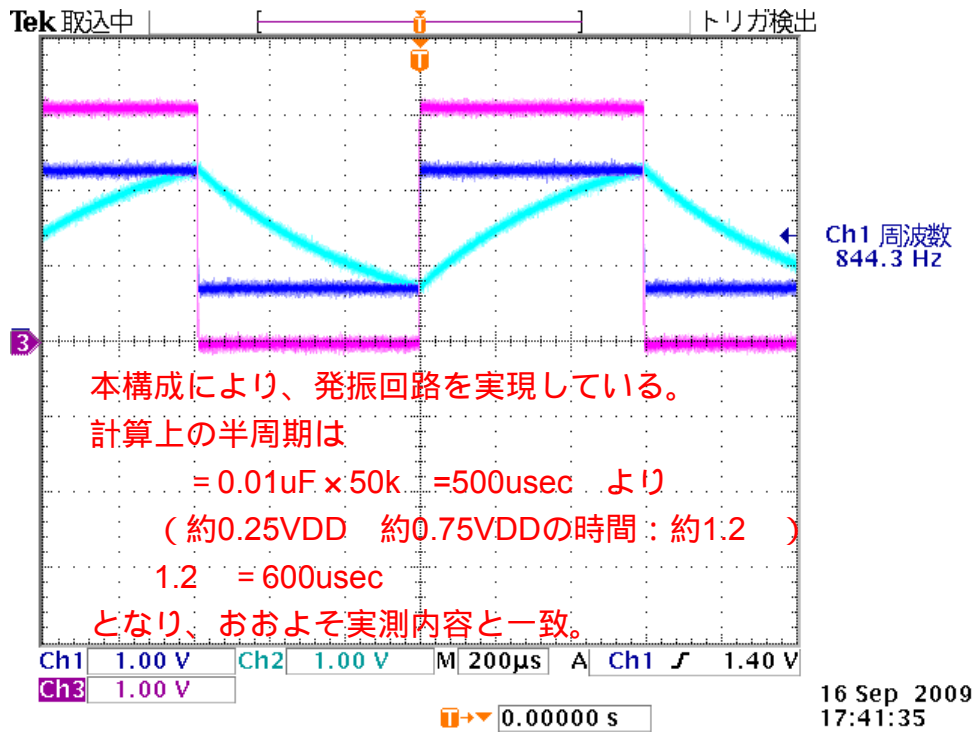


図5-9 CR発振回路1の入出力信号波形

5.10 CR発振回路2

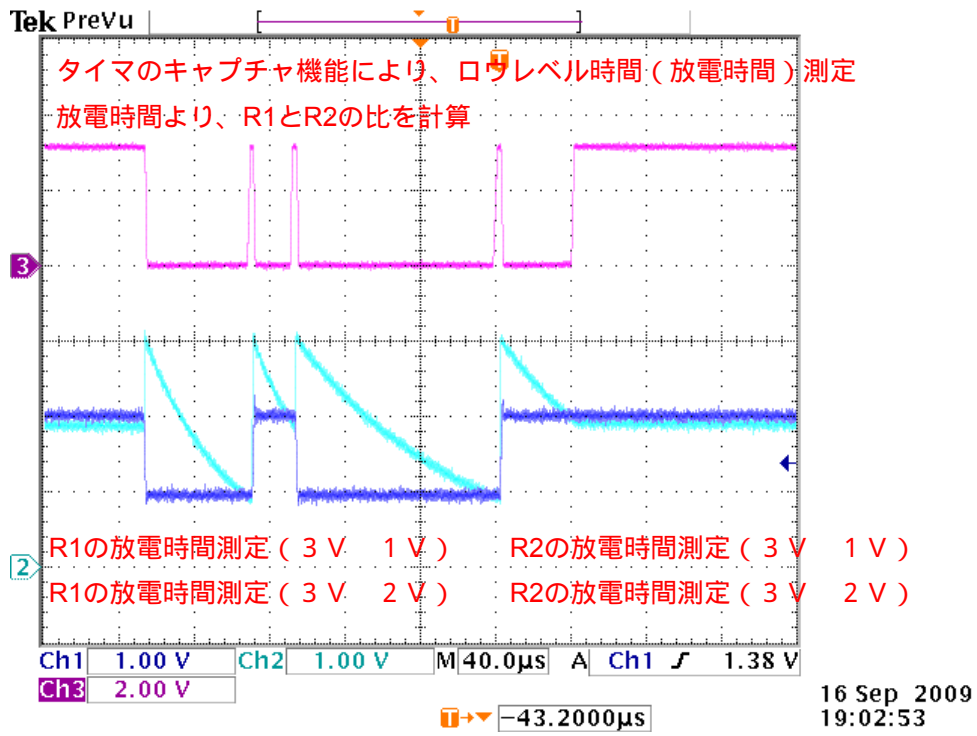


図5-10 CR発振回路2の入出力信号波形

第6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。

(1) 非反転増幅回路

```
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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* 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_InitVr(void);                   /* Setting of
Voltage reference */
void    fn_AdcDaclnit(void);               /* Setting of D/A converter */
void    fn_DmaInnit(void);                 /* Setting of DMA
controler */

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

/*-----*/
/* Local constants
*/
/*-----*/
/*-----*/
/* Global variables
*/
/*-----*/

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

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

```

D/A output data

```
=====*/
static unsigned short ushDacData[100];      /* D/A data for DMA */
const unsigned short aushDacData[100] = {   /* defined D/A data */
/* n      sin(0.2pi*n/1024) */
/*      204,      /* 0      0.100000 */
      217,      /* 1      0.106279 */
      230,      /* 2      0.112533 */
      243,      /* 3      0.118738 */
      255,      /* 4      0.124869 */
      268,      /* 5      0.130902 */
      280,      /* 6      0.136812 */
      292,      /* 7      0.142578 */
      303,      /* 8      0.148175 */
      314,      /* 9      0.153583 */
      325,      /* 10     0.158779 */
      335,      /* 11     0.163742 */
      345,      /* 12     0.168455 */
      354,      /* 13     0.172897 */
      362,      /* 14     0.177051 */
      370,      /* 15     0.180902 */
      377,      /* 16     0.184433 */
      384,      /* 17     0.187631 */
      390,      /* 18     0.190483 */
      395,      /* 19     0.192978 */
      399,      /* 20     0.195106 */
      403,      /* 21     0.196858 */
      406,      /* 22     0.198229 */
      408,      /* 23     0.199211 */
      409,      /* 24     0.199803 */
      409,      /* 25     0.200000 */
      409,      /* 26     0.199803 */
      408,      /* 27     0.199211 */
      406,      /* 28     0.198229 */
      403,      /* 29     0.196858 */
      399,      /* 30     0.195106 */
      395,      /* 31     0.192978 */
      390,      /* 32     0.190483 */
      384,      /* 33     0.187631 */
      377,      /* 34     0.184433 */
      370,      /* 35     0.180902 */
```

362,	/* 36	0.177051 */
354,	/* 37	0.172897 */
345,	/* 38	0.168455 */
335,	/* 39	0.163742 */
325,	/* 40	0.158779 */
314,	/* 41	0.153583 */
303,	/* 42	0.148175 */
292,	/* 43	0.142578 */
280,	/* 44	0.136812 */
268,	/* 45	0.130902 */
255,	/* 46	0.124869 */
243,	/* 47	0.118738 */
230,	/* 48	0.112533 */
217,	/* 49	0.106279 */
204,	/* 50	0.100000 */
191,	/* 51	0.093721 */
179,	/* 52	0.087467 */
166,	/* 53	0.081262 */
153,	/* 54	0.075131 */
141,	/* 55	0.069098 */
129,	/* 56	0.063188 */
117,	/* 57	0.057422 */
106,	/* 58	0.051825 */
95,	/* 59	0.046417 */
84,	/* 60	0.041221 */
74,	/* 61	0.036258 */
64,	/* 62	0.031545 */
55,	/* 63	0.027103 */
47,	/* 64	0.022949 */
39,	/* 65	0.019098 */
31,	/* 66	0.015567 */
25,	/* 67	0.012369 */
19,	/* 68	0.009517 */
14,	/* 69	0.007022 */
10,	/* 70	0.004894 */
6,	/* 71	0.003142 */
3,	/* 72	0.001771 */
1,	/* 73	0.000789 */
0,	/* 74	0.000197 */
0,	/* 75	0.000000 */
0,	/* 76	0.000197 */
1,	/* 77	0.000789 */
3,	/* 78	0.001771 */

```

        6,          /* 79      0.003142 */
       10,          /* 80      0.004894 */
       14,          /* 81      0.007022 */
       19,          /* 82      0.009517 */
       25,          /* 83      0.012369 */
       31,          /* 84      0.015567 */
       39,          /* 85      0.019098 */
       47,          /* 86      0.022949 */
       55,          /* 87      0.027103 */
       64,          /* 88      0.031545 */
       74,          /* 89      0.036258 */
       84,          /* 90      0.041221 */
       95,          /* 91      0.046417 */
      106,          /* 92      0.051825 */
      117,          /* 93      0.057422 */
      129,          /* 94      0.063188 */
      141,          /* 95      0.069098 */
      153,          /* 96      0.075131 */
      166,          /* 97      0.081262 */
      179,          /* 98      0.087467 */
      191,          /* 99      0.093721 */
      204           /* 100     0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAMP */
/*-----*/
fn_AdcDaclnit();

/*-----*/
/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000 : AAAAAAAAAAAAA */
/*||| : 00001 : AAAAAAAAAAAD */
/*||| : 00010 : AAAAAAAAAADD */
/*||| : 00011 : AAAAAAAAAADD */
/*||| : 00100 : AAAAAAAAAADD */
/*||| : 00101 : AAAAAAAAAADD */
/*||| : 00110 : AAAAAAAAAADD */
/*||| : 00111 : AAAAAADD */
/*||| : 01000 : AAAADD */
/*||| : 01001 : AAADD */
/*||| : 01010 : AADD */

```

```

/*||| : 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                   /* Set P33,P32 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 = 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 = 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_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 = 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 */

    /* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
/*|||||||||++++--- : [1]Operation mode of channel 4 */
/*|||||||||          [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1
used as both the start trigger and capture trigger. */

/*||||| : 0 1 0 : Both the edges of TI04 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 4 */
/*||||| : 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 */

TDR04 = 200; /* set interval time to 10us(=0.05us * 200) */
TMMK04 = 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) */
    for(loop = 500; loop > 0; loop--){
        NOP();
    }
    /* wait until VLVI VDD */
    while( LVIF){
        NOP();
    }

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

/*-----*/
/* Module: fn_Dmalnit
*/

/* Description: Setting of DMA contoroler */
/* parameter: --
*/

```

```

        */
/*      return      : --
        */
/*-----*/
void      fn_Dmalnit(void)
{

    DEN0 = 1;          /* Enables operation of DMA channel 0 */

    DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register
0 */

    DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
    DBC0 = 100;          /* DMA Byte Count Register 0 */
    DMC0 = 0b01100100;    /* DMA Mode Control Register 0 */
        /*||||| */
        /*|||++++----- IFC3-IFC0 */
        /*|||          [Selection of DMA start source] */
        /*|||          0000:Disables DMA transfer by interrupt */
        /*|||          (Only software trigger is enabled.) */
        /*|||          0010:INTTM0 */
        /*|||          0011:INTTM01 */
        /*|||          0100:INTTM04 */
        /*|||          0101:INTTM05 */
        /*|||          0110:INTST0/INTCSI00 */
        /*|||          0111:INTSR0/INTCSI01 */
        /*|||          1000:INTST1/INTCSI10/INTIIC10 */
        /*|||          1001:INTSR1 */
        /*|||          1010:INTST3 */
        /*|||          1011:INTSR3 */
        /*|||          1100:INTAD */
        /*|||+----- DWAIT0 */
        /*|||          [Pending of DMA transfer] */
        /*|||          0:Executes DMA transfer upon DMA start request (not held pending).
*/

        /*|||          1:Holds DMA start request pending if any. */
        /*||+----- DS0 */
        /*||          [Specification of transfer data size for DMA transfer] */
        /*||          0:8bits */
        /*||          1:16bits */
        /*|+----- DRS0 */
        /*|          [Selection of DMA transfer direction] */
        /*|          0:SFR to internal RAM */
        /*|          1:Internal RAM to SFR */

```



```

/*+----- STG0 */
/*          [DMA transfer start software trigger] */
/*          0:No trigger operation */
/*          1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

DST0 = 1;          /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}

/*-----*/
/* 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:      fn_AdcDaclnit
*/
/* Description:  Initialization of A/D D/A OPamp module */
/* parameter:   --
*/
/* return      : --
*/

```

```

                                                    */
/*-----*/
void    fn_AdcDaclnit(void)
{
    ADCEN = 1;                                /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

    /*-----*/
    /* Initialization of Voltage reference */
    /*-----*/
    ADVRC = 0b00001000;    /* Analog reference voltage control register */
                            /*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */

                            /*||| |      [2]Operationcontrol of voltage reference */
                            /*||| |      [3]Output voltage selection of voltage reference */
                            /*||| |      [4]Operation control of input gate voltage boost circuit for A/D
converter */

                            /*||| |      [5]Relationship with the conversion mode used */
                            /*||| | */
                            /*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                            /*||| |      [2]Stops operation (Hi-Z) */
                            /*||| |      [3]2.5 V */
                            /*||| |      [4]Stops operation */
                            /*||| |      [5]Can be set in conversion mode 1 */
                            /*||| | */
                            /*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                            /*||| |      [2]Stops operation (Hi-Z) */
                            /*||| |      [3]2.0 V */
                            /*||| |      [4]Enables operation */
                            /*||| |      [5]Can be set in conversion mode 2 or 3 */
                            /*||| | */
                            /*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                            /*||| |      [2]Stops operation (pull-down output) */
                            /*||| |      [3]2.5 V */
                            /*||| |      [4]Stops operation */
                            /*||| |      [5] - */
                            /*||| | */
                            /*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                            /*||| |      [2]Enables operation */
                            /*||| |      [3]2.5 V */
                            /*||| |      [4]Enables operation */
                            /*||| |      [5]Can be set in conversion mode 2 or 3 */
                            /*||| | */

```

```

/*||||| : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*||||| |           [2]Stops operation (pull-down output) */
/*||||| |           [3]2.0 V */
/*||||| |           [4]Enables operation */
/*||||| |           [5] - */
/*||||| | */
/*||||| : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||||| |           [2]Enables operation */
/*||||| |           [3]2.0 V */
/*||||| |           [4]Enables operation */
/*||||| |           [5]Can be set in conversion mode 2 or 3 */
/*||||| | */
/*||||| : Other than the above : Setting prohibited */
/*||||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;           /* Enables operation */
ADVRC.1 = 1;           /* Output 2.0 V */

ADPC = 0b00000000;    /* 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 A */
/*||| : 0 0 0 0 1 : A 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 A A D D */
/*||| : 0 0 0 1 1 : A A A 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 D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*++++----- : Be sure to set 000 */

```

```

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0O) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP1O) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP2O) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (ANO0) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */
DAM.0 = 1; /* D/A converter operation is Real-time output mode */
*/
DAM.2 = 1; /* D/A conversion resolution is 12-bit */
DAM.6 = 1; /* D/A converter positive reference voltage is
VREFOUT */
DACE0 = 0; /* D/A converter CH0 disable */
DACSW0 = aushDacData[99]; /* D/A conversion value setting register 0 */

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

```

```

        */
/*   parameter: --
                                   */
/*   return   : --
                                   */
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    DACE0 = 1;          /* D/A converter CH0 enable */

    /*-----*/
    /*   Initialization of module   */
    /*-----*/
/*   fn_SensorInit();          /* Initialization of Sensor module */

    /*-----*/
    /*   Initialization of variable */
    /*-----*/
/*   uc100msec = C100MSEC;      /* initialize 100msec counter */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TS0L4 = 1;          /* start TAU0 CH4 */
    TMIF04 = 0;        /* reset INTTM04 interrupt request */

    DMAMK0 = 0;        /* enable DMA CH0 interrupt */

    EI();              /* enable all interrupts */

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

```

```
        while (1){
            HALT();
            NOP();
        }
}

/*-----*/
/* Module:      fn_intdma0
*/
/* Description:  DMA ch0 interrupt process
*/
/* parameter:  --
*/
/* return   :  --
*/
/*-----*/
__interrupt void fn_intdma0(void)
{
    DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
    DBC0 = 100;                               /* DMA Byte Count Register 0 */

    DST0 = 1;                                /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */
}

```

(2) 反転増幅回路

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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* Include files
                                     */

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

/* TAU:TDR0n value operation by CK01 (fCLK = 20MHz) */
#define CCK01_PLAY      (2500 - 1) /* 0.125ms = about 8kHz for play */

(0.05[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_InitVr(void);                   /* Setting of
Voltage reference */

void    fn_AdcDaclnit(void);               /* Setting of D/A converter */
void    fn_DmaInit(void);                 /* Setting of DMA
controler */

```

/*-----*/

/* Extern variables/constants

*/

/*-----*/

/*-----*/

/* Local constants

*/

/*-----*/

/*-----*/

/* Global variables

*/

/*-----*/

/*-----*/

/* Local variables

*/

/*-----*/

/*-----*/

/* Code

*/

/*-----*/

/*=====

D/A output data


```

=====*/
static unsigned short ushDacData[100];      /* D/A data for DMA */
const unsigned short aushDacData[100] = {   /* defined D/A data */
                                           /* n      sin(0.2pi*n/100) */
/*      2048,      /* 0      0.100000 */
      2061,      /* 1      0.106279 */
      2074,      /* 2      0.112533 */
      2086,      /* 3      0.118738 */
      2099,      /* 4      0.124869 */
      2111,      /* 5      0.130902 */
      2123,      /* 6      0.136812 */
      2135,      /* 7      0.142578 */
      2147,      /* 8      0.148175 */
      2158,      /* 9      0.153583 */
      2168,      /* 10     0.158779 */
      2179,      /* 11     0.163742 */
      2188,      /* 12     0.168455 */
      2197,      /* 13     0.172897 */
      2206,      /* 14     0.177051 */
      2214,      /* 15     0.180902 */
      2221,      /* 16     0.184433 */
      2227,      /* 17     0.187631 */
      2233,      /* 18     0.190483 */
      2238,      /* 19     0.192978 */
      2243,      /* 20     0.195106 */
      2246,      /* 21     0.196858 */
      2249,      /* 22     0.198229 */
      2251,      /* 23     0.199211 */
      2252,      /* 24     0.199803 */
      2253,      /* 25     0.200000 */
      2252,      /* 26     0.199803 */
      2251,      /* 27     0.199211 */
      2249,      /* 28     0.198229 */
      2246,      /* 29     0.196858 */
      2243,      /* 30     0.195106 */
      2238,      /* 31     0.192978 */
      2233,      /* 32     0.190483 */
      2227,      /* 33     0.187631 */
      2221,      /* 34     0.184433 */
      2214,      /* 35     0.180902 */
      2206,      /* 36     0.177051 */
      2197,      /* 37     0.172897 */
      2188,      /* 38     0.168455 */

```

2179,	/* 39	0.163742 */
2168,	/* 40	0.158779 */
2158,	/* 41	0.153583 */
2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */

```

1863,          /* 82    0.009517 */
1869,          /* 83    0.012369 */
1875,          /* 84    0.015567 */
1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048           /* 100   0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAmp      */
/*-----*/
fn_AdcDaclnit();

```

```

/*-----*/
/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000: AAAAAAAAAAAAA */
/*||| : 00001: AAAAAAAAAAAAAAD */
/*||| : 00010: AAAAAAAAAAAADD */
/*||| : 00011: AAAAAAAAAAADD */
/*||| : 00100: AAAAAAAAAADD */
/*||| : 00101: AAAAAAAAAADD */
/*||| : 00110: AAAAAAAAAADD */
/*||| : 00111: AAAAAADD */
/*||| : 01000: AAAADD */
/*||| : 01001: AAADD */
/*||| : 01010: AADD */
/*||| : 01111: ADDDDDDDDDD */
/*||| : 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 = 0b00001100; /* Set P30-P31,P34 Output latch to Low */

/* Set P33,P32 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 =   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 = 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_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 = 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 */

/* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000;      /* Timer Mode Register 04 */
/*|||||||||+++++ : [1]Operation mode of channel 4 */
/*|||||||||      [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 200; /* set interval time to 10us(=0.05us * 200) */
TMMK04 = 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) */
for(loop = 500; loop > 0; loop--){
    NOP();
}
/* wait until VLVI VDD */
while( LVIF ){
    NOP();
}

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

/*-----*/
/* Module: fn_Dmalnit */
/* Description: Setting of DMA contoroler */
/* parameter: -- */
/* return : -- */
/*-----*/
void fn_Dmalnit(void)

```

```

{

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register
0 */

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100; /* DMA Byte Count Register 0 */
DMC0 = 0b01100100; /* DMA Mode Control Register 0 */
    /*||||| */
    /*|||+----- IFC3-IFC0 */
    /*||| [Selection of DMA start source] */
    /*||| 0000:Disables DMA transfer by interrupt */
    /*||| (Only software trigger is enabled.) */
    /*||| 0010:INTTM0 */
    /*||| 0011:INTTM01 */
    /*||| 0100:INTTM04 */
    /*||| 0101:INTTM05 */
    /*||| 0110:INTST0/INTCSI0 */
    /*||| 0111:INTSR0/INTCSI01 */
    /*||| 1000:INTST1/INTCSI10/INTIIC10 */
    /*||| 1001:INTSR1 */
    /*||| 1010:INTST3 */
    /*||| 1011:INTSR3 */
    /*||| 1100:INTAD */
    /*||+----- DWAIT0 */
    /*|| [Pending of DMA transfer] */
    /*|| 0:Executes DMA transfer upon DMA start request (not held pending).
*/

    /*|| 1:Holds DMA start request pending if any. */
    /*||+----- DS0 */
    /*|| [Specification of transfer data size for DMA transfer] */
    /*|| 0:8bits */
    /*|| 1:16bits */
    /*|+----- DRS0 */
    /*| [Selection of DMA transfer direction] */
    /*| 0:SFR to internal RAM */
    /*| 1:Internal RAM to SFR */
    /*+----- STG0 */
    /* [DMA transfer start software trigger] */
    /* 0:No trigger operation */
    /* 1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

```

```
DST0 = 1;          /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}

/*-----*/
/* 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:      fn_AdcDaclnit
   */
/* Description: Initialization of A/D D/A OPamp module          */
/* parameter: --
   */
/* return  : --
   */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
```

and voltage reference */

/*-----*/

/* Initialization of Voltage reference */

/*-----*/

ADVRC = 0b00001000; /* Analog reference voltage control register */

converters */

/*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A

/*||| | [2]Operationcontrol of voltage reference */

/*||| | [3]Output voltage selection of voltage reference */

/*||| | [4]Operation control of input gate voltage boost circuit for A/D

converter */

/*||| | [5]Relationship with the conversion mode used */

/*||| | */

/*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */

/*||| | [2]Stops operation (Hi-Z) */

/*||| | [3]2.5 V */

/*||| | [4]Stops operation */

/*||| | [5]Can be set in conversion mode 1 */

/*||| | */

/*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */

/*||| | [2]Stops operation (Hi-Z) */

/*||| | [3]2.0 V */

/*||| | [4]Enables operation */

/*||| | [5]Can be set in conversion mode 2 or 3 */

/*||| | */

/*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */

/*||| | [2]Stops operation (pull-down output) */

/*||| | [3]2.5 V */

/*||| | [4]Stops operation */

/*||| | [5] - */

/*||| | */

/*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */

/*||| | [2]Enables operation */

/*||| | [3]2.5 V */

/*||| | [4]Enables operation */

/*||| | [5]Can be set in conversion mode 2 or 3 */

/*||| | */

/*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */

/*||| | [2]Stops operation (pull-down output) */

/*||| | [3]2.0 V */

/*||| | [4]Enables operation */

/*||| | [5] - */

```

/*||| | */
/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||| |          [2]Enables operation */
/*||| |          [3]2.0 V */
/*||| |          [4]Enables operation */
/*||| |          [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 A A */
/*||| : 0 0 0 1 : A 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 A A D D */
/*||| : 0 0 0 1 1 : A A A 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 D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
/*||| */
/*|+++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/

```

```

/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0O) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP1O) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP2O) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (ANO0) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */

/* DAMD0 = 1; /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal
mode */
DAM.2 = 1; /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1; /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting reigster 1 */

}
/*-----*/

```

```
/* Module:      main
                                     */
/* Description:  Main process
                                     */
/*   parameter: --
                                     */
/*   return   : --
                                     */
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    DACE1 = 1;           /* D/A converter CH1 enable */
    DACE0 = 1;           /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TS0L.4 = 1;           /* start TAU0 CH4 */
    TMIF04 = 0;           /* reset INTTM04 interrupt request */

    DMAMK0 = 0;           /* enable DMA CH0 interrupt */

    EI();                 /* enable all interrupts */

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

    while (1){
        HALT();
        NOP();
    }
}
```



```

}

/*-----*/
/* Module:      fn_intdma0
                */
/* Description:  DMA ch0 interrupt process
                */
/* parameter:  --
                */
/* return   :  --
                */
/*-----*/
__interrupt void fn_intdma0(void)
{
    DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
    DBC0 = 100;                               /* DMA Byte Count Register 0 */

    DST0 = 1;                                 /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */
}

```

(3) 反転増幅回路 オフセット調整

```

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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* 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_InitVr(void);                /* Setting of
Voltage reference */
        void    fn_AdcDaclnit(void);            /* Setting of D/A converter */
        void    fn_DmaInit(void);              /* Setting of DMA
controler */

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

/*-----*/
/* Local constants
*/
/*-----*/
/*-----*/
/* Global variables
*/
/*-----*/

/*-----*/
/* Local variables
*/
/*-----*/
static short    shVoltage;                /* voltage (Hex) */
static unsigned short    ushCycleCount;    /* Cycle counter */

/*-----*/
/* Code
*/
/*-----*/
/*=====
D/A output data
=====*/
static unsigned short ushDacData[100];    /* D/A data for DMA */

```

```
const unsigned short aushDacData[100] = {      /* defined D/A data */
                                                /* n      sin(0.2pi*n/100) */
/*      2048,      /* 0      0.100000 */
      2061,      /* 1      0.106279 */
      2074,      /* 2      0.112533 */
      2086,      /* 3      0.118738 */
      2099,      /* 4      0.124869 */
      2111,      /* 5      0.130902 */
      2123,      /* 6      0.136812 */
      2135,      /* 7      0.142578 */
      2147,      /* 8      0.148175 */
      2158,      /* 9      0.153583 */
      2168,      /* 10     0.158779 */
      2179,      /* 11     0.163742 */
      2188,      /* 12     0.168455 */
      2197,      /* 13     0.172897 */
      2206,      /* 14     0.177051 */
      2214,      /* 15     0.180902 */
      2221,      /* 16     0.184433 */
      2227,      /* 17     0.187631 */
      2233,      /* 18     0.190483 */
      2238,      /* 19     0.192978 */
      2243,      /* 20     0.195106 */
      2246,      /* 21     0.196858 */
      2249,      /* 22     0.198229 */
      2251,      /* 23     0.199211 */
      2252,      /* 24     0.199803 */
      2253,      /* 25     0.200000 */
      2252,      /* 26     0.199803 */
      2251,      /* 27     0.199211 */
      2249,      /* 28     0.198229 */
      2246,      /* 29     0.196858 */
      2243,      /* 30     0.195106 */
      2238,      /* 31     0.192978 */
      2233,      /* 32     0.190483 */
      2227,      /* 33     0.187631 */
      2221,      /* 34     0.184433 */
      2214,      /* 35     0.180902 */
      2206,      /* 36     0.177051 */
      2197,      /* 37     0.172897 */
      2188,      /* 38     0.168455 */
      2179,      /* 39     0.163742 */
      2168,      /* 40     0.158779 */
```

2158,	/* 41	0.153583 */
2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */
1863,	/* 82	0.009517 */
1869,	/* 83	0.012369 */

```

1875,          /* 84    0.015567 */
1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048          /* 100   0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAmp */
/*-----*/
fn_AdcDaclnit();

/*-----*/

```

```

/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000 : A A A A A A A A A A */
/*||| : 00001 : A A A A A A A A A D */
/*||| : 00010 : A A A A A A A A D D */
/*||| : 00011 : A A A A A A A A D D D */
/*||| : 00100 : A A A A A A A D D D D */
/*||| : 00101 : A A A A A A A D D D D D */
/*||| : 00110 : A A A A A A D D D D D D */
/*||| : 00111 : A A A A A D D D D D D D */
/*||| : 01000 : A A A A D D D D D D D D */
/*||| : 01001 : A A A D D D D D D D D D */
/*||| : 01010 : A A D D D D D D D D D D */
/*||| : 01111 : A D D D D D D D D D D D */
/*||| : 10000 : 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-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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                   /* Set P33,P32 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 =   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 = 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_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 = 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 */

    /* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
/*|||||||||+++++--- : [1]Operation mode of channel 4 */
/*||||||||| [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 200; /* set interval time to 10us(=0.05us * 200) */
TMMK04 = 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) */
                for(loop = 500; loop > 0; loop--){
                    NOP();
                }
                /* wait until VLVI  VDD */
                while( LVIF){
                    NOP();
                }

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

/*-----*/
/* Module:        fn_Dmalnit
                */
/* Description:   Setting of DMA contoroler
                */
/* parameter: --
                */
/* return  : --
                */
/*-----*/
void fn_Dmalnit(void)
{

```

```

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register
0 */

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100; /* DMA Byte Count Register 0 */
DMC0 = 0b01100100; /* DMA Mode Control Register 0 */
/*||||| */
/*|||++++----- IFC3-IFC0 */
/*||| [Selection of DMA start source] */
/*||| 0000:Disables DMA transfer by interrupt */
/*||| (Only software trigger is enabled.) */
/*||| 0010:INTTM00 */
/*||| 0011:INTTM01 */
/*||| 0100:INTTM04 */
/*||| 0101:INTTM05 */
/*||| 0110:INTST0/INTCSI00 */
/*||| 0111:INTSR0/INTCSI01 */
/*||| 1000:INTST1/INTCSI10/INTIIC10 */
/*||| 1001:INTSR1 */
/*||| 1010:INTST3 */
/*||| 1011:INTSR3 */
/*||| 1100:INTAD */
/*|||+----- DWAIT0 */
/*||| [Pending of DMA transfer] */
/*||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
/*||| 1:Holds DMA start request pending if any. */
/*||+----- DS0 */
/*|| [Specification of transfer data size for DMA transfer] */
/*|| 0:8bits */
/*|| 1:16bits */
/*|+----- DRS0 */
/*| [Selection of DMA transfer direction] */
/*| 0:SFR to internal RAM */
/*| 1:Internal RAM to SFR */
/*+----- STG0 */
/* [DMA transfer start software trigger] */
/* 0:No trigger operation */
/* 1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

```

```
DST0 = 1;          /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}

/*-----*/
/* 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:      fn_AdcDaclnit
                */
/* Description: Initialization of A/D D/A OPamp module
                */
/* parameter:  --
                */
/* return   :  --
                */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */
}
```

```

/*-----*/
/* Initialization of Voltage reference */
/*-----*/
ADVRC = 0b00001000; /* Analog reference voltage control register */
/*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
/*||| | [2]Operationcontrol of voltage reference */
/*||| | [3]Output voltage selection of voltage reference */
/*||| | [4]Operation control of input gate voltage boost circuit for A/D
converter */
/*||| | [5]Relationship with the conversion mode used */
/*||| | */
/*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5]Can be set in conversion mode 1 */
/*||| | */
/*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Enables operation */
/*||| | [3]2.5 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5] - */
/*||| | */

```



```

/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||| |          [2]Enables operation */
/*||| |          [3]2.0 V */
/*||| |          [4]Enables operation */
/*||| |          [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 A */
/*||| : 0 0 0 0 1 : A 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 A D D */
/*||| : 0 0 0 1 1 : A 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 D D D D D */
/*||| : 0 0 1 1 0 : A 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 D */
/*||| : 0 1 0 0 0 : A A A A 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 */
/*||| : 0 1 0 1 0 : A A 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 */
/*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
/*||| */
/*|+++++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */

```

```

/*-----*/
PM2.1 = 1;          /* input port mode (AMP0O) */
PM2.0 = 1;          /* input port mode (AMP0-) */
PM2.2 = 1;          /* input port mode (AMP0+) */

PM2.4 = 1;          /* input port mode (AMP1O) */
PM2.3 = 1;          /* input port mode (AMP1-) */
PM2.5 = 1;          /* input port mode (AMP1+) */

PM2.7 = 1;          /* input port mode (AMP2O) */
PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;         /* input port mode (AMP2+) */

PM11.0 = 1;         /* input port mode (ANO0) */

ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;          /* operational amplifier (AMP0) enable */
OAC.1 = 0;          /* operational amplifier (AMP1) disable */
OAC.2 = 0;          /* operational amplifier (AMP2) disable */

/*-----*/
/*      Initialization of D/A converter      */
/*-----*/
DACEN = 1;          /* supplies input clock to D/A converter */

DAM.0 = 1;          /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.2 = 1;          /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1;          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0;          /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1;          /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[80]; /* D/A conversion value setting reigster 1 */

}
/*-----*/
/* Module:      main
*/

/* Description:  Main process

```

```
        */
/*   parameter: --
                                   */
/*   return   : --
                                   */
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    ushCycleCount = 0;                /* Cycle counter */

    DACE1 = 1;                        /* D/A converter CH1 enable */
    DACE0 = 1;                        /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TS0L.4 = 1;                       /* start TAU0 CH4 */
    TMIF04 = 0;                       /* reset INTTM04 interrupt request */

    DMAMK0 = 0;                       /* enable DMA CH0 interrupt */

    EI();                             /* enable all interrupts */

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

    while (1){
        HALT();
        NOP();
    }
}
```

```

}

/*-----*/
/* Module:      fn_intdma0
                */
/* Description:  DMA ch0 interrupt process
                */
/* parameter:  --
                */
/* return   :  --
                */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TT0L4 = 1;          /* stop TAU0 CH4 */

    /* start A/D conversion */
    /* DI();           /* disable all interrupts */
    ADCS = 1;          /* A/D converter enable */
    ADIF = 0; /* clear A/D interrupt request flag */
    ADMK = 0;          /* enable A/D interrupt */
    HALT();           /* set HALT mode and wait conversion finished */
    ADMK = 1;          /* disable A/D interrupt */
    ADIF = 0; /* clear A/D interrupt request flag */
    /* EI();           /* enable all interrupts */

    shVoltage = (short)(ADCR & 0x0fff); /* get A/D result */
    shVoltage -= (2048 - 1);
    if(shVoltage < 0){
        DACSW1 += 1; /* D/A conversion value setting register 1 */
    }
    else if(shVoltage > 2){
        DACSW1 -= 1; /* D/A conversion value setting register 1 */
    }

    ushCycleCount++; /* Cycle counter */
    if(ushCycleCount > 100){ /* Cycle counter */
        DACSW1 = ushDacData[95]; /* D/A conversion value setting register 1 */
        ushCycleCount = 0; /* reset Cycle counter */
    }
}

```

```

DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
DBC0 = 100;                                     /* DMA Byte Count Register 0 */

DST0 = 1;                                       /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */

TSOL.4 = 1;                                     /* start TAU0 CH4 */
TMIF04 = 0;                                     /* reset INTTM04 interrupt request */

}

```

(4) 反転増幅回路 オフセット調整 + ローパスフィルター

```

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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* 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_InitVr(void);           /* Setting of
Voltage reference */
        void    fn_AdcDaclnit(void);       /* Setting of D/A converter */
        void    fn_DmaInnit(void);        /* Setting of DMA
controler */

```

/*-----*/

/* Extern variables/constants

*/

/*-----*/

/*-----*/

/* Local constants

*/

/*-----*/

/*-----*/

/* Global variables

*/

/*-----*/

/*-----*/

/* Local variables

*/

/*-----*/

static short shVoltage; /* voltage (Hex) */

static unsigned short ushCycleCount; /* Cycle counter */

static unsigned char ucCycleStatus; /* Cycle status */

/*-----*/

/* Code

*/

/*-----*/

/*=====

D/A output data

=====*/

```

static unsigned short ushDacData[100];      /* D/A data for DMA */
const unsigned short aushDacData[100] = {   /* defined D/A data */
                                           /* n      sin(0.2pi*n/100) */
/*      2048,      /* 0      0.100000 */
      2061,      /* 1      0.106279 */
      2074,      /* 2      0.112533 */
      2086,      /* 3      0.118738 */
      2099,      /* 4      0.124869 */
      2111,      /* 5      0.130902 */
      2123,      /* 6      0.136812 */
      2135,      /* 7      0.142578 */
      2147,      /* 8      0.148175 */
      2158,      /* 9      0.153583 */
      2168,      /* 10     0.158779 */
      2179,      /* 11     0.163742 */
      2188,      /* 12     0.168455 */
      2197,      /* 13     0.172897 */
      2206,      /* 14     0.177051 */
      2214,      /* 15     0.180902 */
      2221,      /* 16     0.184433 */
      2227,      /* 17     0.187631 */
      2233,      /* 18     0.190483 */
      2238,      /* 19     0.192978 */
      2243,      /* 20     0.195106 */
      2246,      /* 21     0.196858 */
      2249,      /* 22     0.198229 */
      2251,      /* 23     0.199211 */
      2252,      /* 24     0.199803 */
      2253,      /* 25     0.200000 */
      2252,      /* 26     0.199803 */
      2251,      /* 27     0.199211 */
      2249,      /* 28     0.198229 */
      2246,      /* 29     0.196858 */
      2243,      /* 30     0.195106 */
      2238,      /* 31     0.192978 */
      2233,      /* 32     0.190483 */
      2227,      /* 33     0.187631 */
      2221,      /* 34     0.184433 */
      2214,      /* 35     0.180902 */
      2206,      /* 36     0.177051 */
      2197,      /* 37     0.172897 */
      2188,      /* 38     0.168455 */
      2179,      /* 39     0.163742 */

```

2168,	/* 40	0.158779 */
2158,	/* 41	0.153583 */
2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */
1863,	/* 82	0.009517 */

```

1869,          /* 83    0.012369 */
1875,          /* 84    0.015567 */
1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048           /* 100   0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAmp */
/*-----*/
fn_AdcDaclnit();

```

```

/*-----*/
/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000 : AAAAAAAAAAAAA */
/*||| : 00001 : AAAAAAAAAAAAAAD */
/*||| : 00010 : AAAAAAAAAAAADD */
/*||| : 00011 : AAAAAAAAAAADD */
/*||| : 00100 : AAAAAAAAAADDD */
/*||| : 00101 : AAAAAAAAAADDDDD */
/*||| : 00110 : AAAAAADDDDDDD */
/*||| : 00111 : AAAAAADDDDDDDDD */
/*||| : 01000 : AAAADDDDDDDDDDD */
/*||| : 01001 : AAADDDDDDDDDDDDD */
/*||| : 01010 : AADDDDDDDDDDDDDDD */
/*||| : 01111 : ADDDDDDDDDDDDDDDD */
/*||| : 10000 : DDDDDDDDDDDDDDDDD */
/*||| */
/*+++----- : 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                   /* Set P33,P32 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 =   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 = 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_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 = 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 */

    /* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
/*|||||||||+++++--- : [1]Operation mode of channel 4 */
/*||||||||| [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 200; /* set interval time to 10us(=0.05us * 200) */
TMMK04 = 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) */
                for(loop = 500; loop > 0; loop--){
                    NOP();
                }
                /* wait until VLVI  VDD */
                while( LVIF){
                    NOP();
                }

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

/*-----*/
/* Module:      fn_Dmalnit
                */
/* Description:  Setting of DMA contoroler
                */
/* parameter: --
                */
/* return  : --
                */
/*-----*/
void fn_Dmalnit(void)

```

```

{

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register
0 */

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100; /* DMA Byte Count Register 0 */
DMC0 = 0b01100100; /* DMA Mode Control Register 0 */
    /*||||| */
    /*|||+----- IFC3-IFC0 */
    /*||| [Selection of DMA start source] */
    /*||| 0000:Disables DMA transfer by interrupt */
    /*||| (Only software trigger is enabled.) */
    /*||| 0010:INTTM0 */
    /*||| 0011:INTTM01 */
    /*||| 0100:INTTM04 */
    /*||| 0101:INTTM05 */
    /*||| 0110:INTST0/INTCSI0 */
    /*||| 0111:INTSR0/INTCSI01 */
    /*||| 1000:INTST1/INTCSI10/INTIIC10 */
    /*||| 1001:INTSR1 */
    /*||| 1010:INTST3 */
    /*||| 1011:INTSR3 */
    /*||| 1100:INTAD */
    /*|||+----- DWAIT0 */
    /*||| [Pending of DMA transfer] */
    /*||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/

    /*||| 1:Holds DMA start request pending if any. */
    /*||+----- DS0 */
    /*|| [Specification of transfer data size for DMA transfer] */
    /*|| 0:8bits */
    /*|| 1:16bits */
    /*|+----- DRS0 */
    /*| [Selection of DMA transfer direction] */
    /*| 0:SFR to internal RAM */
    /*| 1:Internal RAM to SFR */
    /*|+----- STG0 */
    /*| [DMA transfer start software trigger] */
    /*| 0:No trigger operation */
    /*| 1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

```

```

        DST0 = 1;                /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

    }

/*-----*/
/* 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:      fn_AdcDaclnit
                */
/* Description:  Initialization of A/D D/A OPamp module                */
/* parameter:   --
                */
/* return      :  --
                */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;                /* supplies input clock to A/D converter, operational amplifier,

```

and voltage reference */

/*-----*/

/* Initialization of Voltage reference */

/*-----*/

ADVRC = 0b00001000; /* Analog reference voltage control register */

converters */

/*| | | | | : [2]Operationcontrol of voltage reference */

/*| | | | | : [3]Output voltage selection of voltage reference */

/*| | | | | : [4]Operation control of input gate voltage boost circuit for A/D

converter */

/*| | | | | : [5]Relationship with the conversion mode used */

/*| | | | | */

/*| | | | | : 0 0 0 : [1]AVREFP (external voltage reference input) */

/*| | | | | : [2]Stops operation (Hi-Z) */

/*| | | | | : [3]2.5 V */

/*| | | | | : [4]Stops operation */

/*| | | | | : [5]Can be set in conversion mode 1 */

/*| | | | | */

/*| | | | | : 0 1 0 : [1]AVREFP (external voltage reference input) */

/*| | | | | : [2]Stops operation (Hi-Z) */

/*| | | | | : [3]2.0 V */

/*| | | | | : [4]Enables operation */

/*| | | | | : [5]Can be set in conversion mode 2 or 3 */

/*| | | | | */

/*| | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */

/*| | | | | : [2]Stops operation (pull-down output) */

/*| | | | | : [3]2.5 V */

/*| | | | | : [4]Stops operation */

/*| | | | | : [5] - */

/*| | | | | */

/*| | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */

/*| | | | | : [2]Enables operation */

/*| | | | | : [3]2.5 V */

/*| | | | | : [4]Enables operation */

/*| | | | | : [5]Can be set in conversion mode 2 or 3 */

/*| | | | | */

/*| | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */

/*| | | | | : [2]Stops operation (pull-down output) */

/*| | | | | : [3]2.0 V */

/*| | | | | : [4]Enables operation */

/*| | | | | : [5] - */

```

/*||| | */
/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||| |          [2]Enables operation */
/*||| |          [3]2.0 V */
/*||| |          [4]Enables operation */
/*||| |          [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 A A */
/*||| : 0 0 0 1 : A A A A A A A A A A A D */
/*||| : 0 0 1 0 : A A A A A A A A A A D D */
/*||| : 0 0 1 1 : A A 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 D D D D D D */
/*||| : 0 0 1 1 1 : 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 */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*|+++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/

```

```

/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0O) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP1O) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP2O) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (ANO0) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */

DAM.0 = 1; /* [DAMD0] D/A converter operation is Real-time
output mode */
/* DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal
mode */
DAM.2 = 1; /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1; /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[90]; /* D/A conversion value setting reigster 1 */

}
/*-----*/

```

```

/* Module:      main
*/

/* Description:  Main process
*/

/* parameter:  --
*/

/* return   :  --
*/

/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    ushCycleCount = 0;                /* Cycle counter */
    ucCycleStatus = 0;                /* Cycle status */

    DACE1 = 1;                        /* D/A converter CH1 enable */
    DACE0 = 1;                        /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TS0L4 = 1;                        /* start TAU0 CH4 */
    TMIF04 = 0;                       /* reset INTTM04 interrupt request */

    DMAMK0 = 0;                       /* enable DMA CH0 interrupt */

    EI();                             /* enable all interrupts */

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

```

```

        while (1){
            HALT();
            NOP();
        }
    }

/*-----*/
/* Module:      fn_intdma0
                                   */
/* Description:  DMA ch0 interrupt process
                                   */
/*   parameter: --
                                   */
/*   return   : --
                                   */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TT0L4 = 1;                      /* stop TAU0 CH4 */

    /* start A/D conversion */
/*   DI();                          /* disable all interrupts */
    ADCS = 1;                       /* A/D converter enable */
    ADIF = 0; /* clear A/D interrupt request flag */
    ADMK = 0;                       /* enable A/D interrupt */
    HALT();                          /* set HALT mode and wait conversion finished */
    ADMK = 1;                       /* disable A/D interrupt */
    ADIF = 0; /* clear A/D interrupt request flag */
/*   EI();                          /* enable all interrupts */

    shVoltage = (short)(ADCR & 0x0fff); /* get A/D result */
    shVoltage -= (2048 - 1);
    if(shVoltage < 0){
        DACSW1 += 1;                /* D/A conversion value setting register 1 */
    }
    else if(shVoltage > 2){
        DACSW1 -= 1;                /* D/A conversion value setting register 1 */
    }

    ushCycleCount++;                /* Cycle counter */
    if(ushCycleCount > 50){         /* Cycle counter */
        switch(ucCycleStatus){      /* Cycle status */

```



```

        case 0:
            TDR04 = 2000;                /* set interval time to
100us(=0.05us * 2000) */

            ucCycleStatus = 1;          /* Cycle status (100Hz) */
            ushCycleCount = 20;         /* reset Cycle counter */
            break;

        case 1:
            TDR04 = 200;                /* set interval time to
10us(=0.05us * 200) */

            ucCycleStatus = 0;          /* Cycle status (1kHz) */
            ushCycleCount = 0;         /* reset Cycle counter */
            break;

    }
    DACSW1 = ushDacData[90];           /* D/A conversion value setting register 1 */
/*    ushCycleCount = 0;                /* reset Cycle counter */

}

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 100;                           /* DMA Byte Count Register 0 */

DST0 = 1;                              /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */

TS0L4 = 1;                             /* start TAU0 CH4 */
TMIF04 = 0;                             /* reset INTTM04 interrupt request */

}

```

(5) 非反転増幅回路 交流

```

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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* 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_InitVr(void); /* Setting of
Voltage reference */
void fn_AdcDaclnit(void); /* Setting of D/A converter */
void fn_DmaInnit(void); /* Setting of DMA
controler */

```

/*-----*/

/* Extern variables/constants

*/

/*-----*/

/*-----*/

/* Local constants

*/

/*-----*/

/*-----*/

/* Global variables

*/

/*-----*/

/*-----*/

/* Local variables

*/

/*-----*/

static short shVoltage; /* voltage (Hex) */

static unsigned short ushCycleCount; /* Cycle counter */

static unsigned char ucCycleStatus; /* Cycle status */

/*-----*/

/* Code

*/

/*-----*/

/*=====

D/A output data

=====*/

```

static unsigned short ushDacData[100];      /* D/A data for DMA */
const unsigned short aushDacData[100] = {   /* defined D/A data */
                                           /* n      sin(0.2pi*n/100) */
/*      2048,      /* 0      0.100000 */
      2061,      /* 1      0.106279 */
      2074,      /* 2      0.112533 */
      2086,      /* 3      0.118738 */
      2099,      /* 4      0.124869 */
      2111,      /* 5      0.130902 */
      2123,      /* 6      0.136812 */
      2135,      /* 7      0.142578 */
      2147,      /* 8      0.148175 */
      2158,      /* 9      0.153583 */
      2168,      /* 10     0.158779 */
      2179,      /* 11     0.163742 */
      2188,      /* 12     0.168455 */
      2197,      /* 13     0.172897 */
      2206,      /* 14     0.177051 */
      2214,      /* 15     0.180902 */
      2221,      /* 16     0.184433 */
      2227,      /* 17     0.187631 */
      2233,      /* 18     0.190483 */
      2238,      /* 19     0.192978 */
      2243,      /* 20     0.195106 */
      2246,      /* 21     0.196858 */
      2249,      /* 22     0.198229 */
      2251,      /* 23     0.199211 */
      2252,      /* 24     0.199803 */
      2253,      /* 25     0.200000 */
      2252,      /* 26     0.199803 */
      2251,      /* 27     0.199211 */
      2249,      /* 28     0.198229 */
      2246,      /* 29     0.196858 */
      2243,      /* 30     0.195106 */
      2238,      /* 31     0.192978 */
      2233,      /* 32     0.190483 */
      2227,      /* 33     0.187631 */
      2221,      /* 34     0.184433 */
      2214,      /* 35     0.180902 */
      2206,      /* 36     0.177051 */
      2197,      /* 37     0.172897 */
      2188,      /* 38     0.168455 */
      2179,      /* 39     0.163742 */

```

2168,	/* 40	0.158779 */
2158,	/* 41	0.153583 */
2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */
1863,	/* 82	0.009517 */

```

1869,          /* 83    0.012369 */
1875,          /* 84    0.015567 */
1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048           /* 100   0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAmp */
/*-----*/
fn_AdcDaclnit();

```

```

/*-----*/
/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000 : AAAAAAAAAAAAA */
/*||| : 00001 : AAAAAAAAAAAAAAD */
/*||| : 00010 : AAAAAAAAAAAADD */
/*||| : 00011 : AAAAAAAAAAADD */
/*||| : 00100 : AAAAAAAAAADDD */
/*||| : 00101 : AAAAAAAAAADDDDD */
/*||| : 00110 : AAAAAADDDDDDD */
/*||| : 00111 : AAAAAADDDDDDD */
/*||| : 01000 : AAAADDDDDDDDD */
/*||| : 01001 : AAADDDDDDDDDDD */
/*||| : 01010 : AADDDDDDDDDDDDD */
/*||| : 01111 : ADDDDDDDDDDDDDD */
/*||| : 10000 : DDDDDDDDDDDDDDD */
/*||| */
/*+++----- : 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                   /* Set P33,P32 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 =   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 = 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_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 = 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 */

    /* 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 TISO 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000;      /* Timer Mode Register 04 */
                                /*|||++++--- : [1]Operation mode of channel 4 */
                                /*|||          [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 2000; /* set interval time to 100us(=0.05us * 2000) */
TMMK04 = 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) */
                for(loop = 500; loop > 0; loop--){
                    NOP();
                }
                /* wait until VLVI  VDD */
                while( LVIF){
                    NOP();
                }

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

/*-----*/
/* Module:        fn_Dmalnit
                */
/* Description:   Setting of DMA contoroler
                */
/* parameter: --
                */
/* return : --
                */
/*-----*/
void fn_Dmalnit(void)
{

```

```

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register
0 */

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
/*
DBC0 = 100; /* DMA Byte Count Register 0 */
DBC0 = 25; /* DMA Byte Count Register 0 */
DMC0 = 0b01100100; /* DMA Mode Control Register 0 */
/*| | | | | | | | */
/*| | | | + + + + + ----- IFC3-IFC0 */
/*| | | | [Selection of DMA start source] */
/*| | | | 0000:Disables DMA transfer by interrupt */
/*| | | | (Only software trigger is enabled.) */
/*| | | | 0010:INTTM00 */
/*| | | | 0011:INTTM01 */
/*| | | | 0100:INTTM04 */
/*| | | | 0101:INTTM05 */
/*| | | | 0110:INTST0/INTCSI00 */
/*| | | | 0111:INTSR0/INTCSI01 */
/*| | | | 1000:INTST1/INTCSI10/INTIIC10 */
/*| | | | 1001:INTSR1 */
/*| | | | 1010:INTST3 */
/*| | | | 1011:INTSR3 */
/*| | | | 1100:INTAD */
/*| | | | + ----- DWAIT0 */
/*| | | | [Pending of DMA transfer] */
/*| | | | 0:Executes DMA transfer upon DMA start request (not held pending).
*/
/*| | | | 1:Holds DMA start request pending if any. */
/*| | | | + ----- DS0 */
/*| | | | [Specification of transfer data size for DMA transfer] */
/*| | | | 0:8bits */
/*| | | | 1:16bits */
/*| | | | + ----- DRS0 */
/*| | | | [Selection of DMA transfer direction] */
/*| | | | 0:SFR to internal RAM */
/*| | | | 1:Internal RAM to SFR */
/*| | | | + ----- STG0 */
/*| | | | [DMA transfer start software trigger] */
/*| | | | 0:No trigger operation */
/*| | | | 1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

```

```

        DST0 = 1;                /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

    }

/*-----*/
/* 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:      fn_AdcDaclnit
               */
/* Description: Initialization of A/D D/A OPAm module
               */
/* parameter:  --
               */
/* return   :  --
               */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;                /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */
}

```

```

/*-----*/
/* Initialization of Voltage reference */
/*-----*/
ADVRC = 0b00001000; /* Analog reference voltage control register */
/*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
/*||| | [2]Operationcontrol of voltage reference */
/*||| | [3]Output voltage selection of voltage reference */
/*||| | [4]Operation control of input gate voltage boost circuit for A/D
converter */
/*||| | [5]Relationship with the conversion mode used */
/*||| | */
/*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5]Can be set in conversion mode 1 */
/*||| | */
/*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Enables operation */
/*||| | [3]2.5 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5] - */
/*||| | */

```



```

/*||||| : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||||| |           [2]Enables operation */
/*||||| |           [3]2.0 V */
/*||||| |           [4]Enables operation */
/*||||| |           [5]Can be set in conversion mode 2 or 3 */
/*||||| | */
/*||||| | : Other than the above : Setting prohibited */
/*||||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;           /* Enables operation */
ADVRC.1 = 1;           /* Output 2.0 V */

ADPC = 0b00000000;    /* 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 A */
/*||| : 0 0 0 0 1 : A 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 A D D */
/*||| : 0 0 0 1 1 : A 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 D D D D D */
/*||| : 0 0 1 1 0 : A 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 D */
/*||| : 0 1 0 0 0 : A A A A 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 */
/*||| : 0 1 0 1 0 : A A 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 */
/*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
/*||| */
/*|++++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */

```

```

/*-----*/
PM2.1 = 1;          /* input port mode (AMP0O) */
PM2.0 = 1;          /* input port mode (AMP0-) */
PM2.2 = 1;          /* input port mode (AMP0+) */

PM2.4 = 1;          /* input port mode (AMP1O) */
PM2.3 = 1;          /* input port mode (AMP1-) */
PM2.5 = 1;          /* input port mode (AMP1+) */

PM2.7 = 1;          /* input port mode (AMP2O) */
PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;         /* input port mode (AMP2+) */

PM11.0 = 1;         /* input port mode (ANO0) */

ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;          /* operational amplifier (AMP0) enable */
OAC.1 = 0;          /* operational amplifier (AMP1) disable */
OAC.2 = 0;          /* operational amplifier (AMP2) disable */

/*-----*/
/*      Initialization of D/A converter      */
/*-----*/
DACEN = 1;          /* supplies input clock to D/A converter */

DAM.0 = 1;          /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.2 = 1;          /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1;          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

}

/*-----*/
/* Module:      main
*/

/* Description:  Main process
*/

/* parameter:  --
*/

/* return   :  --
*/

```

```

/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

/*    ushCycleCount = 0;              /* Cycle counter */
    ucCycleStatus = 0;              /* Cycle status */

/*    DACE1 = 1;                      /* D/A converter CH1 enable */
    DACE0 = 1;                      /* D/A converter CH0 enable */

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

/* initialize D/Aconverter data */
for(count = 0; count<100; count++){
    ushDacData[count] = aushDacData[count];
}

TS0L4 = 1;                          /* start TAU0 CH4 */
TMIF04 = 0;                          /* reset INTTM04 interrupt request */

DMAMK0 = 0;                          /* enable DMA CH0 interrupt */

EI();                                /* enable all interrupts */

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

while (1){
    HALT();
    NOP();
}
}

/*-----*/

```

```

/* Module:      fn_intdma0
*/
/* Description:  DMA ch0 interrupt process
*/
/* parameter: --
*/
/* return  : --
*/
/*-----*/
__interrupt void fn_intdma0(void)
{
    TT0L4 = 1;          /* stop TAU0 CH4 */

    switch(ucCycleStatus){          /* Cycle status */
        case 0:
            TDR04 = 20000;          /* set interval time to
1000us(=0.05us * 20000) */
            ucCycleStatus = 1;      /* Cycle status (Flat) */
            /* DMA RAM
Address Register 0 */
            DBC0 = 1;              /* DMA Byte Count
Register 0 */
            break;

        case 1:
            TDR04 = 200;           /* set interval time to
10us(=0.05us * 200) */
            ucCycleStatus = 2;     /* Cycle status (2/4,3/4 sign) */
            DBC0 = 50 - 1;        /* DMA Byte Count Register 0
*/
            break;

        case 2:
            TDR04 = 20000;        /* set interval time to
1000us(=0.05us * 20000) */
            ucCycleStatus = 3;     /* Cycle status (Flat) */
            DBC0 = 1;             /* DMA Byte Count
Register 0 */
            break;

        case 3:
            TDR04 = 200;         /* set interval time to
10us(=0.05us * 200) */

```

```

ucCycleStatus = 4;          /* Cycle status (1kHz) */
DBC0 = 25 - 1;             /* DMA Byte Count Register 0
*/

break;

case 4:

DRA0 = (unsigned short)&ushDacData[0];      /* DMA RAM
Address Register 0 */

DACSW0 = ushDacData[99];          /* D/A conversion value
setting reigster 0 */

TMR04 = 0b0000000000000000;        /* Timer Mode Register 04 */
TDR04 = 20000;                    /* set interval time to
1000us(=0.05us * 20000) */

ucCycleStatus = 5;          /* Cycle status (10Hz) */
DBC0 = 50;                  /* DMA Byte Count
Register 0 */

break;

case 5:

DRA0 = (unsigned short)&ushDacData[0];      /* DMA RAM
Address Register 0 */

DACSW0 = ushDacData[99];          /* D/A conversion value
setting reigster 0 */

TMR04 = 0b1000000000000000;        /* Timer Mode Register 04 */
TDR04 = 200;                      /* set interval time to
10us(=0.05us * 200) */

ucCycleStatus = 0;          /* Cycle status (1kHz) */
DBC0 = 25;                  /* DMA
Byte Count Register 0 */

break;

}

/* DBC0 = 100;                /* DMA Byte Count Register 0 */

DST0 = 1;                    /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */

TS0L.4 = 1;                  /* start TAU0 CH4 */
TMIF04 = 0;                  /* reset INTTM04 interrupt request */

}

```

(6) 反転増幅回路 交流

```

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

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----*/
/* 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_InitVr(void);          /* Setting of
Voltage reference */
        void    fn_AdcDaclnit(void);      /* Setting of D/A converter */
        void    fn_DmaInnit(void);       /* Setting of DMA
controler */

```

/*-----*/

/* Extern variables/constants

*/

/*-----*/

/*-----*/

/* Local constants

*/

/*-----*/

/*-----*/

/* Global variables

*/

/*-----*/

/*-----*/

/* Local variables

*/

/*-----*/

static short shVoltage; /* voltage (Hex) */

static unsigned short ushCycleCount; /* Cycle counter */

static unsigned char ucCycleStatus; /* Cycle status */

/*-----*/

/* Code

*/

/*-----*/

/*=====

D/A output data

=====*/

```

static unsigned short ushDacData[100];      /* D/A data for DMA */
const unsigned short aushDacData[100] = {   /* defined D/A data */
                                           /* n      sin(0.2pi*n/100) */
/*      2048,      /* 0      0.100000 */
      2061,      /* 1      0.106279 */
      2074,      /* 2      0.112533 */
      2086,      /* 3      0.118738 */
      2099,      /* 4      0.124869 */
      2111,      /* 5      0.130902 */
      2123,      /* 6      0.136812 */
      2135,      /* 7      0.142578 */
      2147,      /* 8      0.148175 */
      2158,      /* 9      0.153583 */
      2168,      /* 10     0.158779 */
      2179,      /* 11     0.163742 */
      2188,      /* 12     0.168455 */
      2197,      /* 13     0.172897 */
      2206,      /* 14     0.177051 */
      2214,      /* 15     0.180902 */
      2221,      /* 16     0.184433 */
      2227,      /* 17     0.187631 */
      2233,      /* 18     0.190483 */
      2238,      /* 19     0.192978 */
      2243,      /* 20     0.195106 */
      2246,      /* 21     0.196858 */
      2249,      /* 22     0.198229 */
      2251,      /* 23     0.199211 */
      2252,      /* 24     0.199803 */
      2253,      /* 25     0.200000 */
      2252,      /* 26     0.199803 */
      2251,      /* 27     0.199211 */
      2249,      /* 28     0.198229 */
      2246,      /* 29     0.196858 */
      2243,      /* 30     0.195106 */
      2238,      /* 31     0.192978 */
      2233,      /* 32     0.190483 */
      2227,      /* 33     0.187631 */
      2221,      /* 34     0.184433 */
      2214,      /* 35     0.180902 */
      2206,      /* 36     0.177051 */
      2197,      /* 37     0.172897 */
      2188,      /* 38     0.168455 */
      2179,      /* 39     0.163742 */

```

2168,	/* 40	0.158779 */
2158,	/* 41	0.153583 */
2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */
1863,	/* 82	0.009517 */

```

1869,          /* 83    0.012369 */
1875,          /* 84    0.015567 */
1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048           /* 100   0.100000 */

```

```
};
```

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

```
/* 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 A/D,D/A,OPAmp */
/*-----*/
fn_AdcDaclnit();

```

```

/*-----*/
/* Initialization of DMA */
/*-----*/
fn_Dmalnit();

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 */
/*||| : 00000 : AAAAAAAAAAAAA */
/*||| : 00001 : AAAAAAAAAAAAAAD */
/*||| : 00010 : AAAAAAAAAAAAAADDD */
/*||| : 00011 : AAAAAAAAAAAAAADDDD */
/*||| : 00100 : AAAAAAAAAADDDDD */
/*||| : 00101 : AAAAAAAAAADDDDDD */
/*||| : 00110 : AAAAAADDDDDDD */
/*||| : 00111 : AAAAAADDDDDDDD */
/*||| : 01000 : AAAADDDDDDDDD */
/*||| : 01001 : AAADDDDDDDDDDD */
/*||| : 01010 : AADDDDDDDDDDDDD */
/*||| : 01111 : ADDDDDDDDDDDDDD */
/*||| : 10000 : DDDDDDDDDDDDDDD */
/*||| */
/*+++----- : 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                   /* Set P33,P32 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 =   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 = 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_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 = 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 */

    /* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                                /*|||||||||+++++--- : [1]Operation mode of channel 4 */
                                /*|||||||||           [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 2000; /* set interval time to 100us(=0.05us * 2000) */
TMMK04 = 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) */
                for(loop = 500; loop > 0; loop--){
                    NOP();
                }
                /* wait until VLVI  VDD */
                while( LVIF){
                    NOP();
                }

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

/*-----*/
/* Module:      fn_Dmalnit
                */
/* Description:  Setting of DMA contoroler
                */
/* parameter:  --
                */
/* return :  --
                */
/*-----*/
void fn_Dmalnit(void)
{

```

```

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register
0 */

DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM Address Register 0 */
DBC0 = 25; /* DMA Byte Count Register 0 */
DMC0 = 0b01100100; /* DMA Mode Control Register 0 */
    /*||||| */
    /*|||+++++----- IFC3-IFC0 */
    /*||| [Selection of DMA start source] */
    /*||| 0000:Disables DMA transfer by interrupt */
    /*||| (Only software trigger is enabled.) */
    /*||| 0010:INTTM00 */
    /*||| 0011:INTTM01 */
    /*||| 0100:INTTM04 */
    /*||| 0101:INTTM05 */
    /*||| 0110:INTST0/INTCSI00 */
    /*||| 0111:INTSR0/INTCSI01 */
    /*||| 1000:INTST1/INTCSI10/INTIIC10 */
    /*||| 1001:INTSR1 */
    /*||| 1010:INTST3 */
    /*||| 1011:INTSR3 */
    /*||| 1100:INTAD */
    /*|||+----- DWAIT0 */
    /*||| [Pending of DMA transfer] */
    /*||| 0:Executes DMA transfer upon DMA start request (not held pending).
*/
    /*||| 1:Holds DMA start request pending if any. */
    /*||+----- DS0 */
    /*|| [Specification of transfer data size for DMA transfer] */
    /*|| 0:8bits */
    /*|| 1:16bits */
    /*|+----- DRS0 */
    /*| [Selection of DMA transfer direction] */
    /*| 0:SFR to internal RAM */
    /*| 1:Internal RAM to SFR */
    /*+----- STG0 */
    /* [DMA transfer start software trigger] */
    /* 0:No trigger operation */
    /* 1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

DST0 = 1; /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

```

```
}

/*-----*/
/* 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:      fn_AdcDaclnit
                */
/* Description:  Initialization of A/D D/A OPamp module          */
/* parameter:   --                                          */
/* return      :  --                                          */
                */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */
}
```

```

/*-----*/
/* Initialization of Voltage reference */
/*-----*/
ADVRC = 0b00001000; /* Analog reference voltage control register */
/*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */

/*||| | [2]Operationcontrol of voltage reference */
/*||| | [3]Output voltage selection of voltage reference */
/*||| | [4]Operation control of input gate voltage boost circuit for A/D
converter */

/*||| | [5]Relationship with the conversion mode used */
/*||| | */
/*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5]Can be set in conversion mode 1 */
/*||| | */
/*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*||| | [2]Stops operation (Hi-Z) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Enables operation */
/*||| | [3]2.5 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */

```

```

/*||| |      [2]Enables operation */
/*||| |      [3]2.0 V */
/*||| |      [4]Enables operation */
/*||| |      [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 */
/*||| : 00000 : A A A A A A A A A A A */
/*||| : 00001 : A A A A A A A A A A A D */
/*||| : 00010 : A A A A A A A A A A A D D */
/*||| : 00011 : A A A A A A A A A A A D D D */
/*||| : 00100 : A A A A A A A A D D D D */
/*||| : 00101 : A A A A A A A A D D D D D */
/*||| : 00110 : A A A A A A A D D D D D D */
/*||| : 00111 : A A A A A A D D D D D D D */
/*||| : 01000 : A A A A D D D D D D D D D */
/*||| : 01001 : A A A D D D D D D D D D D */
/*||| : 01010 : A A D D D D D D D D D D D D */
/*||| : 01111 : A D D D D D D D D D D D D D */
/*||| : 10000 : D D D D D D D D D D D D D D */
/*||| */
/*+++++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/

```

```

PM2.1 = 1;          /* input port mode (AMP0O) */
PM2.0 = 1;          /* input port mode (AMP0-) */
PM2.2 = 1;          /* input port mode (AMP0+) */

PM2.4 = 1;          /* input port mode (AMP1O) */
PM2.3 = 1;          /* input port mode (AMP1-) */
PM2.5 = 1;          /* input port mode (AMP1+) */

PM2.7 = 1;          /* input port mode (AMP2O) */
PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;         /* input port mode (AMP2+) */

PM11.0 = 1;         /* input port mode (ANO0) */

ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;          /* operational amplifier (AMP0) enable */
OAC.1 = 0;          /* operational amplifier (AMP1) disable */
OAC.2 = 0;          /* operational amplifier (AMP2) disable */

/*-----*/
/*      Initialization of D/A converter      */
/*-----*/
DACEN = 1;          /* supplies input clock to D/A converter */

DAM.0 = 1;          /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.2 = 1;          /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1;          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0;          /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1;          /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting reigster 1 */

}

/*-----*/
/* Module:      main
*/

/* Description:  Main process

```

```

        */
/*   parameter: --
                                   */
/*   return   : --
                                   */
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

/*   ushCycleCount = 0;              /* Cycle counter */
    ucCycleStatus = 0;              /* Cycle status */

    DACE1 = 1;                      /* D/A converter CH1 enable */
    DACE0 = 1;                      /* D/A converter CH0 enable */

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

/*   initialize D/Aconverter data */
for(count = 0; count<100; count++){
    ushDacData[count] = aushDacData[count];
}

    TS0L4 = 1;                      /* start TAU0 CH4 */
    TMIF04 = 0;                     /* reset INTTM04 interrupt request */

    DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

    EI();                            /* enable all interrupts */

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

    while (1){
        HALT();
        NOP();
    }
}

```

```

    }
}

/*-----*/
/* Module:      fn_intdma0
                */
/* Description:  DMA ch0 interrupt process
                */
/* parameter: --
                */
/* return  : --
                */
/*-----*/
__interrupt void fn_intdma0(void)
{
    TTOL.4 = 1;                /* stop TAU0 CH4 */

    switch(ucCycleStatus){    /* Cycle status */
        case 0:
            TDR04 = 20000;     /* set interval time to
1000us(=0.05us * 20000) */
            ucCycleStatus = 1; /* Cycle status (Flat) */
            DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM
Address Register 0 */
            DBC0 = 1;         /* DMA Byte Count
Register 0 */
            break;

        case 1:
            TDR04 = 200;      /* set interval time to
10us(=0.05us * 200) */
            ucCycleStatus = 2; /* Cycle status (2/4 sign) */
            DBC0 = 25;        /* DMA
Byte Count Register 0 */
            break;

        case 2:
            DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM
Address Register 0 */
            DACSW0 = ushDacData[99]; /* D/A conversion value
setting register 0 */
            TMR04 = 0b0000000000000000; /* Timer Mode Register 04 */
    }
}

```



```

TDR04 = 20000; /* set interval time to
1000us(=0.05us * 20000) */

ucCycleStatus = 3; /* Cycle status (slow) */
DBC0 = 50; /* DMA
Byte Count Register 0 */

break;

case 3:
DRA0 = (unsigned short)&ushDacData[0]; /* DMA RAM
Address Register 0 */

DACSW0 = ushDacData[99]; /* D/A conversion value
setting register 0 */

TMR04 = 0b10000000000000000; /* Timer Mode Register 04 */
TDR04 = 200; /* set interval time to
10us(=0.05us * 200) */

ucCycleStatus = 0; /* Cycle status (1kHz) */
DBC0 = 25; /* DMA
Byte Count Register 0 */

break;

}

/* DBC0 = 100; /* DMA Byte Count Register 0 */
DST0 = 1; /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */

TS0L4 = 1; /* start TAU0 CH4 */
TMIF04 = 0; /* reset INTTM04 interrupt request */

}

```

(7) インストールメンテーションアンプ

```

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

#pragma interrupt INTRTC fn_intrtc RB1

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

/* TAU:TDR0n value operation by CK01 (fCLK = 20MHz) */
#define CCK01_PLAY      (2500 - 1) /* 0.125ms = about 8kHz for play */
                                     */
(0.05[us/clock] * 2500[count]) */

#define CDAC0_50V      ( 1024 )
#define CDAC0_52V      ( 1065 )
#define CDAC0_54V      ( 1105 )

```

```

/*-----*/
/* Function prototypes
*/
/*-----*/
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_InitVr(void);                   /* Setting of
Voltage reference */
void    fn_InitRtc(void);                  /* Setting of
Real-time counter */
void    fn_AdcDaclnit(void);                /* Setting of D/A converter */

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

/*-----*/
/* Local constants
*/
/*-----*/
/*-----*/
/* Global variables
*/
/*-----*/

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

/*-----*/
/* 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 500ms from when XT1 enable) */
fn_Wait500usBase(500000/500);

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

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

/*-----*/
/* Initialization of A/D,D/A,OPAMP */
/*-----*/
fn_AdcDaclnit();

/* CPU is operating on a High-speed system clock */
CKC = 0b01000000; /* CPU/peripheral hardware clock by fSUB*/

while(CLS != 1){
    NOP();
}

HIOSTOP = 1; /* internal high-speed oscillation stopped */
MSTOP = 1; /* X1 oscillator stopped */

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 */
                        /* ||| : 00000 : AAAAAAAAAAAAA */
                        /* ||| : 00001 : AAAAAAAAAAAAA D */
                        /* ||| : 00010 : AAAAAAAAAAAD D */
                        /* ||| : 00011 : AAAAAAAAAADDD */
                        /* ||| : 00100 : AAAAAAAAAADDDD */
                        /* ||| : 00101 : AAAAAAAAAADDDD */
                        /* ||| : 00110 : AAAAAADDDDDD */
                        /* ||| : 00111 : AAAAAADDDDDD */
                        /* ||| : 01000 : AAAADDDDDDDD */
                        /* ||| : 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:AMP0-2 */

/*-----*/
/*      Setting of Port 3
          */
/*-----*/
      P3 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                              /* Set P33,P32 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 =   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 = 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:AMP2 */
                                     /* P151,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 = 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 */

    /* 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 */

}

/*-----*/
/* 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 : * LWISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
/*||||| * LWISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
/*||||| : 1 : * LWISEL = 0: VDD < VLVI */
/*||||| * LWISEL = 1: EXLVI < VEXLVI */
/*||||| */
/*|||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD)
*/

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

/*||||| * LWISEL = 1: Generates an interrupt signal */
/*||||| when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
/*||||| or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
/*||||| : 1 : * LWISEL = 0: Generates an internal reset signal when VDD < VLVI */
/*||||| and releases the reset signal when VDD >= VLVI. */
/*||||| * LWISEL = 1: Generates an internal reset signal when EXLVI <
VEXLVI */
/*||||| and releases the reset signal when EXLVI >= VEXLVI.
*/

/*||||| */
/*|||||+---- : Voltage detection selection(LWISEL) */
/*||||| : 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) */
for(loop = 500; loop > 0; loop--){
    NOP();
}
/* wait until VLVI = VDD */
while( LVIF ){
    NOP();
}

```



```

    LVIIF = 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 = 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 */
/*    RTCE = 1;                /* starts counter operation */

}

/*-----*/
/* 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:      fn_AdcDaclnit
               */
/* Description:  Initialization of A/D D/A OPAMP module          */
/* parameter:   --
               */
/* return      : --
               */

```

```

/*-----*/
void  fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

    /*-----*/
    /*  Initialization of Voltage reference          */
    /*-----*/
    ADVRC = 0b00001000; /* Analog reference voltage control register */
        /*|||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
        /*||| |      [2]Operationcontrol of voltage reference */
        /*||| |      [3]Output voltage selection of voltage reference */
        /*||| |      [4]Operation control of input gate voltage boost circuit for A/D
converter */

        /*||| |      [5]Relationship with the conversion mode used */
        /*||| | */
        /*||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
        /*||| |      [2]Stops operation (Hi-Z) */
        /*||| |      [3]2.5 V */
        /*||| |      [4]Stops operation */
        /*||| |      [5]Can be set in conversion mode 1 */
        /*||| | */
        /*||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
        /*||| |      [2]Stops operation (Hi-Z) */
        /*||| |      [3]2.0 V */
        /*||| |      [4]Enables operation */
        /*||| |      [5]Can be set in conversion mode 2 or 3 */
        /*||| | */
        /*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
        /*||| |      [2]Stops operation (pull-down output) */
        /*||| |      [3]2.5 V */
        /*||| |      [4]Stops operation */
        /*||| |      [5] - */
        /*||| | */
        /*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
        /*||| |      [2]Enables operation */
        /*||| |      [3]2.5 V */
        /*||| |      [4]Enables operation */
        /*||| |      [5]Can be set in conversion mode 2 or 3 */
        /*||| | */
        /*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */

```

```

/*||| |          [2]Stops operation (pull-down output) */
/*||| |          [3]2.0 V */
/*||| |          [4]Enables operation */
/*||| |          [5] - */
/*||| | */
/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||| |          [2]Enables operation */
/*||| |          [3]2.0 V */
/*||| |          [4]Enables operation */
/*||| |          [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 A */
/*||| : 0 0 0 0 1 : A 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 A A D D */
/*||| : 0 0 0 1 1 : A A A 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 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 */
/*||| : 0 0 1 1 1 : A A A A A A A A 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 */
/*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
/*||| */
/*++++----- : Be sure to set 000 */

```

```

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0O) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP1O) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP2O) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (ANO0) */

OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 1; /* operational amplifier (AMP1) enable */
OAC.2 = 1; /* operational amplifier (AMP2) enable */

/*-----*/
/* Initialization of D/A converter */
/*-----*/
DACEN = 1; /* supplies input clock to D/A converter */

/* DAMD0 = 1; /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.0 = 0; /* [DAMD0] D/A converter operation is Normal
mode */
DAM.2 = 1; /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1; /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = CDAC0_52V; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0; /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1; /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = CDAC0_50V; /* D/A conversion value setting reigster 1 */

```

```
}

/*-----*/
/* Module:      main
                */
/* Description:  Main process
                */
/*   parameter: --
                */
/*   return   : --
                */
/*-----*/
void main(void)
{
    DACE1 = 1;          /* D/A converter CH1 enable */
    DACE0 = 1;          /* D/A converter CH0 enable */

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

    RTCIF = 0;          /* reset RCT interrupt request */
    RTCMK = 0;          /* enable RTC interrupt */
    RTCE  = 1;          /* starts counter operation */

    EI();              /* enable all interrupts */

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

    while (1)
    {
        HALT();
        NOP();
    }
}
```

```
/*-----*/
/* Module:      fn_intrtc
                */
/* Description:  RTC interrupt process
                */
/*   parameter: --
                */
/*   return   : --
                */
/*-----*/
__interrupt void fn_intrtc(void)
{
    switch(DACSW0)
    {
        case    CDAC0_52V:                /* DAC0 output voltage : 0.52V */
            DACSW0 = CDAC0_54V;          /* DAC0 output voltage : 0.52V -> 0.54V
*/
            break;
        case    CDAC0_54V:                /* DAC0 output voltage : 0.54V */
        default:
            DACSW0 = CDAC0_52V;          /* DAC0 output voltage : 0.54V -> 0.52V
*/
            break;
    }
}
}
```

(8) 比較回路

```
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

#pragma interrupt INTDMA0 fn_intdma0 RB1
#pragma interrupt INTPO          fn_intp0 RB2

/*-----*/
/* 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_InitVr(void);           /* Setting of
Voltage reference */

void    fn_AdcDacInit(void);       /* Setting of D/A converter */
void    fn_DmaInit(void);          /* Setting of DMA
controler */

```

/*-----*/

/* Extern variables/constants

*/

/*-----*/

/*-----*/

/* Local constants

*/

/*-----*/

/*-----*/

/* Global variables

*/

/*-----*/

/*-----*/

/* Local variables

*/

/*-----*/

/*-----*/

/* Code

*/

/*-----*/

/*=====*/

D/A output data

=====*/

static unsigned short ushDacData[100]; /* D/A data for DMA */

const unsigned short aushDacData[100] = { /* defined D/A data */

		/* n	sin(0.2pi*n/100) */
/*	2048,	/* 0	0.100000 */
	2061,	/* 1	0.106279 */
	2074,	/* 2	0.112533 */
	2086,	/* 3	0.118738 */
	2099,	/* 4	0.124869 */
	2111,	/* 5	0.130902 */
	2123,	/* 6	0.136812 */
	2135,	/* 7	0.142578 */
	2147,	/* 8	0.148175 */
	2158,	/* 9	0.153583 */
	2168,	/* 10	0.158779 */
	2179,	/* 11	0.163742 */
	2188,	/* 12	0.168455 */
	2197,	/* 13	0.172897 */
	2206,	/* 14	0.177051 */
	2214,	/* 15	0.180902 */
	2221,	/* 16	0.184433 */
	2227,	/* 17	0.187631 */
	2233,	/* 18	0.190483 */
	2238,	/* 19	0.192978 */
	2243,	/* 20	0.195106 */
	2246,	/* 21	0.196858 */
	2249,	/* 22	0.198229 */
	2251,	/* 23	0.199211 */
	2252,	/* 24	0.199803 */
	2253,	/* 25	0.200000 */
	2252,	/* 26	0.199803 */
	2251,	/* 27	0.199211 */
	2249,	/* 28	0.198229 */
	2246,	/* 29	0.196858 */
	2243,	/* 30	0.195106 */
	2238,	/* 31	0.192978 */
	2233,	/* 32	0.190483 */
	2227,	/* 33	0.187631 */
	2221,	/* 34	0.184433 */
	2214,	/* 35	0.180902 */
	2206,	/* 36	0.177051 */
	2197,	/* 37	0.172897 */
	2188,	/* 38	0.168455 */
	2179,	/* 39	0.163742 */
	2168,	/* 40	0.158779 */
	2158,	/* 41	0.153583 */

2147,	/* 42	0.148175 */
2135,	/* 43	0.142578 */
2123,	/* 44	0.136812 */
2111,	/* 45	0.130902 */
2099,	/* 46	0.124869 */
2086,	/* 47	0.118738 */
2074,	/* 48	0.112533 */
2061,	/* 49	0.106279 */
2048,	/* 50	0.100000 */
2035,	/* 51	0.093721 */
2022,	/* 52	0.087467 */
2010,	/* 53	0.081262 */
1997,	/* 54	0.075131 */
1985,	/* 55	0.069098 */
1973,	/* 56	0.063188 */
1961,	/* 57	0.057422 */
1949,	/* 58	0.051825 */
1938,	/* 59	0.046417 */
1928,	/* 60	0.041221 */
1917,	/* 61	0.036258 */
1908,	/* 62	0.031545 */
1899,	/* 63	0.027103 */
1890,	/* 64	0.022949 */
1882,	/* 65	0.019098 */
1875,	/* 66	0.015567 */
1869,	/* 67	0.012369 */
1863,	/* 68	0.009517 */
1858,	/* 69	0.007022 */
1853,	/* 70	0.004894 */
1850,	/* 71	0.003142 */
1847,	/* 72	0.001771 */
1845,	/* 73	0.000789 */
1844,	/* 74	0.000197 */
1843,	/* 75	0.000000 */
1844,	/* 76	0.000197 */
1845,	/* 77	0.000789 */
1847,	/* 78	0.001771 */
1850,	/* 79	0.003142 */
1853,	/* 80	0.004894 */
1858,	/* 81	0.007022 */
1863,	/* 82	0.009517 */
1869,	/* 83	0.012369 */
1875,	/* 84	0.015567 */

```

1882,          /* 85    0.019098 */
1890,          /* 86    0.022949 */
1899,          /* 87    0.027103 */
1908,          /* 88    0.031545 */
1917,          /* 89    0.036258 */
1928,          /* 90    0.041221 */
1938,          /* 91    0.046417 */
1949,          /* 92    0.051825 */
1961,          /* 93    0.057422 */
1973,          /* 94    0.063188 */
1985,          /* 95    0.069098 */
1997,          /* 96    0.075131 */
2010,          /* 97    0.081262 */
2022,          /* 98    0.087467 */
2035,          /* 99    0.093721 */
2048           /* 100   0.100000 */

```

```
};
```

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

```
/* 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 */

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 A/D,D/A,OPAmp      */
/*-----*/
fn_AdcDaclnit();

/*-----*/
/* Initialization of DMA                  */
/*-----*/

```

```

fn_Dmalnit();

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 */
                /*||| : 00000 : AAAAAAAAAAAAAA */
                /*||| : 00001 : AAAAAAAAAAAAAA D */
                /*||| : 00010 : AAAAAAAAAAAAAA DD */
                /*||| : 00011 : AAAAAAAAAAAAAA DDD */
                /*||| : 00100 : AAAAAAAAAA DDDD */
                /*||| : 00101 : AAAAAAAAAA DDDDD */
                /*||| : 00110 : AAAAAA DDDDDDD */
                /*||| : 00111 : AAAAAA DDDDDDDDD */
                /*||| : 01000 : AAAA DDDDDDDDDDD */
                /*||| : 01001 : AA DDDDDDDDDDDDD */
                /*||| : 01010 : A DDDDDDDDDDDDDDD */
                /*||| : 01111 : A DDDDDDDDDDDDDDDDD */
                /*||| : 10000 : D DDDDDDDDDDDDDDDDD */
                /*||| */
                /*+++----- : 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                /* Set P33,P32 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 = 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 = 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_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 = 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 */

    /* 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 */

/* CH4:for DMA */
TMR04 = 0b1000000000000000;      /* Timer Mode Register 04 */
/*|||||||||+++++--- : [1]Operation mode of channel 4 */
/*|||||||||          [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are

```



```

unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 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 TI04 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 4 */
/*||||| : 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 */

TDR04 = 200; /* set interval time to 10us(=0.05us * 200) */
TMMK04 = 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) */
for(loop = 500; loop > 0; loop--){
    NOP();
}
/* wait until VLVI VDD */
while( LVIF ){
    NOP();
}

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

/*-----*/
/* Module: fn_Dmalnit */
/* Description: Setting of DMA contoroler */
/* parameter: -- */
/* return : -- */
/*-----*/
void fn_Dmalnit(void)
{

DEN0 = 1; /* Enables operation of DMA channel 0 */

DSA0 = (unsigned char)&DACSW0; /* DMA SFR Address Register

```

0 */

```

DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
DBC0 = 100;                                     /* DMA Byte Count Register 0 */
DMC0 = 0b01100100;                             /* DMA Mode Control Register 0 */
    /*||||| */
    /*|||+++++----- IFC3-IFC0 */
    /*|||          [Selection of DMA start source] */
    /*|||          0000:Disables DMA transfer by interrupt */
    /*|||          (Only software trigger is enabled.) */
    /*|||          0010:INTTM00 */
    /*|||          0011:INTTM01 */
    /*|||          0100:INTTM04 */
    /*|||          0101:INTTM05 */
    /*|||          0110:INTST0/INTCSI00 */
    /*|||          0111:INTSR0/INTCSI01 */
    /*|||          1000:INTST1/INTCSI10/INTIIC10 */
    /*|||          1001:INTSR1 */
    /*|||          1010:INTST3 */
    /*|||          1011:INTSR3 */
    /*|||          1100:INTAD */
    /*|||+----- DWAIT0 */
    /*|||          [Pending of DMA transfer] */
    /*|||          0:Executes DMA transfer upon DMA start request (not held pending).
*/

    /*|||          1:Holds DMA start request pending if any. */
    /*||+----- DS0 */
    /*||          [Specification of transfer data size for DMA transfer] */
    /*||          0:8bits */
    /*||          1:16bits */
    /*|+----- DRS0 */
    /*|          [Selection of DMA transfer direction] */
    /*|          0:SFR to internal RAM */
    /*|          1:Internal RAM to SFR */
    /*+----- STG0 */
    /*          [DMA transfer start software trigger] */
    /*          0:No trigger operation */
    /*          1:DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

DST0 = 1;                                       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */
}

```

*/

```

/*-----*/
/* 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:      fn_AdcDaclnit
                */
/* Description:  Initialization of A/D D/A OPamp module          */
/* parameter:   --
                */
/* return      :  --
                */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

    /*-----*/
    /* Initialization of Voltage reference          */
    /*-----*/

```

```

ADVRC = 0b00001000; /* Analog reference voltage control register */
/*| | | | + | + + + + : [1]Positive reference voltage supplies selection of A/D and D/A
converters */

/*| | | | | [2]Operationcontrol of voltage reference */
/*| | | | | [3]Output voltage selection of voltage reference */
/*| | | | | [4]Operation control of input gate voltage boost circuit for A/D
converter */

/*| | | | | [5]Relationship with the conversion mode used */
/*| | | | | */
/*| | | | | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*| | | | | [2]Stops operation (Hi-Z) */
/*| | | | | [3]2.5 V */
/*| | | | | [4]Stops operation */
/*| | | | | [5]Can be set in conversion mode 1 */
/*| | | | | */
/*| | | | | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*| | | | | [2]Stops operation (Hi-Z) */
/*| | | | | [3]2.0 V */
/*| | | | | [4]Enables operation */
/*| | | | | [5]Can be set in conversion mode 2 or 3 */
/*| | | | | */
/*| | | | | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*| | | | | [2]Stops operation (pull-down output) */
/*| | | | | [3]2.5 V */
/*| | | | | [4]Stops operation */
/*| | | | | [5] - */
/*| | | | | */
/*| | | | | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*| | | | | [2]Enables operation */
/*| | | | | [3]2.5 V */
/*| | | | | [4]Enables operation */
/*| | | | | [5]Can be set in conversion mode 2 or 3 */
/*| | | | | */
/*| | | | | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*| | | | | [2]Stops operation (pull-down output) */
/*| | | | | [3]2.0 V */
/*| | | | | [4]Enables operation */
/*| | | | | [5] - */
/*| | | | | */
/*| | | | | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*| | | | | [2]Enables operation */
/*| | | | | [3]2.0 V */
/*| | | | | [4]Enables operation */

```

```

/*||| |          [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;          /* Enables operation */
ADVRC.1 = 1;          /* Output 2.0 V */

ADPC = 0b00000000;   /* 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 A A */
/*||| : 0 0 0 0 1 : A 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 A D D */
/*||| : 0 0 0 1 1 : A 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 */
/*||| : 0 0 1 1 0 : A A A A A A A D D D D D */
/*||| : 0 0 1 1 1 : A 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 D */
/*||| : 0 1 0 0 1 : A A A 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 */
/*||| : 0 1 1 1 1 : A 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 */
/*||| */
/*++++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-----*/
/* Initialization of operational amplifier */
/*-----*/

PM2.1 = 1;          /* input port mode (AMP00) */
PM2.0 = 1;          /* input port mode (AMP0-) */
PM2.2 = 1;          /* input port mode (AMP0+) */

```

```

PM2.4 = 1;          /* input port mode (AMP1O) */
PM2.3 = 1;          /* input port mode (AMP1-) */
PM2.5 = 1;          /* input port mode (AMP1+) */

PM2.7 = 1;          /* input port mode (AMP2O) */
PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;        /* input port mode (AMP2+) */

PM11.0 = 1;        /* input port mode (ANO0) */

ADCEN = 1;          /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;          /* operational amplifier (AMP0) enable */
OAC.1 = 0;          /* operational amplifier (AMP1) disable */
OAC.2 = 0;          /* operational amplifier (AMP2) disable */

/*-----*/
/*      Initialization of D/A converter      */
/*-----*/
DACEN = 1;          /* supplies input clock to D/A converter */

/*      DAMD0 = 1;          /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.0 = 0;          /* [DAMD0] D/A converter operation is Normal
mode */
DAM.2 = 1;          /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1;          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99]; /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0;          /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1;          /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[99]; /* D/A conversion value setting reigster 1 */

}

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

```



```

/*      parameter: --
                                     */
/*      return   : --
                                     */
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */

    /* Port setting */
    P12.0 = 0;
    PM12.0 = 1;
    P7.3 = 0;
    PM7.3 = 0;

    /* INTP0 pin valid edge selection */
    EGP0.0 = 1;                       /* Both rising and falling edges */
    EGN0.0 = 1;

    DACE1 = 1;                         /* D/A converter CH1 enable */
    DACE0 = 1;                         /* D/A converter CH0 enable */

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

    /* initialize D/Aconverter data */
    for(count = 0; count<100; count++){
        ushDacData[count] = aushDacData[count];
    }

    TS0L.4 = 1;                       /* start TAU0 CH4 */
    TMIF04 = 0;                       /* reset INTTM04 interrupt request */

    DMAMK0 = 0;                       /* enable DMA CH0 interrupt */

    PIF0 = 0;                         /* reset INTP0 interrupt request */
    PMK0 = 0;                         /* enable INTP0 interrupt */

    EI();                             /* enable all interrupts */

```

```

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

while (1){
    HALT();
    NOP();
}
}

/*-----*/
/* Module:      fn_intdma0
*/
/* Description:  DMA ch0 interrupt process
*/
/* parameter: --
*/
/* return  : --
*/
/*-----*/
__interrupt void fn_intdma0(void)
{
    DRA0 = (unsigned short)&ushDacData[0];    /* DMA RAM Address Register 0 */
    DBC0 = 100;                               /* DMA Byte Count Register 0 */

    DST0 = 1;                                /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */
}

/*-----*/
/* Module:      fn_intp0
*/
/* Description:  INTP0 interrupt process
*/
/* parameter: --
*/
/* return  : --
*/
/*-----*/
__interrupt void fn_intp0(void)
{
    P7.3 = P12.0;
}

```

}

(9) CR発振回路1

```

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_InitVr(void);                /* Setting of
Voltage reference */
                void    fn_AdcDaclnit(void);            /* Setting of D/A converter */

extern void    fn_DisplayInit(void);                    /* Setting of LCD driver */
extern void    fn_Display(unsigned short ushCount); /* LCD display */

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

/*-----*/
/* Local constants
                */
/*-----*/
/*-----*/
/* Global variables
                */
/*-----*/

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

/*-----*/
/* 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 A/D,D/A,OPAMP */
/*-----*/
fn_AdcDaclnit();

/*-----*/
/* 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 A A */

```

```

/*|||:00001:AAAAAAAAAAD*/
/*|||:00010:AAAAAAAAAAD*/
/*|||:00011:AAAAAAAAADD*/
/*|||:00100:AAAAAAAAADD*/
/*|||:00101:AAAAAAAAADD*/
/*|||:00110:AAAAAADDDDD*/
/*|||:00111:AAAAADDDDDD*/
/*|||:01000:AAAADDDDDDDD*/
/*|||:01001:AAADDDDDDDDDD*/
/*|||:01010:AADDDDDDDDDDDD*/
/*|||:01111:ADDDDDDDDDDDD*/
/*|||:10000:DDDDDDDDDDDD*/
/*||| */
/*+++-----: 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 =    0b00001100;    /* Set P30-P31,P34 Output latch to Low */
                                /* Set P33,P32 Output latch High */

PM3 =   0b11110000;    /* Set P30-P33 to output port */
                                /* Set P34(TI06) to input port */
                                /* P30-P33: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 =   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 = 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_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 = 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 */

```

/* 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 */

/* CH4:for capturing interval */
TMR04 = 0b1000000000000000; /* Timer Mode Register 04 */
    /*|||||||||+++++--- : [1]Operation mode of channel 4 */
    /*|||||||||          [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 TI04 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 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are
unselected). */

/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1
used as both the start trigger and capture trigger. */

/*||||| : 0 1 0 : Both the edges of TI04 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 4 */
/*||||| : 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 */

```

```

TDR04 = 20000; /* set interval time to 1ms(=0.05us * 20000) */
TMMK04 = 1; /* disable interrupt */

/* CH6:for TI06 */
TMR06 = 0b1001000101000110; /* 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 6 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 = 0xffff; /* set initial count */
        TMMK06 = 1; /* disable interrupt */
        TIS0.6 = 0; /* select TI06 */

    }

/*-----*/
/* 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) */
for(loop = 500; loop > 0; loop--){
    NOP();
}
/* wait until VLVI VDD */
while( LVIF ){
    NOP();
}

LVIIF = 0;          /* clear LVI 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:          fn_AdcDaclnit
                   */
/* Description:     Initialization of A/D D/A OPamp module          */
/* parameter:      --
                   */
/* return   :      --
                   */
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1;          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

/*-----*/
/* Initialization of Voltage reference          */
/*-----*/
/* ADVRC = 0b00001000;    /* Analog reference voltage control register */
/* |||||+|+---- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
/* |||| |           [2]Operationcontrol of voltage reference */
/* |||| |           [3]Output voltage selection of voltage reference */
/* |||| |           [4]Operation control of input gate voltage boost circuit for A/D
converter */
/* |||| |           [5]Relationship with the conversion mode used */
/* |||| | */
/* |||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/* |||| |           [2]Stops operation (Hi-Z) */
/* |||| |           [3]2.5 V */
/* |||| |           [4]Stops operation */
/* |||| |           [5]Can be set in conversion mode 1 */
/* |||| | */
/* |||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/* |||| |           [2]Stops operation (Hi-Z) */
/* |||| |           [3]2.0 V */

```



```

/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.5 V */
/*||| | [4]Stops operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Enables operation */
/*||| | [3]2.5 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Stops operation (pull-down output) */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5] - */
/*||| | */
/*||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*||| | [2]Enables operation */
/*||| | [3]2.0 V */
/*||| | [4]Enables operation */
/*||| | [5]Can be set in conversion mode 2 or 3 */
/*||| | */
/*||| | : Other than the above : Setting prohibited */
/*||| | */
/*|+++++----- : Be sure to set 0000 */
/*| */
/*+----- : Reference voltage supply (negative side) of A/D converter selection */
/* : 0 : AVSS */
/* : 1 : AVREFM (external voltage reference input) */

/* ADVRC.0 = 1; /* Enables operation */
/* ADVRC.1 = 1; /* Output 2.0 V */

ADPC = 0b00000000; /* 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 */

```

```

/*|||:00000:AAAAAAAAAAAA */
/*|||:00001:AAAAAAAAAAAAAD */
/*|||:00010:AAAAAAAAAAAAADDD */
/*|||:00011:AAAAAAAAAAAAADDDD */
/*|||:00100:AAAAAAAAAADDDDD */
/*|||:00101:AAAAAAAAAADDDDD */
/*|||:00110:AAAAAAAAAADDDDD */
/*|||:00111:AAAAAADDDDDDD */
/*|||:01000:AAAADDDDDDDDD */
/*|||:01001:AAADDDDDDDDDDD */
/*|||:01010:AADDDDDDDDDDDDD */
/*|||:01111:ADDDDDDDDDDDDD */
/*|||:10000:DDDDDDDDDDDDDD */
/*||| */
/*+++----- : Be sure to set 000 */

/* wait for settling time to 17msec */
/* fn_Wait500usBase(17000/500); */

/*-----*/
/* Initialization of operational amplifier */
/*-----*/
PM2.1 = 1; /* input port mode (AMP0O) */
PM2.0 = 1; /* input port mode (AMP0-) */
PM2.2 = 1; /* input port mode (AMP0+) */

PM2.4 = 1; /* input port mode (AMP1O) */
PM2.3 = 1; /* input port mode (AMP1-) */
PM2.5 = 1; /* input port mode (AMP1+) */

PM2.7 = 1; /* input port mode (AMP2O) */
PM2.6 = 1; /* input port mode (AMP2-) */
PM15.0 = 1; /* input port mode (AMP2+) */

PM11.0 = 1; /* input port mode (ANO0) */

ADCEN = 1; /* supplies input clock to OPamp A/D converter */
OAC.0 = 1; /* operational amplifier (AMP0) enable */
OAC.1 = 0; /* operational amplifier (AMP1) disable */
OAC.2 = 0; /* operational amplifier (AMP2) disable */

/*-----*/
/* Initialization of D/A converter */

```

```

/*-----*/
DACEN = 1;                                     /* supplies input clock to D/A converter */

/* set D/A chanel1 */
DAM.1 = 0;                                     /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1;                                     /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = 2048;                                /* D/A conversion value setting reigster 1 */
                                             /* ANO1 = 1.5V (1/2 * 2^12) */

}

/*-----*/
/* Module:      main
                */
/* Description:  Main process
                */
/* parameter:  --
                */
/* return   :  --
                */
/*-----*/
void main(void)
{
    unsigned short    ushCount; /* Pulse counter */

    DACE1 = 1;                                     /* D/A converter CH1 enable */
    ushCount = 0;

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

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

    while (1){
        TSOL.6 = 1;                                /* start TAU0 CH6 (pulse counter) */
        TSOL.4 = 1;                                /* start TAU0 CH4 (interval timer) */
        TMIF04 = 0;                                /* clear interval timer request */

```

```

        while(TMIF04 == 0){          /* wait interval */
            NOP();
        }

        ushCount = TCR06;           /* get pulse count */

        TT0L.6 = 1;                 /* stop TAU0 CH6 (pulse counter)*/
        TT0L.4 = 1;                 /* stop TAU0 CH4 (interval timer) */

        ushCount = 0xffff - ushCount; /* get pulse count */
        fn_Display(ushCount);        /* display pulse count */
    }
}

```

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_ &SEG5 /* tenths place of */
/*#define CLCDPOS_ &SEG10 /* ones place of */
/*#define CLCDPOS_ &SEG18 /* tenths place of */
/*#define CLCDPOS_ &SEG23 /* ones place of */
/*#define CLCDPOS_ &SEG31 /* tenths place of */
/*#define CLCDPOS_ &SEG36 /* ones place of */
/*#define CLCDPOS_ &SEG44 /* tenths place of */
/*#define CLCDPOS_ &SEG49 /* ones place of */

/*#define CLCDPOS_REF &SEG5 /* sign of REF */
/*#define CLCDPOS_REF100 &SEG12 /* hundreds place of REF */
/*#define CLCDPOS_REF10 &SEG17 /* tenths place of REF */
/*#define CLCDPOS_REF1 &SEG22 /* ones place of REF */
/*#define CLCDPOS_TH &SEG32 /* sign of TH */
/*#define CLCDPOS_TH100 &SEG39 /* hundreds place of TH */
/*#define CLCDPOS_TH10 &SEG44 /* tenths place of TH */
/*#define CLCDPOS_TH1 &SEG49 /* ones place of TH */

#define CLCDPOS_REF1000 &SEG5 /* thousands place of REF */
#define CLCDPOS_REF100 &SEG10 /* hundreds place of REF */

```

```
#define CLCDPOS_REF10 &SEG15 /* tenths place of REF */
#define CLCDPOS_REF1 &SEG20 /* ones place of REF */
#define CLCDPOS_1000 &SEG34 /* thousands place of TH */
#define CLCDPOS_100 &SEG39 /* hundreds place of TH */
#define CLCDPOS_10 &SEG44 /* tenths place of TH */
#define CLCDPOS_1 &SEG49 /* ones place of TH */
```

/* Display data */

/*=====

< LCD PANEL >

```

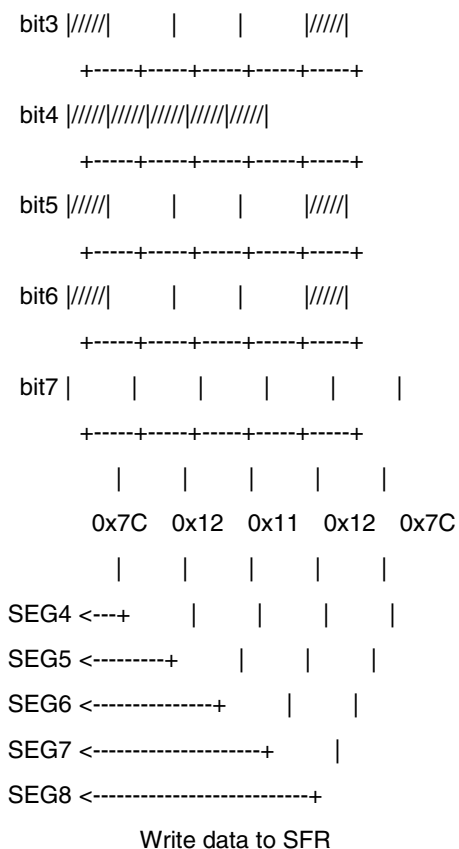
          SEG4  SEG5  SEG6  SEG7  SEG8  = = =  SEG49  SEG50  SEG51  SEG52
SEG53
+-----+-----+-----+-----+-----+  = = =  +-----+-----+-----+-----+
COM0 |      |      |      |      |      |      |      |      |      |      |      |
+-----+-----+-----+-----+-----+  = = =  +-----+-----+-----+-----+
COM1 |      |      |      |      |      |      |      |      |      |      |      |
+-----+-----+-----+-----+-----+  = = =  +-----+-----+-----+-----+
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 |////|      |      |      |      |////|      |      |      |      |
+-----+-----+-----+-----+-----+
```



```

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

```
, {0b01111100          /* '4' */  
,0b01000100  
,0b11111110  
,0b01000000  
,0b00000000}
```

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

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

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

```
, {0b01111110          /* '6' */  
,0b10010010  
,0b10010010  
,0b01100000  
,0b00000000}
```

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

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

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

```
, {0b01101100          /* '8' */  
,0b10010010  
,0b10010010  
,0b01101100
```



```
    ,0b00000000}

/* COM 76543210 */
,    {0b00001100          /* '9' */
    ,0b10010010
    ,0b01010010
    ,0b00111100
    ,0b00000000}
};

/*-----*/
/*    sign of Ref    */
/*-----*/
static const unsigned char aDispRef[] = {
/* COM 76543210 */
    0b11111110          /* R */
    ,0b00010010
    ,0b00110010
    ,0b01010010
    ,0b10001100
};

/*-----*/
/*    sign of Th    */
/*-----*/
static const unsigned char aDispTh[] = {
    0b00000010          /* T */
    ,0b00000010
    ,0b11111110
    ,0b00000010
    ,0b00000010
};

/*-----*/
/*    Button frame    */
/*-----*/
static const unsigned char aDispButtonFrame[] = {
/* COM 76543210 */
    0b00000000
    ,0b01111111          /* */
    ,0b01001001
    ,0b01001001
```

```
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01111111
,0b00000000
,0b01111111      /* */
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,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
,0b01001001
,0b01001001
```



```

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

/*-----*/
/* Global variables
*/
/*-----*/
/*boolean          bDispUpdate;      /* Display update request */
/*boolean          bBlinkOn;         /* Blink status (0:LCD OFF, 1:LCD ON) */

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

/*-----*/
/* Code
*/
/*-----*/
/*=====*/
/*
*/
/*          Common function
*/
/*
*/
/*=====*/
/*-----*/
/* Module:          fn_DisplayInit
*/
/*
*/
/* Description:     Initialization of Display module
*/
/* parameter: --
*/

```

```

/*      return      : --
                                     */

/*-----*/
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 T107 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 = 0b00110011;          /* 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 count to LCD
                                                        */
/*   parameter: Count data(Hex)
                */
/*   return   : --
                                */
/*-----*/
void fn_Display(unsigned short ushCount)
{
    unsigned short    ushNumber;                /* number work for display */
    /* space */
    /* fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_REF - CLCDPOS_START));
    */
    fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_1000 - CLCDPOS_START));

    /*-----*/
    /* display Count data */
    /*-----*/
    /* display sign of TH */
    /* fn_LcdWrite(CLCDPOS_TH, aDispTh, sizeof aDispTh);
    */
    /* space */
    /* fn_LcdWrite((CLCDPOS_TH100 - 2), aDispClear, 2);
```

```
*/

/* display thousands place of Count */
fn_LcdWrite      (CLCDPOS_1000,
                  aDispNumber[(unsigned char)(ushCount / 1000)],
                  CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH100 - 1), aDispClear, 1);
*/

/* display hundreds place of TH */
ushCount %= 1000;
fn_LcdWrite      (CLCDPOS_100,
                  aDispNumber[(unsigned char)(ushCount / 100)],
                  CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH10 - 1), aDispClear, 1);
*/

/* display tenths place of TH */
ushCount %= 100;
fn_LcdWrite      (CLCDPOS_10,
                  aDispNumber[(unsigned char)(ushCount / 10)],
                  CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH1 - 1), aDispClear, 1);
*/

/* display ones place of TH */
fn_LcdWrite      (CLCDPOS_1,
                  aDispNumber[(unsigned char)(ushCount % 10)],
                  CLCDSIZE_NUMBER);

/* space */
/*
fn_LcdWrite((CLCDPOS_TH1 + CLCDSIZE_NUMBER), aDispClear, 1);
*/
}
```

(10) CR発振回路2

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

/* #pragma interrupt INTDMA0 fn_intdma0 RB1 */

/*-----*/
/* 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_InitVr(void);          /* Setting of
Voltage reference */
        void    fn_AdcDaclnit(void);      /* Setting of D/A converter */

extern void    fn_DisplayInit(void);      /* Setting of LCD driver */
extern void    fn_Display(unsigned short ushRef, unsigned short ushTh); /* LCD display */

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

/*-----*/
/* Local constants
*/
/*-----*/
/* Global variables
*/
/*-----*/

/*-----*/
/* Local variables
*/
/*-----*/
static short   shVoltage; /* voltage (Hex) */
static unsigned char ucMeasureStatus; /* measuring status */

static unsigned short ushR_REF1; /* R_REF at 1V */
static unsigned short ushR_REF2; /* R_REF at 2V */
static unsigned short ushR_TH1; /* R_TH at 1V */
static unsigned short ushR_TH2; /* R_TH at 2V */

/*-----*/
/* 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 A/D,D/A,OPAmp      */
/*-----*/
fn_AdcDaclnit();

/*-----*/
/*  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 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 A D D D D */
                                /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                                /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                                /*||| : 0 0 1 1 1 : 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 */
                                /*||| : 0 1 0 0 1 : A A A 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 */
                                /*||| : 0 1 1 1 1 : A 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 */
                                /*||| */
                                /*+++----- : 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 = 0b00001100; /* Set P30-P31,P34 Output latch to Low */
/* Set P33,P32 Output latch High */
PM3 = 0b11110000; /* Set P30-P33 to output port */
/* Set P34(TI06) to input port */
/* P30-P33: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 = 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 = 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 */

```

```

/* 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
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 TI06 */
TMR06 = 0b1000000101000100; /* 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 */
                                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) */
for(loop = 500; loop > 0; loop--){

```

```

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

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

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

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

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

/*-----*/
/* Module:      fn_AdcDaclnit
   */
/* Description:  Initialization of A/D D/A OPamp module          */
/*   parameter: --
   */
/*   return    : --
   */

```

```

*/
/*-----*/
void fn_AdcDaclnit(void)
{
    ADCEN = 1; /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

    ADPC = 0b00000000; /* 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 */
    /*||| : 00000 : A A A A A A A A A A A A */
    /*||| : 00001 : A A A A A A A A A A A D */
    /*||| : 00010 : A A A A A A A A A A D D */
    /*||| : 00011 : A A A A A A A A A D D D */
    /*||| : 00100 : A A A A A A A A D D D D */
    /*||| : 00101 : A A A A A A A D D D D D */
    /*||| : 00110 : A A A A A A D D D D D D */
    /*||| : 00111 : A A A A A D D D D D D D */
    /*||| : 01000 : A A A A D D D D D D D D */
    /*||| : 01001 : A A A D D D D D D D D D */
    /*||| : 01010 : A A D D D D D D D D D D */
    /*||| : 01111 : A D D D D D D D D D D D */
    /*||| : 10000 : D D D D D D D D D D D D */
    /*||| */
    /*+++----- : Be sure to set 000 */

    /* wait for settling time to 17msec */
    /* fn_Wait500usBase(17000/500); */

    /*-----*/
    /* Initialization of operational amplifier */
    /*-----*/
    PM2.1 = 1; /* input port mode (AMP0O) */
    PM2.0 = 1; /* input port mode (AMP0-) */
    PM2.2 = 1; /* input port mode (AMP0+) */

    PM2.4 = 1; /* input port mode (AMP1O) */
    PM2.3 = 1; /* input port mode (AMP1-) */
    PM2.5 = 1; /* input port mode (AMP1+) */

    PM2.7 = 1; /* input port mode (AMP2O) */

```

```

PM2.6 = 1;          /* input port mode (AMP2-) */
PM15.0 = 1;        /* input port mode (AMP2+) */

PM11.0 = 1;        /* input port mode (ANO0) */

ADCEN = 1;         /* supplies input clock to OPamp A/D converter */
OAC.0 = 1;         /* operational amplifier (AMP0) enable */
OAC.1 = 0;         /* operational amplifier (AMP1) disable */
OAC.2 = 0;         /* operational amplifier (AMP2) disable */

/*-----*/
/*      Initialization of D/A converter      */
/*-----*/
DACEN = 1;         /* supplies input clock to D/A converter */

/* set D/A chanel1 */
DAM.1 = 0;         /* [DAMD1] D/A converter operation is Normal
mode */
DAM.3 = 1;         /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = 1365;    /* D/A conversion value setting reigster 1 */
                    /* ANO1 = 1V (1/3 * 2^12) */

}

/*-----*/
/* Module:      main
*/
/* Description:  Main process
*/
/* parameter:  --
*/
/* return   :  --
*/
/*-----*/
void main(void)
{
    unsigned short    count;          /* counter for initializing D/A convert data */
    unsigned short    *pMeasure;     /* pointer for measuring data */

    DACE1 = 1;                       /* D/A converter CH1 enable */

    ucMeasureStatus = 0;              /* measuring status */

```

```

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

PM7.1 = 1;          /* set T_REF HiZ(disconnect) */
PM7.2 = 1;          /* set T_TH HiZ(disconnect) */

fn_Display(0, 0);  /* initial display */

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

while (1){

    switch(ucMeasureStatus){          /* measuring status */
        case 0:                        /* T_REF1 */
            DACSW1 = 1365;             /* D/A conversion
value setting reigster 1 */

            /* ANO1 = 1V (1/3 * 2^12) */

            P7.0 = 1;
            PM7.0 = 0;                 /* charge
CAPACITOR */

            NOP();
            NOP();
            NOP();
            NOP();
            NOP();
            NOP();                     /* wait
charge complete */

            PM7.0 = 1;                 /* end
charge */

            P7.1 = 0;
            PM7.1 = 0;                 /* set
T_REF low(start discharge) */

            TS0L.6 = 1;                /* start

```

```

TIMER06 */
                                TMIF06 = 0;                                /* clear
measure end */
                                while(TMIF06 == 0);                    /* wait measure
end */
                                while(P3.4 == 0);                        /* wait measure
end */
                                ushR_REF1 = TDR06;                      /* get measuring
data */
                                ushR_REF1 = TCR06;                      /* get measuring
data */
                                TT0L.6 = 1;                            /* stop
TAU0 CH6 */
                                PM7.1 = 1;                            /* set
T_REF HiZ(stop discharge) */
                                ucMeasureStatus = 1;                /* set T_REF2 measure */
                                break;
                                case 1:                                /* T_REF2 */
                                DACSW1 = 2730;                        /* D/A conversion
value setting register 1 */
                                /* ANO1 = 2V (2/3 * 2^12) */
                                P7.0 = 1;
                                PM7.0 = 0;                            /* charge
CAPACITOR */
                                NOP();
                                NOP();
                                NOP();
                                NOP();
                                NOP();
                                NOP();                                /* wait
charge complete */
                                PM7.0 = 1;                            /* end
charge */
                                P7.1 = 0;
                                PM7.1 = 0;                            /* set
T_REF low(start discharge) */
                                TS0L.6 = 1;                            /* start
TIMER06 */

```



```

TMIF06 = 0; /* clear
measure end */

while(TMIF06 == 0); /* wait measure end */
ushR_REF2 = TDR06; /* get measuring
data */

TT0L.6 = 1; /* stop
TAU0 CH6 */

PM7.1 = 1; /* set
T_REF HiZ(stop discharge) */

ucMeasureStatus = 2; /* set T_TH1 measure */
break;

case 2: /* T_TH1 */
DACSW1 = 1365; /* D/A conversion
value setting reigster 1 */

/* ANO1 = 1V (1/3 * 2^12) */

P7.0 = 1;
PM7.0 = 0; /* charge
CAPACITOR */

NOP();
NOP();
NOP();
NOP();
NOP();
NOP(); /* wait
charge complete */

PM7.0 = 1; /* end
charge */

P7.2 = 0;
PM7.2 = 0; /* set
T_TH low(start discharge) */

TS0L.6 = 1; /* start
TIMER06 */

TMIF06 = 0; /* clear
measure end */

while(TMIF06 == 0); /* wait measure end */
ushR_TH1 = TDR06; /* get measuring data */
TT0L.6 = 1; /* stop
TAU0 CH6 */

```

```

PM7.2 = 1; /* set
T_TH HiZ(stop discharge) */

ucMeasureStatus = 3; /* set T_TH2 measure */
break;

case 3: /* T_TH2 */
DACSW1 = 2730; /* D/A conversion
value setting register 1 */

/* ANO1 = 2V (2/3 * 2^12) */

P7.0 = 1;
PM7.0 = 0; /* charge
CAPACITOR */

NOP();
NOP();
NOP();
NOP();
NOP();
NOP(); /* wait
charge complete */

PM7.0 = 1; /* end
charge */

P7.2 = 0;
PM7.2 = 0; /* set
T_TH low(start discharge) */

TS0L.6 = 1; /* start
TIMER06 */

TMIF06 = 0; /* clear
measure end */

while(TMIF06 == 0); /* wait measure end */
ushR_TH2 = TDR06; /* get measuring data */
TT0L.6 = 1; /* stop
TAU0 CH6 */

PM7.2 = 1; /* set
T_TH HiZ(stop discharge) */

fn_Display((ushR_REF1 - ushR_REF2), (ushR_TH1 -
ushR_TH2));

/* display T_TH,R_REF data to LCD */

```

```

ucMeasureStatus = 0;          /* set T_REF1 measure */

    }

}

```

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_      &SEG5          /* tenths place of */
/*#define      CLCDPOS_      &SEG10         /* ones place of */
/*#define      CLCDPOS_      &SEG18        /* tenths place of */
/*#define      CLCDPOS_      &SEG23        /* ones place of */
/*#define      CLCDPOS_      &SEG31        /* tenths place of */
/*#define      CLCDPOS_      &SEG36        /* ones place of */
/*#define      CLCDPOS_      &SEG44        /* tenths place of */
/*#define      CLCDPOS_      &SEG49        /* ones place of */

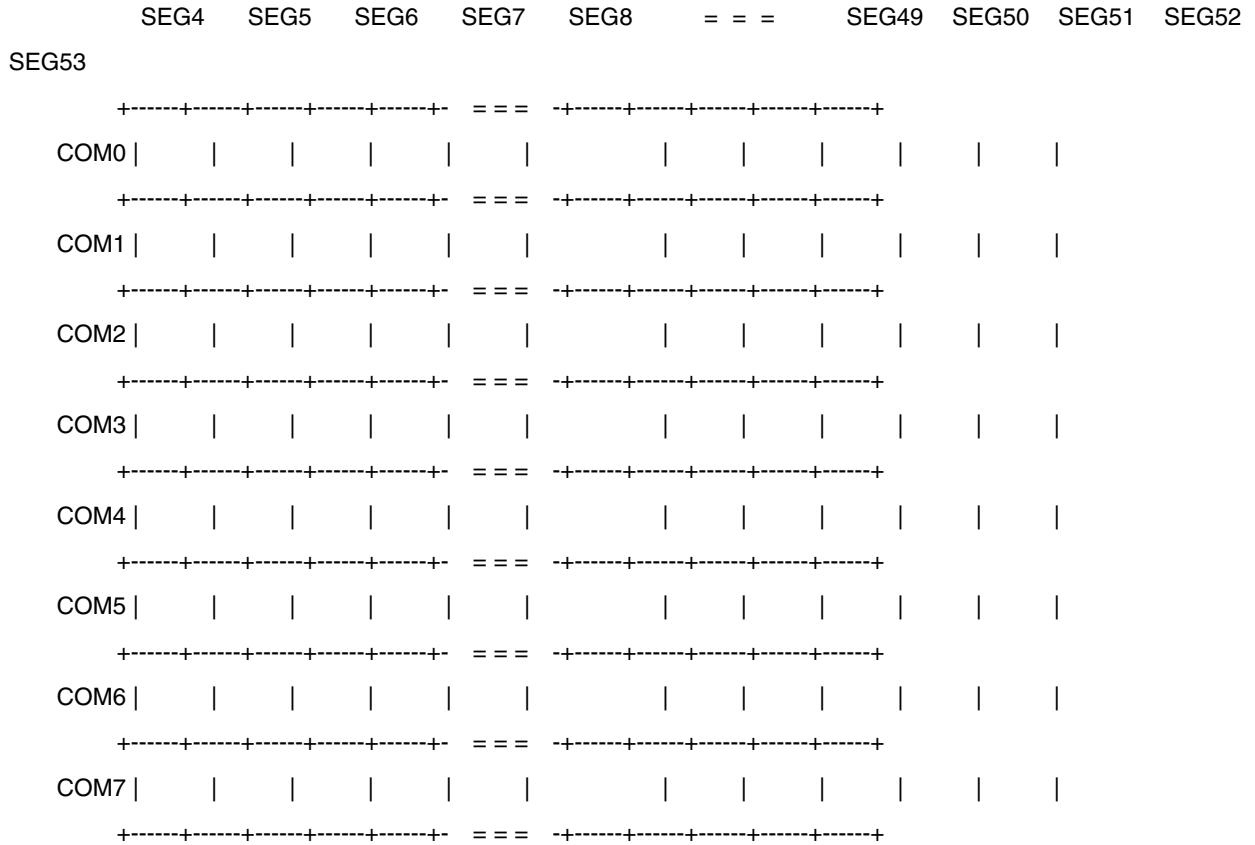
/*#define      CLCDPOS_REF      &SEG5    /* sign of REF */
/*#define      CLCDPOS_REF100 &SEG12 /* hundreds place of REF */
/*#define      CLCDPOS_REF10 &SEG17 /* tenths place of REF */
/*#define      CLCDPOS_REF1  &SEG22 /* ones place of REF */
/*#define      CLCDPOS_TH      &SEG32 /* sign of TH */
/*#define      CLCDPOS_TH100 &SEG39 /* hundreds place of TH */
/*#define      CLCDPOS_TH10  &SEG44 /* tenths place of TH */
/*#define      CLCDPOS_TH1      &SEG49 /* ones place of TH */

#define CLCDPOS_REF1000      &SEG5    /* thousands place of REF */
#define CLCDPOS_REF100 &SEG10 /* hundreds place of REF */
#define CLCDPOS_REF10  &SEG15 /* tenths place of REF */
#define CLCDPOS_REF1   &SEG20 /* ones place of REF */
#define CLCDPOS_TH1000 &SEG34 /* thousands place of TH */
#define CLCDPOS_TH100  &SEG39 /* hundreds place of TH */
#define CLCDPOS_TH10   &SEG44 /* tenths place of TH */
#define CLCDPOS_TH1    &SEG49 /* ones place of TH */
```

/* Display data */

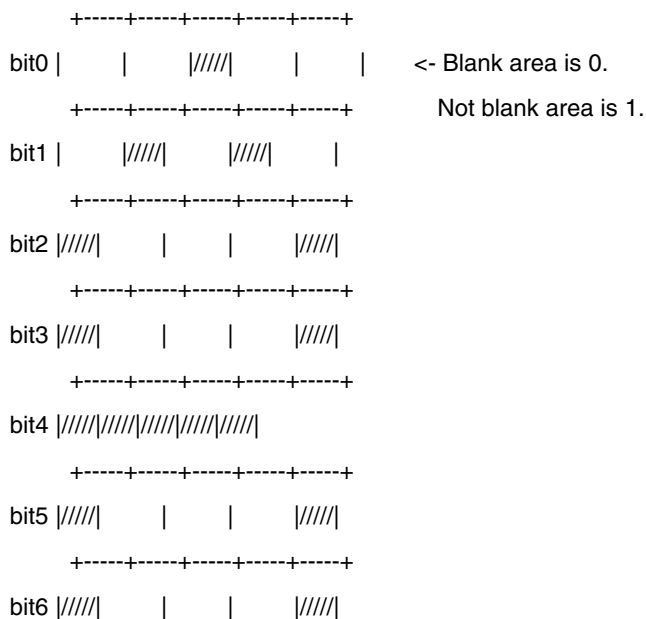
/*=====

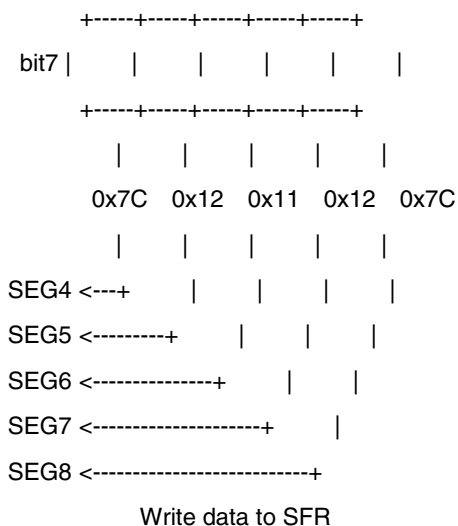
< LCD PANEL >



< example of the data setting >

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





```

=====*/
/*-----*/
/*  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 */
, {0b01101100          /* '8' */
,0b10010010
,0b10010010
,0b01101100
,0b00000000}

/* COM 76543210 */
, {0b00001100          /* '9' */
,0b10010010
,0b01010010
,0b00111100

```

```
        ,0b00000000}
};

/*-----*/
/*   sign of Ref   */
/*-----*/
static const unsigned char aDispRef[] = {
/* COM 76543210 */
        0b11111110          /* R */
        ,0b00010010
        ,0b00110010
        ,0b01010010
        ,0b10001100
};

/*-----*/
/*   sign of Th   */
/*-----*/
static const unsigned char aDispTh[] = {
        0b00000010          /* T */
        ,0b00000010
        ,0b11111110
        ,0b00000010
        ,0b00000010
};

/*-----*/
/*   Button frame */
/*-----*/
static const unsigned char aDispButtonFrame[] = {
/* COM 76543210 */
        0b00000000
        ,0b01111111          /* */
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,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
,0b01111111      /* */
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01001001
,0b01111111
,0b00000000
,0b00000000

```

};

```

static const unsigned char aDispClear[] = {
/* COM 76543210 */

```

0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000
,0b00000000 /*'*/
,0b00000000
,0b00000000
,0b00000000

```

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

/*-----*/
/* Global variables
*/
/*-----*/
/*boolean          bDispUpdate;      /* Display update request */
/*boolean          bBlinkOn;         /* Blink status (0:LCD OFF, 1:LCD ON) */

/*-----*/
/* 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 = 0b00110011;           /* 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 count to LCD                                     */
/*      parameter: REF data(Hex), TH data(Hex)                             */
/*      return      : --
                                     */
/*-----*/
void fn_Display(unsigned short ushRef, unsigned short ushTh)
{
    unsigned short    ushNumber;          /* number work for display */
    /* space */
    /*      fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_REF - CLCDPOS_START));
    */
    fn_LcdWrite(CLCDPOS_START,    aDispClear,    (unsigned    char)(CLCDPOS_REF1000    -
CLCDPOS_START));

    /*-----*/
    /*      display REF data      */
    /*-----*/
    /* display sign of REF */
    /*      fn_LcdWrite(CLCDPOS_REF, aDispRef, sizeof aDispRef);
    */

    /* space */
    /*      fn_LcdWrite((CLCDPOS_REF100 - 2), aDispClear, 2);
    */

    /* display thousands place of REF */
    fn_LcdWrite(CLCDPOS_REF1000,
                aDispNumber[(unsigned char)(ushRef / 1000)],
                CLCDSIZE_NUMBER);

    /* clear digit gap */

```



```
/*      fn_LcdWrite((CLCDPOS_REF100 - 1), aDispClear, 1);
*/

/* display hundreds place of REF */
ushRef %= 1000;
fn_LcdWrite(CLCDPOS_REF100,
            aDispNumber[(unsigned char)(ushRef / 100)],
            CLCDSIZE_NUMBER);

/* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_REF10 - 1), aDispClear, 1);
*/

/* display tenths place of REF */
ushRef %= 100;
fn_LcdWrite(CLCDPOS_REF10,
            aDispNumber[(unsigned char)(ushRef / 10)],
            CLCDSIZE_NUMBER);

/* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_REF1 - 1), aDispClear, 1);
*/

/* display ones place of REF */
fn_LcdWrite(CLCDPOS_REF1,
            aDispNumber[(unsigned char)(ushRef % 10)],
            CLCDSIZE_NUMBER);

/* space */
/*      fn_LcdWrite((CLCDPOS_REF1 + CLCDSIZE_NUMBER), aDispClear, 6);*/
fn_LcdWrite((CLCDPOS_REF1 + CLCDSIZE_NUMBER), aDispClear, 5);

/*****/
/*      display TH data      */
/*****/
/* display sign of TH */
/*      fn_LcdWrite(CLCDPOS_TH, aDispTh, sizeof aDispTh);
*/

/* space */
/*      fn_LcdWrite((CLCDPOS_TH100 - 2), aDispClear, 2);
*/

/* display thousands place of TH */
```

```
fn_LcdWrite(CLCDPOS_TH1000,
            aDispNumber[(unsigned char)(ushTh / 1000)],
            CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH100 - 1), aDispClear, 1);
*/

/* display hundreds place of TH */
ushTh %= 1000;
fn_LcdWrite(CLCDPOS_TH100,
            aDispNumber[(unsigned char)(ushTh / 100)],
            CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH10 - 1), aDispClear, 1);
*/

/* display tenths place of TH */
ushTh %= 100;
fn_LcdWrite(CLCDPOS_TH10,
            aDispNumber[(unsigned char)(ushTh / 10)],
            CLCDSIZE_NUMBER);

/* clear digit gap */
/*
fn_LcdWrite((CLCDPOS_TH1 - 1), aDispClear, 1);
*/

/* display ones place of TH */
fn_LcdWrite(CLCDPOS_TH1,
            aDispNumber[(unsigned char)(ushTh % 10)],
            CLCDSIZE_NUMBER);

/* space */
/*
fn_LcdWrite((CLCDPOS_TH1 + CLCDSIZE_NUMBER), aDispClear, 1);
*/
}
```

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/>