FT60F01X

Application note

目录

<center>**FT60F01x SLEEP 相关寄存器的设置**</center>

1 Sleep 睡眠模式功能及注意事项

产品在待机状态时候，要想达到最低功耗，可以选择睡眠模式，既可以随时唤醒，也可以减少损耗。

为了达到最低睡眠功耗<3 ua)，注意如下：

1、当 IO 口悬空时候，软件可以将所悬空的 IO 口 设为输出并置低，若没有悬空，则以外围电路状态为主，避免 IO 口 耗电；

2、I/O 作为输入的，外部电路应将其拉高或拉低，避免翻转耗电；

3、/MCLR 应该在高电平；

4、选择 LVREN 禁止使能。

2 应用范例

```
//************************************************************
/*   文件名：TEST_60F01x_SLEEP.c
*    功能：    FT60F01x-SLEEP 功能演示
*    IC:      FT60F011A SOP8
*    晶振：    16M/4T
*    说明：    此程序为 FT60F01x-SLEEP 睡眠演示程序，上电之后 PA0、PA1 同时置高约 4s，
*            然后置低，进入睡眠。测试 FT60F01x 的睡眠功耗：
*
*              FT60F011A   SOP8
*            ----------------
*   VDD------- --|1(VDD) (GND)8|------------GND
*   NC------------|2(PA2)    (PA4)7|------------NC
*   led1-------- --|3(PA1)    (PA5)6|------------NC
*   NC------------|4(PA3)    (PA0)5|-----------led2
*            ----------------
*/
//************************************************************
#include "SYSCFG.h"
//************************************************************
#define OSC_16M   0X70
#define OSC_8M     0X60
#define OSC_4M     0X50
#define OSC_2M     0X40
#define OSC_1M     0X30
#define OSC_500K 0X20
#define OSC_250K 0X10
#define OSC_32K   0X00

#define WDT_256K 0X80
#define WDT_32K   0X00
```

```
//****************************************************
//*******************宏定义***********************
#define    unchar        unsigned char
#define    unint         unsigned int
#define    unlong        unsigned long
#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 = WDT_32K|OSC_16M|0X01;    //INROSC
    //OSCCON = 0B01110001;                //WDT 32KHZ IRCF=111=16MHZ/4=4MHZ,0.25US/T
                                        //Bit0=1,系统时钟为内部振荡器(60F01x 保留位)
                                        //Bit0=0,时钟源由 FOSC<2:0>决定即编译选项时选择
    INTCON = 0;                         //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00000000;                 //PA 输入输出 0-输出 1-输入
    WPUA = 0B00100000;                  //PA 端口上拉控制 1-开上拉 0-关上拉
    //WPUA2 = 1;//                        //开 PA2 上拉

    OPTION = 0B00001000;                //Bit3=1 WDT MODE,PS=000=1:1 WDT RATE
                                        //Bit7(PAPU)=0 ENABLED PULL UP PA
    MSCKCON = 0B00000000;               //Bit6->0,禁止 PA4, PC5 稳压输出(60F01x 保留位)
                                        //Bit5->0,TIMER2 时钟为 Fosc(60F01x 保留位)
                                        //Bit4->0,禁止 LVR(60F01x O 版之前)
                                        //Bit4->0, LVREN 使能时,开启 LVR(60F01x O 版及
                                        //O 版后)
                                        //Bit4->1, LVREN 使能时,工作时开启 LVR, 睡眠时自
                                        //动关闭 LVR(60F01x O 版及 O 版之后)

}
/*------------------------------------------------
 *   函数名称：DelayUs
 *   功能：       短延时函数 --16M-2T--大概快 1%左右.
 *   输入参数：Time 延时时间长度   延时时长 Time Us
 *   返回参数：无
 ------------------------------------------------*/
void DelayUs(unsigned char Time)
{
```

```
        unsigned char a;
        for(a=0;a<Time;a++)
        {
            CLRWDT();
        }
}
/*----------------------------------------------
 *   函数名称：DelayMs
 *   功能：     短延时函数
 *   输入参数：Time 延时时间长度  延时时长 Time ms
 *   返回参数：无
 ----------------------------------------------*/
void DelayMs(unsigned char Time)
{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197);              //快 1%
        }
    }
}
/*----------------------------------------------
 *   函数名称：DelayS
 *   功能：     短延时函数
 *   输入参数：Time 延时时间长度 延时时长 Time S
 *   返回参数：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(3);
    led1 = 0;
    led2 = 0;
    while(1)
    {
        CLRWDT();           //清看门狗
        NOP();
        SLEEP();
        NOP();

    }
}
```

Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057
Tel: (86 755) 86117811
Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong
Tel: (852) 27811186
Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539
Tel: (1-510) 668-1321
Fax: (1-510) 226-9918

Web Site: http://www.fremontmicro.com/