



十速

52 系列 MCU

VCC=5V 电压下 IDLE 模式低功耗应用注意事项

Application Note

Rev 1.0

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.



AMENDMENT HISTORY

Version	Date	Description
V1.0	Apr, 2016	New Release.



CONTENTS

AMENDMENT HISTORY	2
PRODUCT NAME	4
TM52 系列 MCU: VCC=5V 电压下, IDLE 模式低功耗应用注意事项	4

PRODUCT NAME

TM52 系列 MCU

TM52 系列 MCU: VCC=5V 电压下, IDLE 模式低功耗应用注意事项

MCU 在 Vcc=5V 电压下运行程序时, 必须打开 LDO 以防止因 ROM 电压过高而导致读取程序出错。IDLE 模式下, MCU 内部 LDO 无法自动关闭, 在此情况下, 因 LDO 模块耗电较大, IC 无法实现低功耗要求。因此, Vcc=5V 电压下, MCU 要实现 IDLE 模式下的低功耗, 首先必须在进入 IDLE 模式前将 LDO 关闭, 然后在 IDLE 模式唤醒后必须第一时间把 LDO 打开, 以保证 MCU 正确读取程序。

注:

1. 5V 电源电压下, IDLE 模式下如无需低功耗要求, 则程序中请不要将 LDO 关闭。
2. 5V 电源电压下, 若 IC 需要进入 STOP 低功耗模式, 则只要 MODE3V=0, PWRSV=1 即可达到低功耗目的, 程序无需按本 AP 方式进行处理。

IDLE 模式下实现低功耗要求的程序处理步骤如下:

1. 切换慢钟为系统主时钟 --> 停止快时钟-->停止外设时钟。
2. 选择 MODE3V 模式使 LDO 关闭 --> 进入 IDLE 模式, 等待唤醒。
3. 在所有能唤醒的代码中, 必须第一时间打开 LDO。如 Timer3 中断、外部中断、Port1 中断等, 中断子程序里面应先打开 LDO, 再运行其他程序。

处理例程如下:

1. 进入 IDLE 模式前需添加的内容

```
void main()
{
    SYSTEM_INIT(); /* 初始化 */
    while(1)
    {
        //主循环
        /*****/
        用户代码
        /*****/
        /* 需添加的内容 */
        SELFCK = 0;
        STPFCK = 1;
        STPPCK = 1;
        OPTION = bx1xx_xxxx; /* OPTION = x1xx_xxxx, MODE3V 使能 */
        PCON = 0x01; /* 芯片进入 IDLE 模式 */
        _nop_(); /* 加一个 nop 等待电压稳定, 不可加多 */
        OPTION = bx0xx_xxxx; /* bit6: MODE3V = 0, 切换到 5V 工作模式 */
        /* 注意, 切 5V 的指令不要用“与或”等多运 */
        /* 算指令, 应直接赋值 */
        /* 下面的五条_nop_() 指令不可省 */
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        STPFCK = 0;
        STPPCK = 0;
        SELFCK = 1;
        _nop_();
        _nop_();
        /* 需添加的内容 */
        /*****/
        用户代码
        /*****/
    }
}
```

2. 唤醒后各唤醒程序中需添加的内容

2.1 定时中断唤醒的处理方法

```
void TIMER3_int (void) interrupt 7
{
    /* 中断唤醒后添加的代码 */

    _nop_();                // 加一个 nop 等待电压稳定, 不可加多
    TION = bx0xx_xxxx;     // bit6: MODE3V = 0, 切换到 5V 工作模式
                           // 注意, 切 5V 的指令不要用“与或”等多运算指令,
                           // , 应直接赋值

    /* 下面的五条_nop_() 指令不可省 */
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    STPFCK = 0;
    STPPCK = 0;
    SELFCK = 1;
    _nop_();
    _nop_();
    /* 中断唤醒后添加的代码 */
}
```

2.2 外部中断唤醒的处理方法

```
void P1_isr() interrupt 8
{
    /* 中断唤醒后添加的代码 */
    _nop_();                // 加一个 nop 等待电压稳定, 不可加多
    OPTION = bx0xx_xxxx;   // bit6: MODE3V = 0, 切换到 5V 工作模式
                           // 注意, 切 5V 的指令不要用“与或”等多运算指令,
                           // , 应直接赋值

    /* 下面的五条_nop_() 指令不可省 */
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    STPFCK = 0;
    STPPCK = 0;
    SELFCK = 1;
    _nop_();
    _nop_();
    /* 中断唤醒后添加的代码 */
}
```