

FT64F0AX

SLEEP Application note

目录

1. 睡眠模式	3
1.1. 进入 SLEEP	3
1.2. 睡眠的唤醒	4
1.3. 睡眠的系统时钟	5
2. 应用范例	6
联系信息	9

FT64F0Ax SLEEP 应用

1. 睡眠模式

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, FT64F0Ax 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、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
TIMER1	SYSON = 1 & TIM1EN = 1	SYSON = 0
TIMER2	SYSON = 1 & TIM2EN = 1	SYSON = 0
TIMER4	SYSON = 1 & TIM4EN = 1	SYSON = 0
PWM	(跟随 Timer1 或 Timer2)	
HIRC / LIRC / EC / LP / XT	(跟随使用它们的外设状态)	
ADC	(当 ADCEN = 1 & ADON = 1 且 ADC 所选时钟源保持运行时, ADC 即可运行)	
SPI	(当 SPICKEN = 1 且 SPI 所选时钟源保持运行时, SPI 即可运行)	
I2C	(当 I2CEN = 1 且 I2C 所选时钟源保持运行时, I2C 即可运行)	
USART1/ USART 2	(当 USARTxEN = 1 且 USARTx 所选时钟源保持运行时, USARTx 即可运行)	
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

器件通过执行 SLEEP 指令进入睡眠模式。

进入睡眠模式时, MCU 的状态如下:

1. WDT 将清零但是保持运行 (如果使能了在休眠期间工作)
2. STATUS 寄存器的 PD 位清零
3. STATUS 寄存器的 TO 位置 1
4. CPU 时钟停止
5. 32kHz LIRC 不受影响, 并且由其提供时钟的外设可以在休眠模式下继续工作
6. LP 晶体振荡器不受影响 (当 TIMx 使用它作为工作时钟时)

7. ADC 不受影响 (如果选择了专用 FRC 时钟)
8. 电容触摸传感振荡器不受影响
9. I/O 端口保持执行 SLEEP 指令之前的状态 (驱动为高电平、低电平或高阻态)
10. WDT 之外的复位不受休眠模式影响

关于外设休眠期间工作的更多详细信息, 请参见各个章节。

要最大程度地降低电流消耗, 应考虑以下条件:

1. I/O 引脚不应悬空, I/O 作为输入时可打开内部的上拉或下拉
2. 外部电路从 I/O 引脚灌电流
3. 内部电路从 I/O 引脚拉电流
4. 内部弱上拉的引脚
5. 模块使用 32kHz LIRC
6. 模块使用 LP 振荡器

1.2. 睡眠的唤醒

可以通过下列任一事件将器件从休眠状态唤醒:

1. MCLR 引脚上的外部复位输入 (如果使能)
2. BOR 复位 (如果使能)
3. POR 复位
4. 看门狗定时器 (如果使能)溢出
5. 任何外部中断
6. 能够在休眠期间运行的外设产生的中断 (更多信息请参见各个外设)

前 3 个事件会使器件复位, 后 3 个事件认为是程序执行的延续。

当执行 SLEEP 指令时, 下一条指令 (PC+1) 被预先取出。如果希望通过中断事件唤醒器件, 则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关, 如果 GIE 位被禁止, 器件将继续执行 SLEEP 指令后的指令。如果 GIE 位被允许, 器件先执行 SLEEP 指令后的指令, 然后将调用中断服务程序。如果不想执行 SLEEP 指令后的指令, 用户应该在 SLEEP 指令后面放置一条 NOP 指令。

器件从休眠模式唤醒时, WDT 清零, 与唤醒的原因无关。

1.2.1. 使用中断唤醒

当禁止全局中断 (GIE 被清零), 并且有任一中断源中断标志位置 1 且其中断被使能时, 将会发生下列某一事件:

- 如果在执行 SLEEP 指令之前发生中断

- ✓ SLEEP 指令将作为 NOP 执行
- ✓ WDT 和 WDT 预分频器不会清零
- ✓ STATUS 寄存器的 TO 位不会置 1
- ✓ STATUS 寄存器的 PD 位不会清零
- 如果在执行 SLEEP 指令期间或之后发生中断
 - ✓ SLEEP 指令将完全执行
 - ✓ 器件将立即从休眠模式唤醒
 - ✓ WDT 和 WDT 预分频器将清零
 - ✓ STATUS 寄存器的 TO 位将置 1
 - ✓ STATUS 寄存器的 PD 位将清零

要确定是否执行了 SLEEP 指令, 可以测试 PD 位。如果 PD 位置 1, 则说明 SLEEP 指令被作为一条 NOP 指令执行了。

1.3. 睡眠的系统时钟

进入睡眠状态后, CPU 时钟停止, PC 停留在 SLEEP 的下一条地址。默认情况下, 系统时钟也会被关闭。但如果 SYSON 位置 1 时, 系统时钟将一直保持运行, 在这种情况下, 被选择为系统时钟的 HIRC, XT, LIRC 振荡器将不会被关闭。

注意: 如果要使用 PROM 或数据 EEPROM 的写完成中断唤醒, SYSON 必须置 1。

2. 应用范例

```
//*****
/* 文件名: TEST_64F0Ax_SLEEP.c
* 功能:    FT64F0Ax_SLEEP 功能演示
* IC:      FT64F0A5  TSSOP20
* 内部:    16M/2T
* 说明:    程序在 PB3 输出 50Hz 方波, 2 秒后进入休眠, 无波形输出
*
*          FT64F0A5  TSSOP20
*          -----
* NC-----|1(PA5)      (PA4)20|-----NC
* NC-----|2(PA6)      (PA3)19|-----NC
* NC-----|3(PA7)      (PA2)18|-----NC
* NC-----|4(PC0)      (PA1)17|-----NC
* NC-----|5(PC1)      (PA0)16|-----NC
* NC-----|6(PB7)      (PB0)15|-----NC
* GND-----|7(GND)     (PB1)14|-----NC
* NC-----|8(PB6)      (PB2)13|-----NC
* VDD-----|9(VDD)     (PB3)12|--DemoPortOut
* NC-----|10(PB5)     (PB4)11|-----NC
*          -----
*/
//*****
#include "SYSCFG.h";
#include "FT64F0AX.h";
//***** 宏定义 *****
#define uchar      unsigned char

#define DemoPortOut  PB3

uchar  outputcounter=0;
/*-----
* 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;    //系统时钟选择为内部振荡器 16MHz,分频比为 1:1

    INTCON=0;             //禁止所有中断

    PORTA=0B00000000;
```

```
PORTB=0B00000000;
PORTC=0B00000000;
```

```
WPUA=0B00000000;      //弱上拉的开关, 0-关, 1-开
WPUB=0B00000000;
WPUC=0B00000000;
```

```
WPDA=0B00000000;      //弱下拉的开关, 0-关, 1-开
WPDB=0B00000000;
WPDC=0B00000000;
```

```
TRISA=0B00000000;
TRISB=0B00000000;      //PB 输入输出设置, 0-输出, 1-输入 PB3-输出
TRISC=0B00000000;
```

```
PSRC0=0B11111111;      //源电流设置最大
PSRC1=0B11111111;
PSRC2=0B00001111;
```

```
PSINK0=0B11111111;      //灌电流设置最大
PSINK1=0B11111111;
PSINK2=0B00000011;
```

```
ANSELA=0B00000000;      //设置对应的 IO 为数字 IO
```

```
}
```

```
/*-----
```

```
* 函数名: DelayUs
```

```
* 功能: 短延时函数
```

```
* 输入: Time 延时时间长度 延时时长 Time μs
```

```
* 输出: 无
```

```
-----*/
```

```
void DelayUs(uchar Time)
```

```
{
```

```
    uchar a;
```

```
    for(a=0;a<Time;a++)
```

```
    {
```

```
        NOP();
```

```
    }
```

```
}
```

```
/*-----
```

```
* 函数名: DelayMs
```

```
* 功能: 短延时函数
```

```
* 输入: Time 延时时间长度 延时时长 Time ms
```

```
* 输出: 无
```

```
-----*/
void DelayMs(uchar Time)
{
    uchar a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);
        }
    }
}
/*-----
 * 函数名: main
 * 功能:   主函数
 * 输入:   无
 * 输出:   无
-----*/
void main(void)
{
    POWER_INITIAL();           //系统初始化

    while(1)
    {
        DemoPortOut=1;
        DelayMs(10);
        DemoPortOut=0;
        DelayMs(10);

        if(++outputcounter>100) //2 秒进入休眠
        {
            outputcounter=0;
            SLEEP();
            NOP();
            NOP();
        }
    }
}
```


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.