

应用注释

78K0S/Kx1+

示例程序（看门狗计时器）

131 ms 时间间隔失控检测

本文件包含示例程序操作概述，怎样使用示例程序及看门狗定时器的设置及使用。在示例程序中，设置失控检测时间为 131ms，在开关输入过程中，看门狗计时器溢出可生成复位信号，生成的几率为 50%。

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.....	4
1.1 初始设置的主要内容.....	5
1.2 主循环之后的内容.....	5
第二章 电路图.....	7
2.1 电路图.....	7
2.2 外围硬件.....	7
第三章 软件.....	8
3.1 文件配置.....	8
3.2 所用内部外围功能.....	9
3.3 初始设置及运行概览.....	9
3.4 流程图.....	11
第四章 设置方法.....	12
4.1 看门狗计时器（WDT）设置.....	12
第五章 用设备进行运行检查.....	20
5.1 构建示例程序.....	20
5.2 随设备运行.....	22
第六章 相关文件.....	24
附录A 程序列表.....	25
附件B 修订记录.....	37

文件号： U18847CA1V0AN00（第一版）
出版日期： 2008 年 03 月 N
© NEC Electronics Corporation 2007

中文版

· 本文档中的信息于 2008 年 3 月开始使用。文档内容可能会不作通知进行修改。实际设计请参阅日电电子最新发布的数据表或数据册等，查看日电电子产品的最新指标。并非所有产品和/或类型在每个国家都能使用。请联系日电电子销售代表，了解可用性信息及其他信息。

· 未经日电电子书面许可，不能以任何形式或方式对本文档的任何部分进行复制或重现。本文档出现的任何错误，日电电子不承担责任。

· 对于在使用本文档列出的日电电子产品时产生的侵犯专利、版权以及其他第三方知识产权的行为，以及对于其他使用这些产品产生的责任，日电电子不承担责任。对于日电电子及其他子公司的任何专利、版权以及其他知识产权，日电电子没有以许可、明示、暗示以及其他任何方式授权。

· 本文档中对电路、软件及其他相关信息的描述旨在说明半导体产品的操作及应用举例。这些电路、软件和信息在客户设备设计中的使用应由客户承担全部责任。如果这些电路、软件和信息导致客户或第三方遭受损失，日电电子不承担责任。

· 日电电子尽力提升日电电子产品质量、可靠性和安全性，但请客户同意并理解这些产品的瑕疵无法完全消除。为了尽量减少由于日电电子产品导致的财产损失或人身伤害（包括死亡），客户必须在其设计中采取足够的安全措施，如冗余、防火、防故障等特性。

· 日电电子产品分为下列三种质量等级：“标准”、“特别”及“专用”。

“专用”质量等级只适用于基于用户设计的“质量保证项目”的特定应用开发的日电电子产品。日电电子的建议应用由其质量级别决定，如下所示。客户在将日电电子产品用于特别用途之前必须检查各产品的质量等级。

“标准”：计算机、办公设备、通信设备、测试测量设备、音频视频设备、家用电子产品、机械工具、个人电子设备、工业机器人。

“特别”：运输设备（汽车、火车、轮船等）、交通控制系统、抗灾系统、防犯罪系统、安全设备、医疗设备（不专为生命救护而设计）。

“专用”：飞机、航空设备、水下中继器、核反应堆控制系统、生命救护系统、用于生命救护的医疗设备等。

除非在日电电子的数据表或数据册等当中有明确说明，日电电子产品质量级别均为“标准”。客户如果希望日电电子产品实现日电电子未预定的应用，必须提前联系日电电子的销售代表以确定日电电子愿意支持给定应用。

（注）

（1）本声明中所用的“日电电子”表示日电电子公司，包括其控股的子公司。

（2）“日电电子产品”表示由日电电子（如上所规定）开发或制造的任何产品。

M&E 02 11-1

第一章 概述

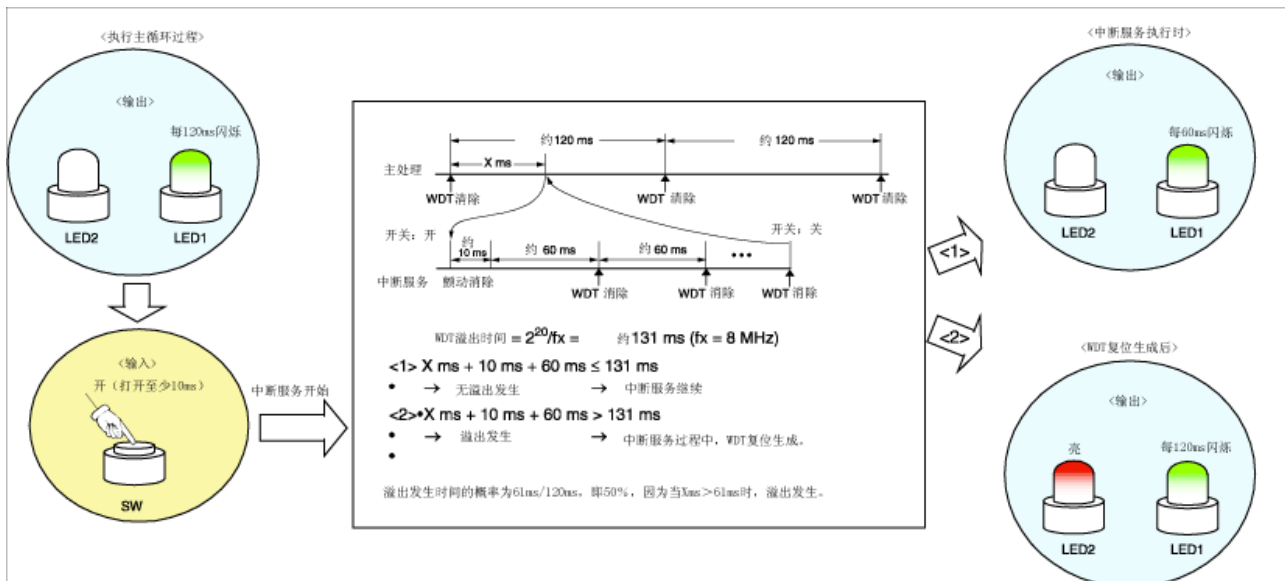
本示例程序展示使用看门狗计时器（WDT）功能的示例。

系统时钟设置为看门狗计时器的运行时钟，失控检测时间设置为约 131 ms。

初始设置完成后，两个 LEDs（LED1、LED2）外的 LED1 每 120 ms 闪烁一次。

根据开关输入计时，可进行下列任何一种操作，发生几率（见下图）。

- 执行中断服务时，开关输入中 LED1 每 60 ms 闪烁一次。
- 中断服务时，看门狗计时器溢出可生成复位信号。复位释放后，LED2 亮，LED1 每 120ms 闪烁一次。



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（8 MHz（典型））作为系统时钟源^注

设置低速内部振荡器的振荡可用软件停止^注

选择系统时钟(fx) 作为看门狗计时器运行时钟，设置溢出时间为 $2^{20}/fx$ （约 131 ms）

当看门狗计时器生成内部复位信号时，LED2 发光

设置 VLVI（低压检测电压）为 2.85 V \pm 0.15 V

在 V_{DD}（供电电压）大于等于 V_{LVI} 后，当检测到 V_{DD} 小于 V_{LVI} 时，生成内部复位（LVI 复位）信号

将 CPU 时钟频率设置为 4 MHz

设置 I/O 端口

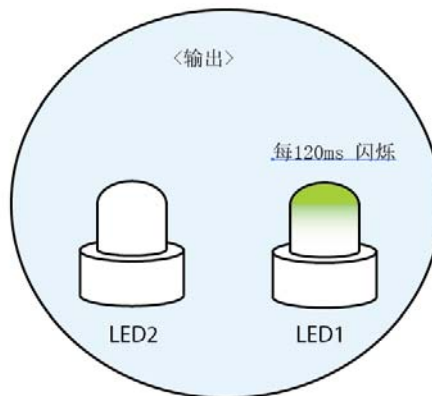
设置 INTP1 的有效边缘（外部中断）设置为下降沿

启用中断

注 通过选项字节进行设置。

1.2 主循环之后的内容

在初始设置完成后，主循环中，两个 LEDs（LED1、LED2）外的 LED1 每 120 ms 闪烁。



检测到开关输入生成的 INTP1 引脚的下降沿后进行中断服务。检测到 INTP1 引脚的下降边缘后经过 10ms，当 INTP1 处于高电平（开关关闭）时发现抖动并返回主循环。当 INTP1 处于低电平时（开关打开）时，边缘检测后经过 10ms，可进行下列处理。



[专栏]抖动

抖动，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。

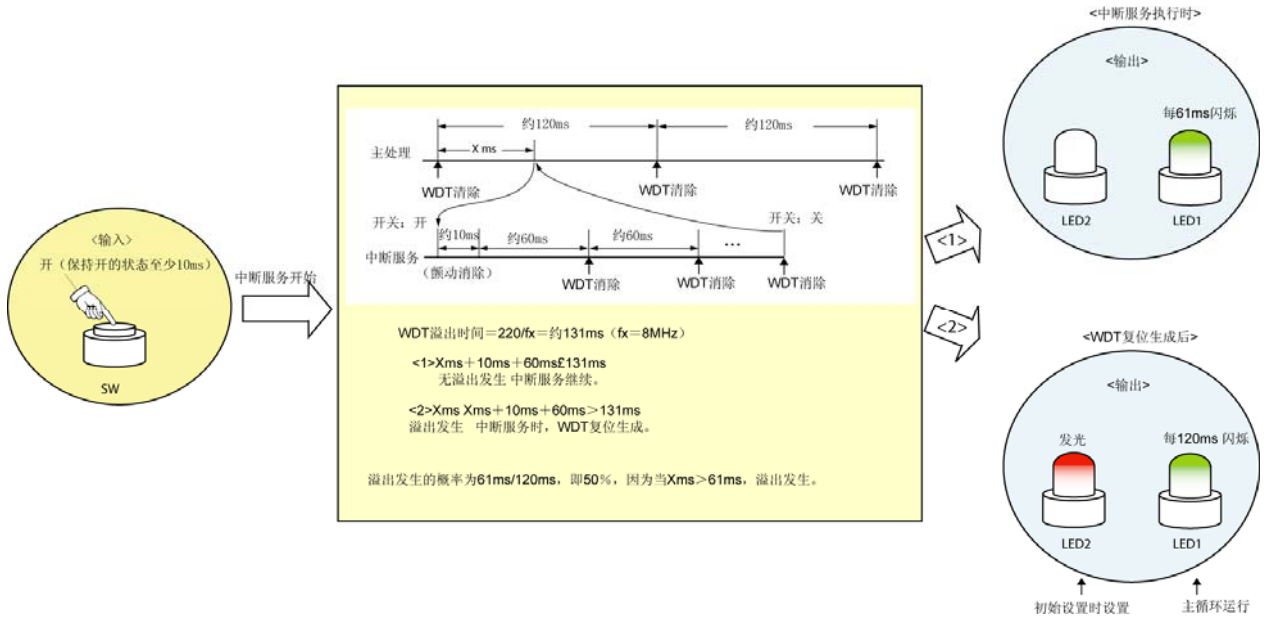
根据开关输入定时器不同，可进行如下操作，发生几率为 50%（见下表）。

<1> 如中断服务时，溢出不是由看门狗计时器引起的

由于中断服务的执行，LED1 在开关输入时每 60 ms 闪烁。当 INTP1 转到高电平（开关关闭），LED1 每 120 ms 闪烁。

<2> 如中断服务时看门狗计时器引起溢出。看门狗计时器生成复位信号。释放复位后，LED2 亮，LED1 每 120ms 闪烁。

备注 非看门狗计时器生成的复位信号关闭 LED2。



注意事项

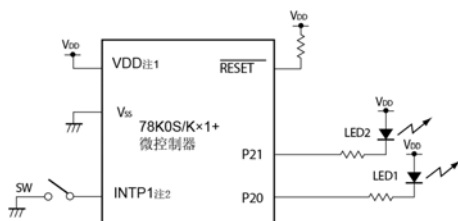
1. 在本示例程序中，看门狗计时器在中断服务开始时不清零，从而使看门狗计时器在中断服务时有 50% 的几率生成复位信号。通用时，看门狗计时器在中断服务开始和结束时清零，从而避免溢出发生。
2. 关于使用设备的注意事项，请参见各产品（[78K0S/KU1+](#)，[78K0S/KY1+](#)，[78K0S/KA1+](#)，[78K0S/KB1+](#)）的用户手册。

第二章 电路图

本章描述本示例程序中所用的电路图和外围硬件。

2.1 电路图

电路图如下所示。



- 注
1. 工作电压范围为 $3.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。
 2. INT P1/P43: 78K0S/KA1+ 和 78K0S/KB1+微控制器
INT P1/P32: 78K0S/KY1+ 和 78K0S/KU1+微控制器

- 注意事项
1. 将 AV_{REF} 引脚连接到 V_{DD} （仅适用于 78K0S/KA1+ 和 78K0S/KB1+ 微控制器）。
 2. 将 AV_{SS} 引脚直接连接到 GND （仅适用于 78K0S/KB1+微控制器）。
 3. 除电路图中所示引脚及 AV_{REF} 、 AV_{SS} 外，其他所有不用的引脚保留开路（不连接）。

2.2 外围硬件

所用外围硬件如下所示。

(1) 开关 (SW)

开关用来作为控制 LED 发光的输入信号。

(2) LED 灯 (LED1, LED2)



LED 灯用来作为对应开关输入和看门狗计时器生成的复位信号的输出。

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)	用于微控制器的硬件初始化处理及主处理的源文件	● ^注	● ^注
main.c (C语言版)			
op.asm	用于设置选项字节（设置系统时钟源）的汇编器源文件	●	●
wdt.prw	用于集成开发环境PM+的工作区文件		●
wdt.prj	用于集成开发环境PM+的项目文件		●

注 “main.asm”包含在汇编语言版中，“main.c”包含在 C 语言版中。

备注 ：仅包含源文件。

：包含与集成开发环境 PM+一起使用的文件。

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。

- 程序失控检测：看门狗计时器
- VDD < VLVI 检测： 低压监测器（LVI）
- 开关输入： INTP1[※]（外部中断）
- LED 输出： P20, P21（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

本示例程序进行的操作有：看门狗计时器设置和低压检测功能、时钟频率的选择、I/O 端口的设置、中断及类似初始设置。

初始设置完成后，主循环中两个 LED（LED1，LED2）灯之外的 LED1 每 120 ms 闪烁。

检测到由开关输入生成的 INTP1 引脚的下降沿时进行中断服务。在检测到 INTP1 引脚的下降沿后经过 10ms，当 INTP1 处于高电平（开关关闭）时发现抖动并返回到主循环。检测到边缘后经过 10ms，当 INTP1 处于低电平（开关打开）时，可进行如下处理。

根据开关输入定时不同，可进行下列操作，发生几率为 50%。

<1> 如在中断服务时，溢出不是由看门狗计时器引起的。

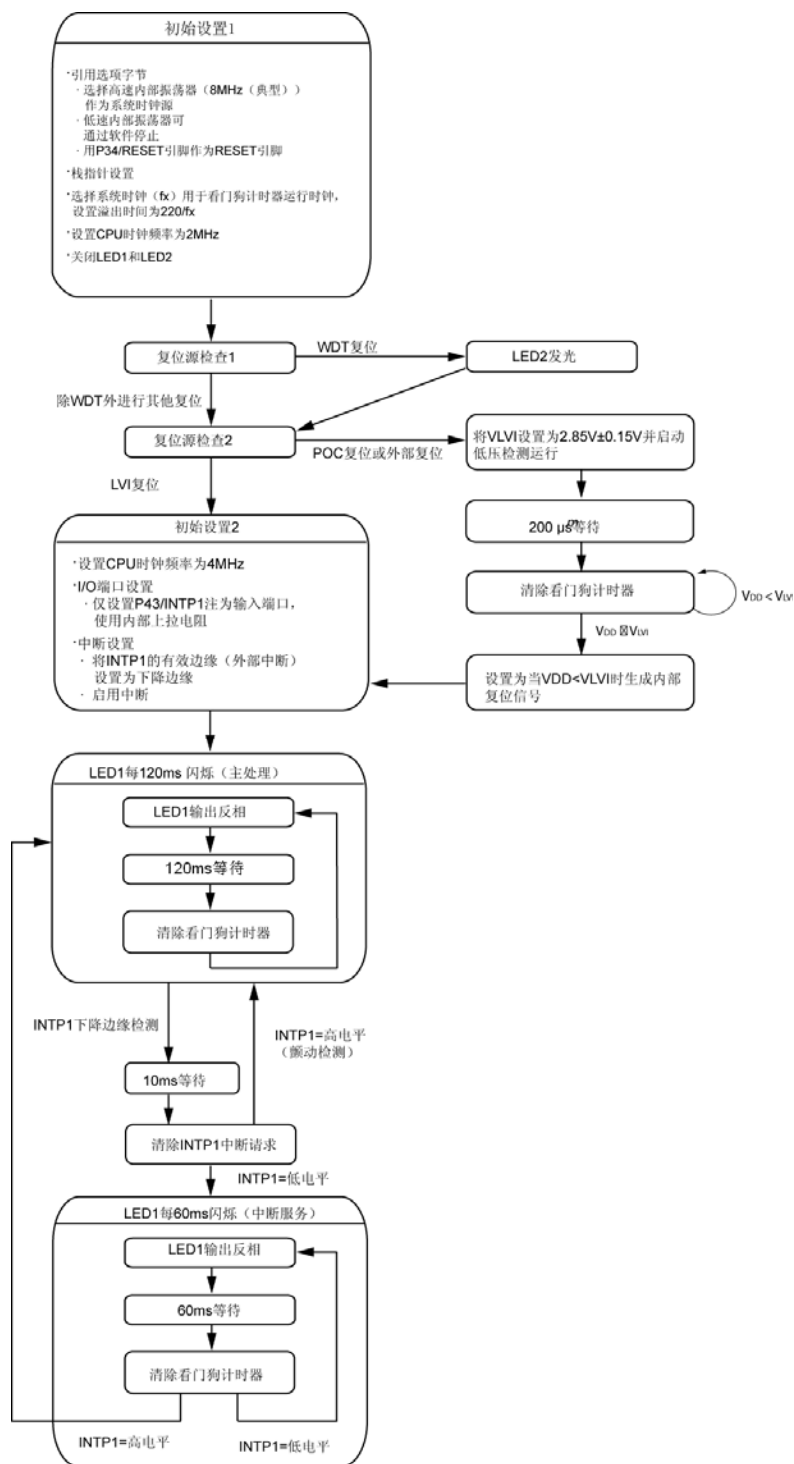
由于执行中断服务，LED1 在开关输入时每 60 ms 闪烁。当 INTP1 转到高电平（开关关闭），LED1 每 120 ms 闪烁。

<2> 如在中断服务时，溢出是由看门狗计时器引起的

看门狗计时器生成复位信号。在复位释放后，LED2 亮，且 LED1 每 120ms 闪烁。

注意事项 在本示例程序中，看门狗计时器在中断服务开始时的计数不清零，从而看门狗计时器在中断服务时有 50 %的几率生成复位信号。通用时，看门狗计时器在中断服务开始和结束时清零，从而防止溢出发生。

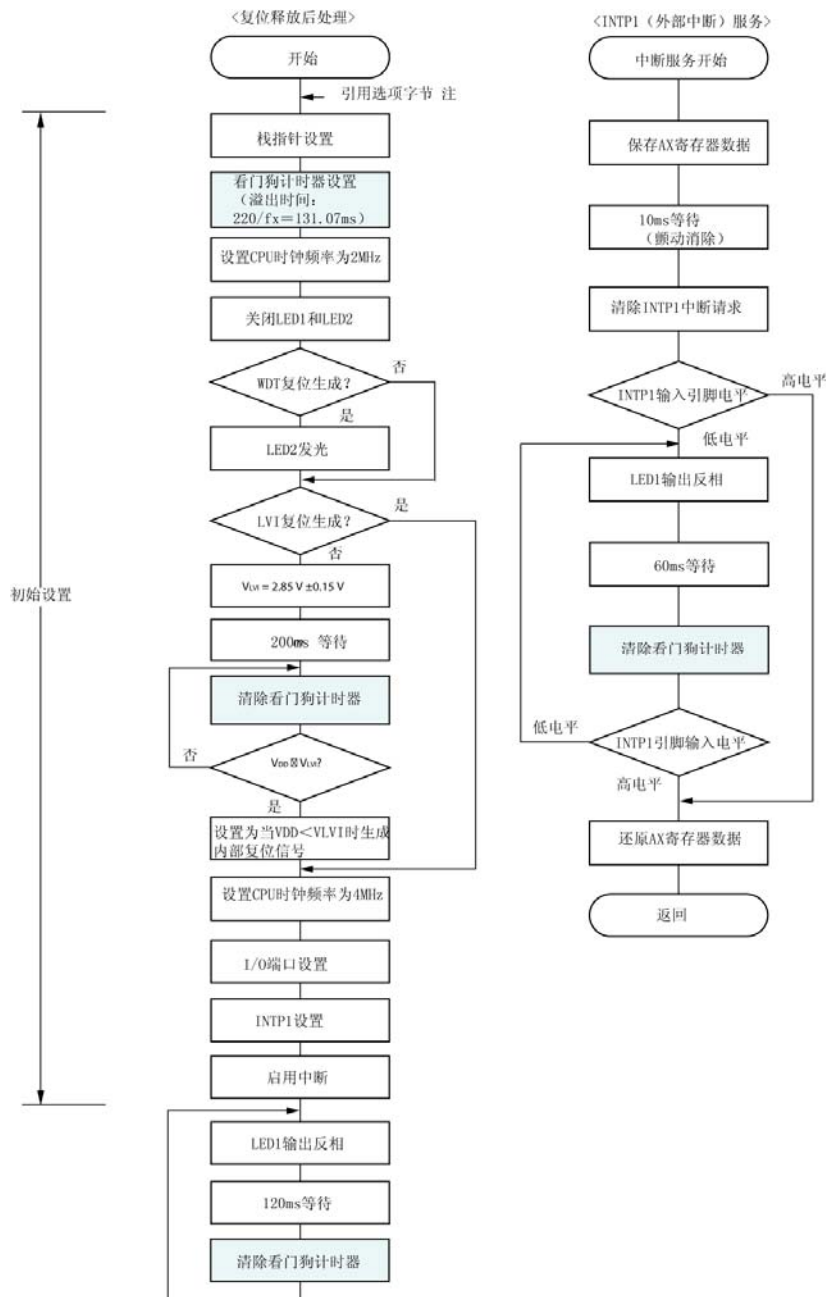
详情在如下所示的状态转换图中描述。



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

注意事项 在本示例程序中，看门狗定时器在中断服务开始时的计数不清零，从而看门狗定时器在中断服务时有 50 % 的几率生成复位信号。通用时，看门狗定时器在中断服务开始和结束时清零，从而防止溢出发生。

第四章 设置方法

本章描述了看门狗定时器。

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](#)。关于中断，请参见[78K0S/Kx1+示例程序（中断）由开关输入生成的外部中断的应用注释](#)。关于低压检测（LVI），请参见[78K0S/Kx1+示例程序（低压检测）在小于 2.7 V检测过程中的复位生成的应用注释](#)。

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 看门狗计时器（WDT）设置

看门狗计时器由下列两类寄存器控制。

- 看门狗计时器模式寄存器（WDTM）
- 看门狗计时器启用寄存器（WDTE）

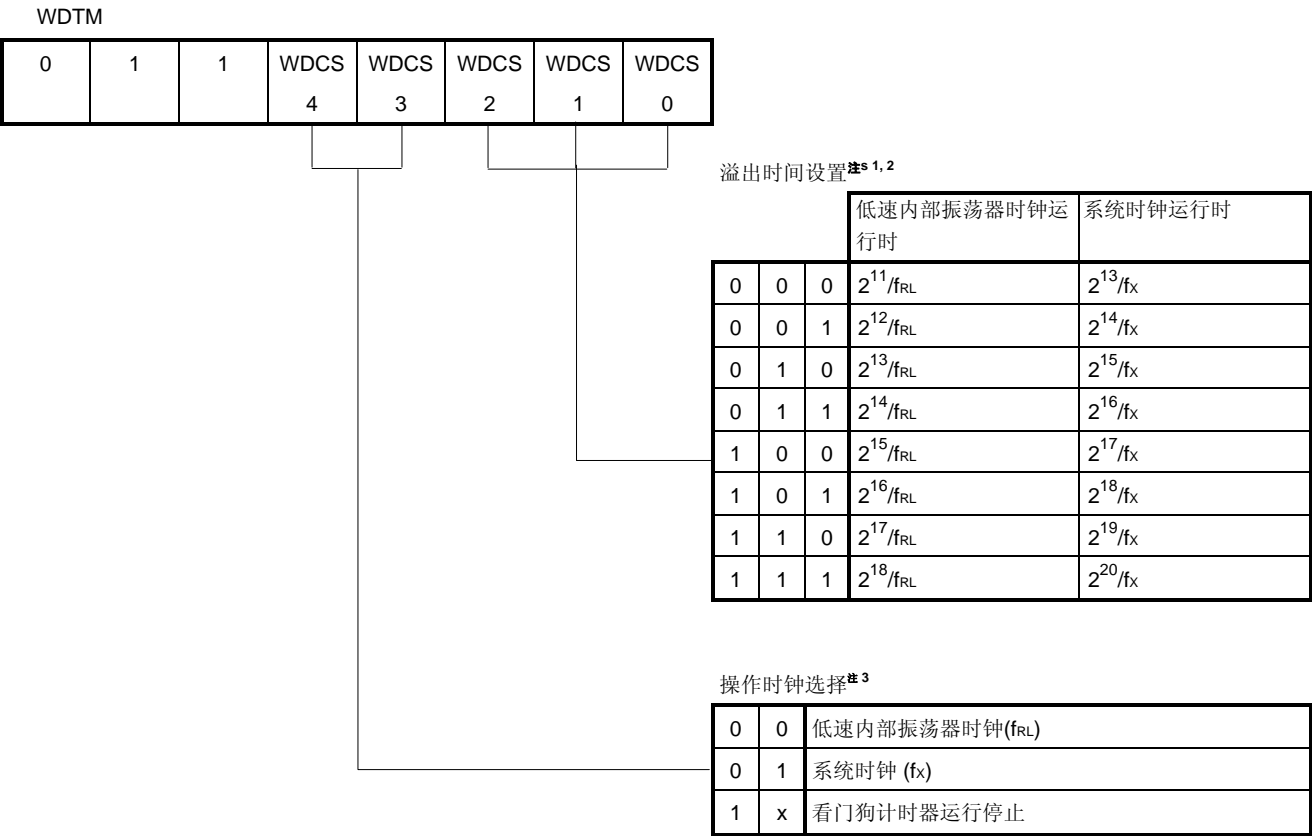
看门狗计时器操作时钟可根据利用选项字节设置的低速内部振荡器振荡控制选择。

(1) 设置看门狗计时器操作时钟和溢出时间

看门狗计时器模式寄存器（WDTM）用于设置看门狗计时器的操作时钟和溢出时间。只有在复位释放后才可写入 WDTM。

注意事项 必须在初始设置时设置看门狗计时器操作时钟和溢出时间。

图 4-1. 看门狗计时器模式寄存器（WDTM）的格式



- 注 1. 如选择看门狗计时器运行停止，溢出时间设置无效。（不关心）。
2. 周期在复位释放时最长（WDCS2、WDCS1、WDCS0 = 1、1、1）。
3. 如低速内部振荡器的振荡控制用选项字节设置为“Cannot be stopped（不能停止）”，运行时不能选。忽视写入值选择低速内部振荡器时钟。请参见[\(3\)设置低速内部振荡器的振荡控制。](#)

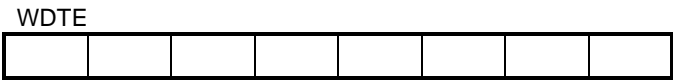
- 注意事项 1. 分别设置位 7、6，和 5 到 0、1，和 1。
2. 复位释放后，WDTM 只可写入一次。如尝试第二次写入，将生成内部复位信号。但是，如果分别将 WDCS4 和 WDCS3 设置为“1”和“x”，且在第一次写入操作时停止看门狗计时器，尽管执行了第二次写入操作，仍无内部复位信号生成。

备注 x: 不关心

(2) 看门狗计时器计数控制

将“ACH”写入看门狗计时器启用寄存器（WDTE）清除看门狗计时器计数并再次启动计数。

图 4-2. 看门狗计时器启用寄存器（WDTE）的格式



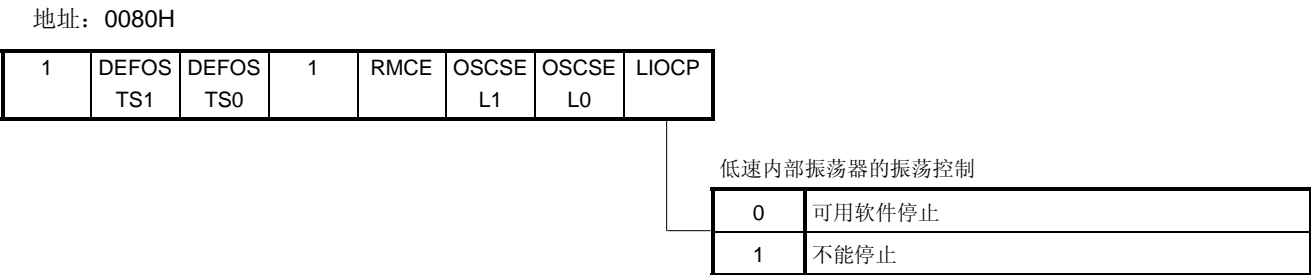
注意事项 如除“ACH”之外的值写入 WDTE，内部复位信号生成。

(3) 设置低速内部振荡器的振荡控制

可根据由利用选项字节设置的低速内部振荡器的振荡控制情况使用看门狗计时器的运行时钟。

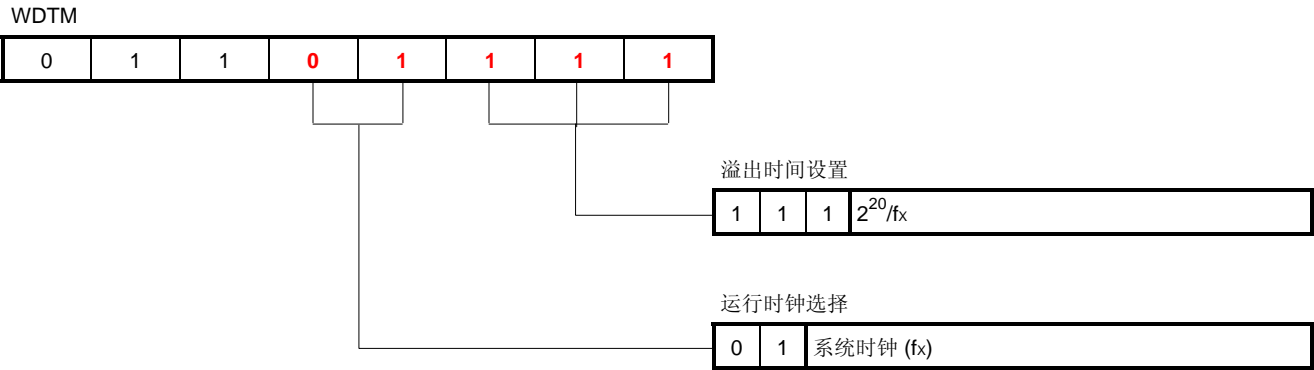
- 将低速内部振荡器的振荡设置为“Cannot be stopped（不能停止）”
运行时钟：仅指低速内部振荡器时钟（运行时钟不可选）
- 将低速内部振荡器的振荡设置为“Can be stopped by using software（可用软件停止）”
运行时钟：可选低速内部振荡器时钟、系统时钟或看门狗计时器运行停止。

图 4-3. 选项字节的格式（仅低速内部振荡器的振荡控制）



备注 关于除低速内部振荡器的振荡控制外的选项字节设置，请参见[78K0S/Kx1+示例程序（初始设置）LED发光开关控制应用注释。](#)

[示例 1] 将系统时钟（fX）用作看门狗计时器运行时钟，设置溢出时间为最大周期（220/fX）（与示例程序设置内容相同）



WDTM 设置值为“01101111”（位 7、6、和 5 必须分别设置为 0、1 和 1”）。

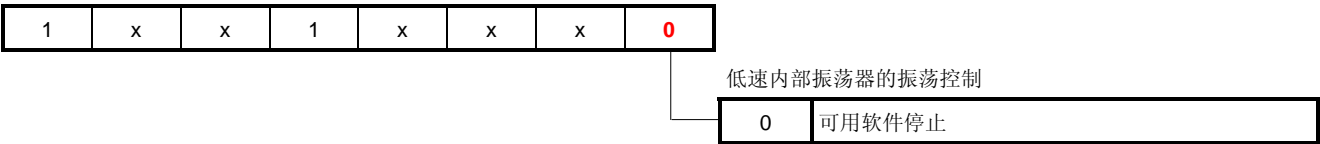
•汇编语言

```
MOV    WDTM,    #01101111B
```

• C 语言

```
WDTM = 0b01101111;
```

选项字节（地址：0080H）

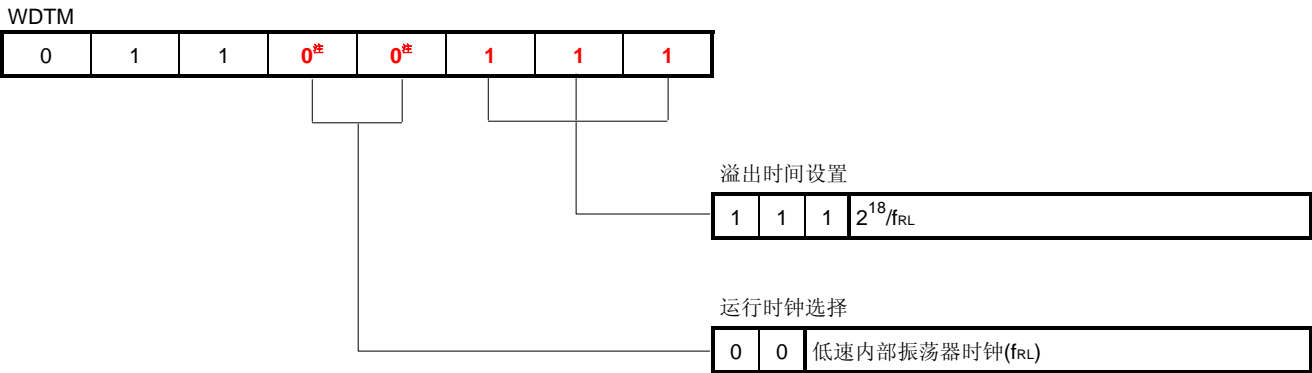


选项字节设置值为“1xx1xxx0（x：不关心，位 7 和 4 必须设置为 1）”。

当与保护字节设置一起描述软件时，下列结果。（如下示例所示，位 6、5 和 1 设置为 0，位 3 和 2 设置为 1。）

```
OPBT    CSEG    AT      0080H
        DB      10011100B
        DB      11111111B
```

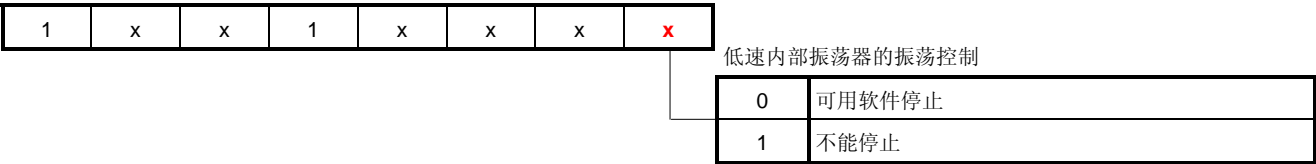
[示例 2] 用低速内部振荡器时钟(fRL) 作为看门狗计时器运行时钟，将溢出时间设置为最大周期 ($2^{18}/f_{RL}$)
(复位释放后与 WDTM 值内容相同)



如上述使用看门狗计时器，计时器不需要用程序设置，因为 WDTM 设置值与复位释放后相同。

注 如用选项字节将低速内部振荡器的振荡控制设置为“Cannot be stopped（不能停止）”，忽略写入值，可选择低速内部振荡时钟。

选项字节（地址： 0080H）

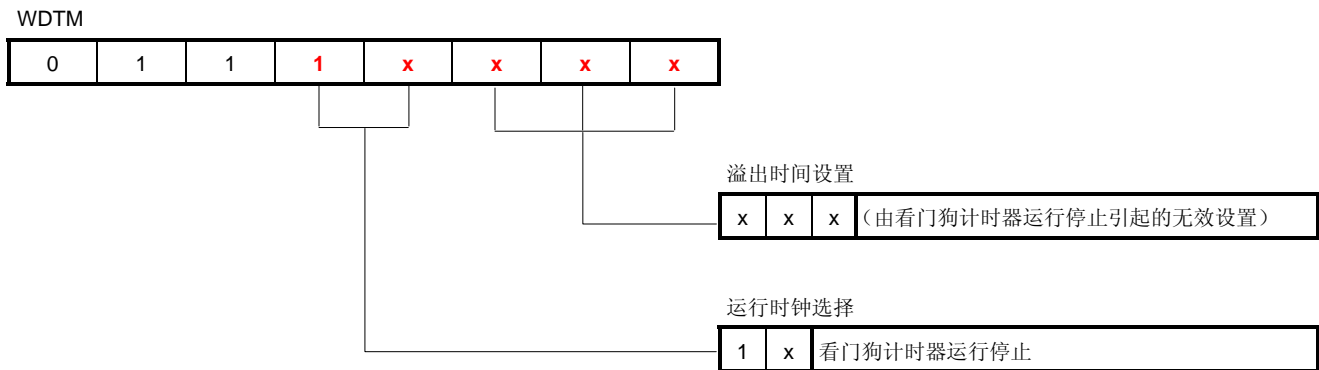


选项设置值为“1xx1xxx^x（x：不关心，位 7 和 4 必须设置为 1）”。

当软件与保护字节设置一起描述时，结果如下。（在如下示例中，位 6、5 和 1 设置为 0，位 3、2 和 0 设置为 1。）

OPBT	CSEG	AT	0080H
	DB		10011101B
	DB		11111111B

[示例 3] 停止看门狗计时器



WDTM 设置值为“0111xxxx (x: 不关心, 位 7、6 和 5 必须分别设置为 0、1 和 1)”。 (在如下示例中, 将位 3 和位 2 到 0 的“x”分别设置为 0 和 1。)

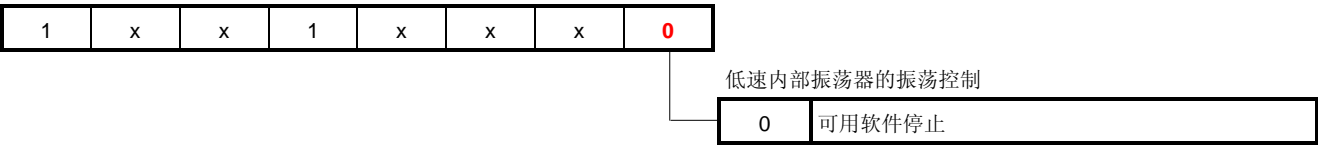
• 汇编语言

```
MOV     WDTM,    #01110111B
```

• C 语言

```
WDTM = 0b01110111;
```

选项字节 (地址: 0080H)



选项字节设置值为“1xx1xxx0 (x: 不关心, 将位 7 和 4 设置为 1)”。
当软件与保护字节设置一起描述时, 结果如下。 (在如下示例中, 将位 6、5 和 1 设置为 0, 将位 3 和 2 设置为 1) 当软件与保护字节设置一起描述时, 结果如下。 (在如下示例中, 将位 6、5 和 1 设置为 0, 将位 3 和 2 设置为 1)

```
OPBT    CSEG    AT      0080H
         DB      10011100B
         DB      11111111B
```


- 汇编语言程序示例（上述 [\[示例 1\]](#) 与示例程序内容相同）

```

XMAIN CSEG UNIT
RESET_START:
    MOVW    AX,    #STACKTOP
    MOVW    SP,    AX                ; 设置栈指针

    MOV      WDTM, #01101111B        ; WDT overflow time = 2^20/fx = 131.07 ms
    .
    .
    .
    MOV     A,     RESF              ; 读取复位源
    BF      A.4,    $CHECK_LVI      ; 如非 WDT 产生的复位, 转到 CHECK_LVI
    MOV     P2,    #00000001B       ; 打开 LED2

CHECK_LVI:
    BT      A.0,    $SET_CLOCK      ; 在 LVI 复位时, 省略后续 LVI 相关处理并转到 SET_CLOCK
    MOV     LVIS,   #00000111B      ; 设置低压检测电压 (VLVI) 为 2.85 V +-0.15 V
    SET1    LVION
    MOV     A,      #40              ; 分配 200 us 等待计数值

WAIT_200US:
    DEC     A
    BNZ     $WAIT_200US             ; 0.5[us/cclk] × 10[cclk] × 40[计数] = 200[us]

WAIT_LVI:
    MOV      WDTE, #0ACH            ; 清除看门狗计时器
    BT      LVIF,   $WAIT_LVI      ; 如果 VDD < VLVI, 则分支执行
    SET1    LVIMD                  ; 设置为当 VDD < VLVI 时内部复位信号生成
    .
    .
    .
    在溢出发生和重新开始计数前
    将 WDT 清零

MAIN_LOOP:
    XOR     P2,    #00000001B       ; LED1 反相输出
    MOV     CNT120, #120            ; 分配 120 ms 等待计数值

WAIT_120MS:
    CALL    !WAIT_1MS              ; 1 ms 等待子例程调用
    DBNZ    CNT120, $WAIT_120MS    ; 1[ms] × 120[计数] = 120[ms]
    MOV      WDTE, #0ACH            ; 将看门狗计时器清零
    BR      $MAIN_LOOP
    .
    .
    .
    在溢出发生和重新开始计数前
    将 WDT 清零

```

备注 上述等待时间（200 μ s）取 f_{CPU} （CPU 时钟频率）为 2 MHz 时计算，与示例程序中相同。

- C语言程序示例（上述 [\[示例 4\]](#) 与示例程序内容相同）

```

void hdwinit(void){
    unsigned char ucCnt200us; /*用于 200 us 等待的 8-位变量*/
    WDTM = 0b01101111;      /* WDT 溢出时间 =  $2^{20}/fx = 131.07\text{ ms}$  */
    •
    •
    •
    /* Check the reset source */
    ucRESF = RESF;           /* 读取复位源*/

    if (ucRESF.4){           /* 由 WDT 生成的复位 */
        P2 = 0b00000001;    /* 打开 LED2 */
    }

    if (!ucRESF.0){          /* 在 LVI 复位时，省略后续 LVI 相关处理*/

        /* 设置低压检测*/
        LVIS = 0b00000111; /* 设置低压检测电压 (VLVI) 为 2.85 V +/-0.15 V */
        LVION = 1;          /* 启用低压监测器运行*/

        for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){ /* 等待约 200 us */
            NOP();
        }

        while (LVIF){        /* 等待 VDD >= VLVI */
            WDTE = 0xAC;    /* 清除看门狗计时器*/
        }

        LVIMD = 1;           /* 设置为当 VDD < VLVI 时内部复位信号生成*/
    }
    •
    •
    •
}

void main(void){
    unsigned int unCnt120ms; /* 用于 120 ms 等待的 16-位变量*/

    EI();                    /* 启用向量中断*/

    while (1){
        P2 ^= 0b00000001;    /* LED1 反相输出*/

        for (unCnt120ms = 0; unCnt120ms < 6666; unCnt120ms++){
            /* 等待约 120 ms */
            NOP();
        }

        WDTE = 0xAC;        /* 将看门狗计时器清零 */
    }
    •
    •
    •
}

```

设置 WDT 溢出时间和运行时钟

在溢出发生和重新开始计数前将 WDT 清零


在溢出发生和重新开始计数前将 WDT 清零

备注 上述等待时间（200 μ s）取 f_{CPU} (CPU 时钟频率)为 2 MHz，与示例程序中相同。

第五章 用设备进行运行检查

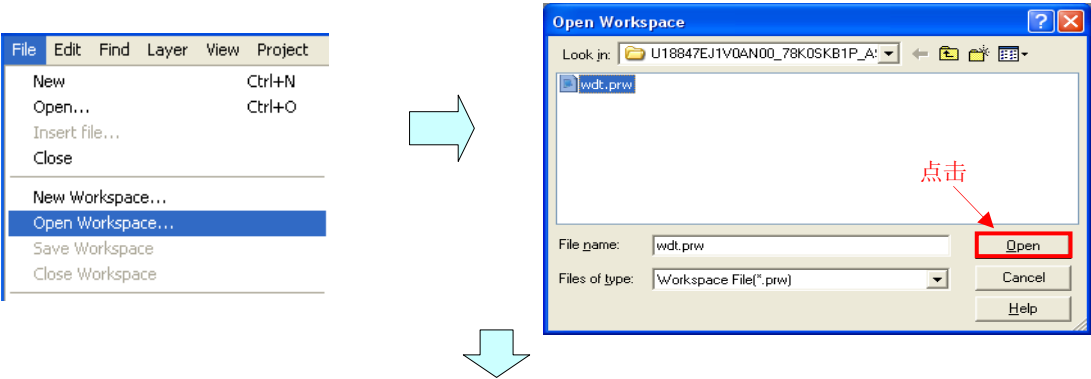
本章描述从构建到利用设备和下载的示例程序进行运行检查的流程。

5.1 构建示例程序

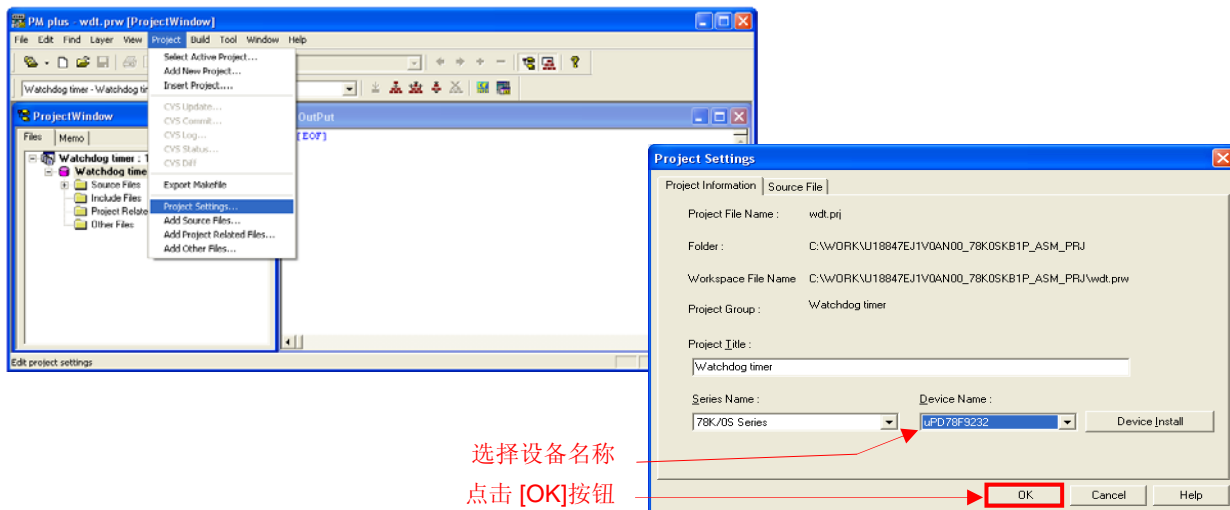
本部分描述了如何利用示例程序（源文件+项目文件）构建示例程序，该示例程序是通过点击图标下载的。关于如何构建其他下载的程序，请参见[78K0S/Kx1+示例程序启动向导应用注释](#)中第三章注册集成开发环境PM+项目和执行构建。


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“wdt.prw”。工作区生成，源文件将自动放入该工作区中。

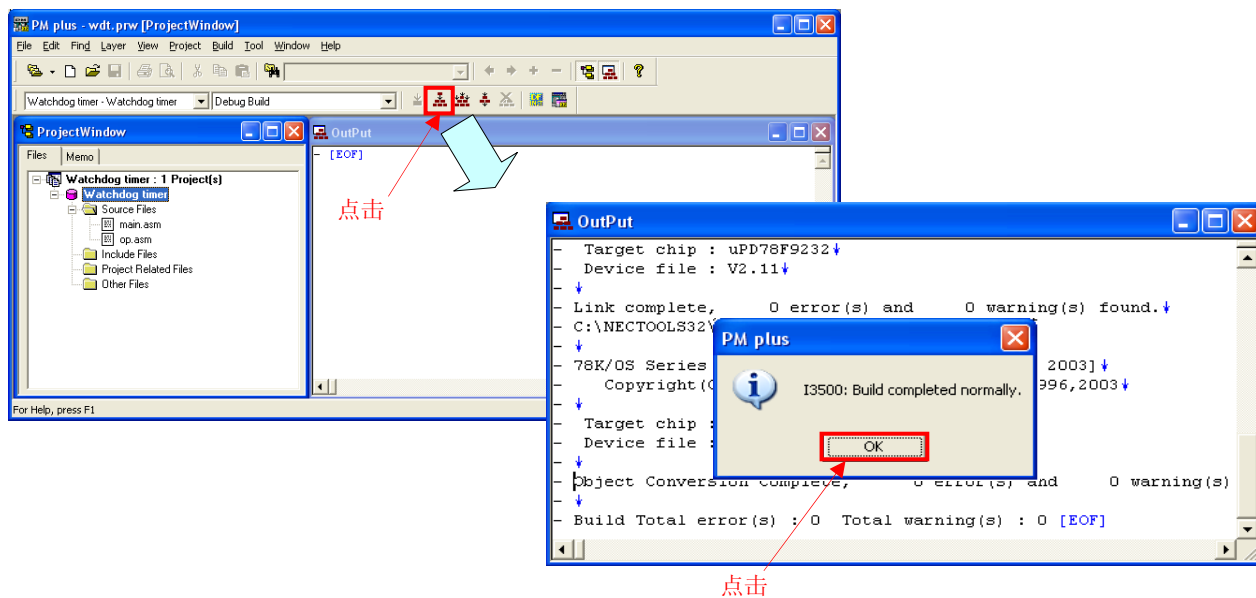


- (3) 从[项目]菜单选择[项目设置]。[项目设置]窗口打开后，选择要用的设备的名称（默认选择具有最大 ROM 或 RAM 的设备），点击[OK]。



- (4) 点击  ([构建]按钮)。当源文件正常构建时，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，将生成用于闪存写入的 HEX 文件。



生成用于闪存写入的HEX 文件 → 转到 [5.2](#)。

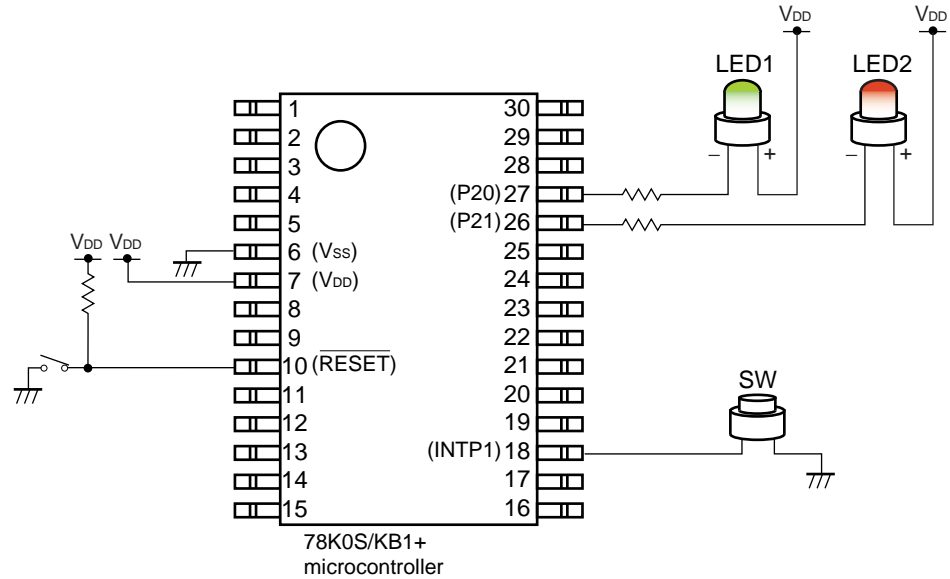
5.2 随设备运行

本部分描述利用设备进行运行检查的示例。

通过执行构建生成的 HEX 文件可写入设备闪存。

关于如何写入设备闪存，请参见 **78K0S/Kx1+简化闪存写入手册信息**（编辑中）。

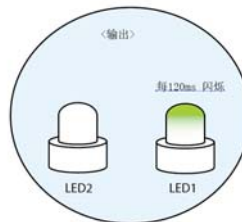
关于如何连接所用设备和外围硬件（开关和 LED 灯）示例如下所示。



本示例程序写入的设备的操作示例描述如下。

(1) 开关处理前的操作（主处理操作）

LED1 每 120 ms 闪烁。



(2) 按下开关后的操作（中断服务后的操作）

如开关按下时间小于 10 ms，可通过颤动辨别开关输入且返回到(1)中描述的操作。

如开关按下时间大于 10 ms，根据开关按下定时时间，可进行下列操作，发生几率为 50%（见下表）。

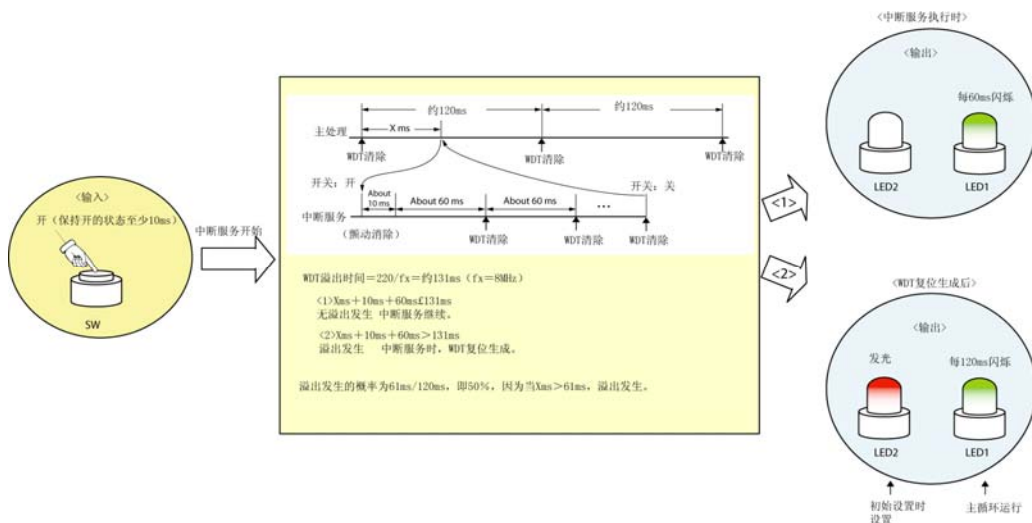
<1> 如开关按下后，溢出不是由看门狗定时器引起的

当开关打开（开关保持按下），LED1 每 60 ms 闪烁。当开关关闭时，LED1 每 120 ms 闪烁。

<2> 如开关按下后，溢出是由看门狗定时器引起的

看门狗定时器生成复位信号。复位释放后，LED2 亮，LED1 每 120 ms 闪烁。

备注 非看门狗定时器生成的复位信号关闭 LED2。



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/KA1+ 简化闪存写入手册MINICUBE2 信息		PDF
78K0S/Kx1+ 应用注释	示例程序启动向导	PDF
	示例程序（初始设置）LED 发光开关控制	PDF
	示例程序（中断）开关输入引起的外部中断	PDF
	示例程序（低压检测）电压低于2.7V检测时复位发生。	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
;
*****
;
;
;      看门狗定时器
;
*****
;
;
;<<历史>>
;      2007.6.--      发布
;
*****
;
;
;<<概述>>
;
;
;本示例程序提供使用看门狗定时器功能的例子。
;通过计算 8 MHz fx 设置看门狗定时器的溢出时间为 131.07 ms 并实施失控检测。(fx:系统时钟源)
;初始设置完成后，LED1 每 120 ms 闪烁，当按下开关时每 60 ms 闪烁。此时，看门狗定时器会产生复位，发生几率
;为 50%，取决于按下 SW 的计时，LED2 亮，因为 SW 输入产生中断 70 ms 后，看门狗定时器将被清除。
;# 通常，在中断开始和结束时，应通过清除看门狗计时器避免溢出。
;
;
;
; <主要设置内容>
;
; - 将低压检查电压(VLVI)设置为 2.85 V +-0.15 V
; -在 VDD >= VLVI 后当 VDD < VLVI 时生成内部复位信号（低压监测器）
; - 设置 CPU 时钟频率为 4 MHz
; - 选择 fx 作为看门狗定时器(WDT)计时时钟并
;   设置溢出时间为 131.07 ms
; - 在主循环中每 120 ms 清除看门狗定时器
; - 开关输入中断产生 70 ms 后，清除看门狗定时器，如果其后开关继续输入，每 60 ms 清除看门狗定时器
; - 设置外部中断 INTP1 有效沿为下降沿
; - 在开关输入时设置颤动解除时间为 10 ms
;
;
;
; <SW 输入和 LED1>
;
;
;      +-----+
;      | SW |   LED1 |
;      | (P43) | (P20) |
;      |-----|
;      | 关 | 每 120 ms 闪烁|
;      |-----|
;      | 开 | 每 60 ms 闪烁 |#
;      +-----+
;
; #当 SW 打开时，WDT 复位产生，产生几率为 50%。
```



```
; 在这种情况下，即使 SW 打开，LED1 每 120 ms 闪烁。
```

```
; <复位源和 LED2>
```

+-----+	
复位源	LED2
	(P21)
+-----+	
除了 WDT	关
+-----+	
WDT	开
+-----+	

```
; <<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)	INTP0
DW	INTERRUPT_P1	;(0A)	INTP1
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
CNT10: DS 1 ; 等待 10 ms
CNT60: DS 1 ; 等待 60 ms
CNT120: DS 1 ; 等待 120 ms

```

```

;=====
;
;
; 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+检测低压+设置时钟

```

```

;-----

```

```

;----- 初始化看门狗计时器 -----

```

```

MOV WDTM, #01101111B ; WDT 溢出时间 =  $2^{20}/f_x = 131.07$  ms

```

```

;----- 设置时钟 <1> -----

```

```

MOV PCC, #00000000B ; 供给 CPU 的时钟 (fcpu) = fxp (= fx/4 = 2 MHz)
MOV LSRM, #00000001B ; 停止低速内部振荡器的振荡

```

```

;----- 初始化端口 2 -----

```

```

MOV P2, #00000011B ; 将 P20, P21 输出锁存器设置为高(关闭 LED1, LED2) 和 设置 P22, P23
为低
MOV PM2, #11110000B ; 设置 P20-P23 为输出模式

```

```

;----- 检查复位源 -----

```

```

MOV A, RESF ; 读取复位源
BF A.4, $CHECK_LVI ; 如非 WDT 产生的复位, 转到 CHECK_LVI

```

```

MOV    P2,    #00000001B    ; 打开 LED2

CHECK_LVI:

BT     A.0,    $SET_CLOCK    ; 在 LVI 复位时, 省略后续 LVI 相关处理, 转到 SET_CLOCK

;---- 设置低压检测 ----

MOV    LVIS,   #00000111B    ; 将低压检测电压(VLVI)设置为 2.85 V +-0.15 V
SET1    LVION   ; 启用低压监测器的运行
MOV    A,      #40           ; 分配 200 us 等待计数值

;---- 200 us 等待 ----

WAIT_200US:
DEC     A
BNZ     $WAIT_200US          ; 0.5[us/clock] x 10[clock] x 40[计数] = 200[us]

;---- 等待 VDD >= VLVI ----

WAIT_LVI:
MOV     WDTE,   #0ACH        ; 清除看门狗计时器
BT      LVIF,   $WAIT_LVI    ; 如果 VDD < VLVI, 分支执行
SET1    LVIMD   ; 设置为当 VDD < VLVI 时生成内部复位信号

;---- 设置时钟 <2> ----

SET_CLOCK:
MOV     PPCC,   #00000001B    ; 设置供给外围硬件的时钟(fxp) = fx/2 (= 4 MHz)
; -> 供给 CPU 的时钟 (fcpu) = fxp = 4 MHz

;-----
;
;
;      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 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    ; 连接片上上拉电阻为 P43
MOV     PM4,    #00001000B    ; 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

```

MOV    P12,    #00000000B    ;将 P120-P123 的输出锁存器设置为低
MOV    PM12,   #11110000B    ;设置 P120-P123 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                                ;启用向量中断

```

主循环

MAIN_LOOP:

----- Reverse LED1 -----

```

XOR    P2,     #00000001B    ;LED1 反相输出

```

```

MOV    CNT120,    #120    ;分配 120 ms 等待计数值

```

----- 120 ms 等待 -----

WAIT_120MS:

```

CALL    !WAIT_1MS          ;等待 1 ms 后, 子例程调用
DBNZ    CNT120,            $WAIT_120MS    ;1[ms] x 120[计数] = 120[ms]

```

----- 清除看门狗计时器 -----

```

MOV    WDTE,   #0ACH          ;清除看门狗计时器
BR     $MAIN_LOOP            ;转到 MAIN_LOOP

```

外部中断 INTP1

INTERRUPT_P1:

```

PUSH    AX                    ;保存 AX 寄存器数据至栈中

```

```

MOV    CNT10,   #10           ;分配 10 ms 等待计数值

```

----- 10 ms 等待处理颤动 -----

WAIT_10MS:

```

CALL    !WAIT_1MS          ;等待 1 ms 子例程调用
DBNZ    CNT10, $WAIT_10MS    ;1[ms] x 10[计数] = 10[ms]

```

```

        CLR1   PIF1                ;清除 INTP1 中断请求

;----- 颤动检测辨识 -----
        BT     P4.3,   $END_INTP1   ;如没有开关输入, 分支执行

SW_ON:
;----- LED1 反相 -----
        XOR    P2,     #00000001B   ;LED1 反相输出

        MOV    CNT60, #60           ;分配 60 ms 等待计数值
;----- 60 ms 等待 -----
WAIT_60MS:
        CALL   !WAIT_1MS           ;等待 1 ms 子例程调用
        DBNZ   CNT60, $WAIT_60MS   ;1[ms] x 60[计数] = 60[ms]

;----- 清除看门狗计时器 -----
        MOV    WDTE, #0ACH         ;清除看门狗计时器

;----- 辨识开关输入状态 -----
        BF     P4.3,   $SW_ON       ;如开关输入继续, 分支执行

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

;-----
;      用于 1 ms 等待的子例程 (用于 4 MHz 操作)
;
;
在 ms 单元中, 通过使用软件运行会合的子例程。
当在 ms 单元中运行会合时, 通过所需要的次数调用该例程。
在 ms 单元中用#num 标明的时间可通过如下设置进行精确测量。(saddr: 短直接可寻址存储区)
;      MOV     saddr, #num
;loopxx:
;      CALL    !WAIT_1MS
;      DBNZ    saddr, $loopxx
;
;-----

WAIT_1MS:
        PUSH   BC                  ;保存 BC 寄存器数据至栈中
        MOV    B,     #220         ;分配 220-时间循环计数值
LOOP_220:
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        DBNZ   B,     $LOOP_220    ;用于 1 ms 等待的 220-时间循环
        NOP
        NOP
        POP    BC                  ;还原 BC 寄存器数据
        RET                       ;从子例程返回

end

```

● main.c (C 语言版)

```

/*****
日电电子 78K0S/KB1+

*****
78K0S/KB1+ 示例程序
*****
看门狗计时器
*****
<<历史>>
2007.6.-- 发布
*****

```

<<概述>>

本示例程序提供使用看门狗计时器功能的例子。

通过计数 8MHz fx，将看门狗计时器的溢出时间设置为 131.07 ms，并执行失控检测 (fx: 系统时钟源)

初始设置完成后，LED1 每 120 ms 闪烁，且当按下开关时，每 60 ms 闪烁。

此时，根据按下 SW 的时间，看门狗计时器有 50% 的几率会产生复位，LED2 亮，因为 SW 输入中断产生后 70 ms，看门狗计时器被清除。

通常，在中断开始和结束时，应通过清除看门狗计时器避免溢出。

<主要设置内容>

- 声明由中断运行的函数：INTP1 -> fn_intp1()
- 选择 fx 为看门狗计时器(WDT)计数时钟，且
设置溢出时间为 131.07 ms
- 将低压检测电压(VLVI)设置为 2.85 V +/-0.15 V
- 在 VDD >= VLVI 后当 VDD < VLVI 时生成内部复位信号（低压监测器）
- 设置 CPU 时钟频率为 4 MHz
- 在主循环中，每 120 ms 清除看门狗计时器
- 开关输入中断产生 70 ms 后清除看门狗计时器，如在其后开关输入继续，每 60 ms 清除看门狗计时器
- 将外部中断 INTP1 有效沿设置为下降沿
- 在开关输入中将时钟解除时间设置为 10 ms

<SW 输入和 LED1>

SW	LED1
(P43)	(P20)
关	每 120 ms 闪烁
开	每 60 ms 闪烁 #

当 SW 开时，WDT 产生复位，产生几率为 50%。
在这种情况下，即使 SW 开，每 120 ms LED1 闪烁。

<复位源和 LED2>

复位源	LED2
	(P21)
除了 WDT	关

```
|-----|-----|
| WDT      | 开    |
+-----+
```

<<I/O 端口设置>>

输入: P43

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

所有未使用端口设置未输出模式。

*****/

/*=====

预处理指令(#pragma)

```

=====*/
#pragmaSFR          /* 可在 C 源程序级描述 SFR 名称*/
#pragmaEI          /* 可在 C 源程序级描述 EI 指令*/
#pragmaNOP          /* 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

*****/
sreg unsigned char ucRESF; /* 用于复位源检查的 8-位变量 (高速内部 RAM 区) */

void hdwinit(void){
    unsigned char ucCnt200us; /* 用于 200 us 等待的 8-位变量*/

/*-----
检测低压+初始化看门狗计时器+设置时钟
-----*/
/* 初始化看门狗计时器*/
WDTM = 0b01101111; /* WDT 溢出时间 =  $2^{20}/f_x = 131.07 \text{ ms}$  */

/* Set the clock <1> */
PCC = 0b00000000; /* 供给 CPU 的时钟(fcpu) =  $f_x/4 = 2 \text{ MHz}$  */
LSRCM = 0b00000001; /* 停止低速内部振荡器的振荡*/

/* 初始化端口 2 */
P2 = 0b00000011; /* 将 P20, P21 输出锁存器设置位高 (关闭 LED1, LED2) 将 P22, P23
设置为低*/
PM2 = 0b11110000; /* 设置 P20-P23 为输出模式*/

/*检查复位源*/
ucRESF = RESF; /* 读取复位源*/

如(ucRESF.4){ /* WDT 产生复位*/
    P2 = 0b00000001; /* 点亮 LED2 */
}

如(!ucRESF.0){ /* 在 LVI 复位时, 省略后续 LVI 相关的处理*/

    /*设置低压检测*/
    LVIS = 0b00000111; /* 设置低压检测电压(VLVI)为  $2.85 \text{ V} \pm 0.15 \text{ V}$  */
    LVION = 1; /* 启用低压监测器运行*/

    对于 (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){ /* 等待 200 us 左右 */
        NOP();
    }

    当 (LVIF){ /* 等待 VDD >= VLVI */
        WDTE = 0xAC; /*清除看门狗计时器*/
    }

    LVIMD = 1; /* 当 VDD < VLVI 时设置从而内部复位信号产生*/
}

/* 设置时钟<2> */
PPCC = 0b00000001; /* 供给外围硬件的时钟(fxp) =  $f_x/2 (= 4 \text{ MHz})$  */
/* -> 供给 CPU 的时钟 (fcpu) =  $f_x = 4 \text{ MHz}$  */

```



```

/*-----
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 3
-----*/
P3  = 0b00000000;    /* 将 P30-P33 的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;   /* 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*/

返回;
}

/******

主循环

*****/
void main(void){
    unsigned int unCnt120ms; /* 用于 120 ms 等待的 16-位变量*/

    EI();                    /* 启用向量中断*/

    当 (1){
        P2 ^= 0b00000001;    /* LED1 反相输出 */

        for (unCnt120ms = 0; unCnt120ms < 6666; unCnt120ms++){ /* 等待 120 ms 左右 */
            NOP();
        }

        WDTE = 0xAC;         /* 清除看门狗计时器*/
    }
}

/******

```

外部中断 INTP1

```
*****/
__interrupt void fn_intp1(){
    unsigned int unCnt;          /* 用于等待的 16-位变量*/

    for (unCnt = 0; unCnt < 555; unCnt++){      /* 等待 10 ms 左右(用于抖动解除) */
        NOP();
    }

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

    当 (!P4.3){                      /* SW 输入时处理*/
        P2 ^= 0b00000001;          /* LED1 反相输出*/

        for (unCnt = 0; unCnt < 3333; unCnt++){    /* 等待 60 ms 左右 */
            NOP();
        }

        WDTE = 0xAC;              /*清除看门狗计时器*/
    }

    返回;
}
```

● op.asm （汇编语言和 C 语言版共用）

```

=====
;
;
;      选项字节
;
;
=====
OPBT  CSEG  AT      0080H
      DB   10011100B    ;选项字节区
;
;      ||||
;      |||+-----低速内部振荡器可用软件停止
;      |++-----高速内部振动时钟(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	2008年02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