

应用笔记

78K0S/Kx1+

示例程序（中断）

开关输入引起的外部中断

本文件内容包含示例程序的操作概要，怎样使用示例程序以及示例程序中断功能的设置和使用。在示例程序中，检测到开关输入的下降边缘会导致中断，并展示了 LED 根据开关输入次数不同时的闪亮模式。

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

目录

第一章 概述.....	3
1.1 初始设置的主要内容.....	3
1.2 主循环后的内容.....	4
第二章 电路图.....	5
2.1 电路图.....	5
2.2 外围硬件.....	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 初始设置及运行概览.....	7
3.4 流程图.....	8
第四章 设置方法.....	9
4.1 中断设置.....	9
4.2 中断时的处理.....	15
第五章 用系统仿真器 SM+进行运行检查.....	18
5.1 构建示例程序.....	18
5.2 随 SM+运行.....	19
第六章 相关文档.....	24
附录 A 程序列表.....	25
附录 B 改版历史.....	34

文档编号： U18812CA1V0AN00 (第一版)
 发行日期： 2008 年 3 月 N

- 本档所登载的内容有效期截止至 2008 年 03 月，信息先于产品的生产周期发布。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开发或制造的产品或为日本电气电子株式会社（定义如上）开发或制造的产品。

M5 02.11-1

第一章 概述

本示例程序提供了一个例子，说明中断功能的使用。展示了在检测到开关输入的下降边缘和进行中断服务时，LED 根据开关输入次数变化的发光模式的变化。

1.1 初始设置的主要内容

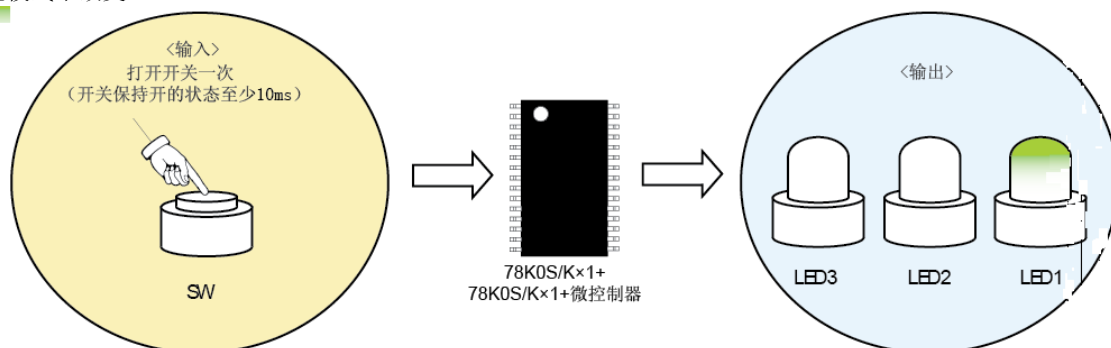
初始设置的主要内容如下。

- 选择高速内部振荡器作为系统时钟源^注
- 停止看门狗计时器的运行
- 设置 CPU 时钟频率和外围硬件时钟频率为 2 MHz
- 设置 I/O 端口
- 设置 INTP1 的有效边缘（外部中断）为下降边缘
- 启用中断

注 用选项字节进行设置。

1.2 主循环后的内容

检测到由开关输入引起的 INTP1 引脚的下降边缘时，中断服务进行。中断服务时，在检测到 INTP1 引脚的下降边缘 10 ms 后，确认开关在开的状态，LED 发光模式改变。如开关为关闭时，经过 10 ms 秒，进行颤动辨别处理，LED 发光模式不改变。



开关输入次数 [#]	LED 输出		
	LED3	LED2	LED1
0	关	关	关
1	关	关	开
2	关	开	关
3	关	开	开
4	开	关	关
5	开	关	开
6	开	开	关
7	开	开	开

注 在第八次开关输入后，重复第零次开关输入时的发光模式。

注意事项 关于使用设备的注意事项，请参见各产品 ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) 的用户手册。



[列] 时钟

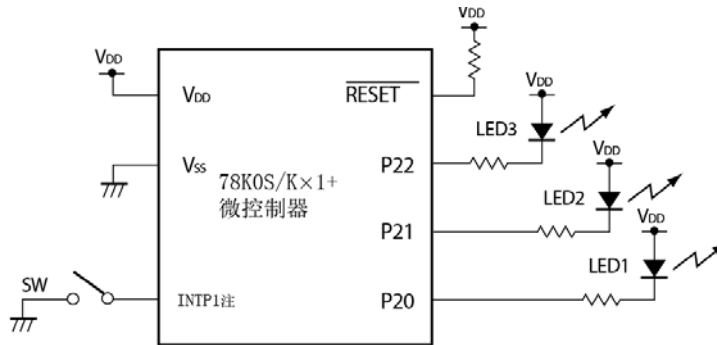
颤动，是在开关按下后瞬间由于接触点的机械开合而导致电信号反复开关的现象。

第二章 电路图

本章描述在本示例程序中所用的电路图和外围硬件。

2.1 电路图

电路图如下。



注 INTP1/P43: 78K0S/KA1+ 和 78K0S/KB1+ 微处理器
INTP1/P32: 78K0S/KY1+ 和 78K0S/KU1+ 微处理器

注意事项 1. 将 AVREF 引脚直接连接到 VDD（仅适用于 78K0S/KA1+ 和 78K0S/KB1+ 微控制器）。
2. 将 AVSS 引脚直接连接到 GND（仅适用于 78K0S/KB1+ 微处理器）。
3. 除电路引脚及其 AVREF 和 AVSS 引脚外，其他所有未使用引脚保留开路（不连接）。

2.2 外围硬件

所用外围硬件如下。

(1) 开关 (SW)

开关用于输入从而控制 LED 点亮。

(2) LEDs (LED1, LED2, LED3)




LEDs 用于相应开关输入的输出。

第三章 软件

本章描述所下载的压缩文件的文件配置、所用微控制器的内部外围功能、示例程序的初始设置和运行概览，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	描述	包含压缩 (*.zip) 文件		
				
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主处理的源文件	● Note 1	● Note 1	
op.asm	用于设置选项字节（设置系统时钟源）的汇编器源文件	●	●	
int.prw	用于集成开发环境PM+的工作区文件		●	
int.prj	用于集成开发环境PM+的项目文件		●	
int.pri int.prs int.prm	用于使用78K0S/Kx1+ 的系统仿真器SM+的项目文件		● Note 2	
int0.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）		● Note 2	●
int0.wvo	适用78K0S/Kx1+的系统仿真器SM+的时间图文件（用于检查波形）			●

- 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。

开关输入：INTP1[#]（外部中断）

LED 输出：P20, P21, P22（输出端口）

注 INTP1/P43: 78K0S/KA1+ 和 78K0S/KB1+微控制器

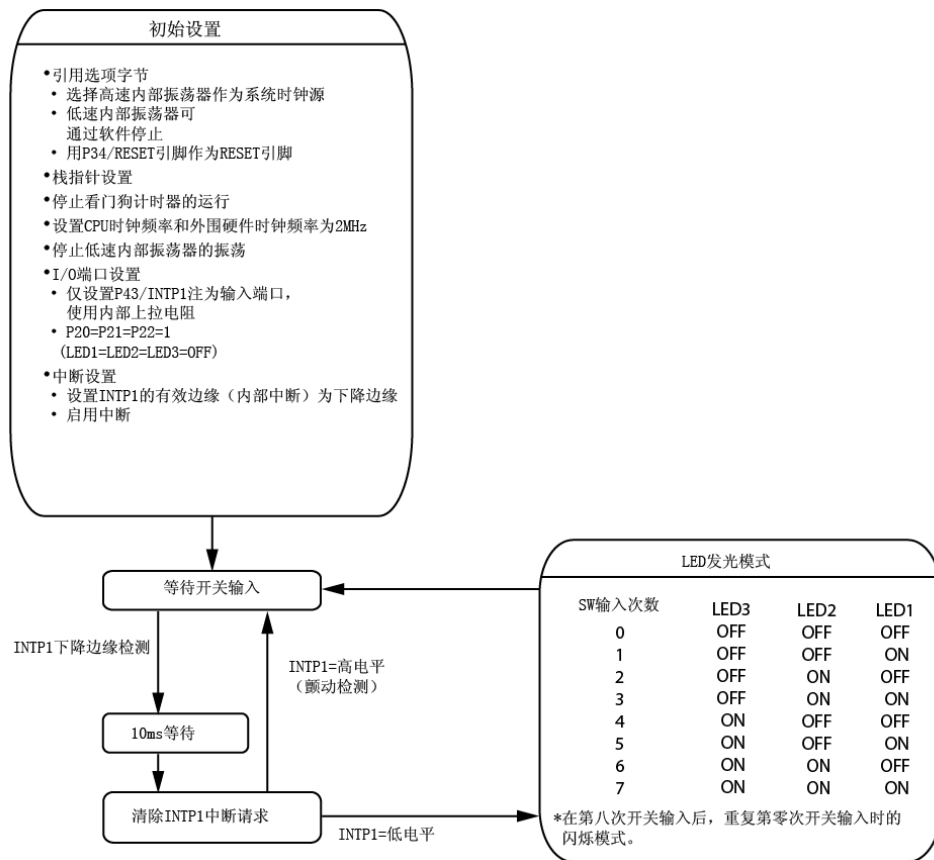
INTP1/P32: 78K0S/KY1+ 和 78K0S/KU1+微控制器

3.3 初始设置及运行概览

在本示例程序中进行的初始设置有：时钟频率的选择、I/O 端口的设置、中断设置及类似设置。

初始设置完成后，利用开关输入(SW)下降边缘的检测进行中断服务，根据开关输入的次數控制三个 LED 灯（LED1, LED2, and LED3）的发光。

详情在如下所示的状态转换图中描述。

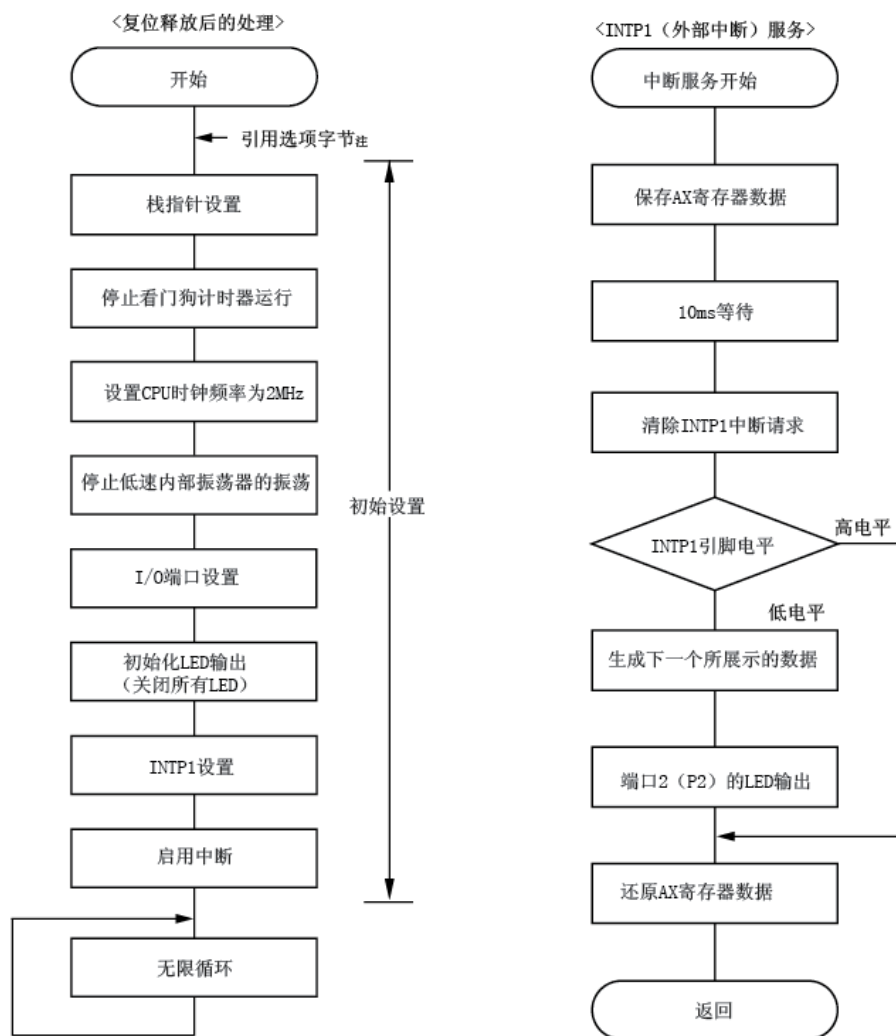


注 INTP1/P43: 78K0S/KA1+ 和 78K0S/KB1+微控制器

INTP1/P32: 78K0S/KY1+ 和 78K0S/KU1+微控制器

3.4 流程图

示例程序的流程图如下所示。



注 对选项字节的引用由微控制器在复位解除后自动进行。在本示例程序中，下面的内容通过引用选项字节进行设置。

- 用高速内部始终（8 MHz（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

本章描述如何设置中断以及中断服务。

关于其他初始设置，请参见[78K0S/Kx1+ 示例程序（初始设置）LED 发光开关控制应用注释](#)。

关于如何设置寄存器，请参见各产品（[78K0S/KU1+](#)、[78K0S/KY1+](#)、[78K0S/KA1+](#)、[78K0S/KB1+](#)）的用户手册。

关于汇编指令，请参见[78K0S系列指令用户手册](#)。

4.1 中断设置

利用下列四种寄存器来控制中断功能。

- 中断请求标志寄存器 0、1^注（IF0、IF1^注）
- 中断屏蔽标志寄存器 0、1^注（MK0、MK1^注）
- 外部中断模式寄存器 0、1^注（INTM0、INTM1^注）
- 程序状态字（PSW）

各种中断请求的中断请求标志和中断屏蔽标志的名称如下。

中断请求信号名称	中断请求标志	中断屏蔽标志
INTLVI	LVIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010
INTAD	ADIF	ADMK
INTP2 ^注	PIF2 ^注	PMK2 ^注
INTP3 ^注	PIF3 ^注	PMK3 ^注
INTTM80 ^注	TMIF80 ^注	TMMK80 ^注
INTSRE6 ^注	SREIF6 ^注	SREMK6 ^注
INTSR6 ^注	SRIF6 ^注	SRMK6 ^注
INTST6 ^注	STIF6 ^注	STMK6 ^注

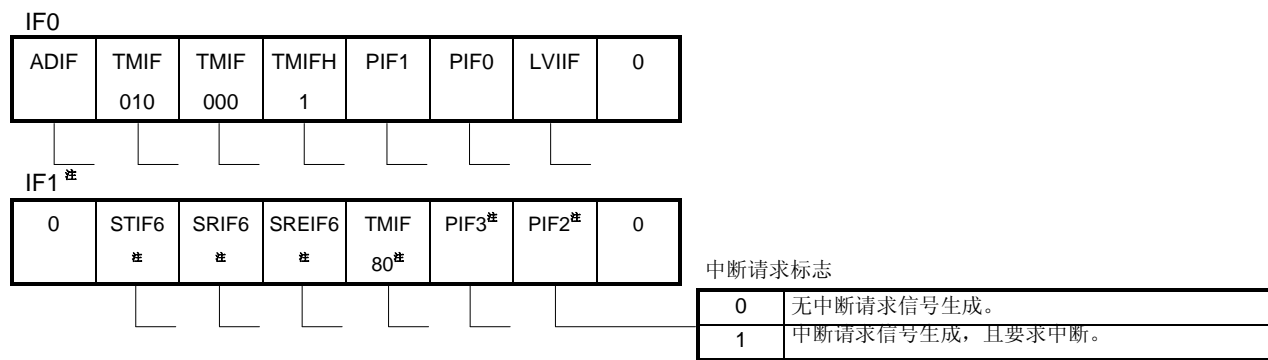
注 仅适用于78K0S/KA1+和78K0S/KB1+ 微控制器

(1) 中断请求标志设置

当相应中断请求生成，或当执行指令时，将中断请求标志设置为 1。当接受中断请求时，复位执行，或指令执行，将中断请求标志清零。

使用中断功能，在生成中断请求前，将所用中断请求标志清零。

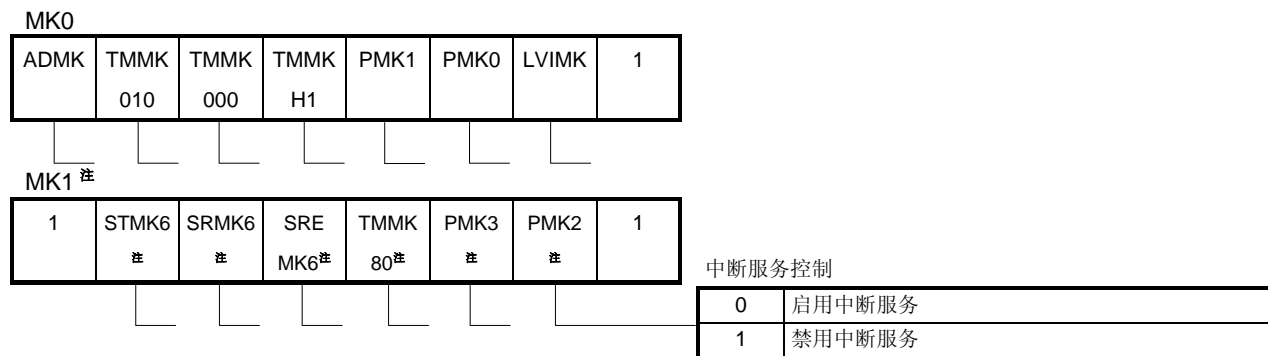
图 4-1. 中断请求标志寄存器 0, 1^注 (IF0, IF1^注)格式

**(2) 中断服务启用/禁用设置**

使用中断屏蔽标志启用（0）或禁用（1）相应中断服务。

使用中断功能，在中断请求生成前，将所用中断屏蔽标志清零。

图 4-2. 中断屏蔽标志寄存器 0, 1^注 (MK0, MK1^注) 格式

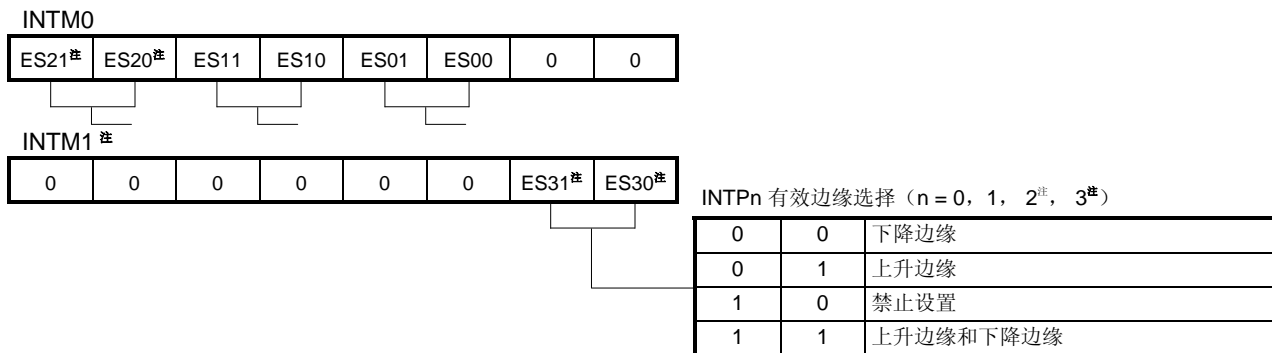


注 仅适用于 78K0S/KA1+ 和 78K0S/KB1+微控制器

(3) 外部中断有效边缘设置

INTM0 和 INTM1^注 用于设置 INTP0、INTP1、INTP2^注、和 INTP3^注 的有效边缘（外部中断）。

图 4-3. 外部中断模式寄存器 0 和 1^注 (INTM0, INTM1^注) 的格式



外部中断组合和有效边缘设置标志如下所示。

设置标志		外部中断
ES31 ^注	ES30 ^注	INTP3 ^注
ES21 ^注	ES20 ^注	INTP2 ^注
ES11	ES10	INTP1
ES01	ES00	INTP0

注 仅适用于 78K0S/KA1+ 和 78K0S/KB1+ 微控制器

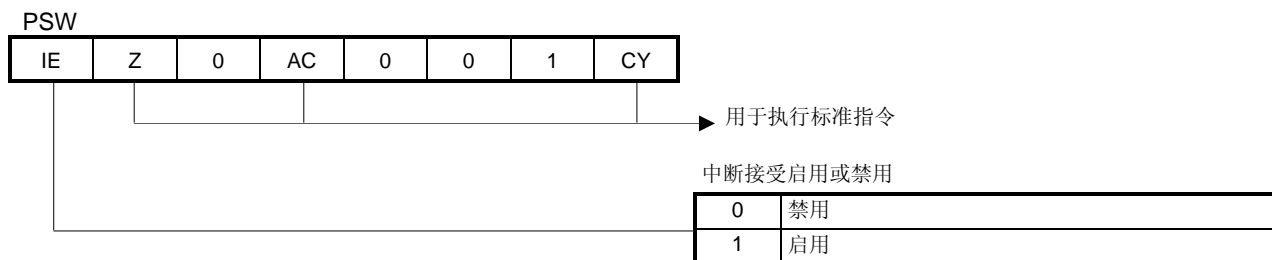
(4) 中断启用/禁用设置

程序状态字 (PSW) 的 IE 标志用于启用 (1) 或禁用 (0) 中断。使用专用指令操作 IE 标志 (EI: 启用中断, DI: 禁用中断)。

当中断屏蔽标志清零且 IE 标志设置为 1, 中断请求生成且当中断请求标志设置为 1 时, 接受中断。

当接受中断后, PSW 自动保存到栈中且将 IE 标志清零。

图 4-4. 程序状态字格式 (PSW)



[示例] 通过 INTP1 下降边缘检测进行中断服务 (与示例程序设置内容相同)

● 寄存器设置

- 将相应 INTP1 的中断请求标志 (PIF1) 清零
- 将相应 INTP1 的中断屏蔽标志 (PMK1) 清零
- 将 INTP1 的有效边缘设置为“下降边缘”
- 通过执行 EI 指令启用中断

IF0

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

中断请求标志

0	无中断请求信号生成
---	-----------

MK0

1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---

中断服务控制

0	启用中断服务
---	--------

INTM0

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

INTP1 有效边缘选择

0	0	下降边缘
---	---	------

PSW

1	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---

启用或禁用中断接受

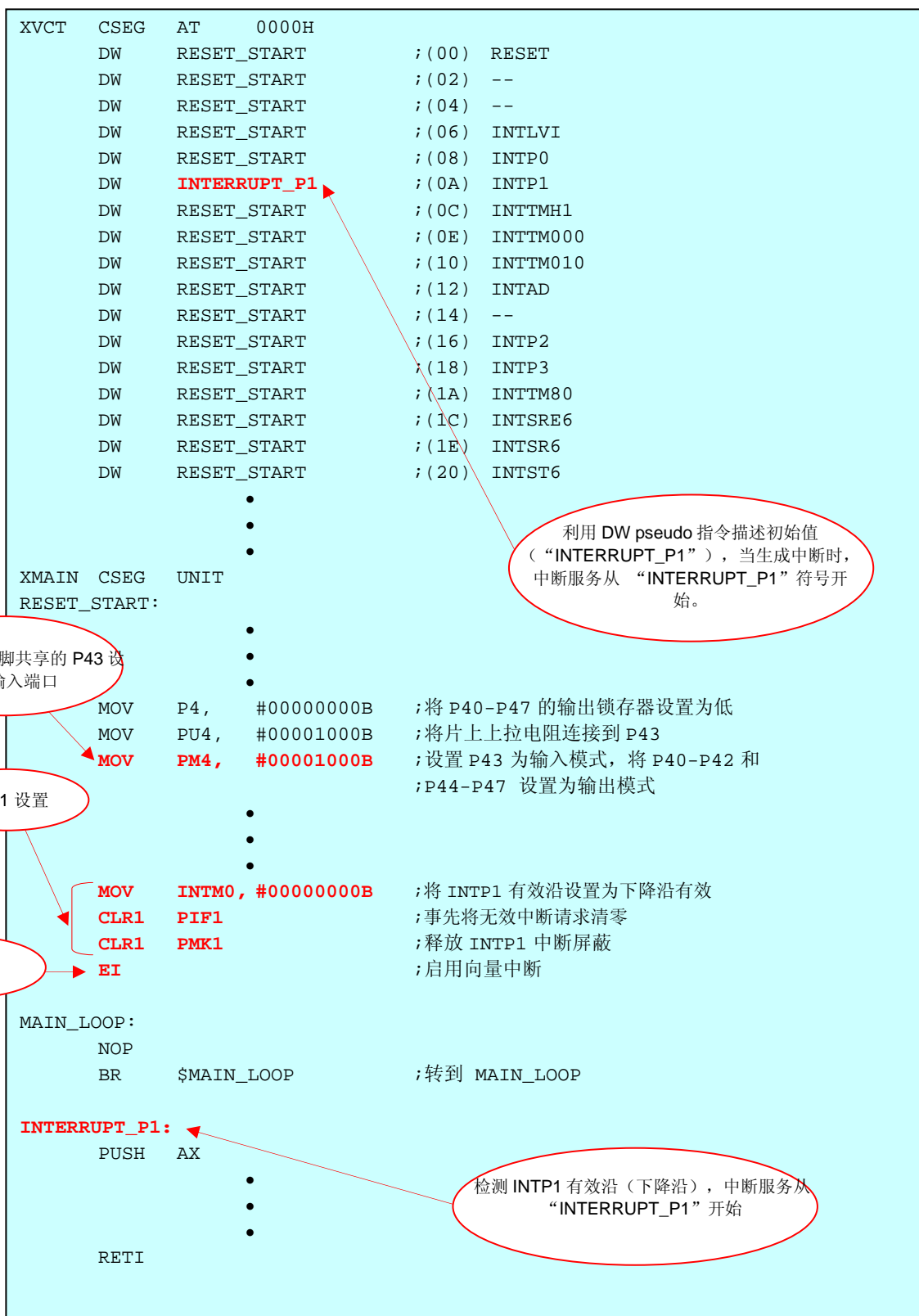
1	启用
---	----

备注

将与 INTP1 共享的端口 (P43: 78K0S/KA1+ 和 78K0S/KB1+ 微控制器 P32: 78K0S/KY1+ 和 78K0S/KU1+微控制器) 设置为“输入端口”。且设置为“使用内部上拉电阻”, 因为本示例程序中微控制器外的电路配置设置未连接上拉电阻。当使用外部中断时, 未要求设置使用内部上拉电阻。关于端口设置, 请参见 [78K0S/Kx1+示例程序 \(初始设置\) LED发光开关控制应用注](#)。

● 汇编语言程序示例

(与 78K0S/KB1+控制器示例程序设置内容相同)



● C 语言程序示例

(与 78K0S/KB1+ 微控制器示例程序设置内容相同)

```

#pragma interrupt INTP1 fn_intp1 /* 中断函数声明: INTP1 */
.
.
.
void hdwinit(void){
.
.
.
P4   = 0b00000000; /* 将 P40-P47 的输出锁存器设置为低 */
PU4  = 0b00001000; /* 将片上上拉电阻连接到 P43 */
PM4  = 0b00001000; /* 设置 P43 为输入模式, 将 P40-P42 和 */
                          /* P44-P47 设置为输出模式 */
.
.
.
INTMO = 0b00000000; /* 将 INTP1 的有效沿设置为下降沿 */
PIF1  = 0;          /* 事先清除无效中断请求 */
PMK1  = 0;          /* 释放 INTP1 中断屏蔽 */
return;
}

void main(void){
.
.
.
EI(); /* 启用向量中断 */

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

__interrupt void fn_intp1(){
    unsigned int unChat; /* 用于颤动消除定时器的 16-位向量 */
    .
    .
    .
    return;
}

```

在预处理指令(#pragma 指令)中声明 INTP1 中断功能 (“fn_intp1”在此情况下)并使用中断修改变量声明中断函数,从而,当中断生成时,从声明的中断函数与中断修改变量启动中断服务。

将与 INTP1 引脚共享的 P43 设置为输入端口

INTP1 设置

启用中断

利用检测 INTP1 的有效沿 (下降沿) 从 “fn_intp1” 开始中断服务。

4.2 中断时的处理

中断时的处理描述如下。

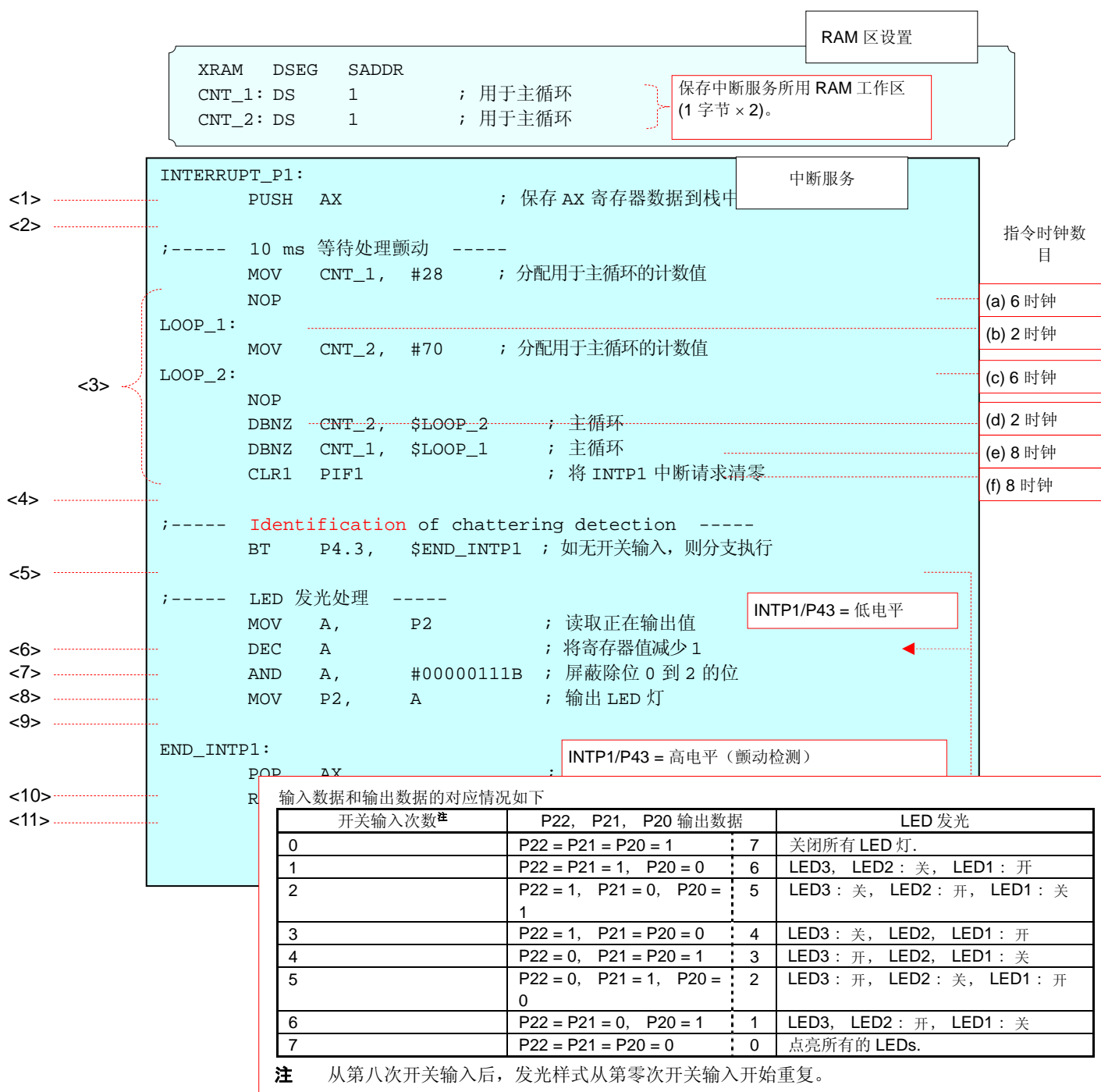
(1) 汇编语言

汇编语言状态下，中断时的处理可进行如下操作。

- <1> 开关输入产生中断请求且中断服务开始。
- <2> AX 寄存器数据保存到栈中。
- <3> 处理等待 10 ms.
- <4> PIF1 (INTP1 中断请求标志) 清零。
- <5> 检查 INTP1/P43^注 引脚状态。
 - INTP1/P43^注 = 低电平 → <5>
 - INTP1/P43^注 = 高电平 (颤动检测) → <9>
- <6> 读取正在输出的 P2 数据。
- <7> 读取数据减少 1。
- <8> 将<7>数据中除位 0 至 2 的位的值的值设置为 0。
- <9> <8>中数据输出到 P2。
- <10> 将保存到栈中的数据还原至 AX 寄存器中。
- <11> 处理从中断服务返回。

注 INTP1/P43: 78K0S/KA1+ 和 78K0S/KB1+微控制器
INTP1/P32: 78K0S/KY1+ 和 78K0S/KU1+ 微控制器

备注 1. 上述<1> 到 <11> 与下页<1>到<11>对应。
2. 下页程序示例的内容与 78K0S/KB1+微控制器示例程序相同。



备注 1. 上述<3>中等待时间 (10 ms) 可通过下述方法计算。

• 1 时钟 = 1/2 MHz = 0.5 μs

• <3>中时钟数目 = (a) + (b) + { (c) + ((d) + (e)) × 70 + (f) } × 28



• <3>等待时间 = [(a) + (b) + { (c) + ((d) + (e)) × 70 + (f) } × 28] × 0.5 μs

$$= [6 + 2 + \{ 6 + (2 + 8) \times 70 + 8 \} \times 28] \times 0.5 \mu s$$

$$= 20,000 \times 0.5 \mu s = 10,000 \mu s = 10 \text{ ms}$$

2. 上述<1>到<11>与前一页中<1>到 <11>对应。

(2) C 语言

C 语言中中断服务的操作与汇编语言中操作相似。

备注 下述程序示例内容与 78K0S/KB1+微控制器示例程序内容相同。

中断服务

```

__interrupt void fn_intp1(){
    unsigned int unChat;          /* 用于颤动解除计时器的 16-位变量*/
    for (unChat = 0; unChat < 278; unChat++){
        NOP();                    /* 等待约 10 ms (用于颤动解除) */
    }
    PIF1 = 0;                     /* 清除 INTp1 中断请求*/
    if (!P4.3){                  /* 如果 SW 开 10 ms 或更长时间则处理进行*/
        P2 = (P2 - 1) & 0b00000111; /* LED 输出根据 SW 输入次数变化*/
    }

    return;
}


```

输入数据与输出数据对应情况如下。

开关输入次数 ^注	P22, P21, P20 输出数据	LED 发光
0	P22 = P21 = P20 = 1	7 关闭所有 LED 灯.
1	P22 = P21 = 1, P20 = 0	6 LED3, LED2 : 关, LED1 : 开
2	P22 = 1, P21 = 0, P20 = 1	5 LED3 : 关, LED2 : 开, LED1 : 关
3	P22 = 1, P21 = P20 = 0	4 LED3 : 关, LED2, LED1 : 开
4	P22 = 0, P21 = P20 = 1	3 LED3 : 开, LED2, LED1 : 关
5	P22 = 0, P21 = 1, P20 = 0	2 LED3 : 开, LED2 : 关, LED1 : 开
6	P22 = P21 = 0, P20 = 1	1 LED3, LED2 : 开, LED1 : 关
7	P22 = P21 = P20 = 0	0 点亮所有的 LEDs.


注 从第八次开关输入后, 发光样式从第零次开关输入开始重复。

第五章 用系统仿真器SM+进行运行检查

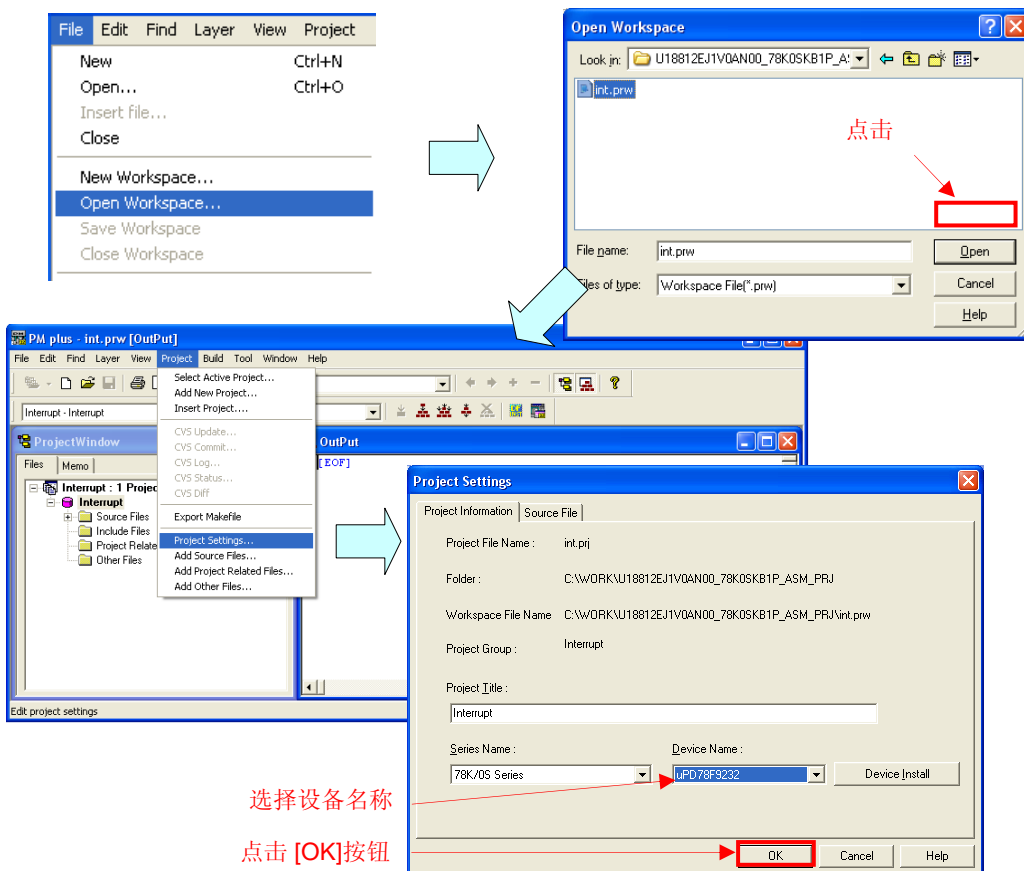
本章描述利用汇编语言文件，示例程序在适用 78K0S/Kx1+的系统仿真器 SM+中如何运行；该汇编语言文件通过选择  图标进行下载。


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 6 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。

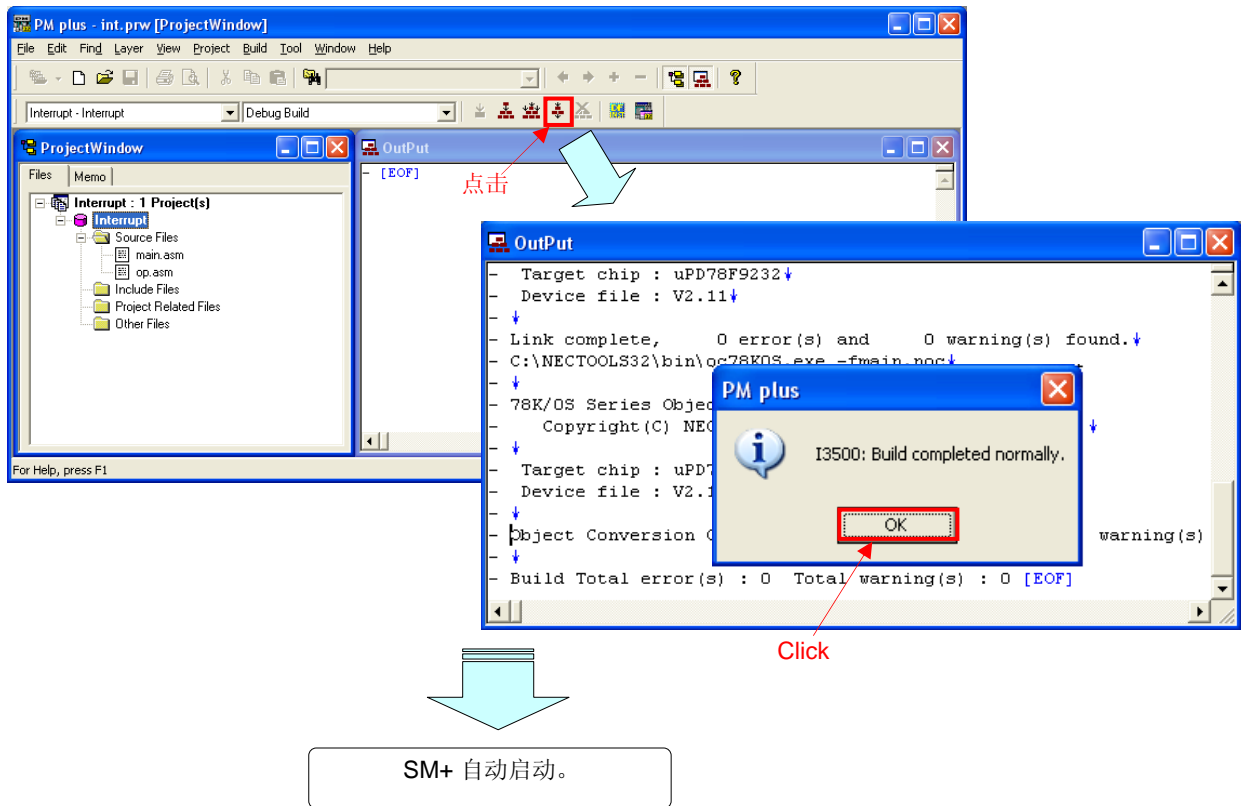
5.1 构建示例程序

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+。本章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  加载的汇编语言文件。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“int.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]。



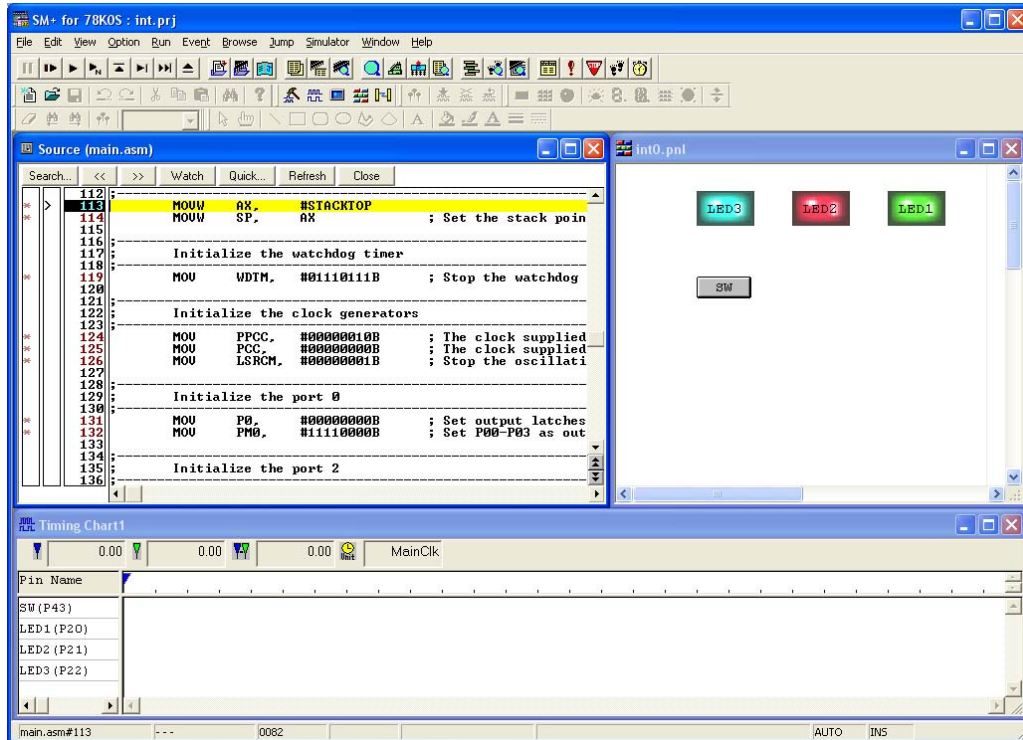
- (4) 点击  ([构建→调试]按钮)。当“main.asm”和“op.asm”源文件正常构建时，将显示信息“I3500: Build completed normally. (构建正常完成)”。
- (5) 点击消息窗口中的[OK]按钮，自动启动 SM+。



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](#)。

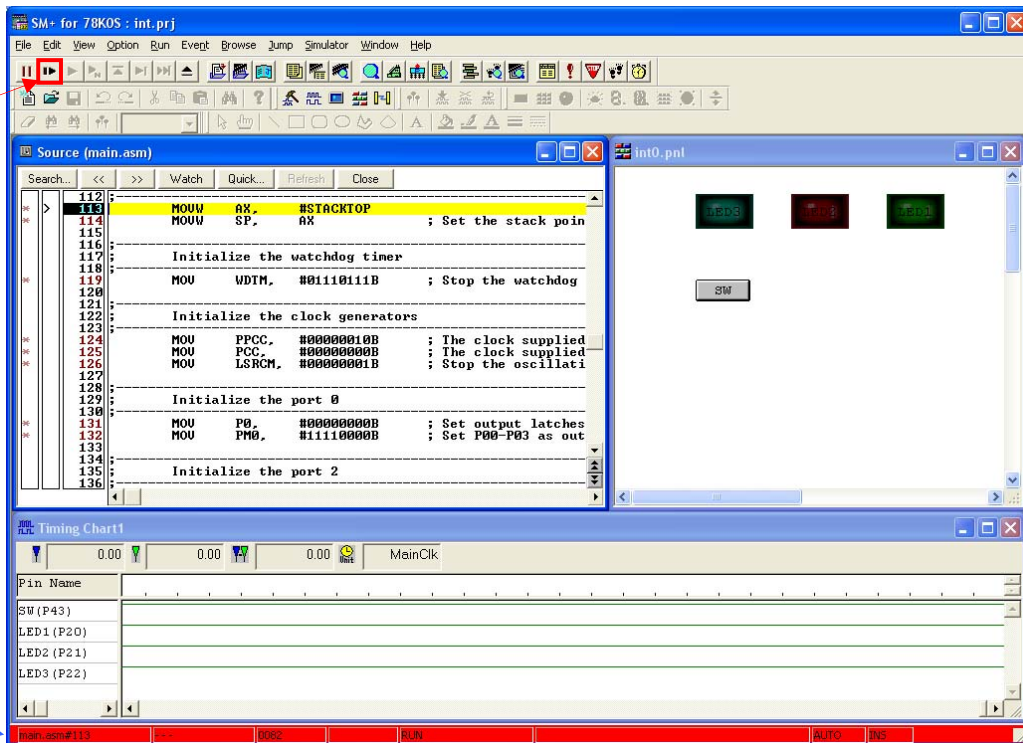
(1) 点击PM+中的[构建→调试]启动SM+（参见5.1），屏幕显示如下。（这是使用汇编语言源文件时的屏幕示例。）



(2) 点击  ([重新启动]按钮)。在 CPU 复位后，程序开始执行，屏幕显示如下。

(3)

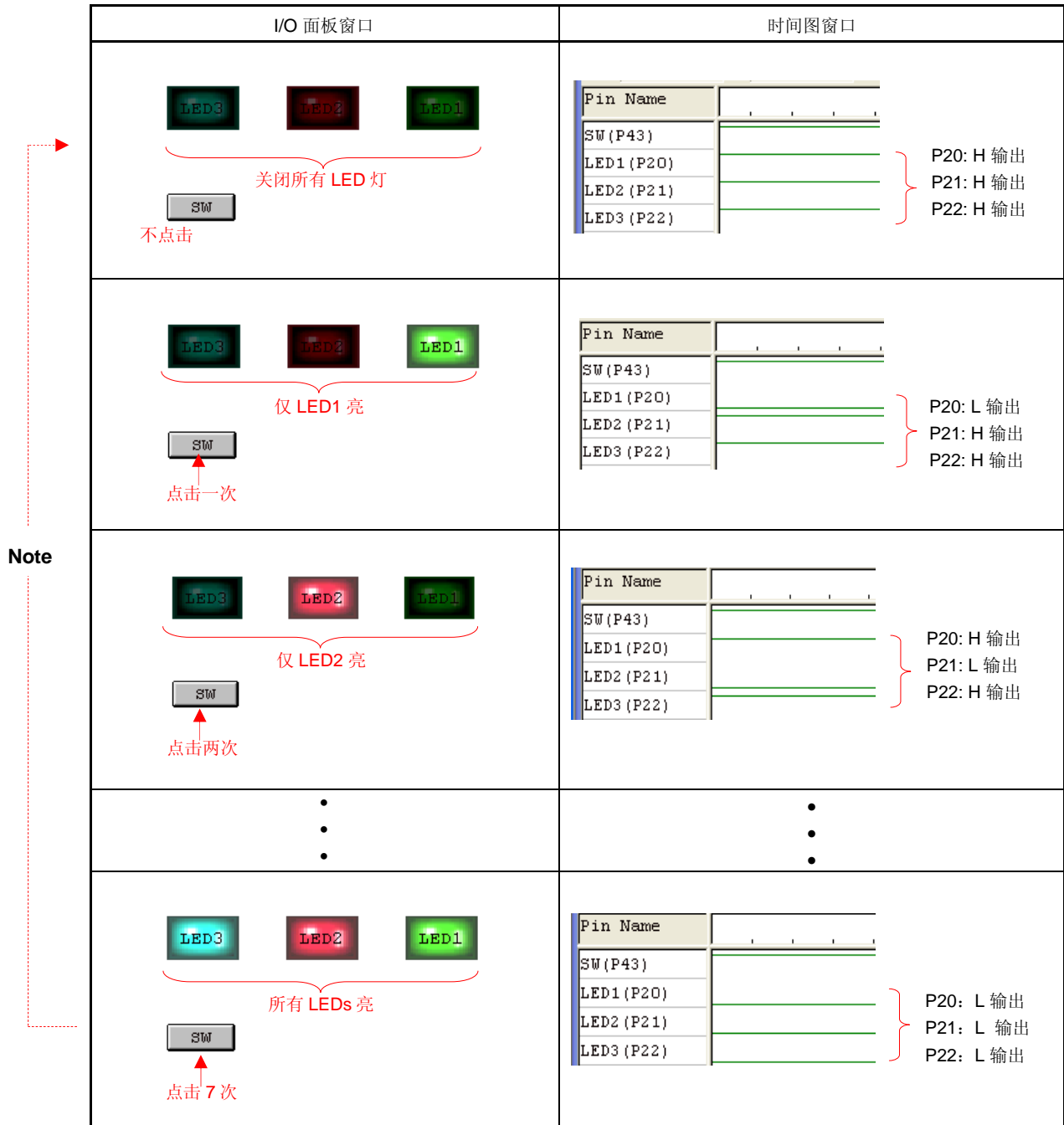
点击



程序执行时变为红色。

在程序执行过程中，在 I/O 面板窗口中点击[SW]按钮。

在 I/O 面板窗口中检查[LED1] 到[LED3]的闪烁周期，在时间图窗口中检查波形，确认二者随[SW]按钮的输入次数而改变。



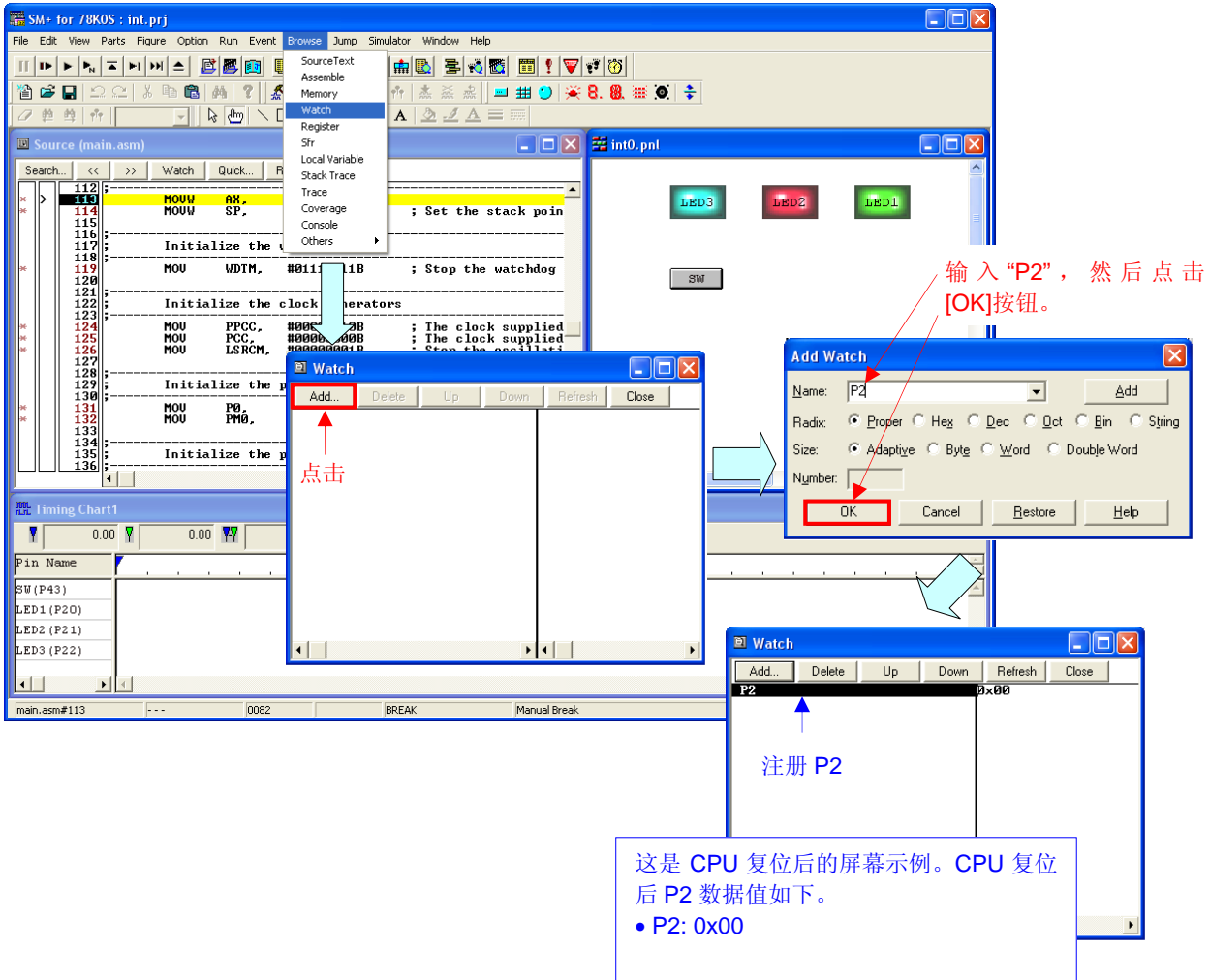
Note

注 在第八次开关输入后，重复第零次开关输入的发光模式。

备注 H: 高电平, L: 低电平

[附 1] 利用 SM+察看功能，可检查端口 2 数据值变化。

- <1> 从[浏览]菜单中选择[察看]按钮可打开[察看]窗口。
- <2> 点击[添加]打开[添加察看]窗口。（此时，[察看]窗口保持打开。）
- <3> 在[名称]区输入“P2”并点击[OK]按钮对 [察看] 窗口中“P2”进行注册，再关闭[添加察看]窗口。



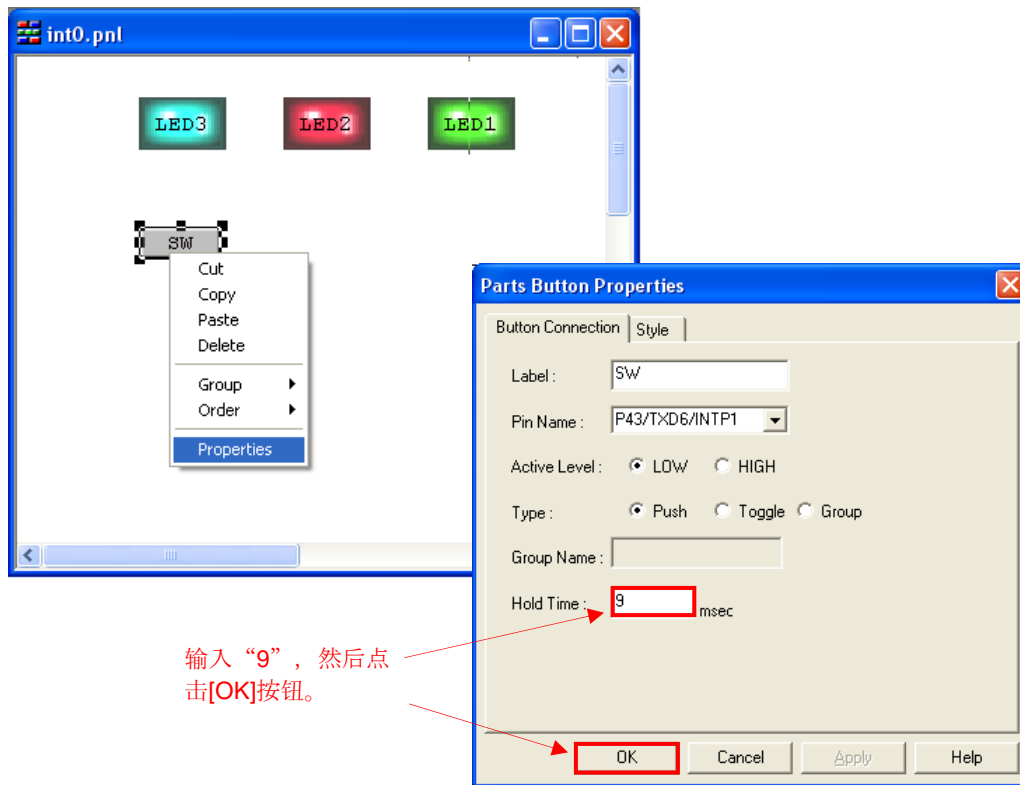
- <4> 执行程序并在 I/O 面板窗口中点击[SW]按钮。检查[察看]窗口中的 P2 数据值随[SW]按钮的输入次数而改变。


[SW] 按钮输入次数 [※]	[察看]窗口的数据值
0	P2: 0x07
1	P2: 0x06
2	P2: 0x05
3	P2: 0x04
4	P2: 0x03
5	P2: 0x02
6	P2: 0x01
7	P2: 0x00

注 第八次开关输入后重复第零次开关输入时的发光模式。

[附 2] [SW]按钮保持时间可设置为小于 10ms，从而可检查是否检测了颤动。

- <1> 在工具栏上选择 
- <2> 在 I/O 面板窗口中右键点击[SW]按钮，选择[属性]。
- <3> 对于保持时间（Hold Time）输入“9”，点击[OK]按钮。



- <4> 在工具栏上选择。 
- <5> 执行程序，点击[SW]按钮。因为按钮保持时间为 9ms，所以即便点击了[SW]按钮，也会识别出颤动，故 LED 发光模式不会改变。

第六章 相关文件

文件名称	日语/英语	
78K0S/KU1+ 用户手册	PDF	
78K0S/KY1+ 用户手册	PDF	
78K0S/KA1+ 用户手册	PDF	
78K0S/KB1+ 用户手册	PDF	
78K/0S 系列指令用户手册	PDF	
RA78K0S 汇编程序包用户手册	语言	PDF
	操作	PDF
CC78K0S C 编译器用户手册	语言	PDF
	操作	PDF
PM+ 工程管理器用户手册	PDF	
SM+ 系统模拟器操作用户手册	PDF	
78K0S/Kx1+ 示例程序（初始设置）LED 照明开关控制应用注释	PDF	
78K0S/Kx1+ 示例程序启动向导应用注释	PDF	

附录A 程序列表

作为程序列表示例，78K0S/KB1+ 微处理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
;*****
;       中断
;*****
;<<历史>>
;       2007.6.--   发布
;*****
;
;
;<<概述>>
;
;本示例程序提供使用中断功能的例子。
;展示了在检测到开关输入的下降沿并生成中断时，LED 发光模式随开关输入次数变化的改变。
;因此，低于 10 ms 的颤动将不计为开关输入，因为中断后立即设置了颤动消除时间为 10 ms。
;
;
;<主要设置内容>
;
; - 设置向量表
; - 设置栈指针
; - 停止看门狗计时器的运行
; - 设置 CPU 时钟频率为 2 MHz
; - 设置外部中断 INTP1 的有效沿为下降沿
; - 在开关输入时，将颤动检测时间设置为 10 ms;
;
;
;<开关输入次数和 LED 发光模式>
;
; +-----+
; | SW 输入 | LED3 | LED2 | LED1 |
; | (P43)   | (P22) | (P21) | (P20) |
; |-----|-----|-----|
; | 0 次   | 关   | 关   | 关   |
; | 1 次   | 关   | 关   | 开   |
; | 2 次   | 关   | 开   | 关   |
; | 3 次   | 关   | 开   | 开   |
; | 4 次   | 开   | 关   | 关   |
; | 5 次   | 开   | 关   | 开   |
; | 6 次   | 开   | 开   | 关   |
; | 7 次   | 开   | 开   | 开   |
; +-----+
; #第八次开关输入后，重复第零次开关输入时的发光模式。
;
;
;<< I/O 端口设置>>
;
; 输入： P43
; 输出： P00-P03, P20-P23, P30-P33, P40-P42, P44-P47, P120-P123, P130
```

```

; # 所有未使用的端口设置为输出模式。
;
;
;*****
;
;=====
;
;
; 向量表
;
;=====
XVCT  CSEG  AT    0000H
      DW   RESET_START      ;(00)  RESET
      DW   RESET_START      ;(02)  --
      DW   RESET_START      ;(04)  --
      DW   RESET_START      ;(06)  INTLVI
      DW   RESET_START      ;(08)  INTPO
      DW   INTERRUPT_P1 ;(0A)  INTTP1
      DW   RESET_START      ;(0C)  INTTMH1
      DW   RESET_START      ;(0E)  INTTM000
      DW   RESET_START      ;(10)  INTTM010
      DW   RESET_START      ;(12)  INTAD
      DW   RESET_START      ;(14)  --
      DW   RESET_START      ;(16)  INTP2
      DW   RESET_START      ;(18)  INTP3
      DW   RESET_START      ;(1A)  INTTM80
      DW   RESET_START      ;(1C)  INTSRE6
      DW   RESET_START      ;(1E)  INTSR6
      DW   RESET_START      ;(20)  INTST6
;
;=====
;
;
; 定义 RAM
;
;=====
XRAM  DSEG  SADDR
CNT_1:    DS    1          ;用于主循环
CNT_2:    DS    1          ;用于主循环
;
;=====
;
;
; 定义存储器栈区
;
;=====
XSTK  DSEG  AT    0FEE0H
STACKEND:
      DS    20H          ;存储器栈区 = 32 字节
STACKTOP:          ;存储器栈区起始地址 = FF00H
;
;*****
;
;
;  RESET 后的初始化
;
;*****
XMAIN  CSEG  UNIT
RESET_START:
;-----
;
; 初始化栈指针
;-----
      MOVW  AX,    #STACKTOP
      MOVW  SP,    AX          ;设置栈指针
;-----
;
; 初始化看门狗计时器
;-----

```

```

MOV    WDTM, #01110111B    ; 停止看门狗计时器的运行
;-----
;
; 初始化时钟发生器
;-----
MOV    PPCC, #0000010B    ; 供给外围硬件的时钟 (fxp) = fx/4 (= 2 MHz)
MOV    PCC, #00000000B    ; 供给 CPU 的时钟 (fcpu) = fxp (= 2 MHz)
MOV    LSRCM, #00000001B  ; 停止低速内部振荡器的振荡
;-----
;
; 初始化端口 0
;-----
MOV    P0, #00000000B    ; 将 P00-P03 的输出锁存器设置为低
MOV    PM0, #11110000B    ; 设置 P00-P03 为输出模式
;-----
;
; 初始化端口 2
;-----
MOV    P2, #00000111B    ; 将 P20-P22 的输出锁存器设置为高 (关闭 LED1-LED3)，将 P23 设
置为低
MOV    PM2, #11110000B    ; 设置 P20-P23 为输出模式
;-----
;
; 初始化端口 3
;-----
MOV    P3, #00000000B    ; 将 P30-P33 的输出锁存器设置为低
MOV    PM3, #11110000B    ; 设置 P30-P33 为输出模式
;-----
;
; 初始化端口 4
;-----
MOV    P4, #00000000B    ; 将 P40-P47 的输出锁存器设置为低
MOV    PU4, #00001000B    ; 连接片上上拉电阻至 P43
MOV    PM4, #00001000B    ; 设置 P43 为输入模式，P40-P42 及 P44-P47 为输出模式
;-----
;
; 初始化端口 12
;-----
MOV    P12, #00000000B    ; 将 P120-P123 的输出锁存器设置为低
MOV    PM12, #11110000B  ; 将 P120-P123 设置为输出模式
;-----
;
; 初始化端口 13
;-----
MOV    P13, #00000001B    ; 将 P130 的输出锁存器设置为高
;-----
;
; 设置中断
;-----
MOV    INTM0, #00000000B  ; 将 INTP1 的有效沿设置为下降沿
CLR1   PIF1                ; 事先清除无效中断请求
CLR1   PMK1                ; 释放 INTP1 中断屏蔽

EI                                ; 启动向量中断

;-----
;
; *****
;
; 主循环
;
; *****
MAIN_LOOP:
NOP

```

```

BR    $MAIN_LOOP          ;转到 MAIN_LOOP

;*****
;
;
;    外部中断 INTP1
;
;*****
INTERRUPT_P1:
    PUSH    AX              ;将 AX 寄存器数据保存到栈中

;----- 10 ms 等待处理颤动 -----
    MOV     CNT_1, #28      ;分配主循环计数值
    NOP

LOOP_1:
    MOV     CNT_2, #70      ;分配副循环计数值
LOOP_2:
    NOP
    DBNZ   CNT_2, $LOOP_2   ;副循环
    DBNZ   CNT_1, $LOOP_1   ;主循环

    CLR1   PIF1             ;清除 INTP1 中断请求

;----- 辨别颤动检测 -----
    BT     P4.3, $END_INTP1 ;若无开关输入，就分支执行

;----- LED 发光处理 -----
    MOV    A,    P2          ;读取当前输出值
    DEC    A              ;将寄存器值减少 1
    AND    A,    #0000111B  ;屏蔽除位 0 到 2 的其他位
    MOV    P2,    A         ;输出 LED 灯

END_INTP1:
    POP    AX              ;还原 AX 寄存器数据
    RETI                   ;从中断服务返回

end

```

● main.c (C 语言版)

```
/******
```

```
日电电子 78K0S/KB1+
```

```
*****
```

```
78K0S/KB1+ 示例程序
```

```
*****
```

```
中断
```

```
*****
```

```
<<历史>>
```

```
2007.6.-- 发布
```

```
*****
```

```
<<概述>>
```

本示例程序提供使用中断功能的例子。

展示了在检测到开关输入的下降沿并生成中断时，LED 随开关输入次数变化的改变。

因此，小于 10 ms 的颤动将不计为开关输入，因为在中断后立即设置颤动消除时间为 10 ms。

<主要设置内容>

- 声明由中断运行的函数：INTP1 -> fn_intp1()
- 停止看门狗定时器的运行
- 设置 CPU 时钟频率为 2 MHz
- 设置外部中断 INTP1 的有效沿为下降沿
- 在开关输入时，将颤动检测时间设置为 10 ms

<开关输入次数和 LED 发光模式>

SW 输入 (P43)	LED3 (P22)	LED2 (P21)	LED1 (P20)
0 次	关	关	关
1 次	关	关	开
2 次	关	开	关
3 次	关	开	开
4 次	开	关	关
5 次	开	关	开
6 次	开	开	关
7 次	开	开	开

#第八次开关输入后，重复第零次开关输入时的发光模式。

<<I/O 端口设置 >>

输入：P43

输出：P00-P03, P20-P23, P30-P33, P40-P42, P44-P47, P120-P123, P130

所有未使用的端口设置为输出模式。

```

*****/

/*=====
    预处理指令 (#pragma)
=====*/
#pragma SFR          /* 可在 C 源程序级描述 SFR 名称 */
#pragma EI           /* 可在 C 源程序级描述 EI 指令*/
#pragma NOP          /* 可在 C 源程序级描述 NOP 指令*/
#pragma interrupt INTP1 fn_intp1 /* 中断函数声明: INTP1 */

/*****

    RESET 后的初始化
*****/

void hdwinit(void){
/*-----

    初始化看门狗计时器
-----*/

    WDTM = 0b01110111; /* 停止看门狗计时器的运行*/

/*-----

    初始化时钟发生器
-----*/

    PPCC = 0b00000010; /* 供给外围硬件的时钟为(fxp) = fx/4 (= 2 MHz) */
    PCC  = 0b00000000; /* 供给 CPU 的时钟为(fcpu) = fxp (= 2 MHz) */
    LSRCM = 0b00000001; /* 停止低速内部振荡器的振荡 */

/*-----

    初始化端口 0
-----*/

    P0  = 0b00000000; /*将 P00-P03 的输出锁存器设置为低 */
    PM0 = 0b11110000; /* 设置 P00-P03 为输出模式*/

/*-----

    初始化端口 2
-----*/

    P2  = 0b00000111; /* 将 P20-P22 的输出锁存器设置为高 (关闭 LED1-LED3)，设置 P23 为
低*/
    PM2 = 0b11110000; /* 设置 P20-P23 为输出模式*/

/*-----

    初始化端口 3
-----*/

```

```

P3 = 0b00000000;          /* 将 P30-P33 输出锁存器设置为低*/
PM3 = 0b11110000;          /* 设置 P30-P33 为输出模式*/

/*-----*/

    初始化端口 4

-----*/

P4 = 0b00000000;          /* 将 P40-P47 的输出锁存器设置为低*/
PU4 = 0b00001000;          /* 连接片上上拉电阻至 P43 */
PM4 = 0b00001000;          /* 设置 P43 为输入模式，设置 P40-P42 和 P44-P47 为输出模式*/

/*-----*/

    初始化端口 12

-----*/

P12 = 0b00000000;          /* 将 P120-P123 的输出锁存器设置为低*/
PM12 = 0b11110000;          /* 设置 P120-P123 为输出模式*/

/*-----*/

    初始化端口 13

-----*/

P13 = 0b00000001;          /* 将 P130 的输出锁存器设置为高*/

/*-----*/

    设置中断

-----*/

INTM0 = 0b00000000;          /* 设置 INTP1 的有效沿为下降沿*/
PIF1 = 0;                    /* 事先清除无效中断请求*/
PMK1 = 0;                    /* 释放 INTP1 中断屏蔽*/

return;
}

/******

    主循环

*****/

void main(void){

    EI();                    /* 启动向量中断 */

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

/******

    外部中断 INTP1

*****/

__interrupt void fn_intp1(){

```

```
unsigned int unChat;    /* 用于消除颤动定时器的 16-位变量*/

for (unChat = 0; unChat < 278; unChat++){ /* 等待约 10 ms (用于消除颤动) */
    NOP();
}

PIF1 = 0;                /* 清除 INTP1 中断请求*/

if (!P4.3){              /* 如 SW 打开 10 ms 或更长时间进行处理 */
    P2 = (P2 - 1) & 0b00000111;    /* 根据 SW 输入数目的 LED 输出*/
}

    返回;
}
```


● op.asm (汇编语言和 C 语言共用)

```

=====
;
;
;   选项字节
;
=====
OPBTCSEG AT      0080H
  DB             10011100B   ;选项字节区
;
;                   |||
;                   |||+----- 低速内部振荡器可用软件停止
;                   |++----- 高速内部振荡器时钟(8 MHz) 选择为系统时钟源
;                   +----- P34/RESET 用作 RESET 引脚
;
  DB             11111111B   ;保护字节区(用于自编程模式)
;
;                   |||||
;                   ++++++++----- 所有模块均可写入或删除
结束

```

附件B 修订记录

版本	出版日期	页码	修订
第一版	2007年10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