中微8051微控制器

应用注意事项

Rev. 1.06

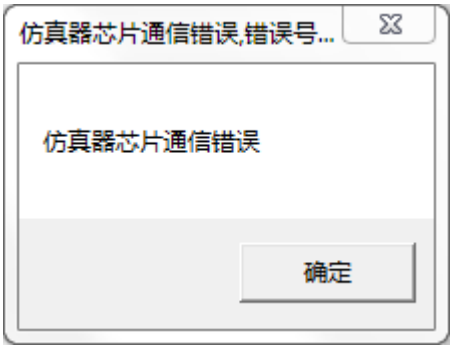
目录

1. 仿真使用注意事项 4
2. BANK 使用 5
3. 低功耗 5
4. ADC 6
5. I2C 8
6. SPI 8
7. UART 9
8. IO 口 9
9. WDT 10
10. 触摸 10
11. 中断 11
11.1 硬件自动清除 11
11.2 软件清除 11
11.3 读写寄存器清除 11
12. 外部振荡器 12
13. DATA FLASH 说明 12
14. FLASH 使用 13
15. BOOT 使用 14
16. 芯片工作电压 14
17. TA 时序操作注意事项 14
18. 运放失调电压 15
19. LVD 15
20. LSE 16
20.1 注意事项 16
20.2 推荐操作 17
21. 其他注意事项 18
22. 版本修订说明 19

本注意事项适用中微 8051 芯片：CMS8S3660、CMS8S3680、CMS8S6980、CMS8S6990、CMS8S5880、CMS8S5887、CMS8S5888、CMS8S5889、CMS80F231x、CMS80F253x、CMS80F261x、CMS80F251x、CMS80F751x、CMS8S7885、CMS8S7895、CMS8S6997、CMS8S6998、CMS8S6999、CMS8S589x、CMS8S369x、CMS8M35xx 等系列。

1. 仿真使用注意事项

- 在配置仿真模式时，建议不要在 DSDA 口接外部下拉电阻；
- 调试口开对应的 LED/LCD SEG 使能，调试模式会失效；
  - CMS80F253x 和 CMS80F751x 系列芯片 LED/LCD SEG 功能优先级比调试功能高，芯片会优先处理 LED/LCD SEG；在调试时，请关闭调试口的 SEG 使能（默认关闭）。（CMS80F261x 系列芯片调试功能优先级比 LED/LCD SEG 功能高）
- DEBUG RST 前如果发生 WDG 复位或寄存器复位，DEBUG RST 后芯片不会 STOP；
  - 如果遇到这个问题，请下载最新的工具；可避免这个问题。
- 如果程序开启 LSE_TIMER 定时功能，在仿真过程中定时器不受仿真相关命令控制（单步，全速等）；
  - 这是正常的现象，保持时钟都是一直运行的；类似的模块还包括 UART、PWM。
- 出现下面图标后，查看是否关闭了调试功能；如果下载程序还报错，切换 HSI 频率，然后执行下载程序操作，这个时候一般就可以调试了。

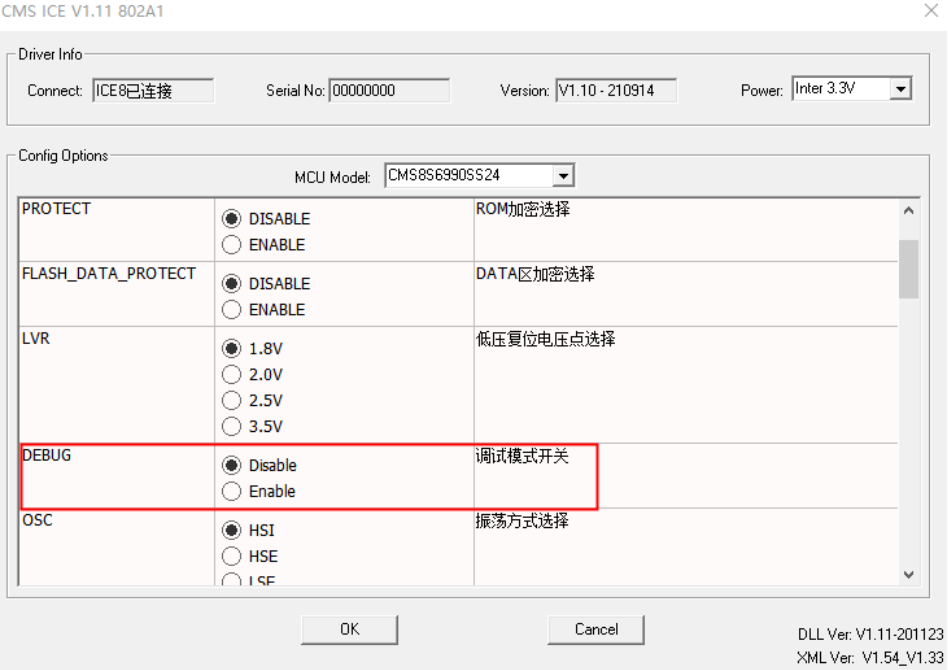


2. BANK 使用

使用 CMS80F261x、CMS80F251x 或 CMS80F751x 时，当使用不同 BANK，如果在中断服务程序里有切换 BANK 的操作，请在进入中断前保存 BANK 的值，退出中断时再恢复。（防止出现退出中断后，BANK 状态不一致的问题）

3. 低功耗

- 使用低功耗的时候请把调试功能关闭。不用的 IO 配置为输出，高电平状态，否则功耗降不下来。若使用中断唤醒系统，需要开启全局中断使能，即对应唤醒中断源的中断使能。



- 使用 CMS8S6990 时，休眠唤醒后建议添加 10ms 的延时。
- 当使用 LSE 唤醒休眠模式时，请在进入休眠前清 LSE 计数使能位后再使能（清 LSE 计数值）。

4. ADC

使用 51 系列芯片的 ADC 时，以下几点需要注意：

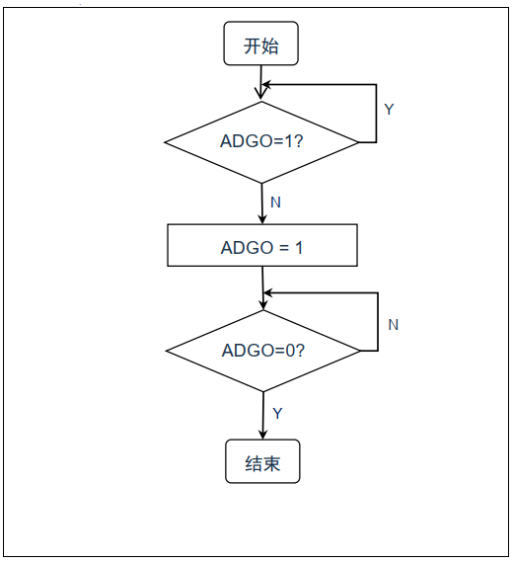
1) 如果遇到采样精度不够，可以适当降低 ADC 速度，或者减小 ADC 采样电路的阻抗会有一定帮助。

2) 使用 ADC 硬件触发时，相邻两触发信号时间间隔应大于 ADC 转换时间，避免在 ADC 转换期间产生触发信号。

3) 当程序中同时使用软件和硬件触发 ADC 时，应在软件触发 ADC 转换前关闭硬件触发。

4) ADC 在转换过程中忽略该时间段内所有硬件和软件触发信号。

5) 使用 ADC 软件触发时，应遵循以下流程图。



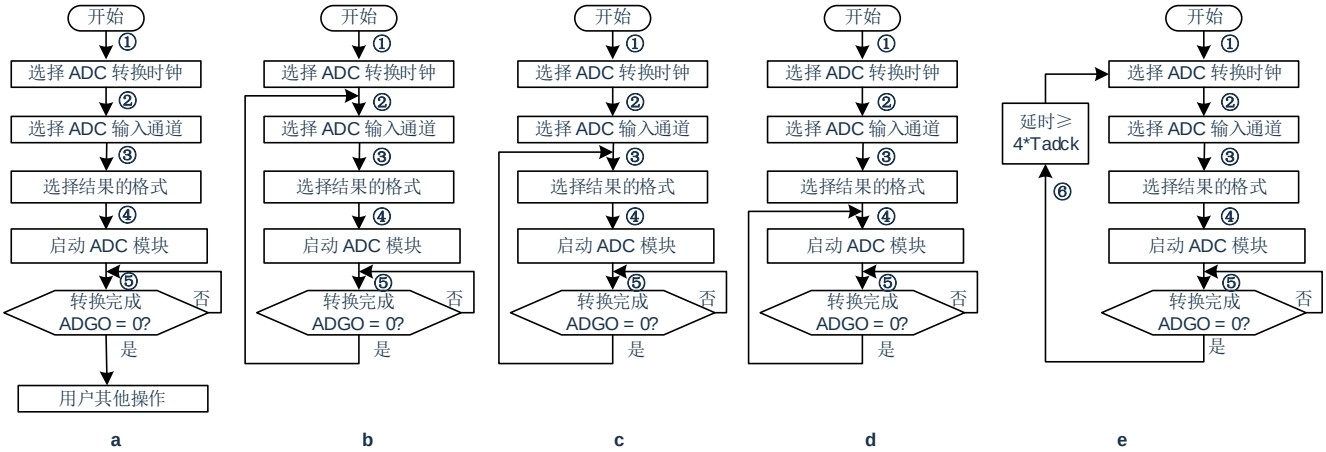
参考代码如下：

```
if(!ADC_IS_BUSY)                    //判断 ADC 是否正在转换
{
        _nop_();
        ADC_GO();                   //软件触发 ADC 转换
        while(ADC_IS_BUSY);         //等待 ADC 转换完毕
}
```

```
if(!ADC_IS_BUSY)                    //判断 ADC 是否正在转换
{
        _nop_();
        ADC_DisableHardwareTrig();  //关闭硬件触发
        ADC_GO();                   //软件触发 ADC 转换
        while(ADC_IS_BUSY);         //等待 ADC 转换完毕
        ADC_EnableHardwareTrig();   //开启硬件触发
}
```

6) ADC 分频比操作注意事项：检测到上一次 ADC 转换完成后，如果需要对 ADC 分频比进行再次操作，建议至少等待 4 个 ADC 转换时钟 Tadck。说明如下：

- ✓ 如图步骤⑥所示：检测到上一次 ADC 转换完成后，如果需要对 ADC 分频比进行再次操作，建议至少等待 4 个 ADC 转换时钟 Tadck；
- ✓ 如 a、b、c、d 所示：再次使用 ADC 时，只要不操作 ADC 分频比，不需要延时。



5. I2C

- 当使用 CMS80F261x 芯片，IO 口复用功能，配置 PS_SCL,PS_SDA 为只写寄存器时，必须采用直接 赋值的方式对这些寄存器赋值，禁止在程序中使用与或操作。
- 当 IO 口作为 SCL/SDA 时，内部上拉电阻和开漏的设置请参考表格：

功能	功能说明	上拉电阻控制寄存器 PxUP	开漏控制寄存器 PxOD	上拉电阻控制寄存器 PxUP	开漏控制寄存器 PxOD
SCL	I2C 时钟输入输出	由 PxUP 设置上拉打开或则关闭	强制开启开漏，与 PxOD 设置无关	由 PxUP 设置上拉打开或者关闭	强制开启开漏，与 PxOD 设置无关
SDA	I2C 数据输入输出	由 PxUP 设置上拉打 开或者关闭	强制开启开漏，与 PxOD 设置无关	由 PxUP 设置上拉打开或者关闭	强制开启开漏，与 PxOD 设置无关
具体芯片型号(不同的芯片 对应的操作不同)		CMS8S3660，CMS8S5880，CMS8M35xx， CMS8S3680，CMS8S6980， CMS8S6990，CMS80F231x， CMS80F261x，CMS80F251x， CMS80F751x，CMS8S78xx， CMS8S588x，CMS8S5887， CMS80F253x		CMS80F761x， CMS8S6998，CMS80F262x， CMS8S369x，CMS8S589x， CMS80F161x，CMS8S6997_8_9	

6. SPI

SPI 只支持 8 位，MSB（高位在前）模式。

当使用 CMS80F261x 芯片，IO 口复用功能，配置 PS_SCLK，PS_MOSI，PS_MISO，PS_NSS 为只写寄存器，必须采用直接赋值的方式对这些寄存器赋值，禁止在程序中使用与或操作。

7. UART

- CMS80F261x 芯片的 PS_RXDI 为只写寄存器，在对这个寄存器赋值时采用直接赋值，禁止在程序中使用与或操作。
- 当 IO 口作为 RX/TX 时，内部上拉电阻和开漏的设置请参考表格：

功能	功能说明	上拉电阻控制寄存器 PxUP	开漏控制寄存器 PxOD	上拉电阻控制寄存器 PxUP	开漏控制寄存器 PxOD
TXD	UART 数据输出	与 PxUP 设置无关	与 PxOD 设置无关	与 PxUP 设置无关	与 PxOD 设置无关
RXD	UART 数据输入，异步模式	与 PxUP 设置无关	与 PxOD 设置无关	由 PxUP 设置上拉打开或者关闭	与 PxOD 设置无关
	UART 同步模式数据输入、输出	强制上拉，与 PxUP 设置无关	与 PxOD 设置无关	强制上拉，与 PxUP 设置无关	与 PxOD 设置无关
具体芯片型号		CMS8S3660， CMS8S5880，CMS8M35xx， CMS8S3680，CMS8S6980， CMS8S6990，CMS80F231x， CMS80F261x，CMS80F251x， CMS80F751x，CMS8S78xx， CMS8S588x，CMS8S5887， CMS80F253x		CMS80F761x， CMS8S6998，CMS80F262x， CMS8S369x，CMS8S589x， CMS80F161x，CMS8S6997_8_9	

8. IO 口

- 当 IO 口的某些功能需要设置几个 IO 口为 1 组来一起使用时，请注意每组 IO 内的各个 IO 口具体使用和作为 1 组功能间的相互关联。
- 使用芯片时请认真阅读芯片的数据手册。当芯片的使用条件超过手册的最大或最小值时，属于超范围工作，不保证运行效果。

9. WDT

在修改 WDCON 时，是有一个逻辑语序的，需要先把 0xAA 和 0x55 写到 TA，然后再给 WDCON 赋值。在使用 C 语言进行定义时，建议将这个操作语序定义为一个宏定义，然后在需要调用的地方直接调用定义的宏定义。在写这个 TA 时序前，将中断关闭 EA=0，等操作完成后根据实际情况再打开 EA=1。

```
TA = 0xAA;
TA = 0x55;
WDCON = 0x80;
```

例如：#define __system_software_reset() TA=0xAA; TA=0x55;WDCON=0X80; 然后在需要执行的地方调用__system_software_reset()；就好了。

```
//        TA = 0xAA;
//        TA = 0x55;
//        WDCON = 0x80;

      __system_software_reset();
```

10. 触摸

触摸模块的启动时间为 25us，请在启动完成后再处理。

11. 中断

本章节主要介绍中断使用注意事项。芯片的中断清除方式有:进入中断服务程序后硬件自动清除、软件清除和读/写寄存器清除三种。

11.1 硬件自动清除

芯片中 INT0（外部中断 0）、INT1（外部中断 1）、T0（定时器 0）、T1（定时器 1）、T3（定时器 3）、T4（定时器 4）产生的中断标志位会在系统进入中断服务程序后被硬件自动清除。

硬件自动清除中断标志的条件：

1) 打开总中断使能（EA=1）。
2) 打开对应的模块中断使能。
3) 产生中断并且进入中断服务程序。

因此，在中断服务程序中获取的中断标志位会一直为 0。

若中断使能关闭，则以上模块的标志位也可使用软件写 0 清除。

11.2 软件清除

系统中存在只能用软件清除的标志位。这些标志位在进入中断服务程序后不会自动清除，需要软件写 0 清除。否则在退出中断服务程序后又会再次进入中断服务程序。

软件清除操作需要注意：当多个中断标志位在同一个寄存器中（如比较器模块的 CNIF、PWM 模块的 PWMPIF、PWMZIF、PWMUIF 等），这些标志位产生的时刻相互关联时，不建议用读-修改-写操作来进行。建议直接写 0 操作，其他不相关的标志位写 1，如清除 bit0：PWMUIF = 0xFE；

该操作对不相关的中断标志写 1 无实际影响。

11.3 读写寄存器清除

系统中有标志位并不是写 0 到该标志位清零，而需要读/写其他寄存器来清除标志位。如 SPI 中断标志寄存器中的传输完成 标志位 SPISIF，置 1 后需要先读 SPSR，再读/写 SPDR 后清零。

12. 外部振荡器

在芯片使用外部晶振的时候，有下面两点需要注意：

1. 晶振不超过 16MHz
2. 在使用 FLASH 时，使用外振做主频的情况下；某些芯片每次做擦除后都需要连续写 256 个 bytes 的 0xFF 到 FLASH，具体的操作请参考章节 14 FLASH 使用。（具体哪些芯片需要这样操作请参考 14 章的具体描述）

如果正在擦除一个 sector，需要这个 sector 擦除完成后再操作其他 sector。

13. Data Flash 说明

➢DATA FLASH 的说明在芯片参考手册或用户手册里。Data Flash 的使用方法和操作与通用 FLASH 一样，只是通过 FLASH 存储器控制寄存器 MCTRL 的第 4 位 MREG 来选择是操作 Data Flash（设置为 1）还是操作程序 Flash（设置为 0）。

Bit4	MREG:	Flash区域选择位；
	1=	选择数据区（低10位地址有效）；
	0=	选择程序区（低16位地址有效）。

➢ 当调用 FLASH 库的时候，在库函数里都用了判断的方式设置 MCTRL 的 MREG 寄存器。但是如果使用的程序只需要操作 DATA FLASH 一个时，建议使用下图右侧的方式设置 MREG。

```
if(codeordata==0)
  MCTRL = 0X19;
else
  MCTRL = 0X09;
```

不建议

```
MCTRL = 0X09;
```

建议

➢ 每次对 FLASH 的操作后，都需要加入_nop6_();以等待 FLASH 执行完成。（加入 6 个 NOP()）

➢ CMS80F261x 系列的芯片，在程序写保护 0-4K 空间时，不能对 DATA FLASH 区进行写操作了。

➢CMS8M35xx、CMS8S3680、CMS8S6980、CMS8S6990 和 CMS80F231x 系列的芯片，在程序写保护 0-2K 空间时，不能对 DATA FLASH 区进行写操作了。

14. FLASH 使用

8051 芯片的 FLASH 使用有几点需要注意：

1) 对于某些芯片，每次对芯片擦除后都需要连续写 256 个 bytes 的 0xFF 到 FLASH（写地址可以是不会用到的 FLASH 某一地址，也可以是擦除 Sector 中的任意地址），具体操作参考下图：

具体的实现参考下图：

```
addr=0x1000;
FLASH_Erase(FLASH_CODE,addr);     //扇区擦除时间约为：4.6ms
for(i=0;i<256;i++)                //连续256 bytes的写Flash实现时间间隔
{
  FLASH_Write(FLASH_CODE,addr, 0xFF); //写地址可以是这个sector里的任意地址
}
```

具体需要做上面操作的芯片，请参考下面的列表：

CMS8M35xx, CMS8S3680, CMS8S6990, CMS8S5885, CMS80F231x, CMS80F261x， CMS80F251x, CMS80F751x，CMS8S78xx，CMS8S588x，CMS8S5887 和 CMS80F253x。

2) 对 FLASH 进行操作前需要对 FLASH 进行解锁，操作完 FLASH 需要给 FLASH 上锁

3) 对于 CMS8S3660、CMS8S3680、CMS8S5880、CMS8S6980、CMS8S6990、CMS80F261x、CMS8M35xx 等芯片，对 FLASH 进行解锁时，可直接操作 MLOCK 寄存器

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
读写	W	W	W	W	W	W	W	W
复位值	0	1	0	1	0	1	0	1

Bit7~Bit0　MLOCK<7:0>: 存储器操作使能位（该寄存器仅支持写操作）；
AAH= 允许存储器相关R/W/E操作；
00H/FFH/55H= 不允许操作；
其他= 禁止写入。

以上图为例，MLOCK = 0xaa，解锁 FLASH；MLOCK = 0x55，将 FLASH 锁住。

4) 对于 CMS8S5887、CMS8S5888、CMS8S5889、CMS80F231x、CMS80F253x、CMS80F251x、CMS80F751x、CMS8S7885、CMS8S7895、CMS8S6997、CMS8S6998、CMS8S6999、CMS8S589x、CMS8S369x 等芯片，对 FLASH 进行解锁时，需要写 TA 时序。写 TA 时序参考下图：

```
MOV    TA,#0AAH
MOV    TA,#055H
MOV    MLOCK,#0AAH
```

需要注意的是：在写入期间，需要保证中间不能插入其他任何操作，包括中断操作，否则写入 MLCOK 的操作会失败，这里给出 2 种处理方式：

a. 建议将这几条语句定义为宏指令进行调用:

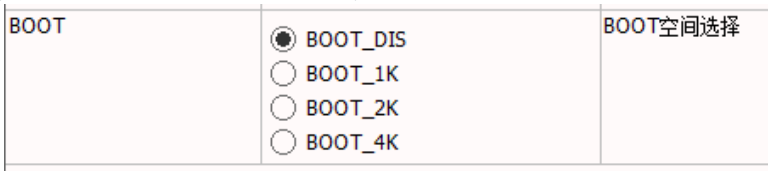
#define MLOCK_Enable() TA=0xAA, TA=0x55, MLOCK=0xAA

b. 在写 TA 时序之前关闭中断（EA=0），操作完成后再打开中断使能（EA=1）

15. BOOT 使用

程序执行的时候，如果把指针指向一个没有的地址或超过芯片的地址（比如使用 BOOT 功能时，下一条执行地址指向一个超过 BOOT 区间的地址），这是不允许的。

不同的芯片 FLASH 程序空间大小不同，FLASH 程序区分为 BOOT 区和 APROM 区， BOOT 区大小由用户配置寄存器分配，在 KEIL 工程的属性里选择，如下图所示：



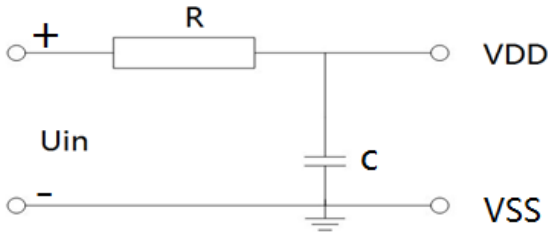
因为 BOOT 是 FLASH 的一部分，所以它的使用也需要遵循 14 章 FLASH 的使用注意事项（解锁、写时序、擦除间间隔擦除时间等）。

16. 芯片工作电压

在设计芯片供电电源电路时，需要注意芯片的复位电压曲线、上电曲线等条件。

在芯片电源电压供电电路设计时（特别是 CMS8S5880）：

- 当芯片工作在 2.1V~4.5V 的电压范围内时，可在 VDD 与 VSS 引脚间并联 0.1uf（104）电容。
- 若需要芯片工作在 4.5V~5.5V 的电压范围内时，可在电源供电电路上串联 4.7R ~ 5.1R 的电阻并与 VSS 并联 2.2uF ~4.7uF 电容，组成 RC 滤波电路（如下图所示）。



RC 滤波电路

17. TA 时序操作注意事项

在操作某些寄存器时，需要先对 TA 写时序；写 TA 时序完成后对寄存器的操作才有效。写 TA 时序要求整个写时序的过程不能被中断，否则写 TA 时序不成功。

在写 TA 时序前请先将中断关闭，等写时序完成后根据需要再打开。

对于 8051 芯片，对 WDCON、CLKDIV、SCKSEL、MLOCK、WWCON0、WWCON1 和 WWCMPD 寄存器操作时需要写 TA 时序（最终寄存器以实际产品为准）。

也可以把写时序连续语句定义为一个宏，如：

```
#define MLOCK_Enable() TA=0xAA, TA=0x55, MLOCK=0xAA;
```

18. 运放失调电压

运放失调电压可以使用 CONFIG 对应的失调电压（出厂时已调至最佳值），也可以由调节位进行再次调节，选择方式由写入 *运放调节位选择寄存器 OPnADJE OPn* 的值决定：

- 写入 AAH：由失调电压调节位决定
- 写入其他值：由 CONFIG 相关位决定

失调电压调节位为*运放控制寄存器 OPnCON1 OPnADJ<4:0>*，其值范围为 0~31。考虑到用户在极少数情况和特定条件下进行修调时，有可能存在难以达到用户期望的修调值情况，建议用户在使用过程中，直接使用内部 CONFIG 对应的失调电压即可（出厂时已调至最佳值）。

19. LVD

中微 8051MCU 自带电源检测功能。若设置 LVD 模块使能（LVDEN=1），同时设置好电压监测点 LVDSEL，当电源电压下降到低于 LVD 设定值时，将会产生中断，提醒用户。

如果休眠前 LVD 模块使能，进入休眠后硬件不会关闭该模块电路，需软件关闭（LVDEN=0）。

在开启 LVD 模块相关功能时，需要注意配置顺序：

- 在开启 LVD 模块中断功能时，需遵循以下配置顺序：
  1)配置 LVD 电压检测点 LVDSEL<2:0>；
  2)打开 LVD 模块使能（LVDEN=1）；
  3)清除 LVD 中断标志位（LVDINTF=0）；
  4)打开 LVD 中断使能（LVDINTE=1）。
- 在使用 LVD 模块中断标志位轮询时，需遵循以下配置顺序：
  1)配置 LVD 电压检测点 LVDSEL<2:0>；
  2)打开 LVD 模块使能（LVDEN=1）；
  3)清除 LVD 中断标志位（LVDINTF=0）；
  4)使用轮询方式查询 LVD 中断标志位 LVDINTF。

适用产品：本应用注意事项适用于以下产品型号：CMS8S589x。

20. LSE

20.1 注意事项

若设置看门狗定时器功能为复位系统，那么设置 LSE 模块使能后，在等待 LSE 时钟稳定的过程中，需要注意及时清除看门狗计数器，否则将产生如下图所示的非期望的看门狗复位：虽然看门狗的溢出时间可配置，但是大部分看门狗溢出时间为 ms 量级，小于 LSE 时钟的稳定时间（约 1.5s）；如果进行了等待超时控制，那么超时控制的时间也是≥1.5s；在正常等待 LSE 时钟稳定的过程中，ms 量级的看门狗溢出将产生非期望的看门狗复位。

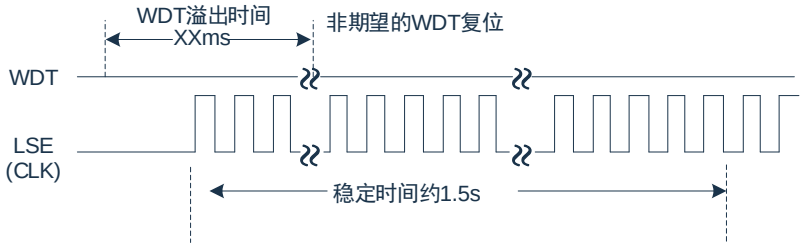
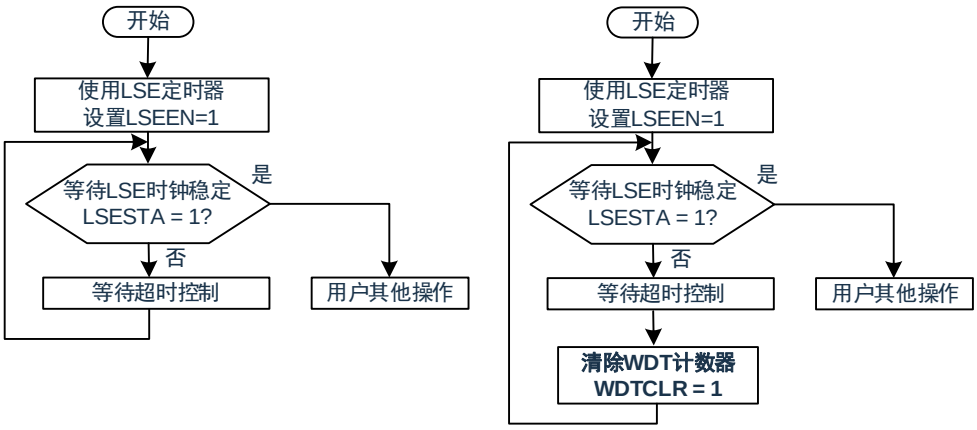


图 20-1：待 LSE 稳定过程中的 WDT 复位

20.2 推荐操作

如图 20-2 b 所示为推荐的操作流程，在等待 LSE 时钟的稳定的过程中及时清除看门狗计数器。



a 常规操作流程 **b 推荐操作流程**

图 20-2：等待 LSE 稳定过程中的 WDT 操作

推荐的程序范例如下：在 LSE 时钟的稳定的过程中及时清除看门狗计数器。

常规范例

```
void LSE_Config(void)
{
    /*(1)开启 LSE*/
    LSE_EnableLSE();
    /*(2)等待 LSE 稳定(约 1.5s)*/
    While( !LSE_GetLSEState())
    {
        ……//等待超时控制
    }
    ……//其他操作
}
```

推荐范例

```
void LSE_Config(void)
{
    /*(1)开启 LSE*/
    LSE_EnableLSE();
    /*(2)等待 LSE 稳定(约 1.5s)*/
    While( !LSE_GetLSEState())
    {
        ……//等待超时控制
        WDT_ClearWDT();//清除看门狗
    }
    ……//其他操作
}
```

21. 其他注意事项

- 对于需要连续设置几个语句来完成一个操作的程序，请不要在连续语句中间加延时或断点；如果中断在中途发生，会中断连续设置的逻辑，从而影响设置的成功率。
- 对程序寄存器的位操作，建议先将寄存器的值读出来，放在一个临时变量里，然后再根据对应的位，来修改临时变量的值；最后将临时变量的值写入到寄存器里。

22. 版本修订说明

版本号	时间	修改内容
V1.00	2020 年 12 月	初始版本
V1.01	2021 年 2 月	添加使用外部振荡器时需要注意的事项
V1.02	2021 年 4 月	添加了操作 FLASH 的时间间隔图示
V1.03	2021 年 11 月	添加了 FLASH 操作注意章节和增加了 ADC 的一些注意事项
V1.04	2021 年 12 月	增加了 TA 写时序的详细描写
V1.05	2022 年 6 月	增加 LSE 唤醒和 IO 口（串口，I2C）上拉、开漏说明以及 FLASH 操作说明
V1.06	2022 年 9 月	1)新增 18 运放失调电压章节 2)章节 4 ADC：新增注意事项 6) 3)新增 19 LVD 章节 4)新增 20 LSE 章节