

FT60F01X

MSCK Application note

目录

1. 慢时钟测量相关寄存器的设置	3
2. 操作步骤	6
3. 应用范例	7
联系信息	10

FT60F01x MSCK 应用

1. 慢时钟测量相关寄存器的设置

慢时钟测量可以比较精准的测量内部慢时钟周期。测量慢时钟时，TIMER2 的预分频、后分频自动变为 1:1，组成一个 12 位的定时器，其时钟源为系统时钟，启动测量后在 1 个或 4 个慢时钟周期进行计数。计数结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟 F_{osc} 的个数。

相关寄存器的各个位定义如下：

1) MSCKCON 寄存器

Bit	7	6	5	4	3	2	1	0
MSCKCON	-	-	-	SLVREN	-	CKMAVG	CKCNTI	-
Reset	-	-	-	0	0	0	0	-

Bit7~Bit5：保留位，读 0

Bit4：O 版之前：

1：编译选项里 LVREN 使能时

1：开启 LVR

0：关闭 LVR

2：编译选项里 LVREN 关闭时

SLVREN=X 均为关闭 LVR

O 版（包括 O 版）以后：

1：编译选项里 LVREN 使能时

1：工作时开启 LVR，睡眠时自动关闭 LVR

0：始终开启 LVR

2：编译选项里 LVREN 关闭时

SLVREN=X 均为关闭 LVR

Bit3：保留位

Bit2：快时钟测量慢时钟周期的测量平均模式

1：打开平均模式（自动测量 4 次，并取其平均值）

0：关闭平均模式

Bit1：使能快时钟测量慢时钟周期

1：使能快时钟测量慢时钟周期

0：关闭快时钟测量慢时钟周期

注：这一位在测量完毕后会自动归零

Bit0：保留位

2) T2CON 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	TOUTPS[3-0]				TMR2ON	T2CKPS[1-0]	
Reset	NA	0000				0	00	

Bit7：保留位

Bit6~Bit3：定时器 2 输出后分频比选择

TOUTPS<3:0>	后分频比	TOUTPS<3:0>	后分频比
0000	1:1	1000	1:9
0001	1:2	1001	1:10
0010	1:3	1010	1:11
0011	1:4	1011	1:12
0100	1:5	1100	1:13
0101	1:6	1101	1:14
0110	1:7	1110	1:15
0111	1:8	1111	1:16

Bit2: 打开定时器 2 位

1: 打开 Timer2

0: 关闭 Timer2

Bit1~Bit0: 定时器 2 驱动时钟预分频比选择

00 = 预分频比是 1: 1

01 = 预分频比是 1: 4

1x = 预分频比是 1: 16

3) INTCON 寄存器

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit7: 全局中断使能

1: 使能所有未屏蔽中断

0: 禁止所有中断

Bit6: 外设中断使能

1: 使能所有未屏蔽中断

0: 禁止所有外设中断

Bit5: 定时器 0 溢出中断使能

1: 使能定时器 0 中断

0: 禁止定时器 0 中断

Bit4: 外部中断使能

1: 使能 PA2/INT 管脚外部中断

0: 禁止 PA2/INT 管脚外部中断

Bit3: PORTA 端口变化中断

1: 使能 PORTA 端口变化中断

0: 禁止 PORTA 端口变化中断

Bit2: 定时器 0 溢出中断标志位

1: Timer0 寄存器溢出 (必须软件清零)

0: Timer0 寄存器未溢出

Bit1: PC1/INT 管脚外部中断标志位

1: PA2/INT 管脚外部中断已发生 (必须软件清零)

0: PA2/INT 管脚外部中断未发生

Bit0: PORTA 端口变化中断标志位

1: PORTA<5:0>至少有一个端口状态发生了改变（必须软件清零）

0: PORTA<5:0>没有一个端口发生状态改变

4) PE1E 寄存器

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMEAIE	-	-	-	-	TMR2IE	-
Reset	0	0	-	-	-	-	0	-

Bit7: EE 写中断使能位

1: 使能 EE 写操作完成中断

0: 关闭 EE 写操作完成中断

Bit6: 快时钟测量慢时钟操作完成中断使能位

1: 使能快时钟测量慢时钟操作完成中断

0: 关闭快时钟测量慢时钟操作完成中断

Bit5~Bit2: 保留位

Bit1: Timer2 与 PR2 比较相等中断使能位

1: 使能 timer2 的值等于 PR2 中断

0: 关闭使能 timer2 的值等于 PR2 中断

Bit0: 保留位

5) PIR1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMEAIF	-	-	-	-	TMR2IF	-
Reset	0	0	-	-	-	-	0	-

Bit7: EE 写中断标志位

1: EE 写操作完成（必须软件清零）

0: EE 写操作未完成

Bit6: 快时钟测量慢时钟操作完成中断标志位

1: 快时钟测量慢时钟操作完成（必须软件清零）

0: 快时钟测量慢时钟未完成

Bit5~Bit2: 保留位

Bit1: Timer2 与 PR2 比较相等中断标志位

1: timer2 的值等于 PR2（必须软件清零）

0: timer2 的值不等于 PR2

Bit0: 保留位

6) SOSCPRL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SOSCPRL[7:0]							
Reset	8'hff							
Type	RW							

Bit7~Bit0: 低速振荡器周期的低七位（单位：快时钟周期数）用于慢时钟测量功能

7) SOSCPRH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SOSCPR[11:8]			
Reset	-	-	-	-	4'hf			
Type	-	-	-	-	RW			

Bit7~Bit4: 保留位

Bit3~Bit0: 低速振荡器周期的高三位（单位：快时钟周期数）用于慢时钟测量功能

2. 操作步骤

1. 把 T2CON.2 置 1，使能 TIMER2；
2. 如果选择 4 次平均，则把 MSCKCON.2 置 1，否则把它清 0；
3. 置位 MSCKCON.1，开始测量；
4. 测量结束后 MSCKCON.1 自动清 0，中断标志置 1；
5. 可以用查询中断标志位 CKMEAIF 或中断的方式等待结果，此时，SOSCPR 的值即为测量结果。

注意：

1. 做慢时钟测量时，最好把系统时钟选择为 16M，这样分辨率更高，测量结果也更接近实际；
2. 当 MSCKCON.1 为 1 时，表示将测量 4 个慢时钟周期，硬件没有做移位操作。故软件应对其右移 2 位以得到 1 个慢时钟周期的结果。

3. 应用范例

```
//*****
/* 文件名: TEST_60F01x_MSCK.c
* 功能:    FT60F01x IO 快时钟测量慢时钟功能演示
* IC:      FT60F011A SOP8
* 晶振:    16M/4T
* 说明:    程序中读取快时钟测量慢时钟数据
*
*          FT60F011A  SOP8
*          -----
* VDD-----|1(VDD)  (GND)8|-----GND
* NC-----|2(PA2)  (PA4)7|-----NC
* NC-----|3(PA1)  (PA5)6|-----NC
* NC-----|4(PA3)  (PA0)5|-----NC
*
*          -----
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define uint      unsigned int

volatile uint     TestBuff;
/*-----
* 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 = 0B01110000;    //IRCF=111=16MHz/4T=4MHz, 0.25μs
    INTCON = 0;             //暂禁止所有中断
    OPTION = 0B00001000;    //Bit3=1, WDT MODE, PS=000=WDT RATE 1:1

    PORTA = 0B00000000;
    TRISA = 0B00000000;     //PA 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

    MSCKCON = 0B00000000;
    //Bit4=0,禁止 LVR(60F01x O 版之前)
    //Bit4=0,LVREN 使能时,开启 LVR(60F01x O 版及 O 版之后)
    //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 版后)
}
/*-----
```

```
* 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  $\mu$ s
* 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--16M-4T--大概快 1%左右.
* 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);          //快 1%
        }
    }
}
```

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

```
* 函数名: SlowTimeTest
* 功能: 快时钟测量慢时钟
* 输入: 无
* 输出: 慢时钟时钟测量值 TestTime
        不开平均模式慢时钟频率=16M/TestTime
        开平均模式慢时钟频率=16M/TestTime/4
```

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

```
uint SlowTimeTest()
```

```
{
    uint TestTime;
    TMR2ON = 1;          //开定时器 2
    CKMEAIF = 0;         //清标志位
    CKMAVG = 0;          //关闭平均模式
```



```

//打开平均模式输出数据为四个周期的时钟数(单周期*4)
CKCNT1 = 1;
while(!CKMEAIF);
CKMEAIF = 0;
TestTime = SOSCPRH << 8;
TestTime = TestTime + SOSCPRL;
return TestTime;
}
/*-----
 * 函数名: main
 * 功能: 主函数
 * 输入: 无
 * 输出: 无
-----*/
void main()
{
    POWER_INITIAL(); //系统初始化
    while(1)
    {
        TestBuff = SlowTimeTest(); //时钟测量值
                                   //32768 该数值≈488(不开平均模式-单周期)
        NOP();
        DelayMs(200); //延时 200ms
    }
}
```

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.