

# 应用注释

## 78K0S/Kx1+

### 示例程序（A/D 转换器）

#### 连续 A/D 转换&平均值计算

本文件内容描述了示例程序的操作概要，怎样使用示例程序以及如何设置和使用 A/D 转换器。在本示例程序中，AN10 引脚和 AN11 引脚模拟输入中，A/D 转换各运行 4 次，各个转换数据和转换数据平均值保存至 RAM 区。

#### 目标设备

- 78K0S/KA1+ 微控制器
- 78K0S/KB1+微控制器
- 78K0S/KU1+微控制器
- 78K0S/KY1+微控制器

文档编号：U18915CA1V0AN00 (第一版)

出版日期：2008 年 03 月 N

© NEC Electronics Corporation 2007  
日本印刷

#### 目录

第一章 概述.....	3
1.1 初始设置主要内容.....	3
1.2 主循环之后的内容.....	4
第二章 电路图.....	5
2.1 电路图.....	5
第三章 软件.....	6
3.1 文件配置.....	6
3.2 所用内部外围功能.....	7
3.3 初始设置和运行概述.....	7
3.4 流程图.....	9
第四章 设置方法.....	10
4.1 设置A/D 转换器.....	10
4.2 输入电压和 A/D 转换结果.....	19
第五章 用系统仿真器SM+进行运行检查.....	20
5.1 构建示例程序.....	20
5.2 随SM+运行.....	21
第六章 相关文档.....	26
APPA 程序列表.....	27
APPB 修订记录.....	39

- 本档所刊登的内容有效期截至 2008 年 03 月。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”：计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”：运输设备（汽车、火车、船舶等），交通用信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”：航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

（注）

（1）本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。

（2）本声明中的“本公司产品”是指所有由日本电气电子株式会社所开发或制造，或为日本电气电子株式会社（定义如上）开发或制造的产品。

## 第一章 概述

在本示例程序中，给出了使用 A/D 转换器的示例。AN10 引脚和 AN11 引脚模拟输入中，A/D 转换各运行 4 次，各个转换数据和转换数据平均值保存到 RAM 区域。

### 1.1 初始设置主要内容

初始设置主要内容如下：

- 选择高速内部振荡器作为系统时钟源\*
- 停止看门狗定时器的运行
- 将  $V_{LVI}$  (低压检测电压) 设为  $4.3\text{ V} \pm 0.2\text{ V}$
- 在  $V_{DD}$  (供电电压) 大于或等于  $V_{LVI}$  时，当检测到  $V_{DD}$  小于  $V_{LVI}$  时，生成内部复位信号 (LVI 复位)。
- 将 CPU 时钟频率设为 8 MHz
- 设置 I/O 端口
- 设置 A/D 转换器
- 设 A/D 转换时间为  $72/f_{XP}$  (9.0 s)

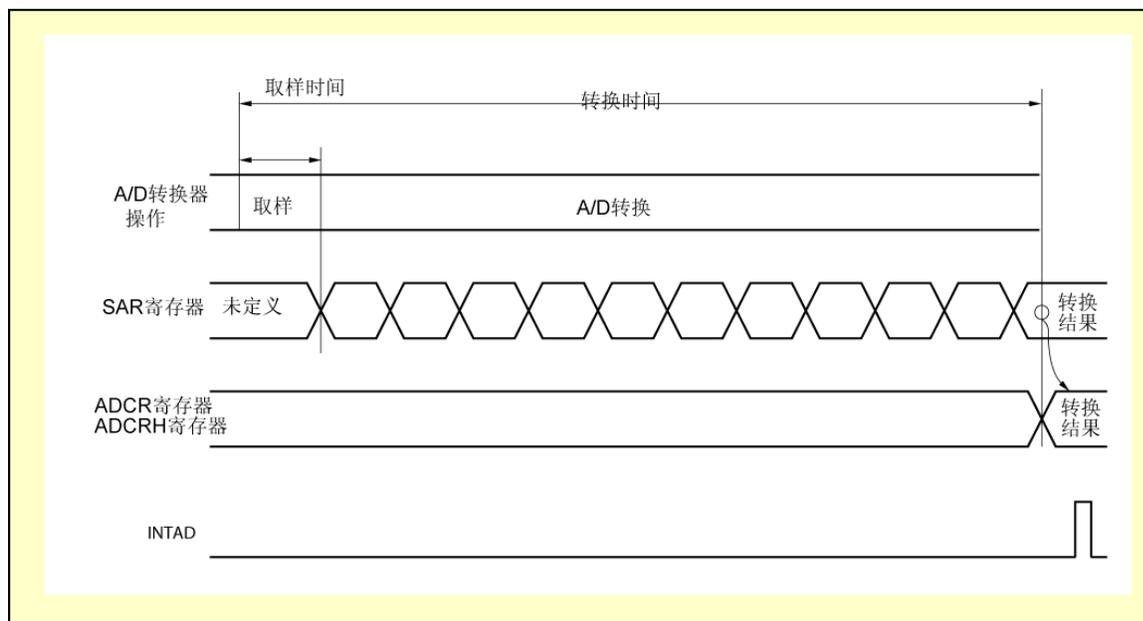
注 通过选项字节进行设置。

## 1.2 主循环之后的内容

初始设置完成以后，ANIO 引脚的模拟输入 A/D 转换运行 4 次，开始 A/D 转换操作，转换数据保存至 RAM 区。ANIO 引脚模拟输入执行同样的处理后，停止 A/D 转换操作。A/D 转换操作停止后，计算 ANIO 引脚和 ANI1 引脚 4 次转换的平均值，并将平均值保存至 RAM 区。

初始设置完成以后，如上所述，重复连续 4 次 A/D 转换处理 (2 ch) 和平均值计算处理(2 ch)。按照这种方式，通过多次运行 A/D 转换和计算转换数据的平均值，可以减低模拟输入变量效果。另外，在计算平均值时，通过停止 A/D 转换操作可以减少电耗。

图 1-1. 基础 A/D 转换器操作 (A/D 转换: 1 次)



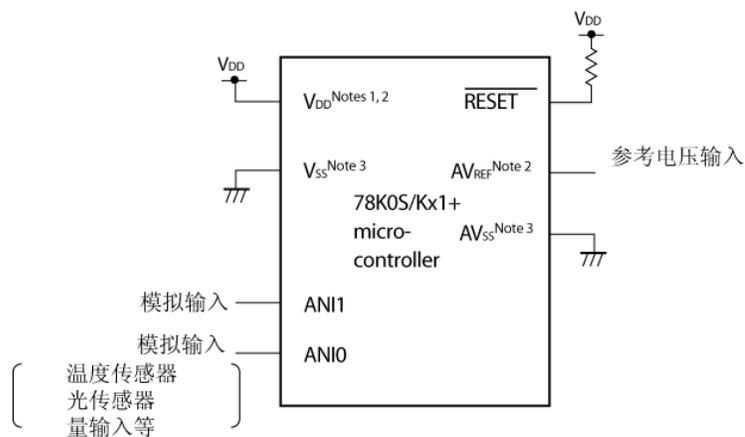
**注意事项** 使用本设备的注意事项，请参阅各个产品的用户手册([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)).

## 第二章 电路图

本章描述了在本示例程序中使用的电路图。

### 2.1 电路图

电路图如下所示。



- 注**
1. 工作电压范围为  $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。
  2. 选择的使用 78K0S/KU1+ 和 78K0S/KY1+ 微控制器的 V<sub>DD</sub> 用作 A/D 转换器的参考电压输入(AV<sub>REF</sub>)。使用 A/D 转换器时，确保 A/D 转换器使用的电源电压稳定。
  3. 选择的使用 78K0S/KA1+、78K0S/KU1+、78K0S/KY1+微控制器的 V<sub>SS</sub> 用作 A/D 转换器的接地电压。确保将 V<sub>SS</sub> 稳定接地(= 0 V)。

**注意事项** 除电路图中所示的引脚外，其他所有不使用的引脚保留开路(不连接)。

### 第三章 软件

本章描述了文件下载的压缩文件的配置、使用的微控制器的内部外围功能、示例程序的初始设置和运行概述，并展示流程图。

#### 3.1 文件配置

下表展示所下载的压缩文件的文件配置。

文件名称	描述	包含压缩文件 (*.zip)		
				
main.asm (汇编语言版) ----- main.c (C语言版)	用于微控制器的硬件初始化处理及源文件主处理	● <sup>注1</sup>	● <sup>注1</sup>	
op.asm	用于设置选项字节(设置系统时钟源)的汇编器源文件	●	●	
ad.prw	用于集成开发环境PM+的工作区文件		●	
ad.prj	用于集成开发环境PM+的项目文件		●	
ad.pri ad.prs ad.prm	用于适用78K0S/Kx1+系统仿真器 SM+的项目文件		● <sup>注2</sup>	
ad0.pnl	适用78K0S/Kx1+的系统仿真器SM+的I/O 面板文件 (用于检查外围硬件的工作)		● <sup>注2</sup>	●

- 注 1. “main.asm” 包含在汇编语言版中，“main.c” 包含在 C 语言版中。  
 2. 这些文件不包含在 78K0S/KU1+微控制器的文件中。

备注:



: 仅包含源文件。



: 包含与集成开发环境 PM+和 78K0S/Kx1+系统仿真器 SM+一起使用的文件。



: 包含与适用 78K0S/Kx1+的系统仿真器 SM+一起使用的微控制器运行仿真文件。

### 3.2 所用内部外围功能

本示例程序使用微控制器的下列内部外围功能。

- 10 位分辨率 A/D 转换: A/D 转换器
- $V_{DD} < V_{LVI}$  检测: 低压检测器 (LVI)
- 模拟输入: ANIO, ANI1 (模拟输入端口)

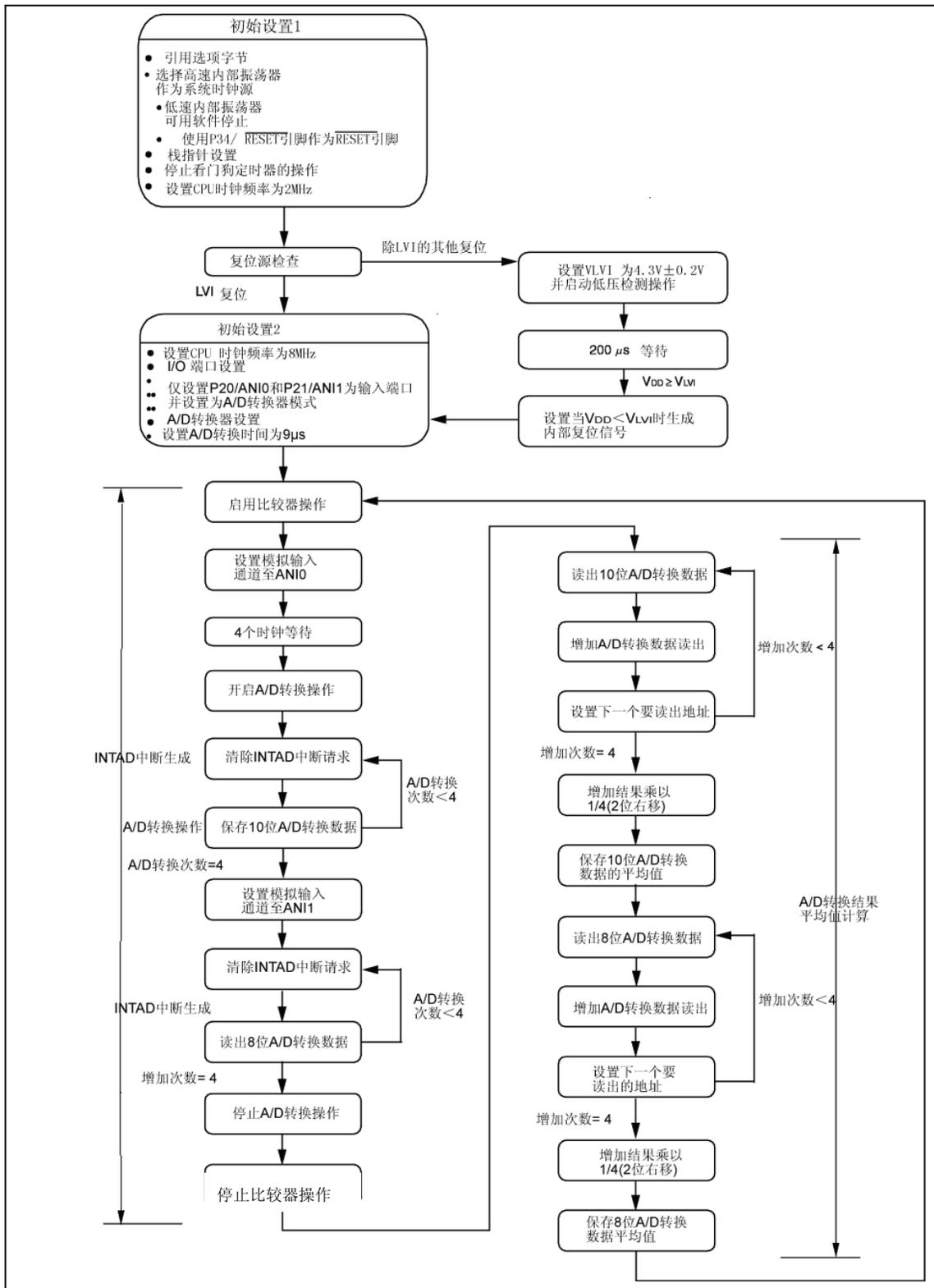
### 3.3 初始设置和操作概述

本示例程序进行的操作有：包括低压检测功能在内的初始设置、时钟频率的选择、I/O 端口的设置、A/D 转换器的设置。

初始设置完成后，开始 A/D 转换操作，在 ANIO 引脚模拟输入，A/D 转换运行 4 次且转换结果保存至 RAM 区。ANIO 引脚模拟输入执行同样处理之后，停止 A/D 转换操作，计算 ANIO 引脚和 ANI1 引脚 4 次 A/D 转换运行的平均值并将平均值保存至 RAM 区。

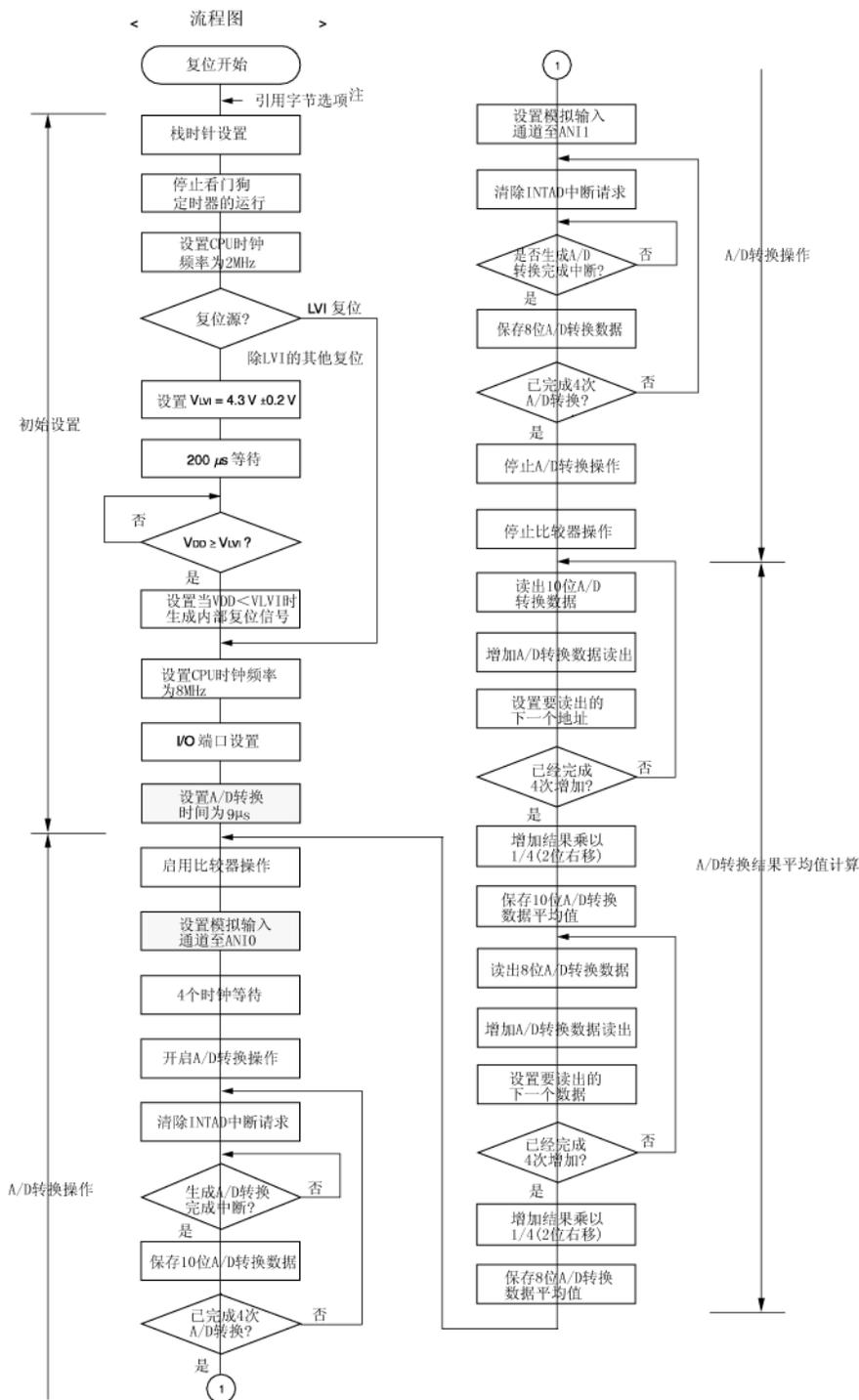
初始设置完成后，正如上文所述，重复连续的 4 次 A/D 转换处理（2ch）和平均值计算处理（2ch）。按照这种方式，多次运行 A/D 转换和使用转换数据计算的平均值可以降低模拟输入变量效果。另外，当计算平均值时，通过停止 A/D 转换操作可以减少电耗。

详情在如下所示的状态转换图中描述。



### 3.4 流程图

示例程序流程图如下所示。



**注** 复位解除后，自动执行微控制器选项字节的引用。在本示例程序中，下面的内容通过引用选项字节进行设置。

- 用高速内部振荡时钟（8MHz（典型））作为系统时钟源。
- 低速内部振荡器可用软件停止。
- 用 P34/RESET 引脚作为 RESET 引脚。

## 第四章 设置方法

本章描述了 A/D 转换器设置。

关于其他初始设置，请阅读 [78K0S/Kx1+ 示例程序 \(初始设置\) LED 发光开关控制的操作说明书](#)。关于中断，请阅读 [78K0S/Kx1+ 示例程序 \(中断\) 由开关输入生成的外部中断的操作说明书](#)。关于低电压检测 (LVI)，请阅读 [78K0S/Kx1+ 示例程序 \(低电压检测\) 在小于 2.7V 检测过程中的复位生成的操作说明书](#)。

关于如何设置寄存器，请阅读各产品的用户手册 ([78K0S/KU1+](#)、[78K0S/KY1+](#)、[78K0S/KA1+](#)、[78K0S/KB1+](#))。关于汇编器指令，请阅读 [78K0S 系列指令用户手册](#)。

### 4.1 设置 A/D 转换器

A/D 转换器使用下列六类寄存器。

- A/D 转换器模式寄存器 (ADM)
- 模拟输入通道规格寄存器 (ADS)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)
- 端口模式寄存器 x (PMx)
- 端口模式控制寄存器 x (PMCx)

<设置基础 A/D 转换器操作步骤的示例>

- <1> 用 FR2 至 FR0 位来设置 A/D 转换时间
- <2> 设置 (1) ADCE 位
- <3> 使用 ADS 寄存器设置模拟输入通道
- <4> 等待四个时钟 (执行两个 NOP 指令或等效于两个机器循环的一个指令)
- <5> 设置(1) ADCS 位: 开始 A/D 转换操作

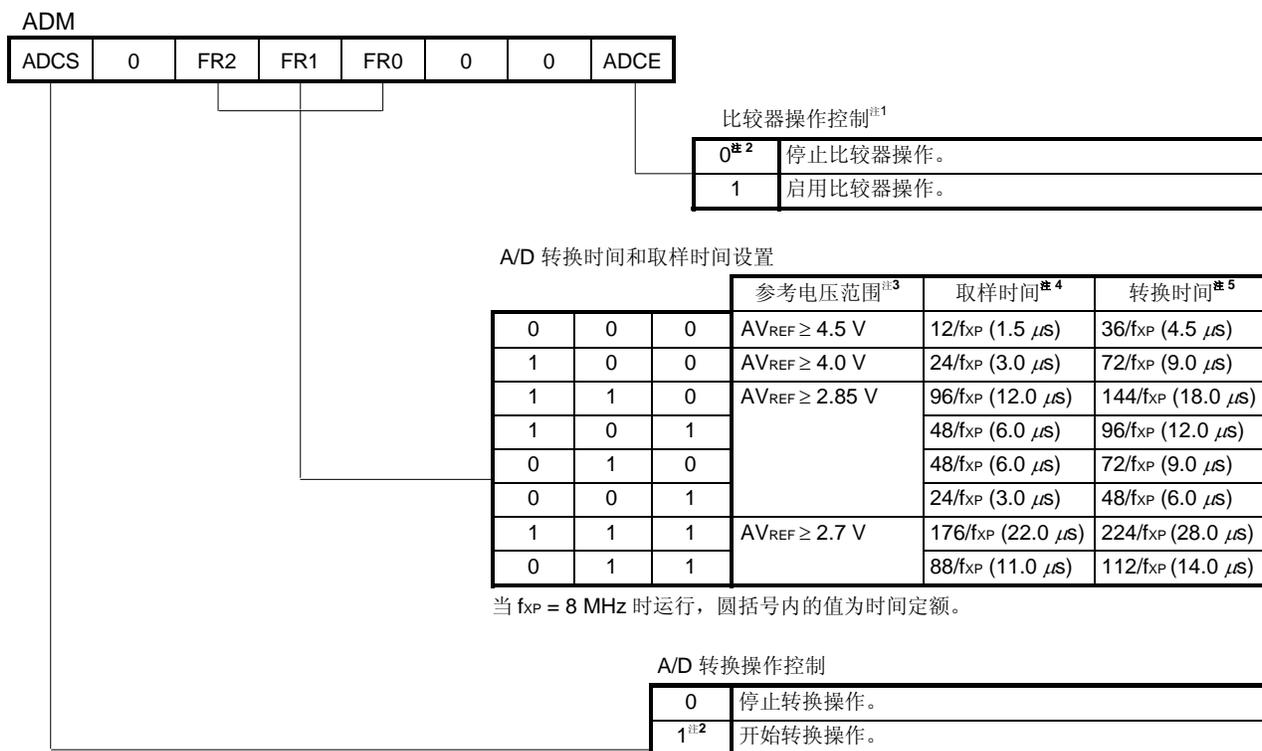
注意事项 1. 可以随意运行第 <1>步至第 <3>步。

2. 在第<2>步和第<5>步之间, 留至少 1  $\mu\text{s}$  的时间间隔。

### (1) ADM 寄存器设置

本寄存器设置需要 A/D 转换的模拟输入的转换时间, 并开始或停止转换操作。

图 4-1. A/D 转换器模式寄存器 (ADM) 的格式



#### 备注

1.  $f_{XP}$ : 提供给外围硬件的时钟振荡频率
2. 转换时间参考取样时间和连续比较取样值的时间的总数, 直到输出转换结果。

(下页列出注和注意事项。)

- 注 1. ADCS 和 ADCE 控制比较器的操作，从开始操作时间直到稳定需要  $1\ \mu\text{s}$ 。因此，自从 ADCE 设为 1，至少  $1\ \mu\text{s}$  流逝后，设置 ADCS 为 1，从第一个转换数据开始转换数据有效。如果没有等待至少  $1\ \mu\text{s}$ ，ADCS 设为 1，忽略第一个转换数据。

表 4-1. ADCS 和 ADCE 设置

ADCS	ADCE	A/D 转换操作
0	0	已停止(不存在DC电耗路径。)
0	1	转换等待模式(仅比较器耗电。)
1	×	转换模式

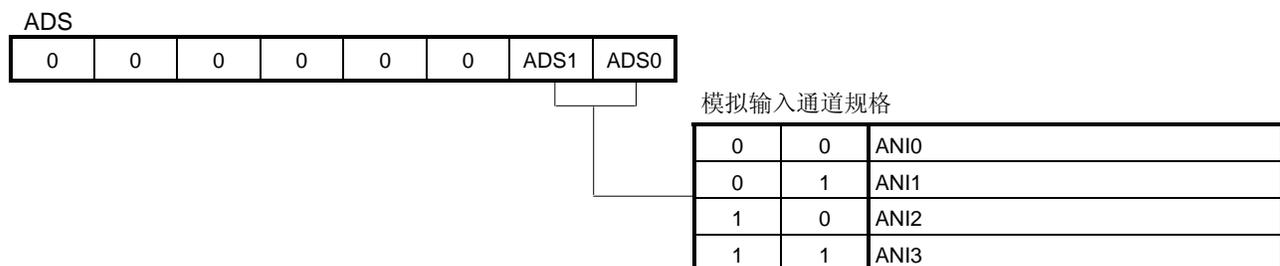
- 即使当 ADCE 为 0 (停止比较器的运行)，如果 ADCS 设为 1，A/D 转换操作启动。然而，忽略第一个转换数据，因为其在担保值范围之外。
- 确信设定 FR2、FR1、FR0 符合参考电压范围，以满足注 4、注 5 和以下。  
示例：当  $AV_{REF} \geq 2.7\ \text{V}$  时，则  $f_{XP} = 8\ \text{MHz}$ 
  - 取样时间至少  $11.0\ \mu\text{s}$ ，A/D 转换时间至少为  $14.0\ \mu\text{s}$  且小于  $100\ \mu\text{s}$ 。
  - 设 FR2、FR1、FR0 为 0、1、1 或 1、1、1。
- 设置取样时间如下：
  - $AV_{REF} \geq 4.5\ \text{V}$ ：至少  $1.0\ \mu\text{s}$
  - $AV_{REF} \geq 4.0\ \text{V}$ ：至少  $2.4\ \mu\text{s}$
  - $AV_{REF} \geq 2.85\ \text{V}$ ：至少  $3.0\ \mu\text{s}$
  - $AV_{REF} \geq 2.7\ \text{V}$ ：至少  $11.0\ \mu\text{s}$
- 设置 A/D 转换时间如下：
  - $AV_{REF} \geq 4.5\ \text{V}$ ：至少  $3.0\ \mu\text{s}$  且小于  $100\ \mu\text{s}$
  - $AV_{REF} \geq 4.0\ \text{V}$ ：至少  $4.8\ \mu\text{s}$  且小于  $100\ \mu\text{s}$
  - $AV_{REF} \geq 2.85\ \text{V}$ ：至少  $6.0\ \mu\text{s}$  且小于  $100\ \mu\text{s}$
  - $AV_{REF} \geq 2.7\ \text{V}$ ：至少  $14.0\ \mu\text{s}$  且小于  $100\ \mu\text{s}$

- 注意事项
- 上述取样时间次数和转换次数不包含时钟频率错误。选择满足注 4 和注 5 中描述的条件取样时间和转换时间，考虑时钟频率错误(使用高速内部振荡器时最大误差范围为  $\pm 5\%$ )。
  - 当 A/D 转换(ADCS = 0)停止时开始 A/D 转换，除 ADM 的 ADCS 外的位操作，执行 2 个 NOP 指令或等效于两个机器工作周期的一个指令之后，设 ADCS 为 1。
  - 重写 FR0 位至 FR2 位之前，停止 A/D 转换(ADCS = 0)。
  - 确保将 6, 2, 1 位清除为“0”。

**(2) ADS 寄存器设置**

寄存器指定了要进行 A/D 转换的模拟电压的输入端口。

图 4-2. 模拟输入通道规格寄存器的格式 (ADS)

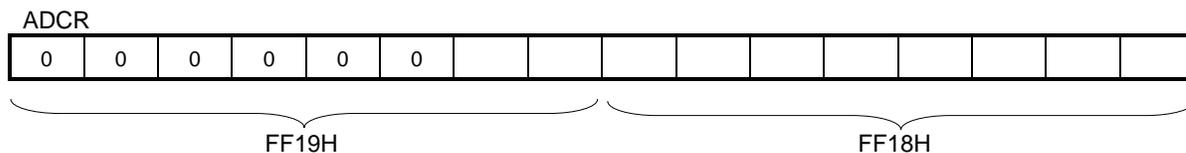


**注意事项** 确保将 2 至 7 位清除为“0”。

**(3) ADCR 寄存器操作**

此寄存器为保留 A/D 转换结果的 16 位只读寄存器。高六位固定至 0。每次 A/D 转换结束，按 FF19H 的第 1 位顺序开始，从连续的近似值的寄存器加载转换结果并将转换结果存储至 ADCR。FF19H 标明转换结果的高两位，FF18H 表明转换结果的低 8 位。

图 4-3. 10 位 A/D 转换结果寄存器的格式(ADCR)

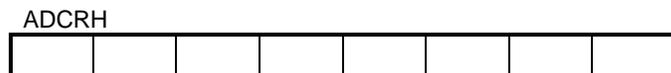


**注意事项** 在写入 ADM 和 ADS 寄存器后，ADCR 寄存器的内容可能变不明确的。完成转换操作后，写入 ADM 和 ADS 寄存器前，读出转换结果。上述提及的除外，正确的转换结果可能不会定时读出。

**(4) ADCRH 寄存器操作**

此寄存器位为保留 A/D 转换结果的 8 位只读寄存器。该寄存器存储了 10 位分辨率结果的高 8 位。

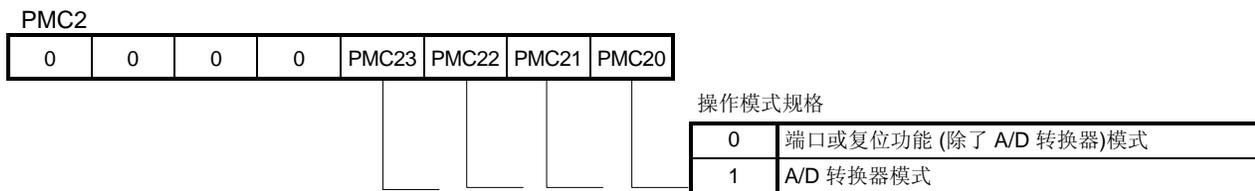
图 4-4. 8 位 A/D 转换结果寄存器的格式(ADCRH)



**(5) PMC2 寄存器和 PM2 寄存器设置**

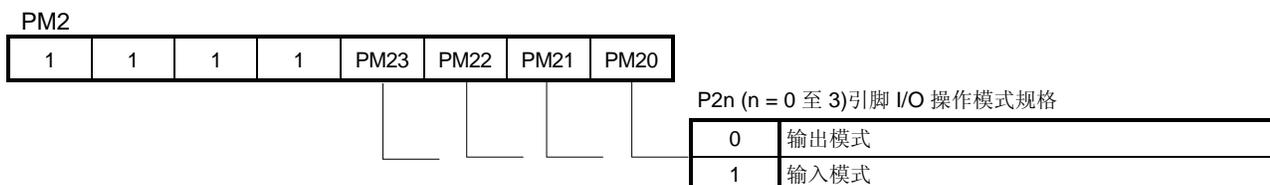
使用 ANI0/P20 至 ANI3/P23 引脚作为模拟输入，设 PMC20 至 PMC23 和 PM20 至 PM23 为 1。

图 4-5. 端口模式控制寄存器 2 的格式 (PMC2)



**注意事项** 当 PMC20 至 PMC23 被设为 1, P20/ANI0 至 P23/ANI3 引脚不能用作端口引脚。 , 确保将 A/D 转换器模式引脚的上拉电阻选项寄存器设 (PU20 至 PU23) 为 0。

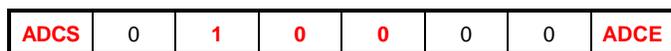
图 4-6. 端口模式寄存器 2 的格式 (PM2)



**[示例]** 通过设模拟输入通道为 ANI0 并设置 A/D 转换时间为  $9\ \mu\text{s}$ ，开始 A/D 转换操作。  
 (提供给外围硬件的时钟振荡频率( $f_{XP}$ ) = 8 MHz)  
 (在本示例程序源内容相同)

**(1) 寄存器设置**

<1> ADM



比较器操作控制

0	停止比较器操作
1	启用比较器操作.

A/D转换时间和取样时间设置

	参考电压范围	取样时间	转换时间
1 0 0	$AV_{REF} \geq 4.0\text{ V}$	$24/f_{XP}$ (3.0 $\mu\text{s}$ )	$72/f_{XP}$ (9.0 $\mu\text{s}$ )

当运行  $f_{XP} = 8\text{ MHz}$ ，圆括号内的值为时间定额。

A/D 转换操作控制

0	停止转换操作
1	开始转换操作

<2> ADS



模拟输入通道规格

0	0	ANI0
---	---	------

<3> PMC2



操作模式规格

1	A/D转换器模式
---	----------

<4> PM2



P2n (n = 0 至 3) 引脚 I/O 操作模式规格

1	输入模式
---	------

## (2) 示例程序

## &lt;1&gt; 汇编语言

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

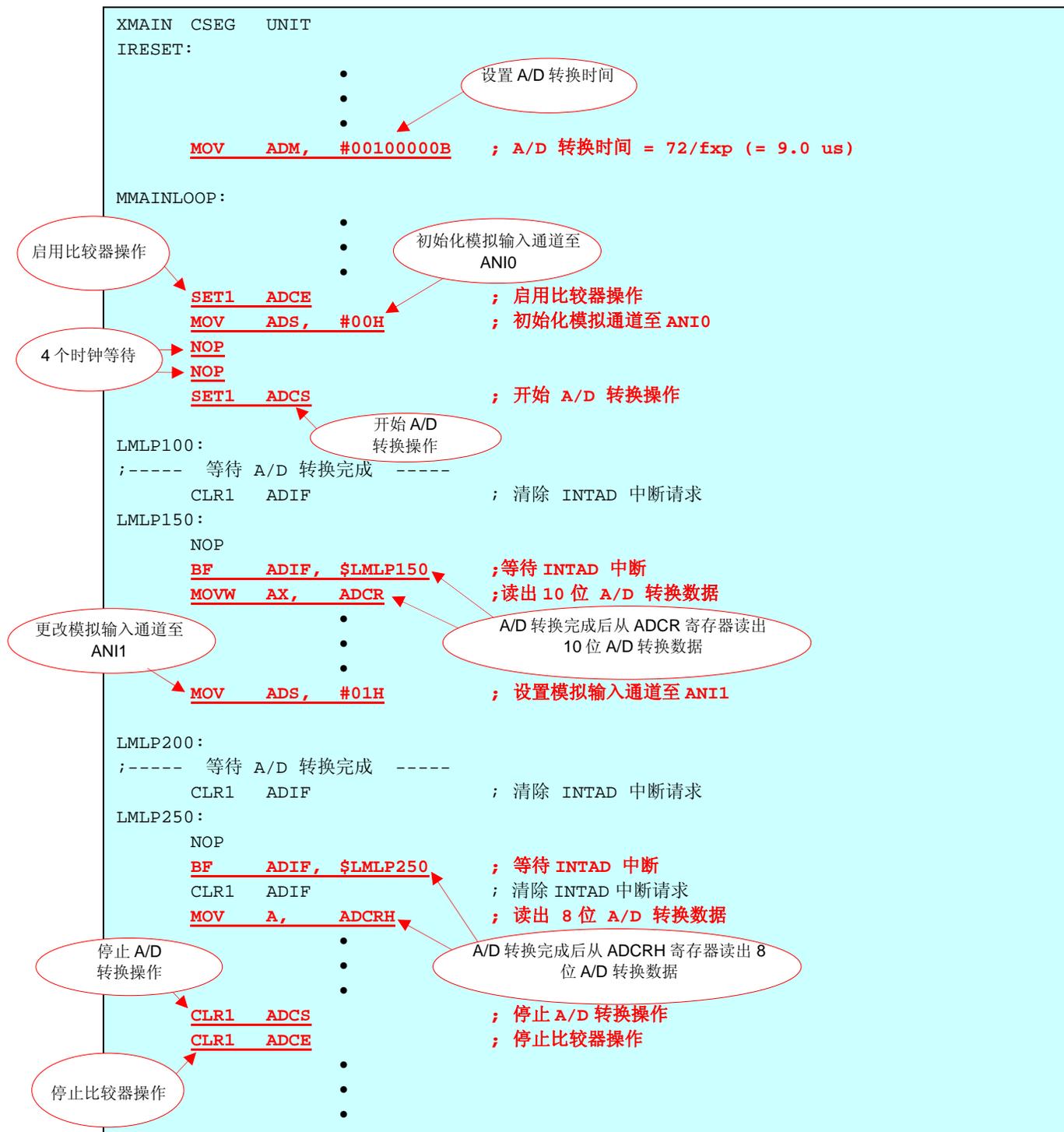
## &lt;2&gt; C 语言

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

[本示例程序源文件节选]

附录 A 程序列表中与 A/D 转换器函数有关的节选如下所示 (与上述 [示例] 中提到的内容相同)。

### (1) 汇编语言



## (2) C 语言

```

void hdwinit(void){
    unsigned char ucCnt200us; /* 200 us 等待的 8 位变量 */
    .
    .
    .
    ADM = 0b00100000; /* A/D 转换时间 = 72/fxp (= 9.0 us) */
    return;
}

void main(void){
    .
    .
    .
    ADCE = 1; /* 启用比较器操作 */
    ADS = 0x00; /* 初始化模拟输入通道至 ANI0 */
    NOP();
    NOP();
    ADCS = 1; /* 开始 A/D 转换操作 */
    /* 开始 A/D 转换操作 */
    /* INTMO =
0b00000000;

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 运行 4 次 A/D 转换处理*/
    {
        ADIF = 0; /* 清除 INTAD 中断请求 */

        while (!ADIF) /* 等待 INTAD 中断 */
        {
            NOP();
        }
        g_ushnAdBuff0[ucTimes] = ADCR; /* 存储 10 位 A/D 转换数据 */
    }

    ADS = 0x01; /* 设模拟输入通道至 ANI1 */

    for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 运行 4 次 A/D 转换处理*/
    {
        ADIF = 0; /* 清除 INTAD 中断请求 */
        while (!ADIF) /* 等待 INTAD 中断 */
        {
            NOP();
        }
        g_ucAdBuff1[ucTimes] = ADCRH; /* 存储 8 位 A/D 转换数据 */
    }

    ADCS = 0; /* 停止 A/D 转换操作 */
    ADCE = 0; /* 停止比较器操作 */
    .
    .
    .

```

设置 A/D 转换时间

启用比较器操作

初始化模拟输入通道至 ANI0

4 个时钟等待

开始 A/D 转换操作

A/D 转换完成后，从 ADCR 寄存器存储 10 位 A/D 转换数据

存储 10 位 A/D 转换数据

改变模拟输入通道至 ANI1

A/D 转换完成后从 ADCRH 寄存器存储 8 位 A/D 转换数据

停止 A/D 转换操作

停止比较器操作

## 4.2 输入电压和 A/D 转换结果

模拟输入引脚的模拟输入电压( $V_{AIN}$ )和理论 A/D 转换结果 (ADCR 寄存器)的关系由下列表达式表达:

### ● ADCR 寄存器(10 位分辨率)

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

或

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

**备注** INT ( ): 圆括号内函数返回值的整数部分  
 $V_{AIN}$ : 模拟输入电压  
 $AV_{REF}$ :  $AV_{REF}$  引脚电压  
 ADCR: 10 位 A/D 转换结果寄存器 (ADCR)值

计算示例: 当模拟输入电压为 1.96 V 时,  $AV_{REF}$  引脚电压为 5 V。

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

**注** 有两类 A/D 转换结果寄存器。

- ADCR 寄存器: 存储 A/D 转换结果 (10 位分辨率)
- ADCRH 寄存器: 存储 A/D 转换结果的高 8 位 (10 位分辨率)

## 第五章 用系统仿真器SM+进行运行检查

本章描述利用汇编语言文件（源文件+项目文件），示例程序在适用 78K0S/Kx1+的系统仿真器 SM+中如何运行；该汇编语言文件通过选择  图标进行下载。

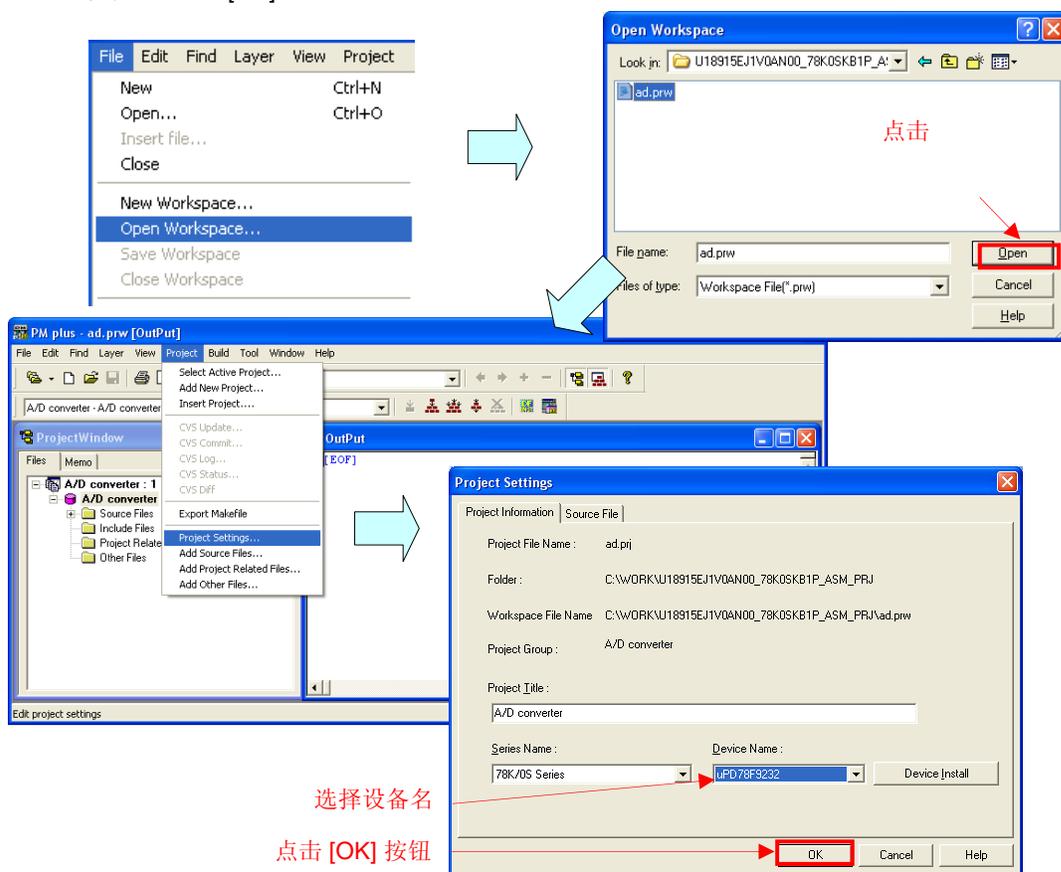
**注意事项** 适用 78K0S/Kx1+的系统仿真器 SM+在 78K0S/KU1+微控制器中不支持（至 2007 年 10 月）。因此，78K0S/KU1+微控制器的运行无法由适用 78K0S/Kx1+的系统仿真器 SM+进行检查。

### 5.1 构建示例程序

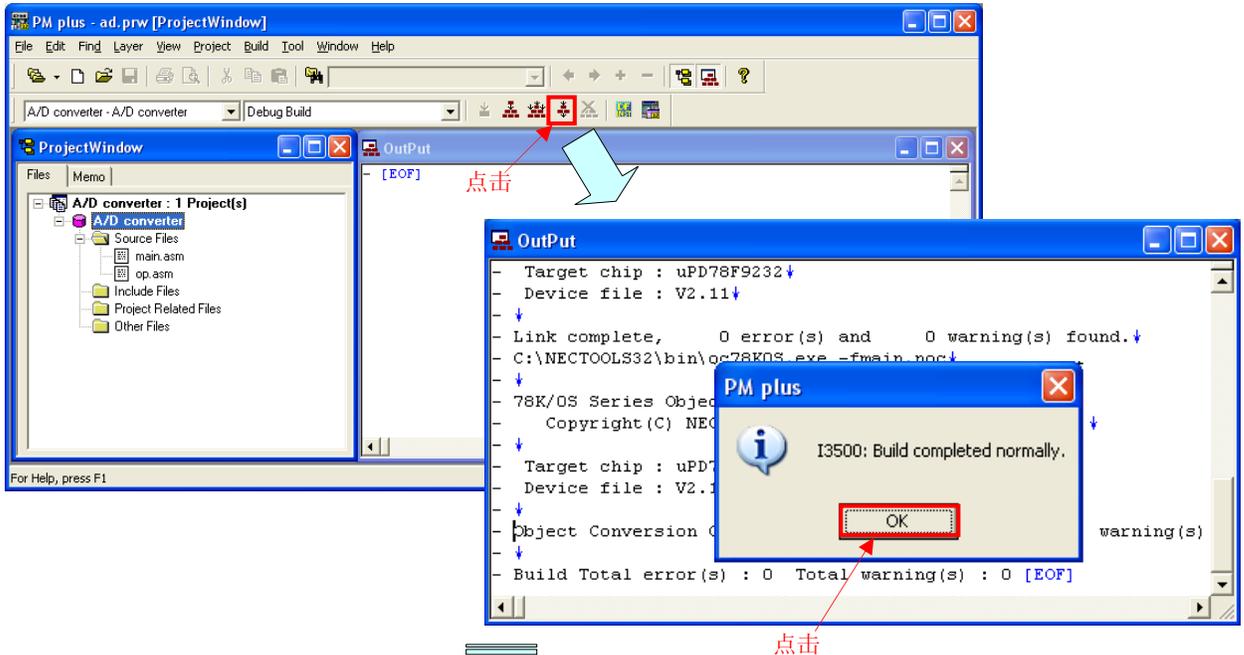
要利用适用 78K0S/Kx1+的系统仿真器SM+（下文称为“SM+”）检查示例程序的运行，在构建示例程序后必须启动SM+。本节描述运行次序的示例，从用集成开发环境PM+构建示例程序到启动SM+；使用  选择下载的汇编语言文件（源文件+项目文件）。关于如何构建其他下载的程序，请参见[78K0S/Kx1+示例程序启动向导的应用注释](#)的“第三章注册集成开发环境PM+项目并执行构建”。

关于如何操作PM+的详情，请参见[PM+项目管理器用户手册](#)。

- (1) 启动 PM+。
- (2) 在[文件]菜单点击[打开]再点击[打开工作区]，选择“ad.prw”。工作区生成，源文件将自动放入该工作区中。
- (3) 从[项目]菜单选择[项目设置]。[项目设置]窗口打开后，选择要用的设备的名称（默认选择具有最大 ROM 或 RAM 的设备），点击[OK]。



- (4) 点击  ([构建→调试]按钮)。当“main.asm”和“op.asm”源文件正常构建时，将显示信息“I3500: Build completed normally. (构建正常完成)”。
- (5) 点击消息窗口中的[OK]按钮，自动启动 SM+。



SM+ 自动启动。

## 5.2 随SM+运行

本节描述在 SM+的 I/O 面板窗口或时间图窗口中对运行进行检查的示例。  
关于如何操作SM+的详情，请参见[SM+系统仿真器操作用户手册](#)。



### [列] 构建错误

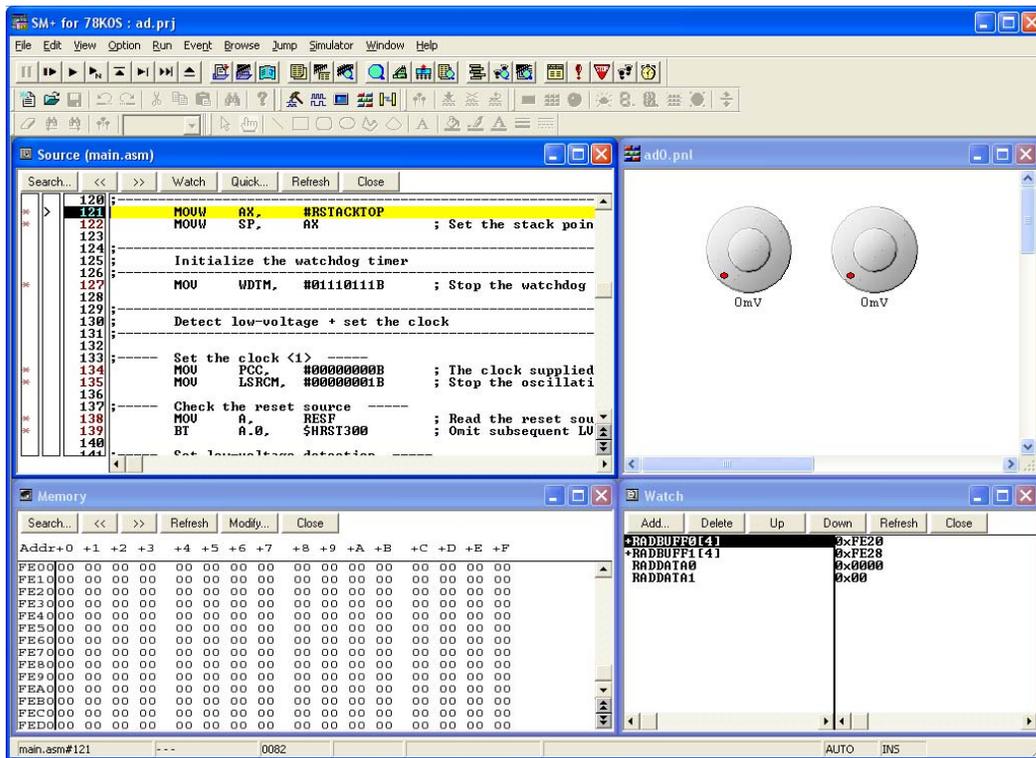
当显示如下错误信息“A006 File not found ‘C:\NECTOOLS32\LIB78K0S\0sl.rel’”或“\*\*\* ERROR F206 Segment ‘@@DATA’ can’t allocate to memory - ignored.”时，根据下列步骤改变编译器设置。当用 PM+进行构建时

- <1> 在[工具]菜单选择 [编译器选项]。
- <2> 显示 [编译器选项]对话框。选择 [启动例程]工具条。
- <3> 未检查 [使用标准库的固定区域]的复选框。(不管其他的复选框。)

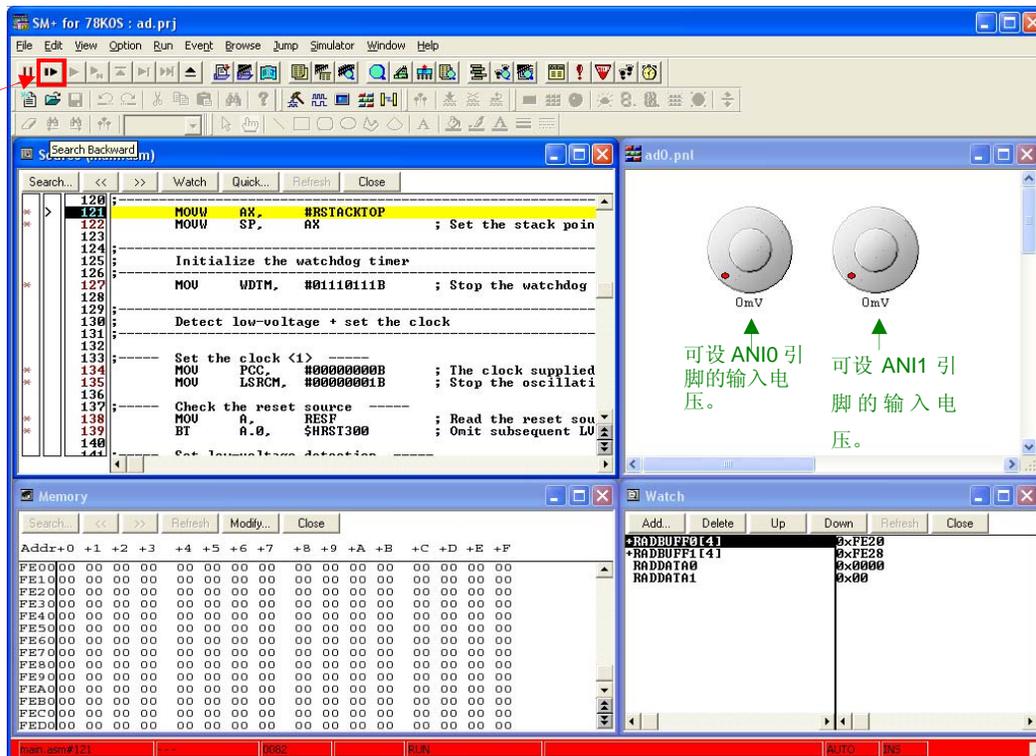
当未检查[使用标准库的固定区]复选框时，启用 118 字节 RAM 区作为一固定标准库区；可是，标准库（getchar 函数和 malloc 函数）将被禁止使用。

当点击  图标下载的文件在本示例程序中使用，默认未经检查的[使用标准库的固定区]复选框。

- (1) 点击PM+中的[构建→调试]启动SM+（参见5.1），屏幕显示如下。  
 (这是汇编语言源文件使用的屏幕示例。)

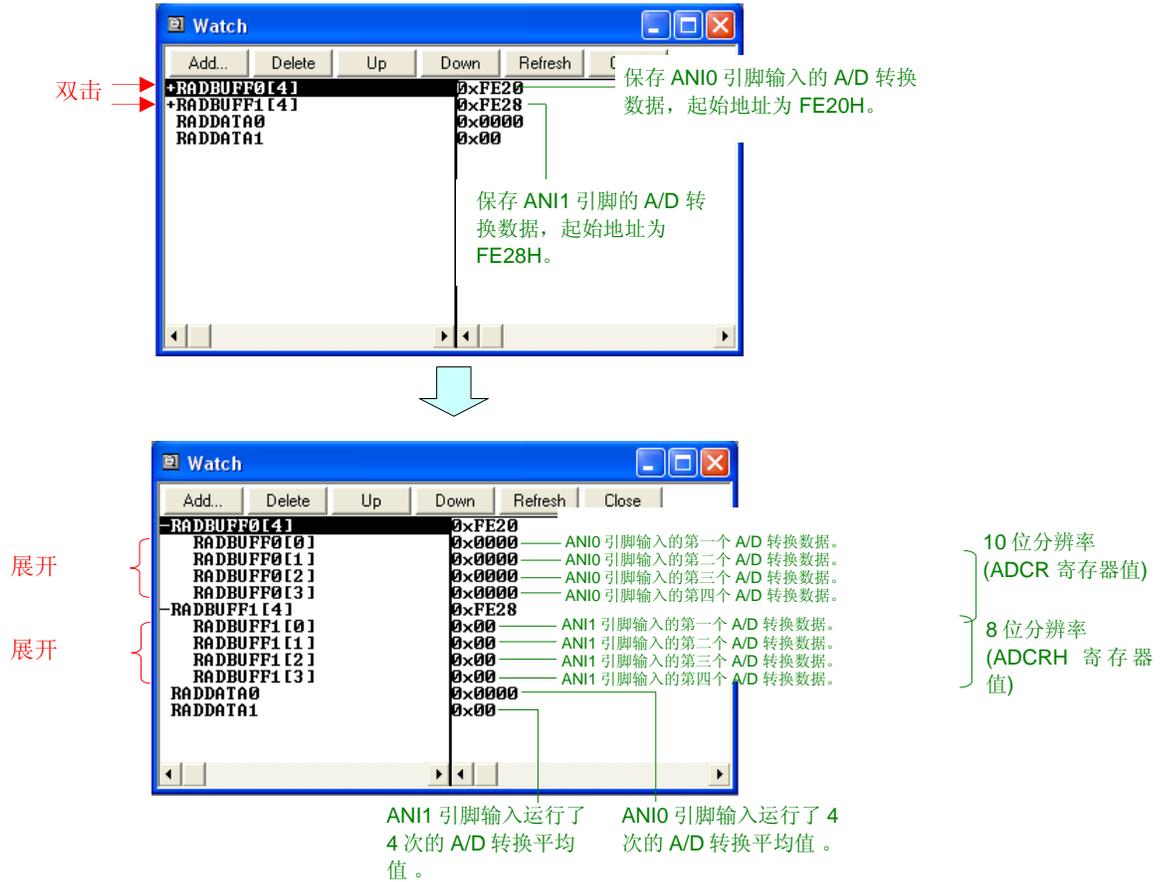


- (2) 点击  ([重启]按钮)。CPU 复位后，将执行本程序，屏幕显示如下。



程序执行过程中变为红色。

(3) 选择[察看]窗口并双击“+RADBUFF0 [4]”和“+RADBUFF1 [4]”，第一个字符从加号 (“+”) 变为减号 (“-”)，且数据将展开并显示在“-RADBUFF0 [4]”和“-RADBUFF1 [4]”下方。



(4) 程序执行期间，通过拖动位于 I/O 面板窗口水准仪上的运动点（红点）来改变输入电压。检查输入电压的改变是否 [察看]窗口和 [存储]窗口 A/D 转换数据有变化。

- 备注**
1. ANI0 引脚和 ANI1 引脚输入电压和 A/D 转换数据之间的关系，请参阅 [4.2 输入电压和 A/D 转换结果](#)。
  2. 设 A/D 转换器参考电压输入 (AVREF) 默认为 5V。要改变 AVREF 电压值，必须在输入面板添加水准仪。关于水准仪的添加，在属性设置中，设连接引脚为“AVREF”，最大输入值为任意值。属性设置后，选择“输入模拟”和拖动位于 I/O 面板窗口上的水准仪运动点进行设置 AVREF 电压值。

示例 1: ANIO pin 输入电压: 1,960 mV, ANI1 引脚输入电压: 0 mV

I/O 面板窗口

ANIO 引脚输入电压为 1,960 mV

[察看] 窗口

[存储] 窗口

Addr	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
FE00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE20	91	01	91	01	91	01	91	01	00	00	00	00	91	01	00	00
FE30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FEA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FEB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FE00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
FED0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

示例 2: ANI0引脚输入电压: 1,960 mV, ANI1 引脚输入电压: 1,960 mV

I/O 面板窗口

[察看] 窗口

[存储] 窗口

## 第六章 相关文档

文档名称		日语/英语
78K0S/KU1+ 用户手册		<a href="#">PDF</a>
78K0S/KY1+ 用户手册		<a href="#">PDF</a>
78K0S/KA1+ 用户手册		<a href="#">PDF</a>
78K0S/KB1+ 用户手册		<a href="#">PDF</a>
78K/0S 系列指令用户手册		<a href="#">PDF</a>
RA78K0S 汇编包用户手册	语言	<a href="#">PDF</a>
	操作	<a href="#">PDF</a>
CC78K0S C 编译器用户手册	语言	<a href="#">PDF</a>
	操作	<a href="#">PDF</a>
PM+ 项目管理器用户手册		<a href="#">PDF</a>
SM+ 系统仿真器操作用户手册		<a href="#">PDF</a>
78K0S/KA1+ 简体闪存写入手册 MINICUBE2 信息		<a href="#">PDF</a>
78K0S/Kx1+ 应用注释	示例程序启动向导	<a href="#">PDF</a>
	示例程序 (初始设置) LED 照明开关控制	<a href="#">PDF</a>
	示例程序 (中断) 开关输入引起的外部中断	<a href="#">PDF</a>
	示例程序 (低电压检测)检测电压低于2.7V时产生复位	<a href="#">PDF</a>

## 附录 A 程序列表

作为程序列表示例，78K0S/KB1+微控制器的源程序如下所示。

### ● main.asm (汇编语言版)

```

;*****
;
;
;       日电电子   78K0S/KB1+
;
;*****
;
;       78K0S/KB1+   示例程序
;*****
;
;       A/D 转换器
;*****
;<<记录>>
;
;       2007.8.--   发布
;*****
;
;<<概述>>
;
;本示例程序展示了使用 A/D 转换器的例子。模拟输入至 ANI0 引脚和 ANI1 引脚的 A/D 转换运行 4 次，
;转换结果保存至 RAM 区。用 INTAD 中断请求标志还原处理读出 A/D 转换结果。另外，ANI0 引脚的
;A/D 转换器数据用 10 位分辨率保存并由 ADCR 寄存器读出。ANI1 引脚 A/D 转换数据用 8 位分辨率保存并由
;ADCRH 寄存器读出。计算各个数据的平均值并保存至 RAM 区。
;
;
;
; <主要设置内容>
;
; - 停止看门狗定时器的运行
; - 将低压检查电压 (VLVI) 设置为 4.3V±0.2V
; - 在 VDD >= VLVI 后当 VDD < VLVI 时生成内部复位信号 (低压检测器)
; - 设置 CPU 时钟为 8MHz
; - 设置提供给外围硬件的时钟为 8MHz
; - 将 A/D 转换器转换时间设置为 9.0 us
;
;
;
; < A/D 转换结构及平均值数据保存位置>
;
;
; +-----+
; | 标签 | 数据 |      数据类型      | A/D 转换端口 |
; | 名称 | 长度 |                    |              |
; |-----|
; |RADBUFF0|16 位|10 位 A/D 转换数据(第 1 次)| P20/ANI0 |
; |      |16 位|10 位 A/D 转换数据(第 2 次)| P20/ANI0 |
; |      |16 位|10 位 A/D 转换数据(第 3 次)| P20/ANI0 |
; |      |16 位|10 位 A/D 转换数据(第 4 次)| P20/ANI0 |
; |-----|
; |RADBUFF1| 8 位| 8 位 A/D 转换数据 (第 5 次)| P21/ANI1 |
; |      | 8 位| 8 位 A/D 转换数据 (第 6 次)| P21/ANI1 |
; |      | 8 位| 8 位 A/D 转换数据 (第 7 次)| P21/ANI1 |

```

```

; |      | 8 位| 8 位 A/D 转换数据 (第 8 次)| P21/ANI1 |
; |-----|
; |RADATA0| 16 位| 1 至 4 次数据平均值| - |
; |RADATA1| 8 位| 5 至 8 次数据平均值| - |
; +-----+
;
;
;
; <<I/O 端口设置>>
;
;
; 输入: P20, P21
; 输出: P00-P03, P22, P23, P30-P33, P40-P47, P120-P123, P130
; # 所有不使用的端口设为输出端口。
;
;
; *****

```

```

;=====
;
; 向量表
;
;=====

```

```

XVCT  CSEG  AT      0000H
      DW  IRESET   ;(00)  RESET
      DW  IRESET   ;(02)  --
      DW  IRESET   ;(04)  --
      DW  IRESET   ;(06)  INTLVI
      DW  IRESET   ;(08)  INTP0
      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

```

```

;=====
;
; 定义 RAM
;
;=====

```

```

DRAM  DSEG  SADDRP
RADBUFF0:  DS   8      ; 存储 10 位 A/D 转换结果
RADBUFF1:  DS   4      ; 存储 8 位 A/D 转换结果
RADATA0:   DS   2      ; 存储 10 位 A/D 转换结果平均值
RADATA1:   DS   1      ; 存储 8 位 A/D 转换结果平均值

```

```

;=====
;
; 定义存储体区
;
;=====

```

```

DSTK  DSEG  AT      0FEE0H

```

```

RSTACKEND: DS    20H           ; 存储体区 = 32 字节
RSTACKTOP:                ; 存储体区开始地址= FF00H

;*****
;
;      复位后初始化
;
;*****
XMAIN CSEG UNIT
IRESET:
;-----
;      初始化栈指针
;-----
      MOVW AX,    #RSTACKTOP
      MOVW SP,    AX           ; 设置栈指针

;-----
;      初始化看门狗定时器
;-----
      MOV  WDTM, #01110111B   ; 停止看门狗定时器运行

;-----
;      检测低电压 + 设定时钟
;-----

;---- 设定时钟 <1> ----
      MOV  PCC,  #00000000B   ; 提供给 CPU 的时钟(fcpu) = fxp (= fx/4 = 2 MHz)
      MOV  LSRCM, #00000001B  ; 停止低速内部振荡器振荡

;---- 检查复位源 ----
      MOV  A,    RESF         ; 读出复位源
      BT   A.0,  $HRST300    ; 省略后面的与 LVI 有关数据处理, 在 LVI 复位期间进行
SET_CLOCK

;---- 设置低压检测 ----
      MOV  LVIS, #00000000B   ; 将低压检测电平(VLVI)设置为 4.3 V +/-0.2 V
      SET1 LVION             ; 启用低压检测器操作

      MOV  A,    #40          ; 指派 200 us 等待计数值
;---- 200 us 等待 ----
HRST100:
      DEC  A
      BNZ  $HRST100          ; 0.5[us/clock] x 10[clock] x 40[count] = 200[us]

;---- VDD >= VLVI 等待处理 ----
HRST200:
      NOP
      BT   LVIF, $HRST200    ; 如果 VDD < VLVI, 就分支执行

      SET1 LVIMD             ; 当 VDD < VLVI 时, 设置以生成内部复位信号
;---- 设置时钟 <2> ----
HRST300:
      MOV  PPCC, #00000000B   ; 提供给外围硬件的时钟(fxp) = fx (= 8 MHz)

```

; -> 提供给 CPU 的时钟 (fcpu) = fxp = 8 MHz

```

;-----
; 初始化端口 0
;-----
MOV  P0,  #00000000B ;将 P00-P03 输出锁存器设为低
MOV  PM0, #11110000B ;将 P00-P03 设为输出模式

;-----
; 初始化端口 2
;-----
MOV  P2,  #00000000B ;将 P20-P23 输出锁存器设为低
MOV  PMC2, #00000011B ;将 P20 和 P21 设置为 A/D 转换器模式
MOV  PM2, #11110011B ;将 P22 和 P23 为输出模式,将 P20 和 P21 设为输入模式

;-----
; 初始化端口 3
;-----
MOV  P3,  #00000000B ;将 P30-P33 输出锁存器设为低
MOV  PM3, #11110000B ;将 P30-P33 设为输出模式

;-----
; 初始化端口 4
;-----
MOV  P4,  #00000000B ;将 P40-P47 输出锁存器设为低
MOV  PM4, #00000000B ;将 P40-P47 设为输出模式

;-----
; 初始化端口 12
;-----
MOV  P12, #00000000B ;将 P120-P123 输出锁存器设为低
MOV  PM12, #11110000B ;将 P120-P123 设为输出模式

;-----
; 初始化端口 13
;-----
MOV  P13, #00000001B ;将 P130 输出锁存器设为高

;-----
; 设置 A/D 转化器
;-----
MOV  ADM, #00100000B ; A/D 转换时间= 72/fxp (= 9.0 us)

*****
;
; 主循环
;
*****
MMAINLOOP:

;-----
; ANI0 引脚 A/D 转换处理(用 10 位分辨率保存转换结果)
;-----
MOVW HL,  #RADBUFF0 ;指定存储 10 位 A/D 转换数据的地址表

```

```

MOV B, #4 ;指定 A/D 转换数据的个数

SET1 ADCE ;启用比较器操作
MOV ADS, #00H ;初始化模拟输出通道至 ANI0
NOP
NOP
SET1 ADCS ;启动 A/D 转换操作

LMLP100:
;----- 等待 A/D 转换完成 -----
CLR1 ADIF ;清除 INTAD 中断请求
LMLP150:
NOP
BF ADIF, $LMLP150 ;等待 INTAD 中断

;----- 存储转换数据 -----
MOVW AX, ADCR ;读出 10 位 A/D 转换数据
MOV [HL+1], A ;存储高 2 位
XCH A, X
MOV [HL], A ;存储低 8 位
INC L ;地址表增加了 2
INC L
DBNZ B, $LMLP100 ;如果 A/D 转换数字 < 4, 就分支执行

;-----
; ANI1 引脚 A/D 转换处理 (用 8 位分辨率保存转换结果)
;-----
MOVW HL, #RADBUFF1 ;指定存储 8 位转换数据的地址表
MOV B, #4 ;指定 A/D 转换个数

MOV ADS, #01H ;设置模拟输入通道至 ANI1

LMLP200:
;----- 等待 A/D 转换完成 -----
CLR1 ADIF ;清除 INTAD 中断请求
LMLP250:
NOP
BF ADIF, $LMLP250 ;等待 INTAD 中断
CLR1 ADIF ;清除 INTAD 中断请求

;----- 存储转换数据 -----
MOV A, ADCRH ;读出 8 位 A/D 转换数据
MOV [HL], A ;存储 A/D 转换数据
INC L ;地址表增加了 1
DBNZ B, $LMLP200 ;如果 A/D 转换数字 < 4, 就分支执行

CLR1 ADCS ;停止 A/D 转换操作
CLR1 ADCE ;停止比较器操作

;-----
; 计算 10 位 A/D 转换数据平均值 (ANI0 引脚)
;-----
MOVW HL, #RADBUFF0 ;指定存储 10 位 A/D 转换数据地址表

```

```

MOVW AX, #0000H; 清除 AX 寄存器
;----- 增加 -----
MOV B, #4 ; 指定增加个数
LMLP300:
XCH A, X
ADD A, [HL] ; 增加低 8 位
XCH A, X
ADDC A, [HL+1] ; 增加高 2 位 (包括进位的低位)
INC L ; 地址表增加了 1
INC L
DBNZ B, $LMLP300 ; 如果新增加数字 < 4, 就分支执行

;----- 计算平均值 -----
MOV B, #2 ; 指定右移位数字(= x1/2)
LMLP350:
ROR A, 1 ; 右移位高位 1
XCH A, X
RORC A, 1 ; 右移位低位 1 (包括高位转换)
XCH A, X
DBNZ B, $LMLP350 ; 如果右移位个数 < 2, 则分支执行

AND A, #00000011B ; 屏蔽位除了高位 0 和 1
MOVW RADATA0, AX ; 存储平均值(10 位数据) 至 RADATA0

;-----
; 计算 8 位 A/D 转换数据平均值(ANI1 引脚)
;-----
MOVW HL, #RADBUFF1 ; 指定存储 8 位 A/D 转换数据地址表
MOVW AX, #0000H; 清除 AX 寄存器

;----- 新增 -----
MOV B, #4 ; 指定新增数字
LMLP400:
ADD A, [HL] ; 新增
BNC $LMLP420 ; 跳转如果不移位, 就分支执行
INC X ; 增加高位 1
LMLP420:
INC L ; 地址表增加了 1
DBNZ B, $LMLP400 ; 如果新增个数< 4, 则分支执行

;----- 计算平均值 -----
MOV B, #2 ; 指定右移位个数(= x1/2)
LMLP450:
XCH A, X
ROR A, 1 ; 右移高位至 1
XCH A, X
RORC A, 1 ; 右移低位至 1 (包括高位移位)
DBNZ B, $LMLP450 ; 如果右移数字< 2, 就执行分支程序

MOV RADATA1, A ; 存储平均值 (8 位数据) 至 RADATA1

BR !MMAINLOOP ; 到 MMAINLOOP

```

结束

## ● main.c (C 语言版)

```

/*****

```

```

    日电电子 78K0S/KB1+

```

```

*****

```

```

    78K0S/KB1+  示例程序

```

```

*****

```

```

    A/D 转换器

```

```

*****

```

```

<<记录>>

```

```

    2007.8.--    发布

```

```

*****

```

```

<<概述>>

```

本实例程序展示了使用 A/D 转换器的例子。模拟输入 ANI0 引脚和 ANI1 引脚 A/D 转换运行 4 次，转换结果保存至 RAM 区。用 INTAD 中断请求标记还原处理读出 A/D 转换结果。此外，ANI0 引脚 10 位 A/D 转换数据用 10 位分辨率保存由 ADCR 寄存器读出，ANI1 引脚 A/D 转换数据用 8 位分辨率保存由 ADCRH 寄存器读出。计算各个数据平均值并保存至 RAM 区。

<主要设置内容>

- 停止看门狗定制器运行
- 设置低电压检测电压(VLVI)至 4.3 V +0.2 V
- 当 VDD < VLVI 时，VDD >= VLVI 后产生一个内部复位信号(低电压检测器)
- 设置 CPU 时钟至 8 MHz
- 将向外围硬件供应的时钟设为 8 MHz
- 将 A/D 转换器转换时间设为 9.0 us

<A/D 转换结果和数据平均值存储位置>

变量名称	数据长度	数据类型	A/D 转换端口
g_ushnAdBuff0	16 位	10 位 A/D 转换数据 (第 1 次)	P20/ANI0
	16 位	10 位 A/D 转换数据 (第 2 次)	P20/ANI0
	16 位	10 位 A/D 转换数据 (第 3 次)	P20/ANI0
	16 位	10 位 A/D 转换数据 (第 4 次)	P20/ANI0
g_ucAdBuff1	8 位	8 位 A/D 转换数据 (第 5 次)	P21/ANI1
	8 位	8 位 A/D 转换数据 (第 6 次)	P21/ANI1
	8 位	8 位 A/D 转换数据 (第 7 次)	P21/ANI1
	8 位	8 位 A/D 转换数据 (第 8 次)	P21/ANI1
g_ushnAdData0	16 位	数据 1 至数据 4 平均值	-
g_unAdData1	8 位	数据 5 至数据 8 平均值	-

<<I/O 端口设置 >>

输入: P20, P21

输出: P00-P03, P22, P23, P30-P33, P40-P47, P120-P123, P130

# 所有不用的端口设为输出端口.

```

*****/

/*=====
    预处理指令 (#pragma)
=====*/
#pragma      SFR          /* SFR 名称可在 C 源电平中描述 */
#pragma      NOP          /* NOP 指令可在 C 源电平中描述 */
/*=====

    定义全局变量
=====*/
sreg static unsigned short int g_ushnAdBuff0[4]; /* 存储 16 位 A/D 转换数据的 16 位变量表 */
sreg static unsigned char g_ucAdBuff1[4];        /* 存储 8 位 A/D 转换数据的 8 位变量表 */
sreg static unsigned short int g_ushnAdData0;    /* 存储 10 位 A/D 转换数据平均值的 16 位变量表 */
*/
sreg static unsigned char g_ucAdData1;          /* 存储 8 位 A/D 转换数据平均值的 8 位变量 */

/*****

    复位后初始化
*****/
void hdwinit(void){
    unsigned char ucCnt200us; /*200 us 等待的 8 位变量 */

/*-----
    初始化看门狗定时器 + 检测低电压 + 设置时钟
-----*/
/* 初始化看门狗定时器*/
WDTM = 0b01110111; /* 停止看门狗定时器操作*/

/* 设置时钟 <1> */
PCC = 0b00000000; /* 供应给 CPU 时钟(fcpu) = fxp (= fx/4 = 2 MHz) */
LSRCM = 0b00000001; /* 停止低速内部振荡器振荡 */

/* 检查复位信号 */
if (!(RESF & 0b00000001)){ /* 在 LVI 复位期间省略后面的与 LVI-有关处理*/

                /* 设置低压检测*/
LVIS = 0b00000000; /* 将低压检测电平 (VLVI) 设为 4.3 V +-0.2 V */
LVION = 1; /* 启用低压检测器操作*/

    for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){ /* Wait of about 200 us */

```

```

        NOP();
    }

    当 (LVIF)时{ /* 等待 VDD >= VLVI */
        NOP();
    }

    LVIMD = 1; /*当 VDD < VLVI 时，设置以生成内部复位信号*/
}

/* 设置时钟 <2> */
PPCC = 0b00000000; /* 供应给外围硬件的时钟 (fxp) = fx (= 8 MHz)
                    -> 供应给 CPU 的时钟(fcpu) = fxp = 8 MHz */

/*-----*/
初始化端口 0
/*-----*/
P0 = 0b00000000; /*将 P30-P33 输出锁存器设为低*/
PM0 = 0b11110000; /*将 P00-P03 设为输出模式 */

/*-----*/
初始化端口 2
/*-----*/
P2 = 0b00000000; /* 将 P20-P23 输出锁存器设为低 */
PMC2 = 0b00000011; /* 将 P20 和 P21 设至 A/D 转换器模式 */
PM2 = 0b11110011; /* 将 P22 和 P23 设为输出模式, P20 和 P21 设为输入模式*/

/*-----*/
初始化端口 3
/*-----*/
P3 = 0b00000000; /* 设 P30-33 输出锁存器为低 */
PM3 = 0b11110000; /* 设 P30-P33 为输出模式 */

/*-----*/
初始化端口 4
/*-----*/
P4 = 0b00000000; /* 将 P40-P47 输出锁存器设为低 */
PM4 = 0b00000000; /* 将 P40-P47 设为输出模式 */

/*-----*/
初始化端口 12
/*-----*/
P12 = 0b00000000; /* 将 P120-P123 输出锁存器设为低 */
PM12 = 0b11110000; /* 将 P120-P123 设为输出模式*/

/*-----*/
初始化端口 13
/*-----*/
P13 = 0b00000001; /* 将 P130 输出锁存器设为高 */

/*-----*/
设置 A/D 转换器
/*-----*/
ADM = 0b00100000; /* A/D 转换时间 = 72/fxp (= 9.0 us) */

```

```

    返回;
}

/*****

    主循环

*****/
void main(void){

unsigned char ucTimes;      /* 计算 A/D 转换个数的 8 位变量 */
unsigned short int ushnAdSum; /* 增加 A/D 转换数据的 16 位变量 */

    while (1)
    {
/*-----
        ANI0 引脚 A/D 转换处理 (用 10 位分辨率保存转换结果)
-----*/
        ADCE = 1;          /* 启用比较器操作 */
        ADS = 0x00;       /* 初始化模拟输入通道至 ANI0 */
        NOP();
        NOP();
        ADCS = 1;        /* 开始 A/D 转换操作 */

        for (ucTimes = 0; ucTimes < 4; ucTimes++) /* 运行 4 次 A/D 转换处理*/
        {
            ADIF = 0;          /* 清除 INTAD 中断请求 */

            while (!ADIF)      /* 等待 INTAD 中断 */
            {
                NOP();
            }

            g_ushnAdBuff0[ucTimes] = ADCR; /* 存储 10 位 A/D 转换数据 */
        }

/*-----
        ANI1 引脚 A/D 转换处理(用 8 位分辨率保存转换结果)
-----*/
        ADS = 0x01;          /* 设模拟输入通道至 ANI1 */

```

```

for (ucTimes = 0; ucTimes < 4; ucTimes++)    /* 运行 4 次 A/D 转换处理 */
{
    ADIF = 0;                                /* 清除 INTAD 中断请求 */

    while (!ADIF) /* 等待 INTAD 中断 */
    {
        NOP();
    }

    g_ucAdBuff1[ucTimes] = ADCRH;            /* 存储 8 位 A/D 转换数据 */
}

ADCS = 0;                                    /* 停止 A/D 转换操作 */
ADCE = 0;                                    /* 停止比较器操作 */

/*-----
   计算 10 位 A/D 转换数据平均值 (ANI0 引脚)
-----*/
ushnAdSum = 0x0000; /* 增加 A/D 转换数据清除变量 */

for (ucTimes = 0; ucTimes < 4; ucTimes++)    /* 增加 4 个 A/D 转换数据 */
{
    ushnAdSum += g_ushnAdBuff0[ucTimes];      /* 增加 10 位 A/D 转换数据 */
}

g_ushnAdData0 = ushnAdSum >> 2; /* 保存 10 位 A/D 转换数据平均值 */

/*-----
   计算 8 位 A/D 转换数据平均值 (ANI1 引脚)
-----*/
ushnAdSum = 0x0000; /* 增加 A/D 转换数据清除变量 */

for (ucTimes = 0; ucTimes < 4; ucTimes++)    /* 增加 4 个 A/D 转换数据 */
{
    ushnAdSum += g_ucAdBuff1[ucTimes]; /* 增加 8 位 A/D 转换数据 */
}

g_ucAdData1 = ushnAdSum >> 2; /* 保存 8 位 A/D 转换数据平均值 */
}
}

```

## ● op.asm (汇编语言和 C 语言共用部分)

```

=====
;
;
;   选项字节
;
;
=====
OPBT      CSEG   AT   0080H
          DB   10011100B ;选项字节区
;
;           ||||
;           |||+-----  软件可以停止低速内部振荡器
;           |++-----  选择告诉内部振荡器时钟(8 MHz) 作为系统时钟源
;           +-----    P34/引脚用作复位引脚

          DB   11111111B ;保护字节区域 (适用于自编程模式)
;
;           |||||
;           ++++++-----所有的模块都可以被写入或擦除
结束

```

## 附录 B 修订记录

版本	出版日期	页码	修订
第一版	2008年2月	-	-

详细信息请联系：

中国区

**MCU 技术支持热线：**

电话：+86-400-700-0606 (普通话)

服务时间：9:00-12:00，13:00-17:00 (不含法定节假日)

网址：

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

**[北京]**

日电电子（中国）有限公司

中国北京市海淀区知春路 27 号

量子芯座 7, 8, 9, 15 层

电话：(+86) 10-8235-1155

传真：(+86) 10-8235-7679

**[深圳]**

日电电子（中国）有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901, 3902, 3909 室

电话：(+86) 755-8282-9800

传真：(+86) 755-8282-9899

**[上海]**

日电电子（中国）有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话：(+86) 21-5888-5400

传真：(+86) 21-5888-5230

**[香港]**

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话：(+852) 2886-9318

传真：(+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话：(+86) 21-5888-5400

传真：(+86) 21-5888-5230

**[成都]**

日电电子（中国）有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话：(+86)28-8512-5224

传真：(+86)28-8512-5334