

# **FT61F02X**

## **SLEEP Application note**

## 目录

1. SLEEP 睡眠模式(POWER-DOWN) .....	3
1.1. 进入 SLEEP .....	3
1.2. 从 SLEEP 中唤醒 .....	4
2. 应用范例 .....	5
联系信息 .....	8

## FT61F02x SLEEP 应用

### 1. SLEEP 睡眠模式(POWER-DOWN)

睡眠模式下，指令时钟关闭，指令执行停止，大多数模块掉电以降低功耗。如表 1-1 所示，FT61F02x 可根据实际需求在睡眠时有选择地开启各个模块，而无须指令介入，以使其相应功能如 LVR、LVD、WDT、Timers、比较器、PWM 和 ADC 能在 SLEEP 模式下保持运行。一些模块也可配置成进入 SLEEP 后自动关闭，而无须由指令关闭。

模块	SLEEP 模式下的各模块配置条件	
	运行	自动关闭?
指令时钟	(始终关闭)	Yes
LVR	LVREN = 00 or (LVREN = 01 & SLVREN=1)	LVREN = 10
LVD	LV DEN = 1	No
WDT	WDTE or SWDTEN	No
TIMER0	(始终关闭)	Yes
TIMER1	T1SYNC = 1 & TMR1CS = 1 & T1OSCEN = 1 & TMR1ON = 1	TMR1ON = 0
TIMER2	(始终关闭)	Yes
PWM3/4/5	PxCKSRC = 010 / 011 & TMRxON = 1	No
增强型 ECCP	(跟随 TIMER2)	
HIRC / LIRC / EC / LP / XT	(跟随使用它们的外设状态)	
ADC	(当 ADON = 1 且 ADC 所选时钟源保持运行时，ADC 即可运行)	
比较器	(CM ≠ 000 或 111 时，比较器保持运行)	
稳压器	(VREG_OE = 1 时，稳压器保持运行)	
I/O	(除非 SLEEP 时使能 PWM，否则 I/O 将保持其进入 SLEEP 前的状态)	

表 1-1 除指令时钟外，其他模块可根据需求在 SLEEP 模式下保持运行

#### 1.1. 进入 SLEEP

CPU 通过执行 SLEEP 指令进入睡眠模式。进入睡眠时：

- 如果 WDT 使能，则 WDT 的后分频器(如果分配给 WDT)和定时器将被清零，并重新开始计时。
- 超时标志位 (/TF) = 1。
- 掉电标志位 (/PF) = 0。
- 时钟源
  - 指令时钟自动关闭；
  - 如果 Timer 在 SLEEP 下保持运行，则其所选用的时钟源 HIRC, LIRC 或外部振荡器(EC, LP, XT)也将保持运行。如果某个 Timer 在睡眠中自动关闭，则其时钟源也将自动关闭，除非此时钟源同时被另一个保持运行的 Timer 所使用。
  - 指令时钟自动停止，因此即便配置了输出内部指令时钟，进入睡眠后其输出也将停止。
- I/O 端口

- 如果 Timer2 在 SLEEP 下保持运行，则 PWM 输出也将保持。如果 Timer2 自动关闭，那么 PWM 的输出将保持其进入 SLEEP 前的状态。
- 其他数字输出端口，将保持其进入 SLEEP 之前的状态(高阻态，低电平或高电平)。
- 当“ADON = 1”且 ADC 所选时钟源保持运行时，ADC 即可运行。如果其时钟源自动关闭，则 ADC 也将自动关闭。

有关各个外设 in SLEEP 状态下如何工作的详细信息，请查阅相应章节。

## 1.2. 从 SLEEP 中唤醒

从睡眠中唤醒有 2 个基本原则：

- 基于时间，即 CPU 在一定的时间后醒来。建议选择 LIRC 作为计时的时钟源，因为 LIRC 比 HIRC 的功耗更低。
  - 基于事件，即触发 POR，系统复位，仅唤醒而不产生中断，以及产生中断的事件，如 LVD，ADC，端口变化中断，PC1 边沿中断。
1. 如果使能，看门狗定时器可触发唤醒 (参阅 [章节 错误!未找到引用源。](#) 看门狗定时器)。
  2. 完全复位和系统复位：
    - POR 完全复位 (不能关闭)
    - 通过/MCLR 进行外部系统复位 (如果使能)
    - LVR 复位 (如果使能)
  3. 使能中断 (关闭“全局中断使能”不影响唤醒功能)。请参阅 [章节 错误!未找到引用源。](#) 中断。

注：

1. 从睡眠中唤醒将清零 WDT。
2. 紧跟 SLEEP 指令后必须写为 NOP 指令

使用非中断方式 (即未执行“中断服务程序) 从 SLEEP 中唤醒时，比如 WDT 唤醒或全局中断控制位 (GIE)未使能时的中断事件唤醒，下一条指令将被执行两次。为了避免重复执行，紧跟 SLEEP 的后面必须为 NOP 指令。

*SLEEP*

*NOP           // 非中断方式唤醒时，NOP 将执行两遍。*

## 2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_SLEEP.c
* 功能:    FT61F02x-SLEEP 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/4T
* 说明:    此程序为 FT61F02x-SLEEP 睡眠演示程序, 上电之后 LED1 与 LED2
*          同时亮一下熄灭, 进入睡眠, 测试 FT61F02x 的睡眠功耗。
*
*          FT61F023  SOP16
*          -----
* VDD-----|1(VDD)  (VSS)16|-----GND
* NC-----|2(PA7)   (PA0)15|-----led2
* NC-----|3(PA6)   (PA1)14|-----led1
* NC-----|4(PA5)   (PA2)13|-----NC
* NC-----|5(PC3)   (PA3)12|-----NC
* NC-----|6(PC2)   (PC0)11|-----NC
* NC-----|7(PA4)   (PC1)10|-----NC
* NC-----|8(PC5)   (PC4)09|-----NC
*
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define led1    RA1
#define led2    RA0
/*-----
* 函数名: POWER_INITIAL
* 功能:    上电系统初始化
* 输入:    无
* 输出:    无
*-----*/
void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;          //IRCF=111=16MHz/4T=4MHz,0.25µs
    INTCON = 0;                   //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B00000000;           //PA 输入输出 0-输出 1-输入
    PORTC = 0B00000000;
    TRISC = 0B00000000;           //PC 输入输出 0-输出 1-输入
    WPUA = 0B00000000;            //PA 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00000000;            //PC 端口上拉控制 1-开上拉 0-关上拉
}
```

```

ANSEL = 0B00000000;
OPTION = 0B00001000;          //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
MSCKCON = 0B00000000;
//Bit6->0,禁止 PA4, PC5 稳压输出
//Bit5->0,TIMER2 时钟为 Fosc
//Bit4->0,禁止 LVR
CMCON0 = 0B00000111;          //关闭比较器, CxIN 为数字 IO 口

```

```

}
/*-----
* 函数名: DelayUs
* 功能:   短延时函数 --16M-2T--大概快 1%左右.
* 输入:   Time 延时时间长度 延时时长 Time μs
* 输出:   无

```

```

-----*/

```

```

void DelayUs(unsigned char Time)

```

```

{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}

```

```

/*-----
* 函数名: DelayMs
* 功能:   短延时函数, 快 1%
* 输入:   Time 延时时间长度 延时时长 Time ms
* 输出:   无

```

```

-----*/

```

```

void DelayMs(unsigned char Time)

```

```

{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(98);
        }
    }
}

```

```

/*-----

```

\* 函数名: DelayS  
\* 功能: 短延时函数  
\* 输入: Time 延时时间长度 延时时长 TimeS  
\* 输出: 无

-----\*/

void DelayS(unsigned char Time)

```
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<10;b++)
        {
            DelayMs(100);
        }
    }
}
```

/\*-----

\* 函数名: main  
\* 功能: 主函数  
\* 输入: 无  
\* 输出: 无

-----\*/

void main()

```
{
    POWER_INITIAL();           //系统初始化
    led1 = 1;
    led2 = 1;
    DelayS(4);
    led1 = 0;
    led2 = 0;
    while(1)
    {
        SLEEP();
    }
}
```

## 联系信息

### **Fremont Micro Devices Corporation**

#5-8, 10/F, Changhong Building  
Ke-Ji Nan 12 Road, Nanshan District,  
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

### **Fremont Micro Devices (HK) Limited**

#16, 16/F, Block B, Veristrong Industrial Centre,  
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

\* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.