

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

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

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

2010年4月1日  
ルネサスエレクトロニクス株式会社

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

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

## ご注意書き

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

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

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

# アプリケーション・ノート

## 78K0S/Kx1+

### サンプル・プログラム (A/Dコンバータ)

### 連続A/D変換&平均値算出編

この資料は、サンプル・プログラムの動作概要や使用方法、およびA/Dコンバータの設定方法や活用方法を説明したものです。サンプル・プログラムでは、ANI0端子とANI1端子からのアナログ入力に対して、A/D変換を4回ずつ行い、それぞれの変換データとその平均値をRAM領域に保存します。

#### 対象デバイス

78K0S/KA1+マイクロコントローラ  
 78K0S/KB1+マイクロコントローラ  
 78K0S/KU1+マイクロコントローラ  
 78K0S/KY1+マイクロコントローラ

#### 目次

第1章 概要 ... 3	
1.1 初期設定の主な内容 ... 3	
1.2 メイン・ループ以降の内容 ... 4	
第2章 回路図 ... 5	
2.1 回路図 ... 5	
第3章 ソフトウェアについて ... 6	
3.1 ファイル構成 ... 6	
3.2 使用する内蔵周辺機能 ... 7	
3.3 初期設定と動作概要 ... 7	
3.4 フロー・チャート ... 9	
第4章 設定方法について ... 10	
4.1 A/Dコンバータの設定 ... 10	
4.2 入力電圧とA/D変換結果 ... 19	
第5章 システム・シミュレータ SM+での動作確認 ... 20	
5.1 サンプル・プログラムのビルド ... 20	
5.2 SM+での動作 ... 22	
第6章 関連資料 ... 26	
付録A プログラム・リスト ... 27	
付録B 改版履歴 ... 42	

- 本資料に記載されている内容は2008年7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。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。

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

特別水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器

特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等

当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

M8E 02.11

# 第1章 概 要

このサンプル・プログラムでは、A/Dコンバータの使用例を示しています。ANI0端子とANI1端子からのアナログ入力に対して、A/D変換を4回ずつ行い、それぞれの変換データとその平均値をRAM領域に保存します。

## 1.1 初期設定の主な内容

初期設定の主な内容は、次のとおりです。

システム・クロック・ソースとして、高速内蔵発振器を選択<sup>注</sup>

ウォッチドッグ・タイマの動作停止

V<sub>LVI</sub> (低電圧検出電圧) を4.3 V ± 0.2 Vに設定

V<sub>DD</sub> (電源電圧) < V<sub>LVI</sub>になったあとに、V<sub>DD</sub> < V<sub>LVI</sub>を検出した場合、内部リセット (LVIリセット) 信号を発生

CPUクロック周波数を8 MHzに設定

入出力ポートの設定

A/Dコンバータの設定

・A/D変換時間を72/f<sub>XP</sub> (9.0 μs) に設定

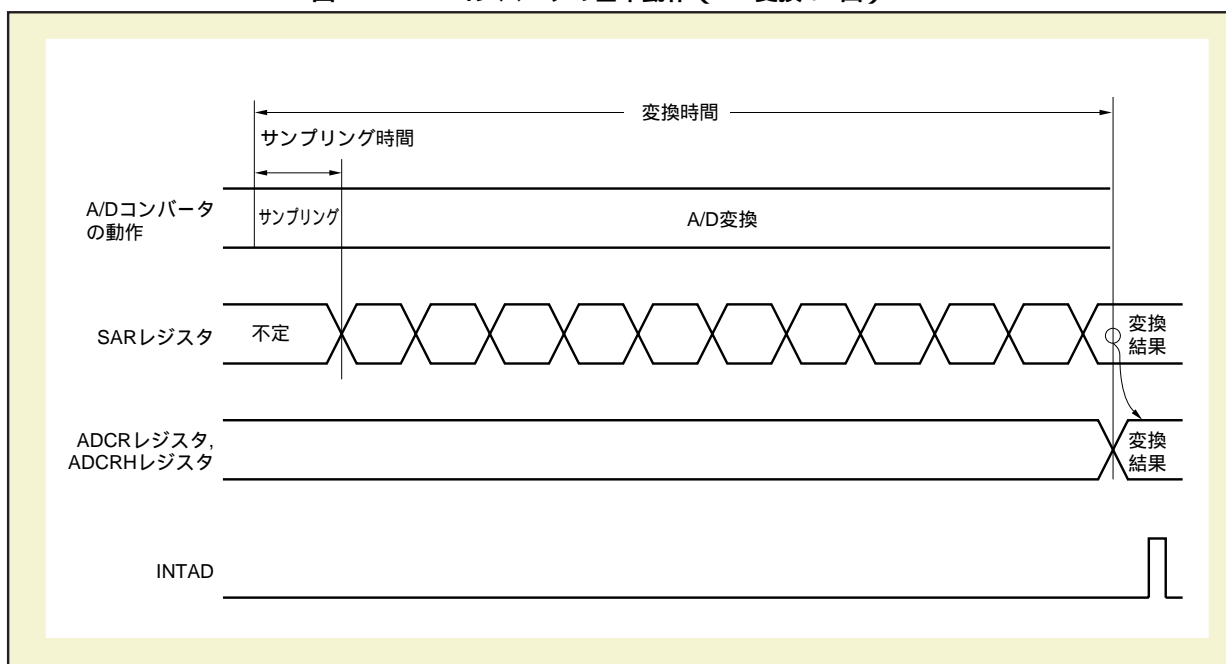
**注** オプション・バイトで設定します。

## 1.2 メイン・ループ以降の内容

初期設定完了後は、A/D変換動作を開始し、ANI0端子からのアナログ入力に対して、A/D変換を4回ずつ行い、それらの変換データをRAM領域に保存します。ANI1端子からのアナログ入力に対しても、同様の処理を行ったあと、A/D変換動作を停止します。A/D変換動作停止後、ANI0端子とANI1端子の4回のA/D変換データの平均値をそれぞれ算出し、RAM領域に保存します。

初期設定完了後は、上記のように連続4回のA/D変換処理(2ch)とその平均値算出処理(2ch)を繰り返します。このように、A/D変換を複数回行い、その変換データから算出した平均値を使用することで、アナログ入力のばらつきの影響を抑えることができます。また、平均値算出時にA/D変換動作を停止することで、消費電力を抑えることができます。

図1-1 A/Dコンバータの基本動作 (A/D変換: 1回)



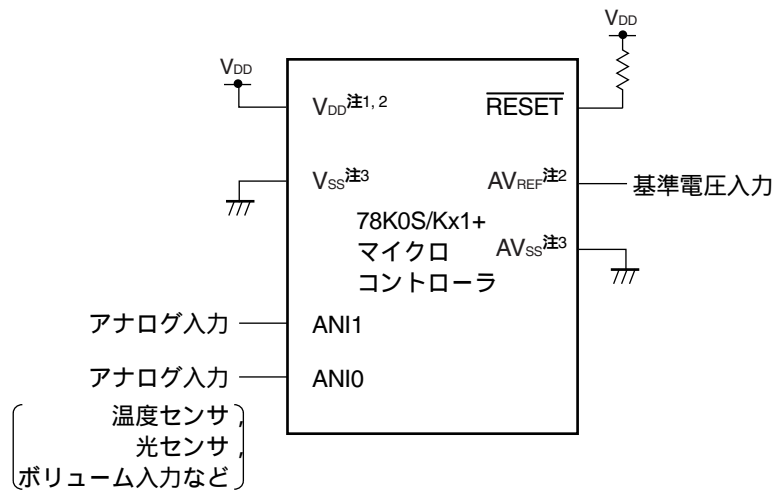
**注意** デバイス使用上の注意事項については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

## 第2章 回路図

この章では、このサンプル・プログラムで使用する回路図を説明します。

### 2.1 回路図

回路図を次に示します。



注1. 4.5 V  $V_{DD}$  5.5 Vの電圧範囲で使用してください。

2. 78K0S/KU1+, 78K0S/KY1+マイクロコントローラの $V_{DD}$ は、A/Dコンバータの基準電圧入力 ( $AV_{REF}$ ) と兼用しています。A/Dコンバータを使用する場合は、使用する電源電圧で安定するようにしてください。

3. 78K0S/KA1+, 78K0S/KU1+, 78K0S/KY1+マイクロコントローラの $V_{SS}$ は、A/Dコンバータのグラウンド電位 ( $AV_{SS}$ ) と兼用しています。 $V_{SS}$ を必ず安定しているGND (= 0 V) に接続してください。




注意 回路図中の端子以外の未使用端子はすべて出力ポートのため、オープン (未接続) にしてください。

## 第3章 ソフトウェアについて

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

### 3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル		
				
main.asm (アセンブリ言語版) ----- main.c (C言語版)	マイコンのハードウェア初期化処理とメイン処理のソース・ファイル	注1	注1	
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 (システム・クロック・ソースなどを設定)			
ad.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
ad.prj	統合開発環境 PM+用プロジェクト・ファイル			
ad.pri ad.prs ad.prm	システム・シミュレータ SM+ for 78K0S/Kx1+用プロジェクト・ファイル		注2	
ad0.pnl	システム・シミュレータ SM+ for 78K0S/Kx1+用入出力パネル・ファイル (周辺ハードウェア動作を確認するために使用)		注2	

注1. アセンブリ言語版には「main.asm」、C言語版には「main.c」が同封されています。

2. 78K0S/KU1+マイクロコントローラには、同封されていません。

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+とシステム・シミュレータ SM+ for 78K0S/Kx1+で使用するファイルを同封



: システム・シミュレータ SM+ for 78K0S/Kx1+で使用するマイコン動作シミュレーション・ファイルを同封



## 3.2 使用する内蔵周辺機能

このサンプル・プログラムでは、マイコンに内蔵する次の周辺機能を使用します。

- ・ 10ビット分解能A/D変換 : A/Dコンバータ
- ・  $V_{DD} < V_{LVI}$ 検出 : 低電圧検出 (LVI) 回路
- ・ アナログ入力 : ANI0, ANI1 (アナログ入力ポート)

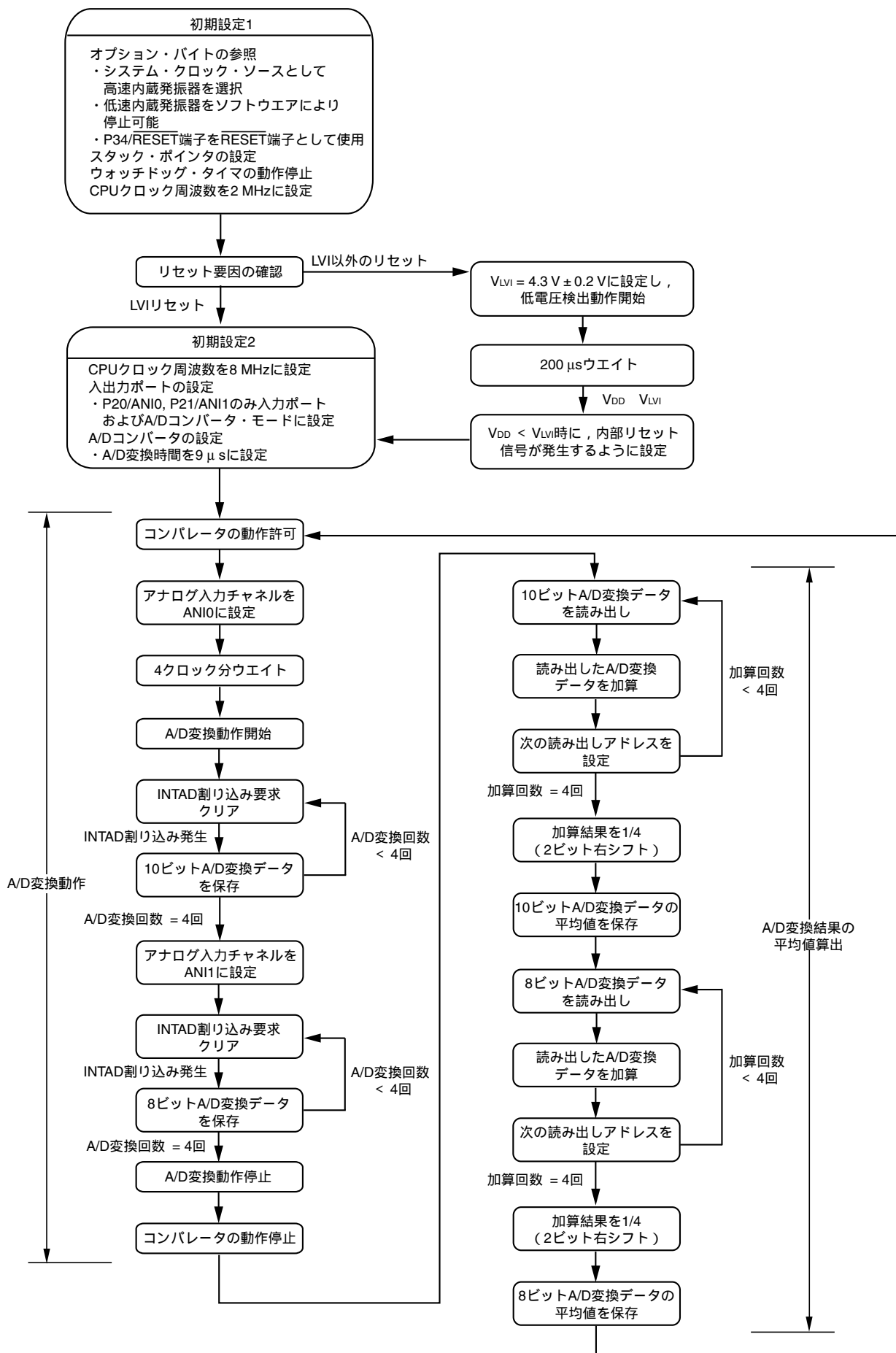
## 3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、低電圧検出機能の設定、クロック周波数の選択、入出力ポートの設定、A/Dコンバータの設定などを行います。

初期設定完了後は、A/D変換動作を開始し、ANI0端子からのアナログ入力に対して、A/D変換を4回ずつ行い、それらの変換データをRAM領域に保存します。ANI1端子からのアナログ入力に対しても、同様の処理を行ったあと、A/D変換動作を停止します。A/D変換動作停止後、ANI0端子とANI1端子の4回のA/D変換データの平均値をそれぞれ算出し、RAM領域に保存します。

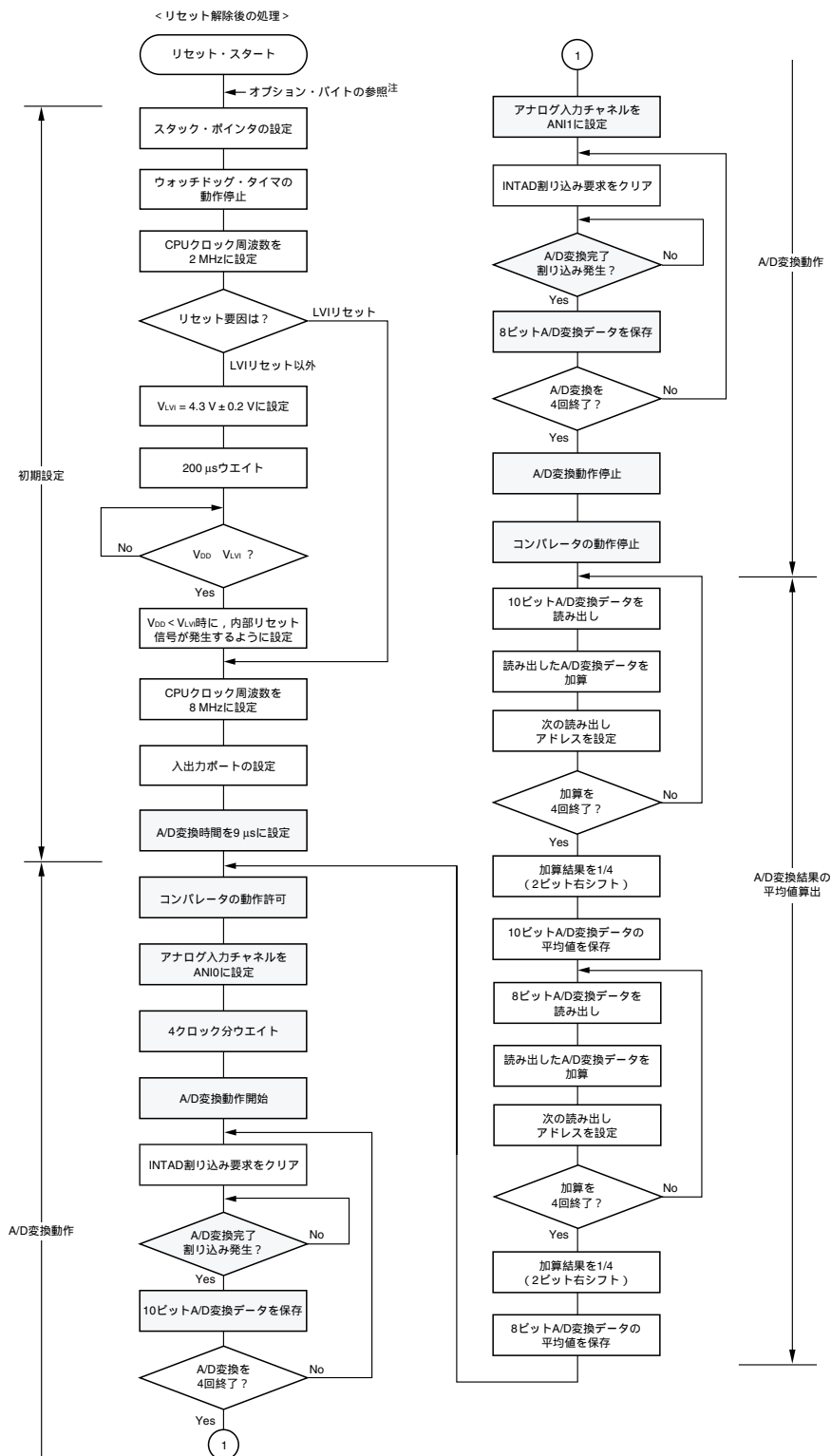
初期設定完了後は、上記のように連続4回のA/D変換処理 (2 ch) とその平均値算出処理 (2 ch) を繰り返します。このように、A/D変換を複数回行い、その変換データから算出した平均値を使用することで、アナログ入力のばらつきの影響を抑えることができます。また、平均値算出時にA/D変換動作を停止することで、消費電力を抑えることができます。

詳細については、次の状態遷移図（ステート・チャート）に示します。



### 3.4 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。



注 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・バイトの参照により、次の内容が設定されます。

- ・システム・クロック・ソースとして、高速内蔵発振クロック（8 MHz (TYP.)）を使用
- ・低速内蔵発振器をソフトウェアで停止可
- ・P34/RESET端子をRESET端子として使用

## 第4章 設定方法について

この章では、A/Dコンバータの設定について説明します。

その他の初期設定については、[78K0S/Kx1+ サンプル・プログラム\(初期設定\)](#) [LED点灯のスイッチ制御編 アプリケーション・ノート](#)を、割り込みについては、[78K0S/Kx1+ サンプル・プログラム\(割り込み\)](#) [スイッチ入力による外部割り込み編 アプリケーション・ノート](#)を、低電圧検出(LVI)については、[78K0S/Kx1+ サンプル・プログラム\(低電圧検出\)](#) [2.7 V未満検出時リセット発生編 アプリケーション・ノート](#)を参照してください。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

アセンブラ命令については、[78K/0Sシリーズ 命令編 ユーザーズ・マニュアル](#)を参照してください。

### 4.1 A/Dコンバータの設定

A/Dコンバータは、主に次の6種類のレジスタを使用します。

- ・A/Dコンバータ・モード・レジスタ (ADM)
- ・アナログ入力チャンネル指定レジスタ (ADS)
- ・10ビットA/D変換結果レジスタ (ADCR)
- ・8ビットA/D変換結果レジスタ (ADCRH)
- ・ポート・モード・レジスタx (PMx)
- ・ポート・モード・コントロール・レジスタx (PMCx)

< A/Dコンバータの基本的な動作設定手順例 >

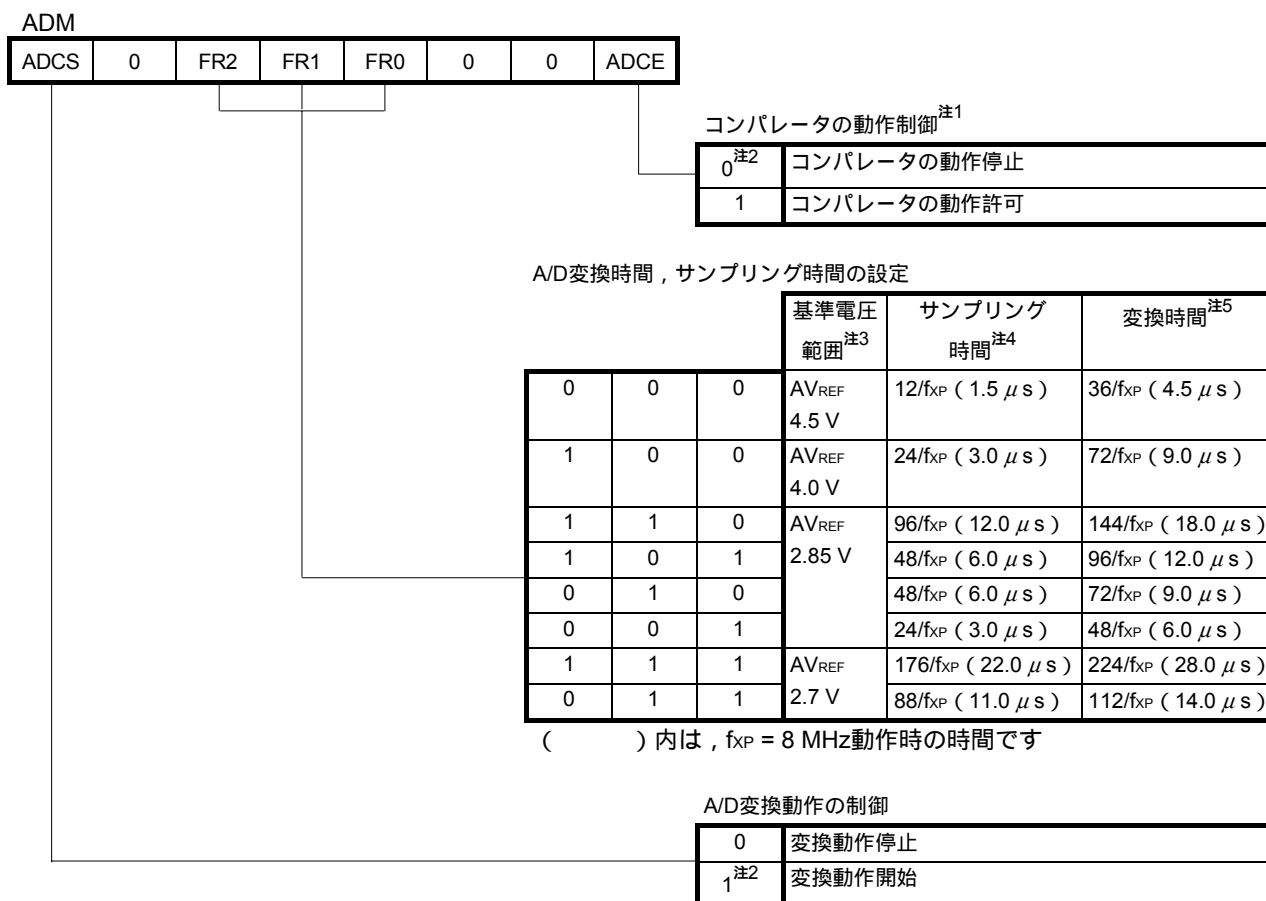
- FR2-FR0ビットでA/D変換時間を設定
- ADCEビットをセット(1)
- ADSレジスタでアナログ入力チャンネルを設定
- 4クロック分ウエイト(NOP命令を2つ実行, または2マシン・サイクル相当の命令を実行)
- ADCSビットをセット(1) : A/D変換動作開始

- 注意1. ~ は順不同です。  
 2. から までの間は, 1 μs以上空けてください。

(1) ADMレジスタの設定

ADMレジスタは, A/D変換するアナログ入力の変換時間, 変換動作の開始 / 停止を設定するレジスタです。

図4 - 1 A/Dコンバータ・モード・レジスタ (ADM) のフォーマット



- 備考1. f<sub>XP</sub> : 周辺ハードウェアへのクロックの発振周波数  
 2. 変換時間は, サンプリング時間と, サンプリング値を逐次比較し変換結果が出力されるまでの時間の合計です。

(注と注意は次ページ)

注1. コンパレータは、ADCSとADCEで動作制御され、動作開始から安定するまでに、 $1\mu\text{s}$ かかります。このため、ADCEに1を設定してから $1\mu\text{s}$ 以上経過したあとに、ADCSに1を設定することで、最初の変換データより有効となります。 $1\mu\text{s}$ 以上ウエイトしないでADCSに1を設定した場合は、最初の変換データを無視してください。

表4 - 1 ADCSとADCEの設定

ADCS	ADCE	A/D変換動作
0	0	停止状態（DC電力消費パスは存在しません）
0	1	変換待機モード（コンパレータのみ電力を消費）
1	x	変換モード

- ADCE = 0(コンパレータ動作停止)時の場合でも、ADCSに1を設定するとA/D変換動作を開始します。ただし、最初の変換データは、保証値の範囲外のため、無視してください。
- 基準電圧範囲に応じて、下記の注4と注5の条件を満たすように、FR2, FR1, FR0を設定する必要があります。

例 AVREF 2.7 V, f<sub>XP</sub> = 8MHzの場合

- ・ サンプリング時間は $11.0\mu\text{s}$ 以上、A/D変換時間は $14.0\mu\text{s}$ 以上 $100\mu\text{s}$ 未満
- ・ FR2, FR1, FR0 = 0, 1, 1または1, 1, 1を設定

- サンプリング時間が次の時間になるように設定してください。

- ・ AVREF 4.5 V :  $1.0\mu\text{s}$ 以上
- ・ AVREF 4.0 V :  $2.4\mu\text{s}$ 以上
- ・ AVREF 2.85 V :  $3.0\mu\text{s}$ 以上
- ・ AVREF 2.7 V :  $11.0\mu\text{s}$ 以上

- A/D変換時間が次の時間になるように設定してください。

- ・ AVREF 4.5 V :  $3.0\mu\text{s}$ 以上 $100\mu\text{s}$ 未満
- ・ AVREF 4.0 V :  $4.8\mu\text{s}$ 以上 $100\mu\text{s}$ 未満
- ・ AVREF 2.85 V :  $6.0\mu\text{s}$ 以上 $100\mu\text{s}$ 未満
- ・ AVREF 2.7 V :  $14.0\mu\text{s}$ 以上 $100\mu\text{s}$ 未満

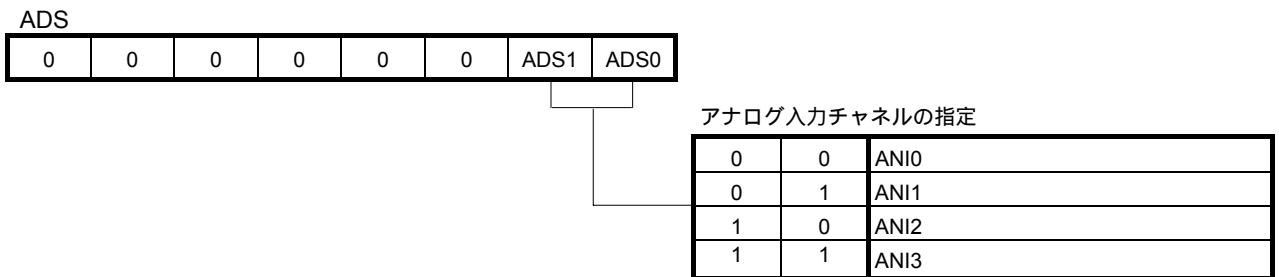
注意1. 前述のサンプリング時間および変換時間は、クロック周波数の誤差を含んでいませんので、クロック周波数の誤差を考慮して、注4, 5の条件を満たすサンプリング時間および変換時間を選択してください（高速内蔵発振器使用時は、最大 $\pm 5\%$ の誤差）。

- A/D変換停止（ADCS = 0）状態でADMのADCS以外のビットを操作したあとに、A/D変換開始する場合、NOP命令を2つまたは2マシン・サイクル相当の命令を実行してから、ADCSを1に設定してください。
- FR0-FR2を書き換える場合は、いったんA/D変換動作を停止（ADCS = 0）させたのちに行ってください。
- ビット6, 2, 1には、必ず0を設定してください。

(2) ADSレジスタの設定

ADSレジスタは、A/D変換するアナログ電圧の入力ポートを指定するレジスタです。

図4 - 2 アナログ入力チャンネル指定レジスタ (ADS) のフォーマット

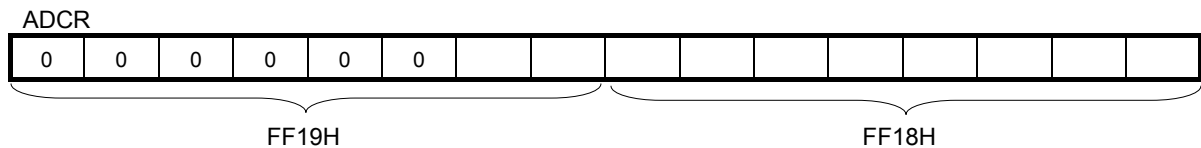


注意 ビット2-7には必ず0を設定してください。

(3) ADCRレジスタの動作

ADCRレジスタは、A/D変換結果を保持するリード・オンリーの16ビット・レジスタです。上位6ビットは“0”固定です。A/D変換が終了するたびに、逐次変換レジスタから変換結果がロードされます。ADCRにはFF19Hのビット1から順に格納されます。FF19Hには変換結果の上位2ビットが、FF18Hには変換結果の低位8ビットが入ります。

図4 - 3 10ビットA/D変換結果レジスタ (ADCR) のフォーマット

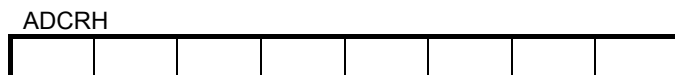


注意 ADMレジスタ、ADSレジスタに対して書き込み動作を行ったとき、ADCRレジスタの内容は不定となることがあります。変換結果は、変換動作終了後、ADMレジスタ、ADSレジスタに対して書き込み動作を行う前に読み出してください。上記以外のタイミングでは、正しい変換結果が読み出されないことがあります。

(4) ADCRHレジスタの動作

ADCRHレジスタは、A/D変換結果を保持するリード・オンリーの8ビット・レジスタです。10ビット分解能の上位8ビットを格納します。

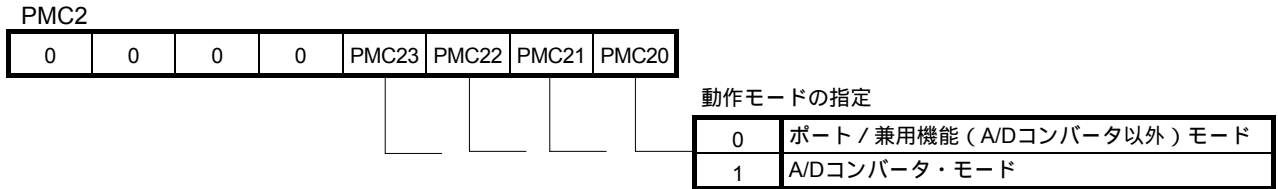
図4 - 4 8ビットA/D変換結果レジスタ (ADCRH) のフォーマット



(5) PMC2レジスタとPM2レジスタの設定

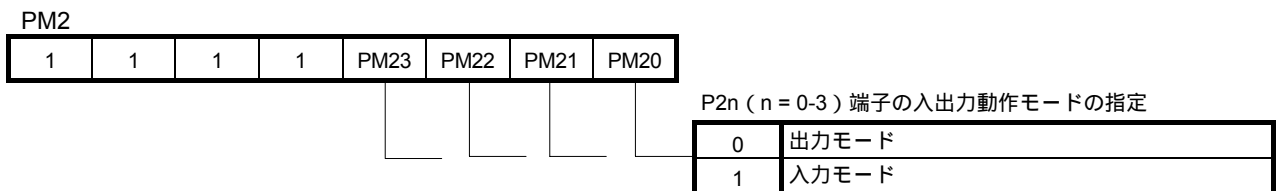
ANI0/P20-ANI3/P23端子をアナログ入力として使用するとき，PMC20-PMC23およびPM20-PM23にそれぞれ1を設定してください。

図4 - 5 ポート・モード・コントロール・レジスタ2 (PMC2) のフォーマット



**注意** PMC20-PMC23に1を設定した場合，P20/ANI0-P23/ANI3端子をポート機能として使用できません。  
また，A/Dコンバータ・モードに設定した端子のプルアップ抵抗オプション・レジスタ(PU20-PU23)は，必ず0を設定してください。

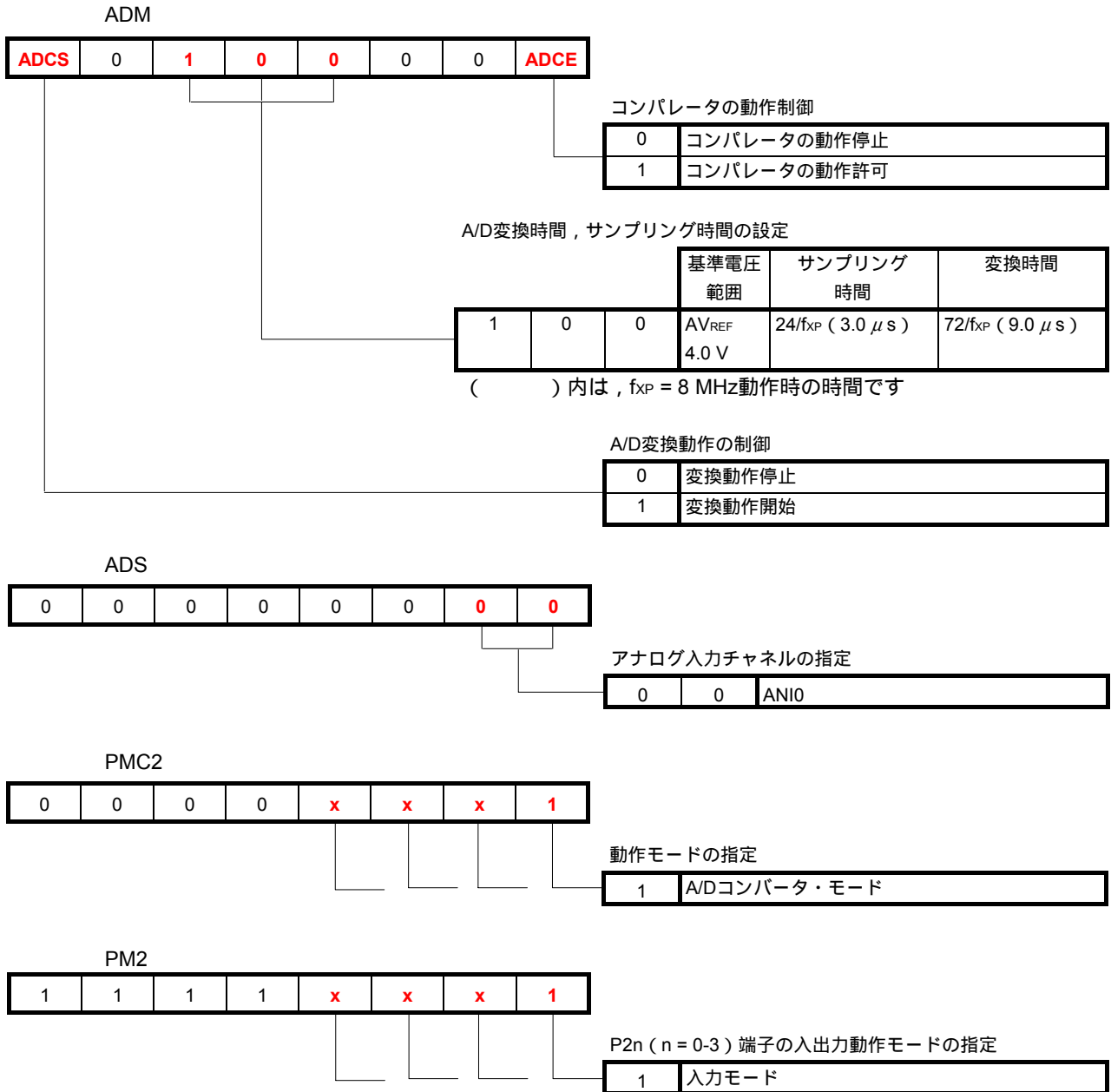
図4 - 6 ポート・モード・レジスタ2 (PM2) のフォーマット





【 例 】アナログ入力チャンネルをANI0に，A/D変換時間を $9\mu\text{s}$ に設定し，A/D変換動作を開始する場合  
 (周辺ハードウェアへのクロックの発振周波数 ( $f_{XP}$ ) = 8 MHz)  
 (本サンプル・プログラム・ソースと同内容)

(1) レジスタの設定内容



## (2) サンプル・プログラム

## アセンブリ言語の場合

```
SET1  PMC2.0
SET1  PM2.0
MOV   ADM,  #00100000B
SET1  ADCE
MOV   ADS,  #00H
NOP
NOP
SET1  ADCS
```

## C言語の場合

```
PMC2.0 = 1;
PM2.0 = 1;
ADM = 0b00100000;
ADCE = 1;
ADS = 0x00;
NOP();
NOP();
ADCS = 1;
```

## 【本サンプル・プログラム・ソースからの抜粋】

付録A プログラム・リストから，A/Dコンバータの機能に関する抜粋部分を示します（前述の【例】と同内容）。

## (1) アセンブリ言語

```

XMAIN CSEG UNIT
IRESET:
    .
    .
    .
    MOV    ADM, #00100000B ; A/D変換時間 = 72/fxp (= 9.0us)
    ; A/D変換時間を設定

MMAINLOOP:
    .
    .
    .
    SET1  ADCE ; コンパレータの動作許可
    MOV   ADS, #00H ; アナログ入力チャネルを
    ; アナログ入力チャネルをANI0に初期化
    ; コンパレータの動作許可
    ; アナログ入力チャネルをANI0に初期化

    NOP ; 4クロック分
    NOP ; ウェイト
    SET1  ADCS ; A/D変換動作開始
    ; A/D変換動作開始

LMLP100:
;----- A/D変換完了待ち -----
    CLR1  ADIF ; INTAD割り込み要求をクリア
LMLP150:
    NOP
    BF    ADIF, $LMLP150 ; INTAD割り込み待ち
    MOVW  AX, ADCR ; 10ビットA/D変換データを読み出し
    ; A/D変換終了後に、10ビットA/D変換
    ; データをADCRレジスタより読み出し
    .
    .
    MOV   ADS, #01H ; アナログ入力チャネルをANI1に設定
    ; アナログ入力チャネルをANI1に変更

LMLP200:
;----- A/D変換完了待ち -----
    CLR1  ADIF ; INTAD割り込み要求をクリア
LMLP250:
    NOP
    BF    ADIF, $LMLP250 ; INTAD割り込み待ち
    CLR1  ADIF ; INTAD割り込み要求をクリア
    MOV   A, ADCRH ; 8ビットA/D変換データを読み出し
    ; A/D変換終了後に、8ビットA/D変換
    ; データをADCRHレジスタより読み出し
    .
    .
    CLR1  ADCS ; A/D変換動作停止
    ; A/D変換動作停止
    CLR1  ADCE ; コンパレータの動作停止
    ; コンパレータの動作停止
    .
    .
    .

```

## (2) C言語

```

void hdwinit(void) {
    unsigned char ucCnt200us; /* 200usウェイト用8ビット変数 */
    .
    .
    .
    ADM = 0b00100000; /* A/D変換時間 = 72/fxp (= 9.0us) */
    return;
}

void main(void) {
    .
    .
    .
    ADCE = 1; /* コンパレータの動作許可 */
    ADS = 0x00; /* アナログ入力チャンネルをANI0に初期化 */
    NOP();
    NOP();
    ADCS = 1; /* A/D変換動作開始 */
    INTM0 = 0b00000000;

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回A/D変換処理を行う */
    {
        ADIF = 0; /* INTAD割り込み要求をクリア */

        while (!ADIF) /* INTAD割り込み待ち */
        {
            NOP();
            g_ushnAdBuff0[ucTimes] = ADCR; /* 10ビットA/D変換データを格納 */
        }

        ADS = 0x01; /* アナログ入力チャンネルをANI1に設定 */
    }

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回A/D変換処理を行う */
    {
        ADIF = 0; /* INTAD割り込み要求をクリア */
        while (!ADIF) /* INTAD割り込み待ち */
        {
            NOP();
            g_ucAdBuff1[ucTimes] = ADCRH; /* 8ビットA/D変換データを格納 */
        }
    }

    ADCS = 0; /* A/D変換動作停止 */
    ADCE = 0; /* コンパレータの動作停止 */
    .
    .
    .

```

A/D変換時間を設定  
 コンパレータを動作許可に設定  
 アナログ入力チャンネルをANI0に初期化  
 4クロック分ウェイト  
 A/D変換動作開始  
 アナログ入力チャンネルをANI1に変更  
 A/D変換終了後に、10ビットA/D変換データをADCRレジスタより格納  
 A/D変換終了後に、8ビットA/D変換データをADCRHレジスタより格納  
 A/D変換動作停止  
 コンパレータを動作停止に設定

## 4.2 入力電圧とA/D変換結果

アナログ入力端子 (ANI0-ANI3) より入力されたアナログ入力電圧と理論上のA/D変換結果 (ADCRレジスタ)<sup>注</sup>には、次式に示す関係があります。

ADCRレジスタ (10ビット分解能)

$$\text{ADCR} = \text{INT} \left( \frac{V_{\text{AIN}}}{V_{\text{VREF}}} \times 1024 + 0.5 \right)$$

または、

$$\left( \text{ADCR} - 0.5 \right) \times \frac{V_{\text{VREF}}}{1024} < V_{\text{AIN}} < \left( \text{ADCR} + 0.5 \right) \times \frac{V_{\text{VREF}}}{1024}$$

**備考** INT ( ) : ( ) 内の値の整数部を返す関数  
 $V_{\text{AIN}}$  : アナログ入力電圧  
 $V_{\text{VREF}}$  :  $V_{\text{VREF}}$ 端子電圧  
 ADCR : 10ビットA/D変換結果レジスタ (ADCR) の値


**計算例** アナログ入力電圧が1.96V,  $V_{\text{VREF}}$ 端子電圧が5 Vの場合

$$\cdot \text{ADCR} = \text{INT} \left( \frac{1960}{5000} \times 1024 + 0.5 \right) = \text{INT} (401.908) = 401 = 0191\text{H}$$

**注** A/D変換結果レジスタは2種類あります。


- ・ ADCRレジスタ : A/D変換結果 (10ビット分解能) を格納
- ・ ADCRHレジスタ : A/D変換結果 (10ビット分解能) の上位8ビットを格納

## 第5章 システム・シミュレータ SM+での動作確認

この章では、のアイコンを選択してダウンロードしたアセンブリ言語用のファイル（ソース・ファイル+プロジェクト・ファイル）を用い、サンプル・プログラムが、システム・シミュレータ SM+ for 78K0S/Kx1+でどのように動作するかを説明します。

**注意** SM+ for 78K0S/Kx1+は、78K0S/KU1+マイクロコントローラには対応していません（2008年7月現在）。  
したがって、78K0S/KU1+マイクロコントローラはSM+ for 78K0S/Kx1+で動作確認することはできません。

### 5.1 サンプル・プログラムのビルド

サンプル・プログラムをSM+ for 78K0S/Kx1+（以降、「SM+」と表記します）で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、サンプル・プログラムのビルド方法について、のアイコンからダウンロードしたアセンブリ言語用サンプル・プログラム（ソース・プログラム+プロジェクト・ファイル）を使用し、説明します。その他のダウンロードしたプログラムのビルド方法については、[78K0S/Kx1+ サンプル・プログラム スタートアップ・ガイド アプリケーション・ノート](#)を参照してください。

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

#### 【コラム】ビルドのエラー


PM+でビルドしているときに「A006 File not found 'C:¥NECTOOLS32¥LIB78K0S¥s0sl.rel'」または、「\*\*\* ERROR F206 Segment '@@DATA' can't allocate to memory - ignored.」というエラー・メッセージが出た場合、次の手順にてコンパイラオプションの設定を変更してください。

[ツール] [コンパイラオプションの設定]を選択してください。

[コンパイラオプションの設定]ダイアログが開いたら、「スタートアップ・ルーチン」タグを選択してください。

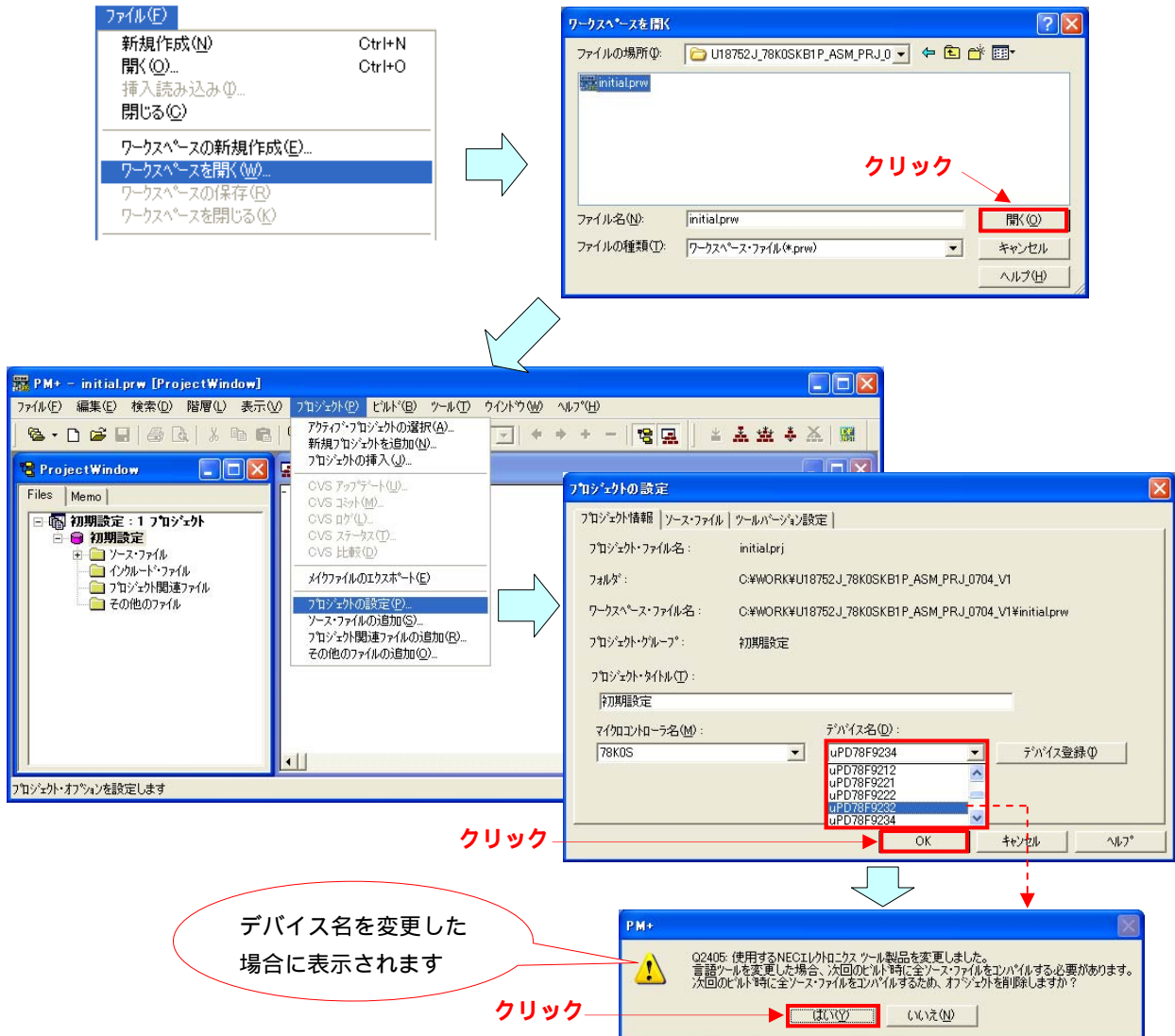
「標準ライブラリ固定領域を使用する」のチェックを外してください（それ以外のチェックは、そのまま）。


「標準ライブラリ固定領域を使用する」のチェックを外すと、標準ライブラリ固定領域として確保されていた118バイトのRAM領域が使用可能になりますが、標準ライブラリ（getchar関数やmalloc関数など）を使用できなくなります。

このサンプル・プログラムでは、のアイコンを選択してダウンロードしたファイルを使用する場合、デフォルトで「標準ライブラリ固定領域を使用する」のチェックが外されています。

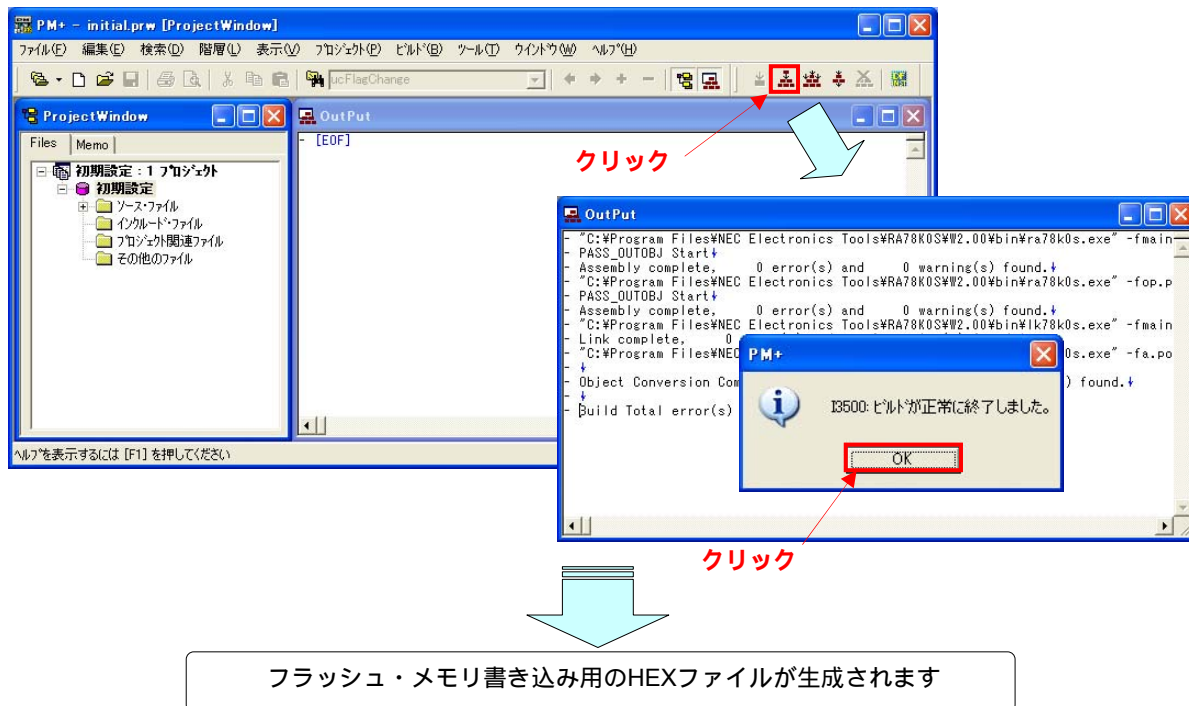
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から, 「ad.prw」を選択し, [開く] ボタンをクリックしてください。ワークスペースが作成され, その中にソース・ファイルが自動的に読み込まれます。
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が立ち上がったら, 使用するデバイス名を選択(デフォルトでは, ROM/RAMサイズの最も大きいデバイスが選択)し, [OK] ボタンをクリックしてください。

**備考** 下の図は, 「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。



- (4)  (「ビルド」ボタン)をクリックしてください。ソース・ファイルが正常にビルドされると、「I3500: ビルドが正常に終了しました」というメッセージ画面が立ち上がります。
- (5) メッセージ画面にある [ OK ] ボタンをクリックしてください。フラッシュ・メモリ書き込み用のHEXファイルが作成されます。

**備考** 下の図は、「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。



## 5.2 SM+での動作


ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。

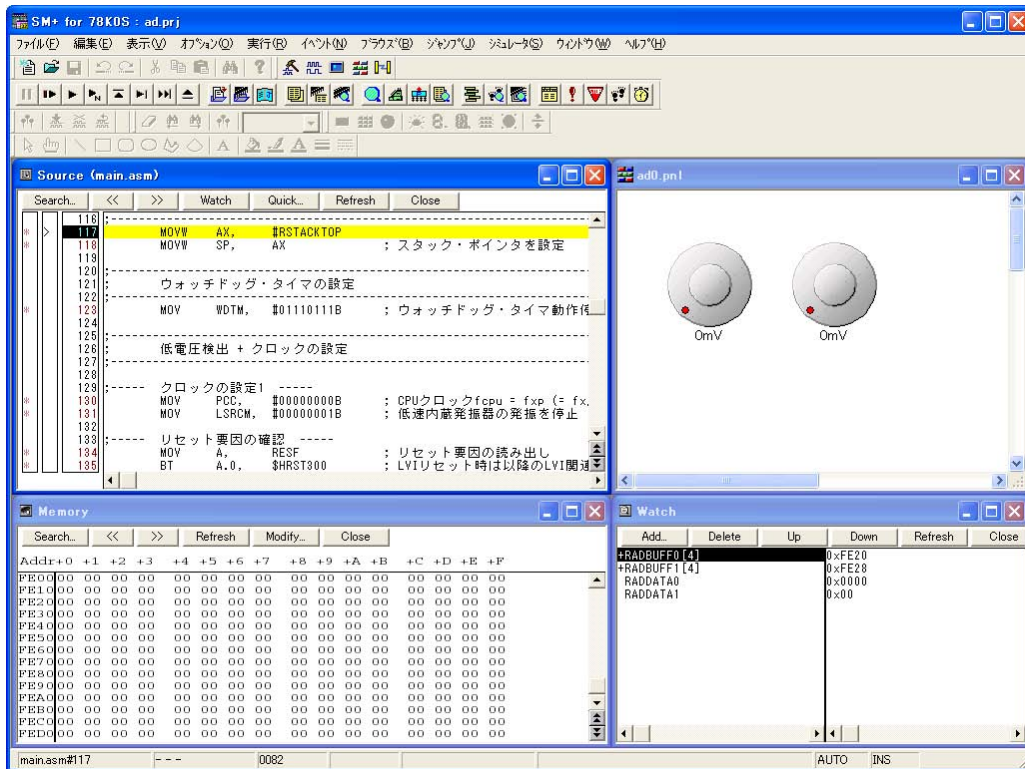
SM+操作方法の詳細については、[SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル](#)を参照してください。


- (1) SM+ for 78K0S/Kx1+ W1.02をPM+ Ver6.30の環境で使用するために、次のサイトにあるPDFファイルを参照して、「外部ツールの登録」を行い、SM+を起動してください。

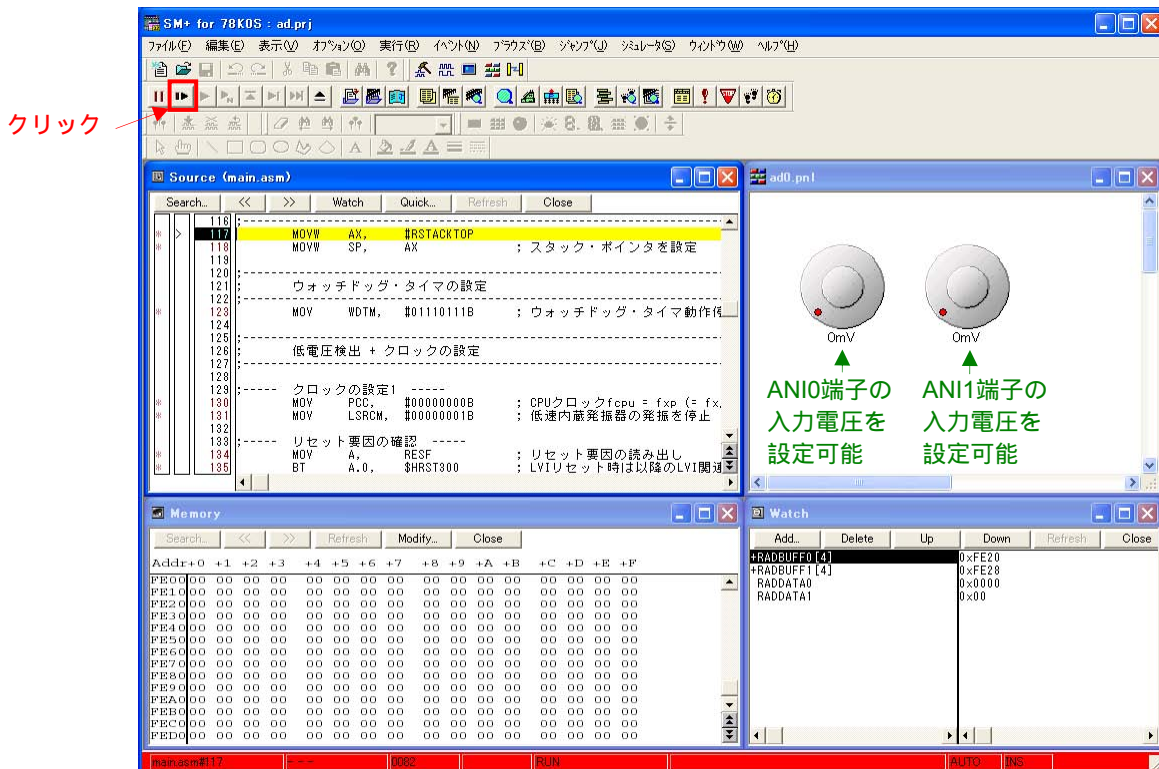
・ <http://www.necel.com/micro/ja/freesoft/pdf/ZUD-CD-07-0189.pdf>



- (2) SM+を起動すると、次のような画面になります（これは、のアイコンを選択してダウンロードしたアセンブリ言語のソース・ファイルを使用した場合の画面例です）。

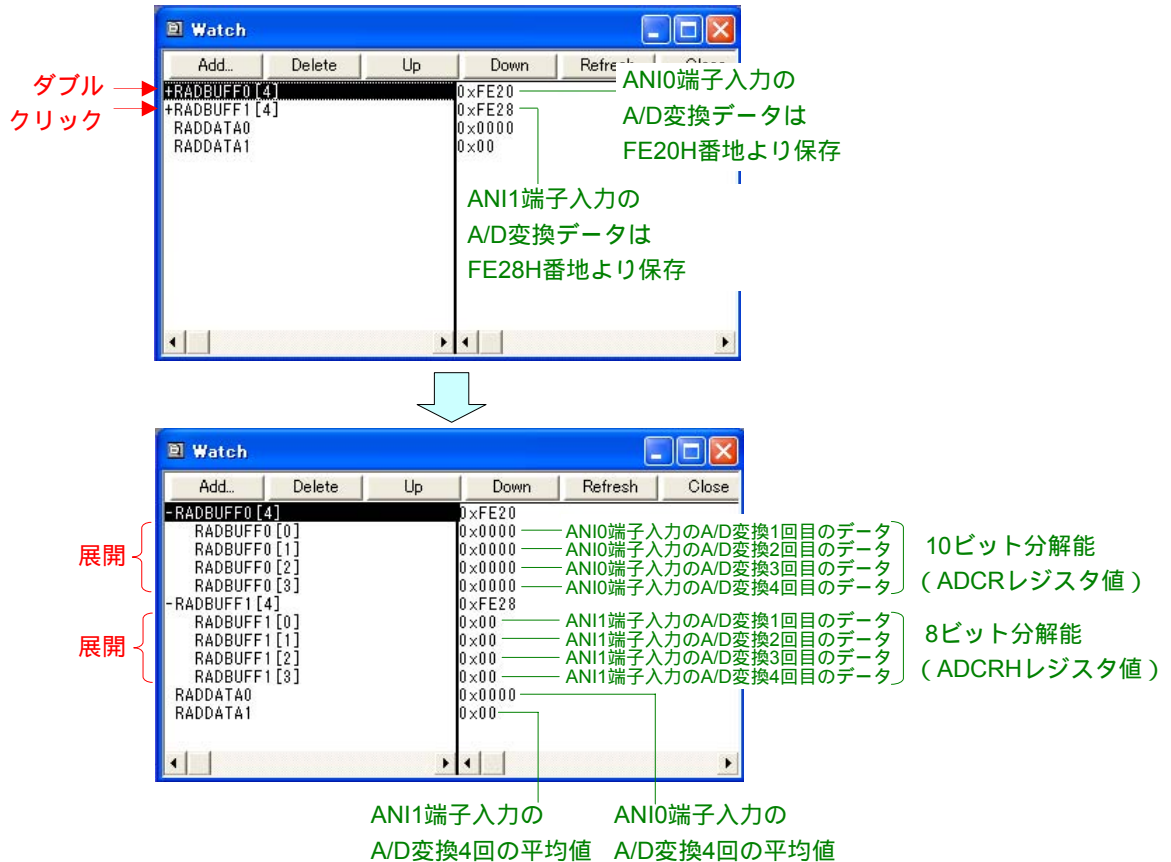


- (3) （「リスタート」ボタン）をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。



プログラム  
実行中は、  
赤になります。

- (4) [ Watch ] ウィンドウ上の「 +RADBUFF0 [ 4 ] 」と「 +RADBUFF1 [ 4 ] 」をダブルクリックすると、先頭文字が「 + 」から「 - 」に切り替わり、「 - RADBUFF0 [ 4 ] 」と「 - RADBUFF1 [ 4 ] 」の下に、データが展開表示されます。



- (5) プログラム実行中に、入出力パネル・ウィンドウ上のレベルゲージにある動作点（赤い丸）をドラッグすると、入力電圧を変化させることができます。入力電圧の変化により、[ Watch ]ウィンドウ上および[ Memory ]ウィンドウ上のA/D変換データが変化することを確認してください。

- 備考1. ANI0, ANI1端子からの入力電圧と、A/D変換データの関係については、[4.2 入力電圧とA/D変換結果](#)を参照してください。
2. A/Dコンバータの基準電圧入力（AVREF）は、デフォルトで5 Vに設定されています。AVREFの電圧値を変更したい場合は、入力パネルにレベルゲージを追加する必要があります。追加したレベルゲージは、プロパティにて、接続端子を「AVREF」に、最大入力値を任意の値に設定してください。プロパティ設定後、[ 入力シミュレーション ]を選択し、入出力パネル・ウィンドウ上のレベルゲージにある動作点をドラッグして、AVREFの電圧値を設定してください。

例1. ANI0端子からの入力電圧：1960 mV, ANI1端子からの入力電圧：0 mV

入出力パネル・ウィンドウ

[ Watch ] ウィンドウ

[ Memory ] ウィンドウ

例2. ANI0端子からの入力電圧：1960 mV, ANI1端子からの入力電圧：1960 mV

入出力パネル・ウィンドウ

[ Watch ] ウィンドウ

[ Memory ] ウィンドウ

## 第6章 関連資料

資料名		和文 / 英文
78K0S/KU1+ ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0S/KY1+ ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0S/KA1+ ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0S/KB1+ ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0Sシリーズ 命令編 ユーザーズ・マニュアル		<a href="#">PDF</a>
RA78K0S アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
CC78K0S Cコンパイラ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		<a href="#">PDF</a>
SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル		<a href="#">PDF</a>
フラッシュ書き込み簡単マニュアル (MINICUBE2編) インフォメーション	78K0S/KU1+	<a href="#">PDF</a>
	78K0S/KY1+	<a href="#">PDF</a>
	78K0S/KA1+	<a href="#">PDF</a>
	78K0S/KB1+	<a href="#">PDF</a>
78K0S/Kx1+ アプリケーション・ ノート	サンプル・プログラム スタートアップ・ガイド	<a href="#">PDF</a>
	サンプル・プログラム (初期設定) LED点灯のスイッチ制御編	<a href="#">PDF</a>
	サンプル・プログラム (割り込み) スイッチ入力による外部割り込み編	<a href="#">PDF</a>
	サンプル・プログラム (低電圧検出) 2.7V未滴検出時リセット発生編	<a href="#">PDF</a>

## 付録A プログラム・リスト

プログラム・リスト例として、78K0S/KB1+マイクロコントローラのソース・プログラムを次に示します。

```
main.asm (アセンブリ言語版)
;*****
;
; NEC Electronics      78K0S/KB1+シリーズ
;
;*****
; 78K0S/KB1+シリーズ      サンプル・プログラム
;*****
; A/Dコンバータ
;*****
; 【履歴】
; 2007.8.--      新規作成
;*****
;
; 【概要】
;
; 本サンプルプログラムは、A/Dコンバータの使用例を示すものである。AN10端子とAN11端
; 子へのアナログ入力に対してA/D変換を4回ずつ行い、RAM領域へ保存する。ここで、A/D
; 変換結果はINTAD割り込み要求フラグのポーリング処理を利用して読み出す。また、AN10
; 端子のA/D変換データは、ADCRレジスタを読み出すことにより10ビット分解能で保存し、
; AN11端子のA/D変換データは、ADCRHレジスタを読み出すことにより8ビット分解能で保存
; する。これらのデータは、それぞれ平均値を求めてRAM領域へ保存する。
;
;
; < 主な設定内容 >
;
; ・ウォッチドッグ・タイマの動作停止
; ・低電圧検出電圧VLVIを4.3V±0.2Vに設定
; ・VDD VLVIとなった後にVDD < VLVIとなった場合、内部リセット信号発生(低電圧検出回路)
; ・CPUクロックを8MHzに設定
; ・周辺ハードウェアへの供給クロックを8MHzに設定
; ・A/Dコンバータの変換時間を9.0usに設定
;
;
```

; < A/D変換結果と平均値のデータ格納場所 >

```

;
; +-----+
; | ラベル名 | データ長 |          データ種別          | A/D変換ポート |
; |-----|-----|-----|-----|
; | RADBUFF0 | 16ビット | 10ビットA/D変換データ (1回目) | P20/ANI0 |
; |          | 16ビット | 10ビットA/D変換データ (2回目) | P20/ANI0 |
; |          | 16ビット | 10ビットA/D変換データ (3回目) | P20/ANI0 |
; |          | 16ビット | 10ビットA/D変換データ (4回目) | P20/ANI0 |
; |-----|-----|-----|-----|
; | RADBUFF1 | 8ビット | 8ビットA/D変換データ (5回目) | P21/ANI1 |
; |          | 8ビット | 8ビットA/D変換データ (6回目) | P21/ANI1 |
; |          | 8ビット | 8ビットA/D変換データ (7回目) | P21/ANI1 |
; |          | 8ビット | 8ビットA/D変換データ (8回目) | P21/ANI1 |
; |-----|-----|-----|-----|
; | RADATA0  | 16ビット | 1-4回目データの平均値          | -          |
; | RADATA1  | 8ビット  | 5-8回目データの平均値          | -          |
; +-----+

```

; 【ポート入出力の設定】

```

;
; 入力ポート : P20, P21
; 出力ポート : P00-P03, P22, P23, P30-P33, P40-P47, P120-P123, P130
; 未使用のポートは全て出力ポートに設定しておく
;

```

; \*\*\*\*\*

; =====

; ベクタ・テーブルの設定

; =====

XVCT	CSEG	AT	0000H		
		DW	IRESET	;(00)	RESET
		DW	IRESET	;(02)	--
		DW	IRESET	;(04)	--
		DW	IRESET	;(06)	INTLVI
		DW	IRESET	;(08)	INTPO
		DW	IRESET	;(0A)	INTP1
		DW	IRESET	;(0C)	INTTMH1
		DW	IRESET	;(0E)	INTTM000

```

DW      IRESET      ;(10)  INTTM010
DW      IRESET      ;(12)  INTAD
DW      IRESET      ;(14)  --
DW      IRESET      ;(16)  INTP2
DW      IRESET      ;(18)  INTP3
DW      IRESET      ;(1A)  INTTM80
DW      IRESET      ;(1C)  INTSRE6
DW      IRESET      ;(1E)  INTSR6
DW      IRESET      ;(20)  INTST6

```

```

;=====
;
;
;   RAMの定義
;
;=====

```

```

DRAM  DSEG  SADDRP
RABUFF0:  DS    8      ; 10ビットA/D変換結果格納用
RABUFF1:  DS    4      ; 8ビットA/D変換結果格納用
RADATA0:  DS    2      ; 10ビットA/D変換結果の平均値格納用
RADATA1:  DS    1      ; 8ビットA/D変換結果の平均値格納用

```

```

;=====
;
;
;   スタック領域の確保
;
;=====

```

```

DSTK  DSEG  AT      OFEE0H
RSTACKEND:  DS    20H      ; スタック領域を32バイト確保
RSTACKTOP:      ; スタック領域の先頭アドレス = FF00H

```

```

;*****
;
;
;   リセット解除後の初期化処理
;
;*****

```

```

XMAIN CSEG  UNIT
IRESET:
;-----
;   スタック・ポインタの設定
;-----
MOVW  AX,    #RSTACKTOP
MOVW  SP,    AX      ; スタック・ポインタを設定

```

```

;-----
;   ウォッチドッグ・タイマの設定
;-----
MOV   WDTM,  #01110111B   ; ウォッチドッグ・タイマ動作停止

;-----
;   低電圧検出 + クロックの設定
;-----

;----- クロックの設定1 -----
MOV   PCC,    #00000000B   ; CPUクロックfcpu = fxp (= fx/4 = 2MHz)
MOV   LSRCM,  #00000001B   ; 低速内蔵発振器の発振を停止

;----- リセット要因の確認 -----
MOV   A,      RESF         ; リセット要因の読み出し
BT    A.0,    $HRST300     ; LVIリセット時は以降のLVI関連処理を省略し、SET_CLOCKへ

;----- 低電圧検出の設定 -----
MOV   LVIS,   #00000000B   ; 低電圧検出レベルVLVI = 4.3V ± 0.2Vに設定
SET1  LVION   ; 低電圧検出回路の動作許可

MOV   A,      #40         ; 200usウェイト用のカウント値を代入
;----- 200usウェイト -----
HRST100:
DEC   A
BNZ  $HRST100             ; 0.5[us/cik]×10[cik]×40[count] = 200[us]

;----- VDD VLVI待ち処理 -----
HRST200:
NOP
BT   LVIF,    $HRST200    ; VDD < VLVIなら分岐

SET1 LVIMD     ; VDD < VLVI時に内部リセット信号が発生するように設定

;----- クロックの設定2 -----
HRST300:
MOV   PPCC,   #00000000B   ; 周辺ハードウェアへの供給クロックfxp = fx (= 8MHz)
; -> CPUクロックfcpu = fxp = 8MHz

;-----
;   ポート0の設定
;-----
MOV   P0,     #00000000B   ; P00-P03の出力ラッチLow

```



```

MOV    PM0,    #11110000B    ; P00-P03を出力ポートに設定

;-----
;   ポート2の設定
;-----
MOV    P2,     #00000000B    ; P20-P23の出力ラッチLow
MOV    PMC2,   #00000011B    ; P20,P21をA/Dコンバータ・モードに設定
MOV    PM2,    #11110011B    ; P22,P23を出力ポートに、P20,P21を入力ポートに設定

;-----
;   ポート3の設定
;-----
MOV    P3,     #00000000B    ; P30-P33の出力ラッチLow
MOV    PM3,    #11110000B    ; P30-P33を出力ポートに設定

;-----
;   ポート4の設定
;-----
MOV    P4,     #00000000B    ; P40-P47の出力ラッチLow
MOV    PM4,    #00000000B    ; P40-P47を出力ポートに設定

;-----
;   ポート12の設定
;-----
MOV    P12,    #00000000B    ; P120-P123の出力ラッチLow
MOV    PM12,   #11110000B    ; P120-P123を出力ポートに設定

;-----
;   ポート13の設定
;-----
MOV    P13,    #00000001B    ; P130の出力High

;-----
;   A/Dコンバータの設定
;-----
MOV    ADM,    #00100000B    ; A/D変換時間 = 72/fXP (= 9.0us)

;-----
;
;   メイン・ループ
;-----
;-----
MAINLOOP:

```

```
;-----  
; ANI0端子のA/D変換処理 (変換結果を10ビット分解能で保存)  
;-----  
MOVW HL, #RADBUFF0 ; 10ビットA/D変換データ格納用テーブル・アドレスを指定  
MOV B, #4 ; A/D変換回数を指定  
  
SET1 ADCE ; コンパレータの動作許可  
MOV ADS, #00H ; アナログ入力チャネルをANI0に初期化  
NOP  
NOP  
SET1 ADCS ; A/D変換動作開始  
  
LMLP100:  
;----- A/D変換完了待ち -----  
CLR1 ADIF ; INTAD割り込み要求をクリア  
LMLP150:  
NOP  
BF ADIF, $LMLP150 ; INTAD割り込み待ち  
  
;----- 変換データの格納 -----  
MOVW AX, ADCR ; 10ビットA/D変換データを読み出し  
MOV [HL+1], A ; 上位2ビットを格納  
XCH A, X  
MOV [HL], A ; 下位8ビットを格納  
INC L ; テーブル・アドレスを+2  
INC L  
DBNZ B, $LMLP100 ; A/D変換回数 < 4なら分岐  
  
;-----  
; ANI1端子のA/D変換処理 (変換結果を8ビット分解能で保存)  
;-----  
MOVW HL, #RADBUFF1 ; 8ビットA/D変換データ格納用テーブル・アドレスを指定  
MOV B, #4 ; A/D変換回数を指定  
  
MOV ADS, #01H ; アナログ入力チャネルをANI1に設定  
  
LMLP200:  
;----- A/D変換完了待ち -----  
CLR1 ADIF ; INTAD割り込み要求をクリア  
LMLP250:  
NOP
```

```

BF      ADIF, $LMLP250      ; INTAD割り込み待ち
CLR1    ADIF                ; INTAD割り込み要求をクリア

;----- 変換データの格納 -----
MOV     A,      ADCRH      ; 8ビットA/D変換データを読み出し
MOV     [HL],   A          ; A/D変換データを格納
INC     L                ; テーブル・アドレスを+1
DBNZ   B,      $LMLP200   ; A/D変換回数 < 4なら分岐

CLR1    ADCS              ; A/D変換動作停止
CLR1    ADCE              ; コンパレータの動作停止

;-----
;      10ビットA/D変換データの平均値を算出 (AN10端子)
;-----
MOVW   HL,     #RADBUFF0   ; 10ビットA/D変換データ格納用テーブル・アドレスを指定
MOVW   AX,     #0000H      ; AXレジスタをクリア

;----- 加算する -----
MOV     B,      #4          ; 加算回数を指定
LMLP300:
XCH    A,      X
ADD    A,      [HL]        ; 下位8ビットを加算
XCH    A,      X
ADDC   A,      [HL+1]      ; 上位2ビットを加算(下位の繰り上がりを含む)
INC    L                ; テーブル・アドレスを+2
INC    L
DBNZ   B,      $LMLP300   ; 加算回数 < 4なら分岐

;----- 平均値を求める -----
MOV     B,      #2          ; 右シフト(= 1/2倍)の回数を指定
LMLP350:
ROR    A,      1          ; 上位のビットを1右シフト
XCH    A,      X
RORC   A,      1          ; 下位のビットを1右シフト(上位からのシフトを含む)
XCH    A,      X
DBNZ   B,      $LMLP350   ; 右シフト回数 < 2なら分岐

AND    A,      #00000011B  ; 上位のビット0,1以外をマスク
MOVW   RADATA0, AX        ; 平均値(10ビット・データ)をRADATA0に格納

```

```

;-----
;      8ビットA/D変換データの平均値を算出 (ANI1端子)
;-----
      MOVW   HL,    #RADBUFF1    ; 8ビットA/D変換データ格納用テーブル・アドレスを指定
      MOVW   AX,    #0000H      ; AXレジスタをクリア

;----- 加算する -----
      MOV    B,     #4          ; 加算回数を指定
LMLP400:
      ADD    A,     [HL]        ; 加算
      BNC   $LMLP420          ; 繰り上がりがなければ分岐
      INC   X                ; 上位ビットを+1
LMLP420:
      INC   L                ; テーブル・アドレスを+1
      DBNZ  B,     $LMLP400     ; 加算回数<4なら分岐

;----- 平均値を求める -----
      MOV    B,     #2          ; 右シフト(= 1/2倍)の回数を指定
LMLP450:
      XCH   A,     X
      ROR   A,     1          ; 上位のビットを1右シフト
      XCH   A,     X
      RORC  A,     1          ; 下位のビットを1右シフト(上位からのシフトを含む)
      DBNZ  B,     $LMLP450     ; 右シフト回数<2なら分岐

      MOV   RADATA1, A        ; 平均値(8ビット・データ)をRADATA1に格納

      BR   !MAINLOOP        ; MAINLOOPへ

end

```

main.c (C言語版)

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

NEC Electronics 78K0S/KB1+シリーズ

\*\*\*\*\*

78K0S/KB1+シリーズ サンプル・プログラム

\*\*\*\*\*

A/Dコンバータ

\*\*\*\*\*

【履歴】

2007.8.-- 新規作成

\*\*\*\*\*

【概要】

本サンプルプログラムは、A/Dコンバータの使用例を示すものである。AN10端子とAN11端子へのアナログ入力に対してA/D変換を4回ずつ行い、RAM領域へ保存する。ここで、A/D変換結果はINTAD割り込み要求フラグのポーリング処理を利用して読み出す。また、AN10端子のA/D変換データは、ADCRレジスタを読み出すことにより10ビット分解能で保存し、AN11端子のA/D変換データは、ADCRHレジスタを読み出すことにより8ビット分解能で保存する。これらのデータは、それぞれ平均値を求めてRAM領域へ保存する。

< 主な設定内容 >

- ・ウォッチドッグ・タイマの動作停止
- ・低電圧検出電圧VLVIを4.3V±0.2Vに設定
- ・VDD VLVIとなった後にVDD < VLVIとなった場合、内部リセット信号発生(低電圧検出回路)
- ・CPUクロックを8MHzに設定
- ・周辺ハードウェアへの供給クロックを8MHzに設定
- ・A/Dコンバータの変換時間を9.0usに設定

< A/D変換結果と平均値のデータ格納場所 >

変数名	データ長	データ種別	A/D変換ポート
g_ushnAdBuff0	16ビット	10ビットA/D変換データ (1回目)	P20/AN10
	16ビット	10ビットA/D変換データ (2回目)	P20/AN10
	16ビット	10ビットA/D変換データ (3回目)	P20/AN10
	16ビット	10ビットA/D変換データ (4回目)	P20/AN10

-----				
g_ucAdBuff1	8ビット	8ビットA/D変換データ (5回目)	P21/ANI1	
	8ビット	8ビットA/D変換データ (6回目)	P21/ANI1	
	8ビット	8ビットA/D変換データ (7回目)	P21/ANI1	
	8ビット	8ビットA/D変換データ (8回目)	P21/ANI1	
-----				
g_ushnAdData0	16ビット	1-4回目データの平均値	-	
g_unAdData1	8ビット	5-8回目データの平均値	-	
+-----+				

【ポート入出力の設定】

入力ポート：P20, P21

出力ポート：P00-P03, P22, P23, P30-P33, P40-P47, P120-P123, P130

未使用のポートは全て出力ポートに設定しておく

\*\*\*\*\*/

/\*=====

前処理指令

=====\*/

#pragma SFR /\* 特殊機能レジスタ(SFR)名を記述可能にする \*/

#pragma NOP /\* NOP命令を記述可能にする \*/

/\*=====

グローバル変数の定義

=====\*/

sreg static unsigned short int g\_ushnAdBuff0[4]; /\* 10ビットA/D変換データ格納用16ビット変数テーブル \*/

sreg static unsigned char g\_ucAdBuff1[4]; /\* 8ビットA/D変換データ格納用8ビット変数テーブル \*/

sreg static unsigned short int g\_ushnAdData0; /\* 10ビットA/D変換データの平均値格納用16ビット変数 \*/

sreg static unsigned char g\_ucAdData1; /\* 8ビットA/D変換データの平均値格納用8ビット変数 \*/

```

/*****

リセット解除後の初期化処理

*****/
void hdwinit(void){
    unsigned char ucCnt200us;    /* 200usウェイト用8ビット変数 */

/*-----
    ウォッチドッグ・タイマの設定 + 低電圧検出 + クロックの設定
-----*/

/* ウォッチドッグ・タイマの設定 */
    WDTM = 0b01110111;    /* ウォッチドッグ・タイマ動作停止 */

/* クロックの設定1 */
    PCC = 0b00000000;    /* CPUクロックfcpu = fxp (= fx/4 = 2MHz) */
    LSRCM = 0b00000001;    /* 低速内蔵発振器の発振を停止 */

/* リセット要因の確認 */
    if (!(RESF & 0b00000001)){    /* LVIリセット時は以降のLVI関連処理を省略 */

        /* 低電圧検出の設定 */
        LVIS = 0b00000000;    /* 低電圧検出レベルVLVI = 4.3V ± 0.2Vに設定 */
        LVION = 1;    /* 低電圧検出回路の動作許可 */

        for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){    /* 約200usウェイト */
            NOP();
        }

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

        LVIMD = 1;    /* VDD < VLVI時に内部リセット信号が発生するように設定 */
    }

/* クロックの設定2 */
    PPCC = 0b00000000;    /* 周辺ハードウェアへの供給クロックfxp = fx (= 8MHz)
        -> CPUクロックfcpu = fxp = 8MHz */

```

```

/*-----
    ポート0の設定
-----*/
P0   = 0b00000000;    /* P00-P03の出力ラッチLow */
PM0  = 0b11110000;    /* P00-P03を出力ポートに設定 */

/*-----
    ポート2の設定
-----*/
P2   = 0b00000000;    /* P20-P23の出力ラッチLow */
PMC2 = 0b00000011;    /* P20,P21をA/Dコンバータ・モードに設定 */
PM2  = 0b11110011;    /* P22,P23を出力ポートに、P20,P21を入力ポートに設定 */

/*-----
    ポート3の設定
-----*/
P3   = 0b00000000;    /* P30-P33の出力ラッチLow */
PM3  = 0b11110000;    /* P30-P33を出力ポートに設定 */

/*-----
    ポート4の設定
-----*/
P4   = 0b00000000;    /* P40-P47の出力ラッチLow */
PM4  = 0b00000000;    /* P40-P47を出力ポートに設定 */

/*-----
    ポート12の設定
-----*/
P12  = 0b00000000;    /* P120-P123の出力ラッチLow */
PM12 = 0b11110000;    /* P120-P123を出力ポートに設定 */

/*-----
    ポート13の設定
-----*/
P13  = 0b00000001;    /* P130の出力High */

/*-----
    A/Dコンバータの設定
-----*/
ADM  = 0b00100000;    /* A/D変換時間 = 72/fxp (= 9.0us) */

return;
}

```



```
/******  
  
    メイン・ループ  
  
*****/  
void main(void){  
  
    unsigned char ucTimes;          /* 回数カウント用8ビット変数 */  
    unsigned short int ushnAdSum; /* A/D変換データ加算用16ビット変数 */  
  
    while (1)  
    {  
/*-----  
        ANI0端子のA/D変換処理 (変換結果を10ビット分解能で保存)  
-----*/  
        ADCE = 1;          /* コンパレータの動作許可 */  
        ADS = 0x00;        /* アナログ入力チャンネルをANI0に初期化 */  
        NOP();  
        NOP();  
        ADCS = 1;          /* A/D変換動作開始 */  
  
        for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回A/D変換処理を行う */  
        {  
            ADIF = 0;          /* INTAD割り込み要求をクリア */  
  
            while (!ADIF)      /* INTAD割り込み待ち */  
            {  
                NOP();  
            }  
  
            g_ushnAdBuff0[ucTimes] = ADCR; /* 10ビットA/D変換データを格納 */  
        }  
  
/*-----  
        ANI1端子のA/D変換処理 (変換結果を8ビット分解能で保存)  
-----*/  
        ADS = 0x01;          /* アナログ入力チャンネルをANI1に設定 */  
  
        for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回A/D変換処理を行う */  
        {  
            ADIF = 0;          /* INTAD割り込み要求をクリア */
```

```
        while (!ADIF)                /* INTAD割り込み待ち */
        {
            NOP();
        }

        g_ucAdBuff1[ucTimes] = ADCRH; /* 8ビットA/D変換データを格納 */
    }

    ADCS = 0;                        /* A/D変換動作停止 */
    ADCE = 0;                        /* コンパレータの動作停止 */

/*-----
10ビットA/D変換データの平均値を算出 (ANI0端子)
-----*/

    ushnAdSum = 0x0000;              /* A/D変換データ加算用変数をクリア */

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回分のデータを加算する */
    {
        ushnAdSum += g_ushnAdBuff0[ucTimes]; /* 10ビットA/D変換データを加算 */
    }

    g_ushnAdData0 = ushnAdSum >> 2; /* 10ビットA/D変換データの平均値を保存 */

/*-----
8ビットA/D変換データの平均値を算出 (ANI1端子)
-----*/

    ushnAdSum = 0x0000;              /* A/D変換データ加算用変数をクリア */

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 4回分のデータを加算する */
    {
        ushnAdSum += g_ucAdBuff1[ucTimes]; /* 8ビットA/D変換データを加算 */
    }

    g_ucAdData1 = ushnAdSum >> 2; /* 8ビットA/D変換データの平均値を保存 */
}
}
```

op.asm (アセンブリ言語版とC言語版共通)

```

;=====
;
; オプション・バイトの設定
;
;=====
OPBT      CSEG  AT      0080H
          DB      10011100B      ; オプション・バイトの設定
;
;          |||
;          |||+----- 低速内蔵発振器はソフトウェアで停止可能
;          |++----- 高速内蔵発振クロック(8MHz)を使用
;          +----- P34/RESET端子をリセット端子として使用

          DB      11111111B      ; プロテクト・バイトの設定(セルフプログラミング用)
;
;          ||| |||
;          ++++++----- 全てのブロックへの書き込み許可

end

```

## 付録B 改版履歴

本文欄外の 印は、本版で改訂された主な箇所を示しています。この" "をPDF上でコピーして「検索する文字列」に指定することによって、改版箇所を容易に検索できます。

版 数	発行年月	改版箇所	改版内容
第1版	October 2007	-	-
第2版	July 2008	p.20	<b>第5章 システム・シミュレータ SM+での動作確認</b> ・注意文中の（2007年10月現在）を（2008年7月現在）に変更
		pp.20-22	5.1 サンプル・プログラムのビルドを変更
		p.22	5.2 SM+での動作 ・（1）を追加
		p.26	<b>第6章 関連資料</b> ・フラッシュ書き込み簡単マニュアル（MINICUBE2編） インフォメーションを追加

## 【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

---

## 【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

---

## 【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : [info@necel.com](mailto:info@necel.com)

---

## 【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。

---