

# **FT61F13X**

## **TIMER1 Application note**

## 目录

1. 定时器(TIMERS) .....	3
1.1. 定时器 1 (TIMER1).....	4
1.1.1. Timer1 相关寄存器汇总 .....	5
1.1.2. TMR1 寄存器的读/写操作 .....	6
2. 应用范例.....	7
联系信息 .....	10

## FT61F13x TIMER1 应用

### 1. 定时器(TIMERS)

共有 4 个定时器，包括看门狗定时器(WDT)在内。

	WDT	Timer0	Timer1	Timer2
预分频器 (位)	–	8 (与 WDT 共用)	4 (1x, 4x, 16x, Timer1 和 Timer2 共用)	
计数器 (位)	16	8	12	16
后分频器 (位)	7 (与 Timer0 共用)	–	–	4 (1 – 16x)
时钟源	<ul style="list-style-type: none"> <li>• LP</li> <li>• XT</li> <li>• HIRC</li> <li>• <u>LIRC</u></li> </ul>	<ul style="list-style-type: none"> <li>• LP</li> <li>• XT</li> <li>• HIRC</li> <li>• <u>指令时钟</u></li> <li>• PA2/T0CKI (转变沿计数器)</li> </ul>	<ul style="list-style-type: none"> <li>• LP</li> <li>• XT</li> <li>• HIRC</li> <li>• <u>指令时钟</u></li> </ul>	<ul style="list-style-type: none"> <li>• LP</li> <li>• XT</li> <li>• HIRC</li> <li>• <u>指令时钟</u></li> <li>• LIRC</li> <li>• 2x 指令时钟</li> <li>• 2x HIRC</li> <li>• 2x (EC, LP or XT)</li> </ul>

**表 1-1** 定时器资源

注：如果定时器的时钟源不是指令时钟，在更改 TMRx 之前需先设置“TMRxON = 0”。

当定时器使能时，其所选的时钟源会自动开启。指令时钟在 SLEEP 模式下被关闭，因此不能用于 WDT。当定时器选择 LP / XT 振荡器作为时钟源时，FOSC 必须相应配置成 LP / XT 模式或选择 INTOSCIO 模式，否则 LP / XT 振荡器将处于关闭状态，不会产生计数。

WDT 的后分频器(postscaler)和 Timer0 的预分频器(prescaler)共用同一个硬件分频电路。该硬件电路由指令选择分配给 WDT 或 Timer0，但二者不能同时使用。未被分配分频器的定时器，其分频比值为“1”。同样的规则也适用于共用同一个预分频器电路的 Timer1 和 Timer2。

在 POR 或系统复位时，除 Timer0 的计数器(counter)外，其他所有定时器的计数器、预分频器和后分频器都将复位。以下事件也将复位相应定时器的计数器和分频器：

	WDT	Timer0	Timer1	Timer2
预分频器	–	<ul style="list-style-type: none"> <li>• 写 TMR0</li> <li>• PSA 切换</li> </ul>	<ul style="list-style-type: none"> <li>• T1ON = 0 &amp; T1CKPSA = 1</li> <li>• LIRC 和 HIRC 交叉校准启动</li> <li>• T1ON = 0 &amp; TMR2ON = 0</li> <li>• 写 T2CON0, TMR1L/H, TMR2L/H</li> </ul>	<ul style="list-style-type: none"> <li>• TMR2ON = 0 &amp; T1CKPSA = 0</li> </ul>
计数器	<ul style="list-style-type: none"> <li>• WDT, OST 溢出</li> <li>• 进入/退出 SLEEP</li> <li>• CLRWDT</li> <li>• 写 WDTCON</li> </ul>	<ul style="list-style-type: none"> <li>• Timer0 溢出</li> </ul>	<ul style="list-style-type: none"> <li>• TMR1 = PR1 (匹配)</li> </ul>	<ul style="list-style-type: none"> <li>• TMR2 = PR2 (匹配)</li> </ul>
后分频器	<ul style="list-style-type: none"> <li>• 除写 WDTCON 外的以上所有条件</li> <li>• PSA 切换</li> </ul>	–		<ul style="list-style-type: none"> <li>• 除(T1ON = 0 &amp; TMR2ON = 0)外的以上所有条件</li> </ul>

**表 1-2** 定时器的计数器和分频器的重置事件

注意：对 TMR1L/H 进行写操作将会重置 Timer2 的预分频器。可通过以下步骤规避此问题：

1. 设置 “T1ON = 0”，从而停止 Timer1 计数器；
2. 读取 TMR1L 和 TMR1H 值；
3. 将读到的值分别写入 PR1L 和 PR1H，强制 “TMR1 = PR1” 产生匹配；
4. 将最初的预期值写回 PR1L 和 PR1H；
5. TMR1L 和 TMR1H 将自动重置；

一旦发生 PWM 故障刹车，且刹车事件一直保持，那么 Timer2 的计数器、预分频器和后分频器将停止递增，在故障刹车事件清除后将自动恢复。

PWM 单脉冲输出结束时将自动设置 “TMR2ON = 0”，通过设置 “TMR2ON = 1” 可重新启动 Timer2。

### 1.1. 定时器 1 (TIMER1)

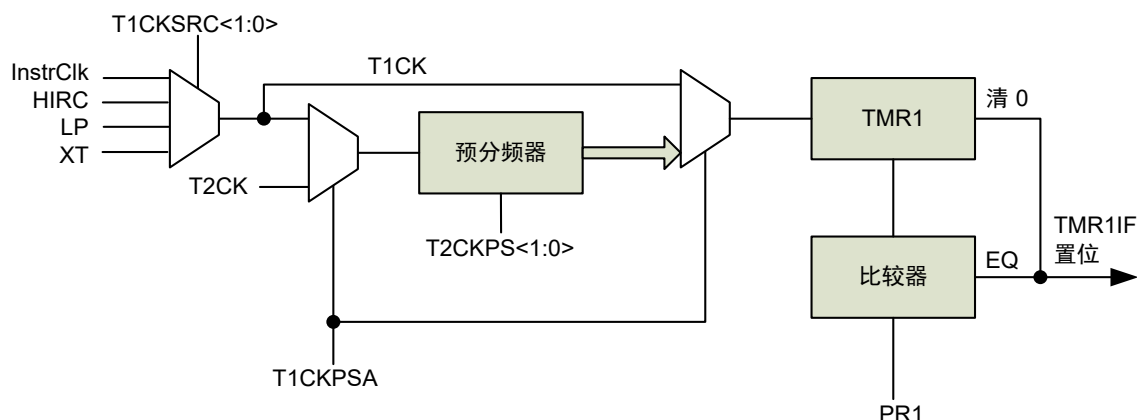


图 1-1 Timer1 结构框图

Timer1 用作计时功能。

共用的预分频器（由 T2CKPS 设置预分频比值 1, 4 或 16）可分配给 Timer1 或 Timer2 使用，如果未分配给 Timer1，其分频比将为“1”。

Timer1 将从 0x00 开始递增，直至递增到设置时间 =

$$\text{Timer1\_预分频} * \text{PR1} / (\text{Timer1 时钟频率})$$

此时，TMR1 与 PR1 匹配，并置位中断标志位(TMR1IF)，而 TMR1 在下一个递增周期复位为 0x00。匹配后是否触发中断和/或从睡眠中唤醒则取决于相应的使能/关闭控制位(GIE, PEIE 和 TMR1IE)。

如需从睡眠中唤醒，需设置 “T1CKRUN = 1” 和 “T1CKSRC ≠ 00”，以使 Timer1 的时钟源不是指令时钟且在 SLEEP 模式下保持运行，否则 Timer1 将停止计数，维持其进入睡眠前的计数值。

### 1.1.1. Timer1 相关寄存器汇总

名称	状态		寄存器	地址	复位值
T1CKPSA	预分频器分配位	1 = Timer1 (此时即使TMR2ON=0, Timer2仍启动计数) 0 = <u>Timer2</u>	T1CON0[4]	0x11A	RW-0
T1CKRUN	睡眠时T1CK保持工作	1 = Yes (时钟源非指令时钟) 0 = <u>No</u>	T1CON0[3]		RW-0
T1ON	Timer1	1 = 使能 0 = <u>关闭</u>	T1CON0[2]		RW-0
T1CKSRC	<u>Timer1 时钟源</u> 00 = <u>指令时钟</u> 10 = LP <sup>(*)</sup> 01 = HIRC                              11 = XT <sup>(*)</sup> <sup>(*)</sup> FOSC 应相应配置成 LP/XT 模式或选择 INTOSCIO 模式, 否则振荡器将不会运行。		T1CON0[1:0]		RW-00
T2CKPS (T1CKPS)	<u>Timer2/Timer1 预分频比</u> 00 = <u>1</u> 01 = 4                      1x = 16		T2CON0[1:0]	0x12	RW-00
PR1L	PR1 周期寄存器低 8 位		PR1L[7:0]	0x116	RW- 1111
PR1H	PR1 周期寄存器高 4 位		PR1H[3:0]	0x117	1111 1111
TMR1L	TMR1 计数结果寄存器低 8 位		TMR1L[7:0]	0x118	RW- 0000
TMR1H	TMR1 计数结果寄存器高 4 位		TMR1H[3:0]	0x119	0000 0000

表 1-3 Timer1 相关用户控制寄存器

名称	状态		寄存器	地址	复位值
GIE	全局中断	1 = 使能 (PEIE, TMR1IE 适用) 0 = <u>全局关闭</u> (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B	RW-0
PEIE	外设总中断	1 = 使能 (TMR1IE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]	0x18B	RW-0
TMR1IE	Timer1与PR1匹配中断	1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[3]	0x8C	RW-0
TMR1IF	Timer1与PR1匹配中断标志位	1 = 匹配 (锁存) 0 = <u>不匹配</u>	PIR1[3]	0x0C	RW-0

表 1-4 Timer1 中断使能和状态位

### 1.1.2. TMR1 寄存器的读/写操作

TMR1H 和 TMR1L 不能同时读或写。通过 TMR1H 的内部缓存 TMR1H\_buf 可解决此问题，必须遵循以下读写顺序：

- 读 TMR1 时，先读 TMR1L，此时 TMR1H 的值将被锁存到 TMR1H\_buf，然后读 TMR1H。当 Timer1 的时钟源不是指令时钟时，需设置 “TMR1ON = 0” 以停止计数，然后在读 TMR1 之前执行 1 条 NOP 指令。

写 TMR1 时，先写 TMR1H，此时 TMR1H 的值将被储存在 TMR1H\_buffer 中。然后写 TMR1L，此时 TMR1H 和 TMR1L 将同时更新到计数值中。另外，为了避免写入操作和计数之间的竞争，在写操作前，应设置 “TMR1ON = 0” 以停止计数。

## 2. 应用范例

```

/*****
* 文件名: test_61f13x_Timer1.prj
* 功能:   FT61F13x-TMR1 功能演示
* IC:    FT61F135 SOP20
* 晶振:   16M/2T
* 说明:   DemoPortOut 输出 1KHz 占空比 50%的波形-Timer1 实现
*
*
*           FT61F135  SOP20
*           -----
* VDD-----|1(VDD) (VSS)20|-----VSS
* NC-----|2(PC1)  (PA0)19|-----NC
* NC-----|3(PC0)  (PA1)18|-----NC
* NC-----|4(PB7)  (PA2)17|-----NC
* NC-----|5(PB6)  (PA3)16|-----NC
* NC-----|6(PB5)  (PA4)15|---DemoPortOut
* NC-----|7(PB4)  (PA5)14|-----NC
* NC-----|8(PB3)  (PA6)13|-----NC
* NC-----|9(PB2)  (PA7)12|-----NC
* NC-----|10(PB1) (PB0)11|-----NC
*
*           -----
*
*****/

#include "SYSCFG.h"

//=====
//DEFINE
//=====

#define unchar      unsigned char
#define DemoPortOut PA4

/*-----
* 函数名: interrupt ISR
* 功能:   定时器 1 中断
* 输入:   无
* 输出:   无
*-----*/

void interrupt ISR(void)
{
    if(TMR1IF)                //500μs 中断一次 = 1KHz
    {
        TMR1IF = 0;          //清零中断标志位
        DemoPortOut = ~DemoPortOut; //翻转电平
    }
}

/*-----

```

\* 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/2T=8MHz, 0.125μs
    OPTION = 0B00001000;           //Bit3=1 WDT, Bit[2:0]=000=WDT RATE 1:1
    INTCON = 0;                     //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B00000000;             //PA 输入输出 0-输出 1-输入
                                    //PA4-OUT

    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
}
```

/\*-----

\* 函数名称: TIMER1\_INITIAL  
 \* 功能: 初始化设置定时器 1  
 \* 相关寄存器: T1CON0 T2CON0 TMR1 PR1 PIR1  
 \* 设置 TMR1 定时时长=PR1\*T1ck\*TMR1 预分频  
 \*  $=250*(1/16000000)*2*16=500\mu s$

-----\*/

void TIMER1\_INITIAL (void)

```
{
    T1CON0 = 0B00010000;           //预分频器分配给 TMR1, 时钟源为指令时钟
    //Bit4: 预分频器分配位 0-Timer2 1-Timer1
    //Bit3: 睡眠时 T1CK 保持工作 0-NO 1-YES
    //Bit2: TMR1 使能位 0-关闭 1-使能
    //Bit[1:0]: Timer1 时钟源 00-指令时钟

    T2CON0 = 0B00000011;           // Bit[1:0]=11 TMR1 预分频 1:16

    TMR1L = 0;
    TMR1H = 0;

    PR1L = 250;
    PR1H = 0;

    TMR1IF = 0;
```



```
    T1ON = 1;                //使能 TIMER1 启动
}
/*-----
 * 函数名:  main
 * 功能:   主函数
 * 输入:   无
 * 输出:   无
-----*/
void main()
{
    POWER_INITIAL();          //系统初始化
    TIMER1_INITIAL();

    GIE = 1;                  //使能全局中断
    PEIE = 1;                 //使能外设中断
    TMR1IE = 1;               //使能定时器 1 中断
    while(1)
    {
        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.