

# **FT61F02X**

## **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. Timer1 寄存器的读/写操作 .....	6
1.1.3. Timer1 时钟源 .....	7
1.1.4. Timer1 预分频器.....	8
1.1.5. Timer1 工作于异步计数器模式下 .....	8
1.1.6. Timer1 门控 .....	8
1.1.7. Timer1 捕捉/比较时基 .....	8
2. 应用范例.....	9
联系信息 .....	12

## FT61F02x TIMER1 应用

### 1. 定时器(TIMERS)

共有 7 个定时器，包括看门狗定时器(WDT)在内。

	WDT	Timer0	Timer1	Timer2	Timer3/4/5
预分频器 (位)	—	8 (与 WDT 共用)	3 (1x, 2x, 4x, 8x)	4 (1x, 4x, 16x)	7 (1x, 2x, 4x, 8x, 16x, 32x, 64x, 128x)
计数器 (位)	16	8	16	8	12
后分频器 (位)	7 (与 Timer0 共用)	—	—	4 (1 – 16x)	—
时钟源	<ul style="list-style-type: none"> <li>LIRC</li> </ul>	<ul style="list-style-type: none"> <li>指令时钟</li> <li>PA2/T0CKI (转变沿计数器)</li> </ul>	<ul style="list-style-type: none"> <li>指令时钟</li> <li>LP</li> <li>PA7/T1CKI (上升沿计数器)</li> </ul>	<ul style="list-style-type: none"> <li><b>2x 指令时钟</b></li> <li>2x HIRC</li> </ul>	<ul style="list-style-type: none"> <li>HIRC</li> <li><b>2x 指令时钟</b></li> <li>PA2/T0CKI (转变沿计数器)</li> <li>PA7/T1CKI (上升沿计数器)</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。当定时器选择 LP 振荡器作为时钟源时，FOSC 必须相应配置成 LP 模式或选择 INTOSCIO 模式，否则 LP 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”。

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	Timer3/4/5
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>TMR1ON = 0</li> <li>写 TMR1L/H</li> </ul>	<ul style="list-style-type: none"> <li>LIRC 和 HIRC 交叉校准启动</li> <li>写 T2CON, TMR2L/H</li> <li>任何复位动作</li> </ul>	<ul style="list-style-type: none"> <li>写 TMRxL/H</li> <li><b>写 TxCKDIV</b></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>CLRWDWT</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, 特殊事件触发)</li> <li>ECCP 触发特殊事件</li> </ul>	<ul style="list-style-type: none"> <li>TMR2 = PR2 (匹配)</li> </ul>	<ul style="list-style-type: none"> <li>TMRx = PRx (BUZZER 模式下匹配)</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>写 T2CON, TMR2L/H</li> <li>任何复位动作</li> </ul>	—

表 1-2 定时器的计数器和分频器的重置事件

## 1.1. 定时器 1 (TIMER1)

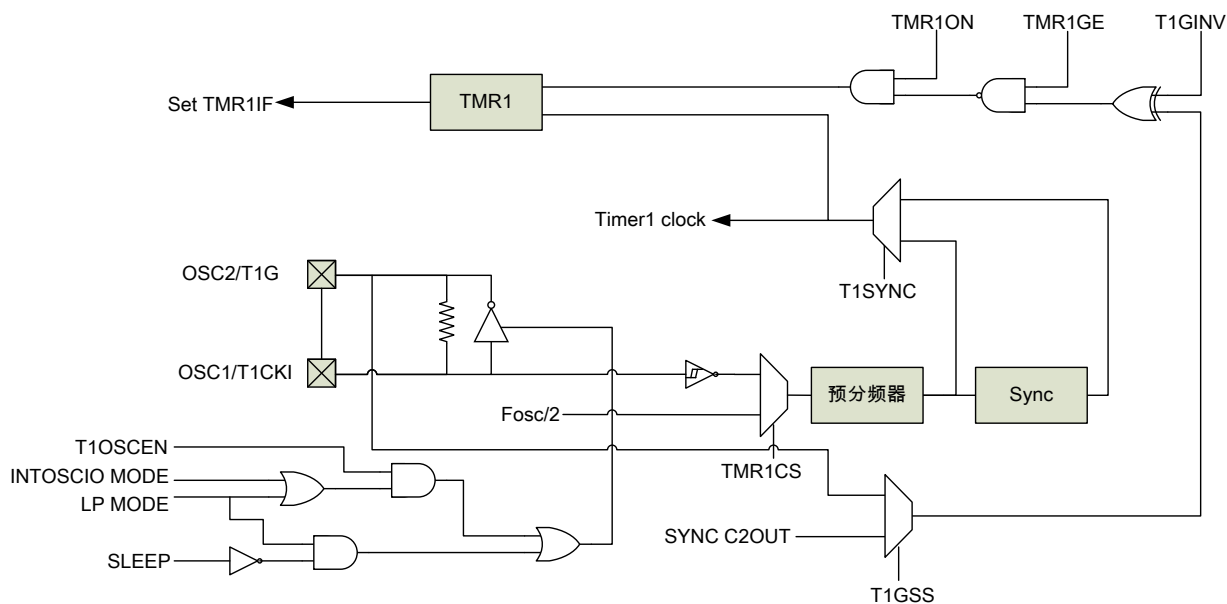


图 1-1 Timer1 结构框图

TIMER1 是一个 16 位的定时器、计数器，有以下特性：

- 一对 16 位定时器/计数器寄存器 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- 通过比较器或 T1G 引脚的 Timer1 门控 (计数使能)
- 溢出中断
- 溢出时唤醒 (仅限外部时钟且异步模式时)
- 捕捉/比较功能的时基
- 特殊事件触发 (带 ECCP)
- 比较器输出与 Timer1 时钟同步

Timer1 模块是 16 位递增计数器。对 TMR1H 或 TMR1L 的写操作将直接更新计数器。

与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

Timer1 的一对寄存器 (TMR1H:TMR1L) 递增至 FFFFh 后返回 0000h。Timer1 计满返回时，PIR1 寄存器的 Timer1 中断标志位 (TMR1IF) 被置 1。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 (TMR1ON, GIE, PEIE 和 TMR1IE)。在中断服务程序中将 TMR1IF 位清零将清除中断。

如需从睡眠中唤醒，需设置为异步计数器模式。在该模式下，可使用外部晶振或时钟源信号使计数器递增。否则 Timer1 将停止计数，维持其进入睡眠前的计数值。所需寄存器配置如下：

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1

- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须将 T1CON 寄存器的 TMR1CS 位置 1
- 可将 T1CON 寄存器的 T1OSCEN 位置 1

### 1.1.1. Timer1 相关寄存器汇总

名称	状态		寄存器	地址	复位值
T1GINV	门控翻转位	1 = 高电平有效(门控为高电平时启动计数) 0 = 低电平有效	T1CON[7]	0x10	RW-0
TMR1GE	门控使能位	1 = 使能 (时钟源非指令时钟) 0 = No	T1CON[3]		RW-0
T1CKPS	Timer1 预分频比 00 = 1    01 = 2    10 = 4    11 = 8		T1CON[5:4]		RW-00
T1OSCEN	LP振荡器使能位 1 = LP振荡器被使能用于Timer1 0 = LP振荡器关闭 注: 当时钟配置为LP或LIRC模式时此位有效, 否则忽略此位。		T1CON[3]		RW-0
T1SYNC	Timer1外部时钟输入同步控制位 当TMR1CS = 1: 1 = 不同步 0 = 同步 注: 当TMR1CS = 0时, 此位被忽略, Timer1使用内部时钟		T1CON[2]		RW-0
TMR1CS	Timer1时钟源选择	1 = PA7/T1CKI(上升沿) 0 = 指令时钟	T1CON[1]		RW-0
TMR1ON	使能位	1 = 使能 0 = 关闭	T1CON[0]		RW-0
T1GSS	门控源选择位	1 = T1G引脚 (配置为数字输入) 0 = 比较器C2的输出	CMCON1[1]	0x1A	RW-1
C2SYNC	比较器 2 输出同步位 1 = 输出与 Timer1 时钟的下降沿同步 0 = 异步输出		CMCON1[0]		RW-0
TMR1L	TMR1 定时、计数结果寄存器低 8 位		TMR1L[7:0]	0x0E	RW- 0000 0000
TMR1H	TMR1 定时、计数结果寄存器高 8 位		TMR1H[7:0]	0x0F	

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]		RW-0
TMR1IE	Timer1与PR1溢出中断 1 = 使能 0 = <u>关闭</u> (无唤醒)	PIE1[0]	0x8C	RW-0
TMR1IF	Timer1与PR1溢出中断标志位 1 = 溢出 (锁存) 0 = <u>未溢出</u>	PIR1[0]	0x0C	RW-0

表 1-4 Timer1 中断使能和状态位

### 1.1.2. Timer1 寄存器的读/写操作

定时器运行于外部异步时钟时，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。

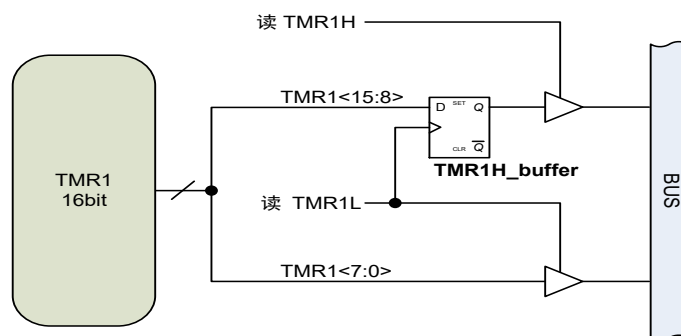


图 1-2 TMR1 读操作结构框图

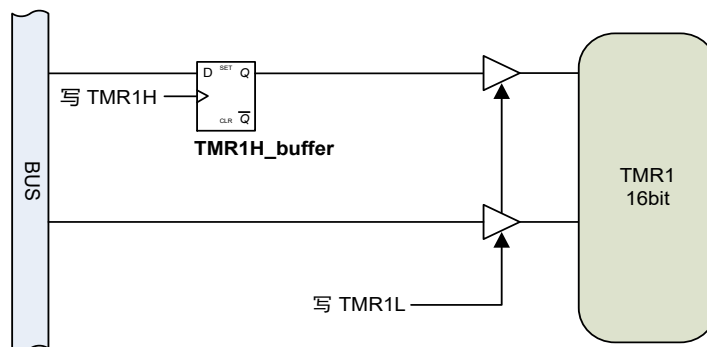


图 1-3 TMR1 写操作结构框

### 1.1.3. Timer1 时钟源

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS=0 时,为指令时钟(2T 模式下)。当 TMR1CS=1 时,时钟源由外部提供(T1CKI 管脚)。

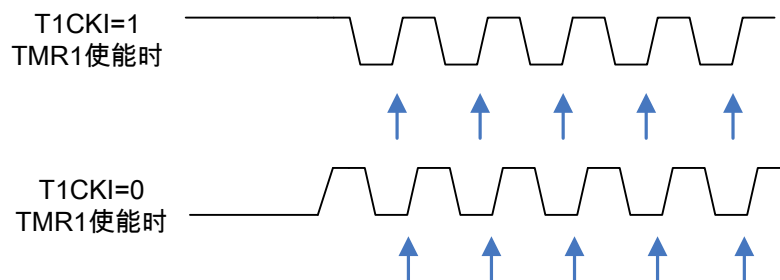
当选择外部时钟源时,Timer1 模块可作为定时器、计数器工作。计数时,Timer1 在外部时钟输入 T1CKI 的上升沿递增。

当时钟配置为 LP 或 INTOSCIO 模式时,Timer1 可使用 LP 振荡器作为时钟源。

注意:

在计数器模式下,发生以下任何一个或多个情况后,计数器在首个上升沿递增前,必须先经过一个下降沿:

- POR 复位后使能 Timer1
- 写入 TMR1H 或 TMR1L
- Timer1 被禁止
- T1CKI 为高电平时 Timer1 被禁止(TMR1ON=0),然后在 T1CKI 为低电平时 Timer1 被使能(TMR1ON=1)。



注意:

1. 箭头所指边沿为计数器递增;
2. 计数器模式下,计数器递增之前必须先经过一个下降沿。

图 1-4 TIMER1 边沿递增示意

OSC1(输入)引脚与 OSC2(输出)引脚之间连接低功耗 32.768kHz 晶振,将 T1CON 寄存器的 T1OSCEN 控制位置 1 使能振荡器,休眠时振荡器将继续工作。

因为 Timer1 振荡器与系统 LP 振荡器共用,则 Timer1 只能在主系统时钟来自内部振荡器或振荡器处于 LP 模式时才能使用此模式。用户必须提供软件延时以确保振荡器正常起振。

当 Timer1 振荡器被使能时,PORTA[7]、PORTA[6]的输出驱动被禁止,且 PA7 和 PA6 位读为 0。但 TRISA7、TRISA6 保持原来的值。

注意:

1. 因振荡器需要一定的起振和稳定时间。所以在将 T1OSCEN 置 1 且在使能 Timer1 之前应当增加适当的延时;
2. 配置为振荡器模式时,T1G 固定输出 1,因此不能用它来门控 TIMER1。

#### 1.1.4. Timer1 预分频器

Timer1 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写；但当发生对 TMR1H 或 TMR1L 的写操作时，或者 TIMER1 被关闭时(TMR1ON 为 0)，预分频计数器被清零。

#### 1.1.5. Timer1 工作于异步计数器模式下

当 T1CON 寄存器的控制位 T1SYNC 置 1 时，外部时钟输入不同步。定时器与内部相位时钟异步递增。若选择了外部时钟源，则定时器将在休眠时继续运行，并可在溢出时产生中断，唤醒处理器。然而，读写定时器时应特别谨慎（见[章节 1.1.2](#)）。

注意：

从同步模式切换为异步模式时，有可能错过一次递增。从异步模式切换为同步模式时，有可能多产生一次递增。

#### 1.1.6. Timer1 门控

Timer1 门控源可软件配置为 T1G 引脚或比较器 C2 的输出，器件可直接使用 T1G 为外部事件定时，或使用比较器 C2 对模拟事件定时。Timer1 门控源的选择请参见 CMCON1 寄存器。

使用 T1CON 寄存器的 T1GINV 位可翻转 Timer1 门控，不论其来自 T1G 引脚还是比较器 C2 的输出。这将配置 Timer1 以确保事件之间存在低电平有效或高电平有效的的时间。

#### 1.1.7. Timer1 捕捉/比较时基

工作于捕捉或比较模式时，ECCP 模块使用一对 TMR1H:TMR1L 寄存器作为时基。

1. 在捕捉模式下，TMR1H:TMR1L 这对寄存器的值在发生某个配置好的事件时被复制到 CCPR1H:CCPR1L 这对寄存器中。
2. 在比较模式下，当 CCPR1H:CCPR1L 这对寄存器的值与 TMR1H:TMR1L 的值匹配时，将触发一个事件。该事件可以是特殊事件触发。

详细信息请参见[章节错误!未找到引用源。](#)“增强型捕捉/比较/PWM 模块”。

## 2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_Timer1.c
* 功能:    FT61F02x-Timer1 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/2T
* 说明:    当 DemoPortIn 悬空或者高电平时,
*          DemoPortOut 输出 1kHz 占空比 50%的波形-Timer1 实现
*          当 DemoPortIn 接地时,DemoPortOut 输出高电平.关定时器中断
*
*          FT61F023  SOP16
*          -----
* VDD-----|1(VDD)  (VSS)16|-----GND
* NC-----|2(PA7)   (PA0)15|-----NC
* NC-----|3(PA6)   (PA1)14|-----NC
* NC-----|4(PA5)   (PA2)13|-----NC
* DemoPortIn---|5(PC3)  (PA3)12|---DemoPortOut
* NC-----|6(PC2)   (PC0)11|-----NC
* NC-----|7(PA4)   (PC1)10|-----NC
* NC-----|8(PC5)   (PC4)09|-----NC
*
*          -----
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define DemoPortOut  PA3
#define DemoPortIn   PC3
/*-----
* 函数名: interrupt ISR
* 功能:    定时器 1 的中断处理
* 设置 Timer1 定时时长=(1/系统时钟周期)*指令周期*预分频值*(65536-TMR1H:TMR1L)
*          =(1/16000000)*2*1*4000=500μs
*          -----*/
void interrupt ISR(void)
{
    if(TMR1IF)
    {
        TMR1IF = 0;
        TMR1L = 0X60;                //定时 500μs=>TMR1=4000*0.125μs=500μs
                                    //初值=65536-4000=61536=>0XF060
        TMR1H = 0XF0;                //赋初值=>TMR1H=0XF0;TMR1L=0X60
        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
    INTCON = 0;                    //暂禁止所有中断
    PORTA = 0B00000000;
    TRISA = 0B00000000;            //PA 输入输出 0-输出 1-输入
                                   //PA3->输出

    PORTC = 0B00000000;
    TRISC = 0B00001000;            //PC 输入输出 0-输出 1-输入
                                   //PC3->输入

    WPUA = 0B00000000;            //PA 端口上拉控制 1-开上拉 0-关上拉
    WPUC = 0B00001000;            //PC 端口上拉控制 1-开上拉 0-关上拉

    OPTION = 0B00001000;           //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,禁止 PA4, PC5 稳压输出
    //Bit5->0,TIMER2 时钟为 Fosc
    //Bit4->0,禁止 LVR
    CMCON0 = 0B00000111;           //关闭比较器, CxIN 为数字 IO 口
}

/*-----
* 函数名: TIMER1_INITIAL
* 功能:   初始化设置定时器 1
* 设置定时时长=(1/系统时钟周期)*指令周期*预分频值*(65536-TMR1H:TMR1L)
*          =(1/16000000)*2*1*4000=500μs
*-----*/

void TIMER1_INITIAL (void)
{
    //需要在中断里重新赋初始值
    T1CON = 0B00000000;
    //Bit[5:4]=00,T2 时钟分频 1:1
    //Bit1=0,T1 时钟源选择内部时钟
    TMR1L = 0X60;                  //TMR1 低八位赋初始值
    TMR1H = 0XF0;                  //TMR1 高八位赋初始值

    TMR1IE = 1;                   //使能 TMER1 的中断
    TMR1ON = 1;                   //使能 TMER1 启动
    PEIE=1;                       //使能外设中断

```

```
    GIE = 1;                                //使能全局中断
}
/*-----
* 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();                       //初始化 T1
    while(1)
    {
        if(DemoPortIn == 1)                //判断输入是否为高电平
        {
            TMR1IE = 1;                     //开定时器 1 中断
        }
        else
        {
            TMR1IE = 0;                     //关定时器 1 中断
            DemoPortOut = 1;
        }
    }
}
```

## 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.