

FT64F0AX

MSCK Application note

目录

1. 慢时钟测量	3
1.1. 慢时钟测量相关寄存器汇总.....	3
1.2. 测量原理.....	4
1.3. 上电自动测量.....	4
1.4. 操作步骤.....	6
2. 应用范例.....	7
联系信息	11

FT64F0Ax MSCK 应用

1. 慢时钟测量

芯片集成了两个内部 RC 振荡器，一个是经过出厂校准的高速高精度的 16M 快时钟 HIRC，一个是低功耗的 32K 时钟 LIRC，利用慢时钟测量功能可以把 LIRC 的周期用系统时钟计算出来。此功能可以比较精准的测量内部慢时钟周期。

1.1. 慢时钟测量相关寄存器汇总

名称	状态	寄存器	地址	复位值
CKMAVG	<u>LIRC 和 HIRC 交叉校准时 4 次平均测量模式</u> 1 = 使能 0 = <u>关闭</u>	MSCKCON[1]	0x41D	RW-0
CKCNTI	<u>启动 LIRC 和 HIRC 的交叉校准功能</u> 1 = <u>启动</u> 0 = 完成 (测量完毕自动清零)	MSCKCON[0]		RW-1
SOSCPR	<u>校准 LIRC 周期所需的 HIRC 周期数</u>	SOSCPR[11:0]	0x41F[3:0] 0x41E[7:0]	RW-FFF

表 1-1 LIRC 和 HIRC 交叉校准控制/状态位

名称	状态	寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, CKMIE 适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断 1 = 使能 (CKMIE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
CKMIE	LIRC 和 HIRC 交叉校准完成中断 1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[1]	0x91	RW-0
CKMIF ¹	LIRC 和 HIRC 交叉校准完成标志位 1 = Yes (锁存) 0 = <u>No</u>	PIR1[1]	0x11	R_W1C-0

表 1-2 LIRC 和 HIRC 交叉校准中断使能/状态位

¹ 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

1.2. 测量原理

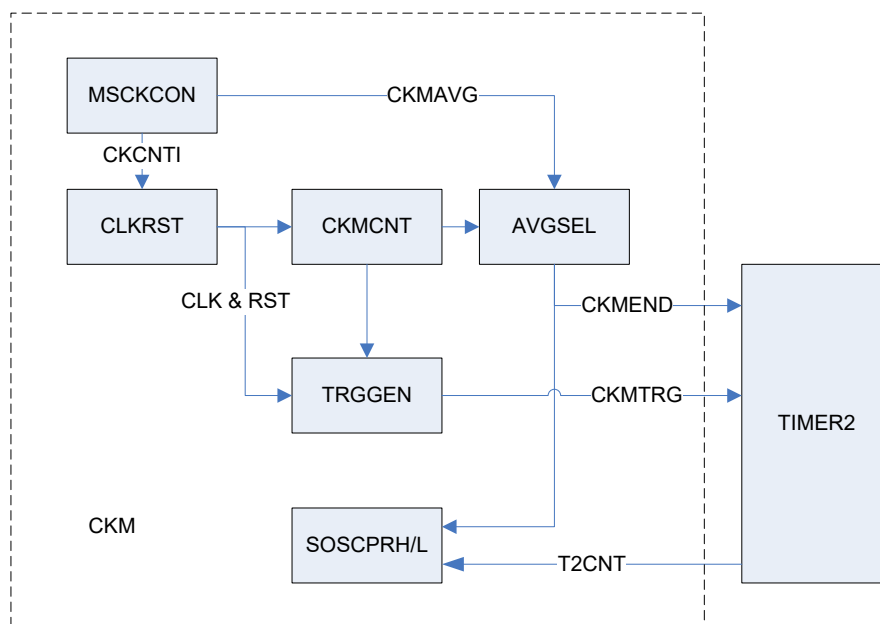


图 1-1 慢时钟测量模式原理框图

慢时钟测量类似于定时器的捕捉模式，处于这种模式下，被测量时钟 LIRC 的边沿（任意沿）将会触发定时器，在另一高速时钟（如 HIRC）的作用下开始计数，在此后的第 2 个（或第 8 个，平均模式时）LIRC 边沿到来时，定时器停止计数，同时把定时器的值锁存到 SOSCPRH/L 寄存器。

慢时钟测量使用的定时器是 TIM2。

注意：

1. 在慢时钟测量过程中软件不要写 SOSCPRH/L；
2. 不要在单步调试下做慢时钟测量，因为暂停模式下 TIM2 被停止，这样会导致测量结果不正确；
3. 若 SYSON = 0 时，慢时钟测量无法在 SLEEP 模式下进行，不要在测量运行时进入 SLEEP 模式。

1.3. 上电自动测量

在上电后慢时钟测量将会自动启动，此时 CKCNTI 置 1，CKMAVG 为 0，打开 LIRC 和 HIRC。TIM2 的时钟被自动配置为 16M 的内部高速时钟，即类似设置 T2CKSRC 为 001、TIM2EN=1 的功能，但未配置这些位。TIM2 使用默认配置，无需置位 CEN 使能 TIM2 计数，此时不能配置 TIM2。

在测量过程中，实际应用程序已在运行，若要使用 TIM2 则需要查询 CKCNTI。若 CKCNTI 为 0 即可使用 TIM2，此时 SOSCPR 寄存器的值为有效值，其单位为 F_{HSI} 时钟的个数。

注意：上电自动测量不会置位 CKM 中断标志。

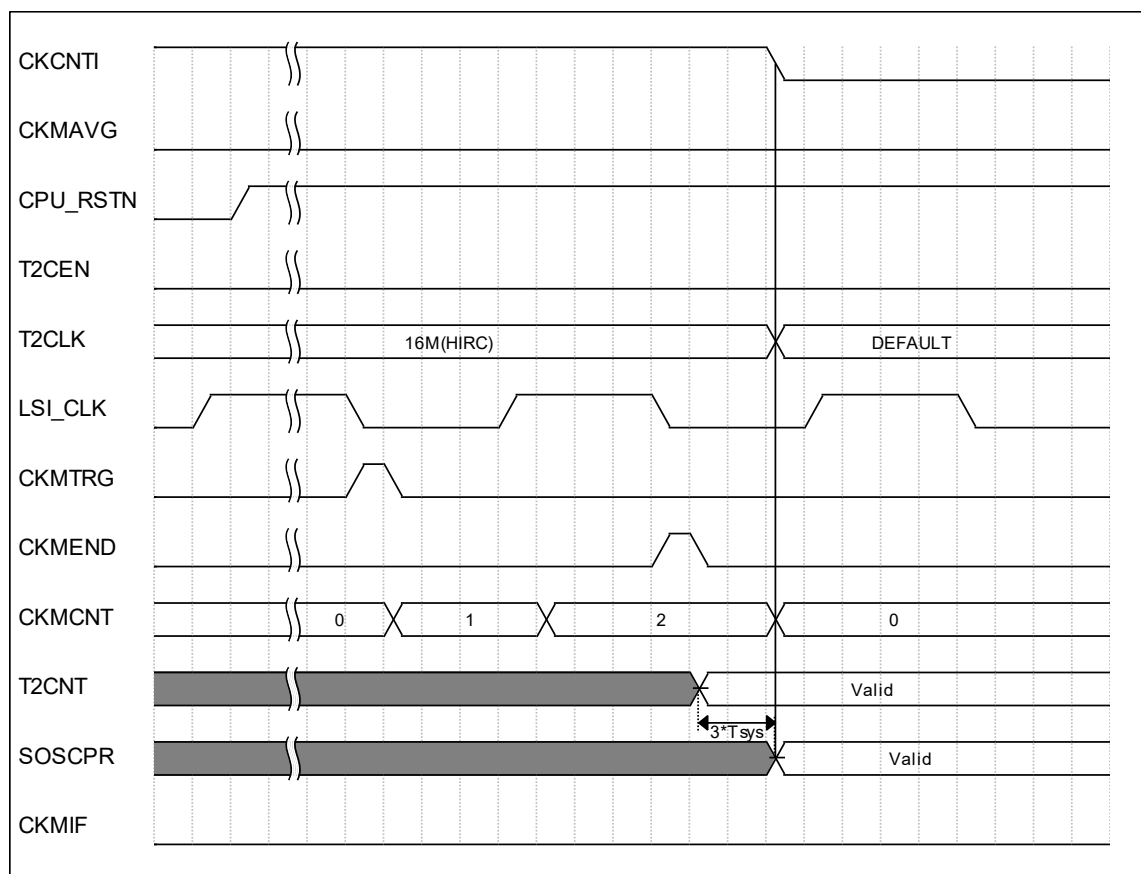


图 1-2 上电慢时钟自动测量时序图

1.4. 操作步骤

1. 为提高计量精度，建议设置 T2CKSRC = 001，TIM2EN=1，选择 16M 的内部高速时钟；
2. 关闭 TIM2 的相关中断使能，设置 TIM2ARRH/L 为最大值，设置 TIM2PSC = 0000；
3. 设置 TIM2CR1 为复位值，再将 CEN 置 1，使能 TIM2；
4. 如果选择 4 次平均，则把 MSCKCON.1 置 1，否则把它清 0；
5. 置位 MSCKCON.0，开始测量；
6. 测量结束后 MSCKCON.0 自动清 0，中断标志置 1；
7. 可以用查询或中断的方式等待结束；
8. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

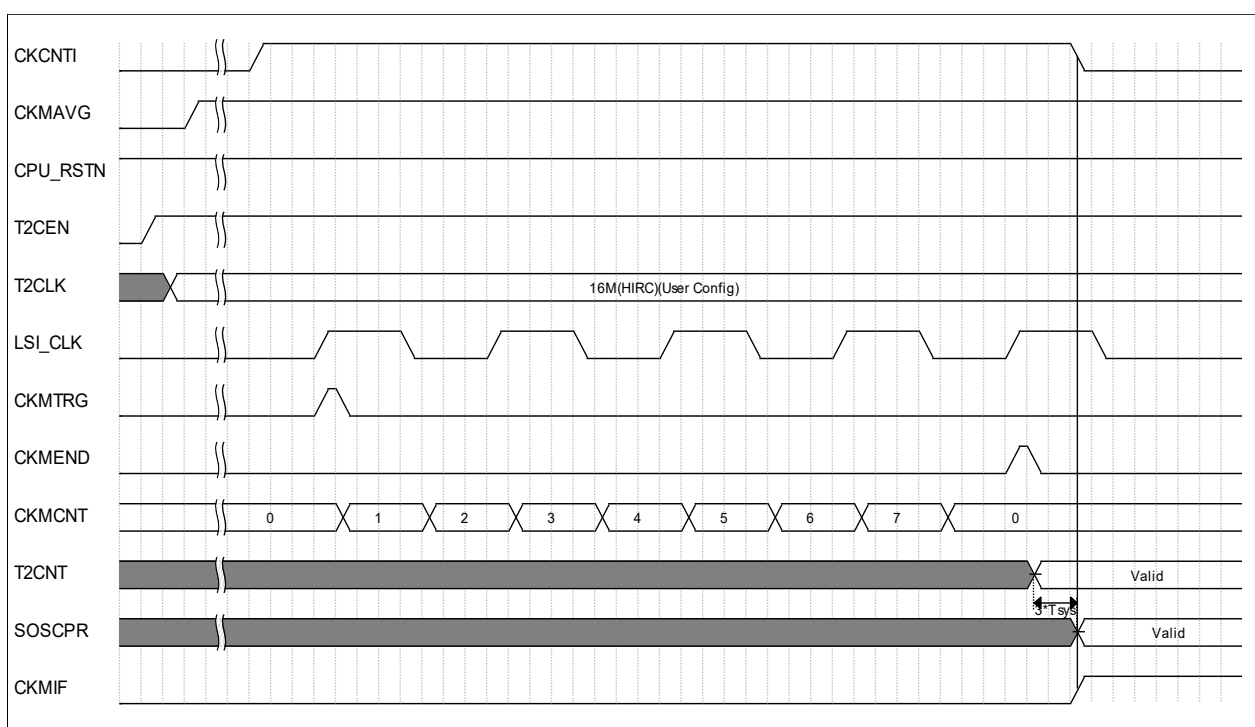


图 1-3 慢时钟测量模式时序图

2. 应用范例

```
//*****
/* 文件名: TEST_64F0Ax_MSCK.c
* 功能:   FT64F0Ax_MSCK 功能演示
* IC:    FT64F0A5  TSSOP20
* 内部:   16M/2T
* 说明:   测量慢时钟
*
*          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 uint       unsigned int

#define DemoPortOut PB3

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=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;           //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 Us
* 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: 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(2T)
          开平均模式慢时钟频率=16M/TestTime/4(2T)
-----*/
uint SlowTimeTest(void)
{
    PCKEN|=0B00000100;    //使能 TIM2 模块时钟
    TCKSRC=0B00010000;    //TIM2 时钟源 HIRC
    T2CEN=1;              //使能 Timer2 计数器
    CKMAVG=1;              //打开平均模式
    CKCNT1=1;              //使能快时钟测量慢时钟周期
    NOP();
    while(!CKMIF);         //等待慢时钟测量完成
    CKMIF=0;               //软件清 0 标志位
    return (uint)(SOSCPRH<<8|SOSCPRL);
}
/*-----
* 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)
    {

```

```
    TestBuff=SlowTimeTest(); //时钟测量值

    NOP();
    NOP();
    NOP();
    DelayMs(200);
  }
}
```

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.