

FT62F08X

TIM4 Application note

目录

1. 基本定时器 TIM4	3
1.1. 特性	3
1.2. TIM4 相关寄存器汇总	3
1.3. TIM4 时钟源	5
1.4. 预分频器	5
1.5. TIM4 中断	5
2. 应用范例	6
联系信息	10

FT62F08x TIM4 应用

1. 基本定时器 TIM4

1.1. 特性

- 8bit 自动重载向上计数器
- 计数时钟可编程预分频
- 计数器溢出中断

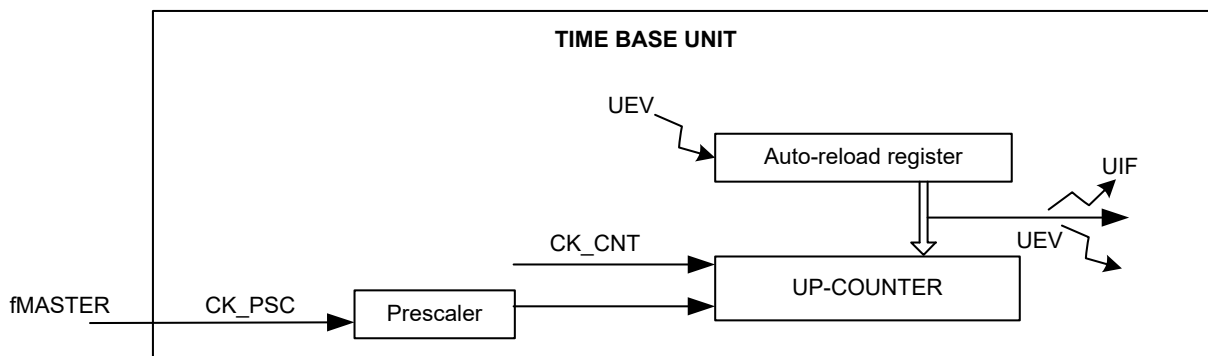


图 1-1 TIM4 原理框图

1.2. TIM4 相关寄存器汇总

[illegible]

表 1-1 Timer4 相关用户寄存器汇总

名称	状态		寄存器	地址	复位值
TIM4EN	<u>TIM4 模块时钟</u>	1 = 使能 0 = 关闭	PCKEN[3]	0x9A	RW-0
SYSON	睡眠模式下，系统时钟控制	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T4ARPE	<u>周期的自动预装载</u> 1 = 使能 (T4ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T4ARR 立即被加载)		TIM4CR1[7]	0x111	RW-0
T4CKS	<u>Timer4 时钟源</u> 00 = Sysclk 10 = LP ^(*) 01 = HIRC 11 = XT ^(*) ^(*) FOSC 应相应配置成 LP/XT 或选择 INTOSCIO 模式，否则振荡器将不会运行		TIM4CR1[5:4]		RW-00
T4OPM	<u>单脉冲模式</u> 1 = 使能 (下一次更新事件到来时，计数器停止) 0 = 关闭 (发生更新事件时，计数器不停止)		TIM4CR1[3]		RW-0
T4URS	<u>当 T4UDIS = 0 时，更新事件中断源</u> 1 = 计数器上溢 0 = 软件设置 T4UG 位或计数器上溢		TIM4CR1[2]		RW-0
T4UDIS	<u>产生更新事件控制</u> 1 = 禁止 0 = 允许		TIM4CR1[1]		RW-0
T4CEN	TIM4 计数器	1 = 使能 0 = 关闭	TIM4CR1[0]		RW-0
T4PSC	<u>Timer4 预分频器</u> 000 = 1 100 = 16 001 = 2 101 = 32 010 = 4 110 = 64 011 = 8 111 = 128 注：必须产生更新事件或 T4CEN=0，更新的预分频值才生效		TIM4PSCR[2:0]	0x116	RW-000
T4CNT	Timer4 计数值		TIM4CNTR[7:0]	0x115	RW-0000 0000
T4ARR	<u>周期的自动重载寄存器(预装载值)</u> 注：此值为 0 时，计数器不工作		TIM4ARR[7:0]	0x117	RW-1111 1111

表 1-2 Timer4 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, T4UIE, T4UG 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (T4UIE, T4UG 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
T4UIE	允许更新中断	1 = 使能	TIM4IER[0]	0x112	RW-0
T4UG ¹	允许更新软件中断	0 = 关闭	TIM4EGR[0]	0x114	WO-0
T4UIF	更新中断标志位	1 = 更新事件等待响应 0 = 无更新事件	TIM4SR[0]	0x113	R_W1C-0

表 1-3 Timer4 中断使能和状态位

1.3. TIM4 时钟源

TIM4 有 4 种时钟源可选，由寄存器位 T4CKS 设置。在 TIM4 的被使能 (PCKEN.TIM4EN=1) 的情况下，所选择的时钟源被自动使能。

注意：

1. 如果要选择 LP 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 LP 模式，否则对应的时钟源将不被使能；
2. 同理，如果要选择 XT 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 XT 模式，否则对应的时钟源将不被使能；

SLEEP 模式下，如果 SYSON 为 1，且 TIM4EN=1，则所选择的时钟源将保持振荡，TIM4 将继续工作；否则，所选的时钟源取决于其他模块的设置情况。

1.4. 预分频器

计数时钟可以进行 3bit 的时钟预分频：

$$f_{CK_CNT} = F_{master} / 2^{(PSCR[2:0])}$$

预分频支持分频自动更新，即在更新事件发生后，能够自动改变预分频值。当 T4CEN 为 0 时，写入预分频寄存器的值也能直接加载实际应用的预分频寄存器中。

1.5. TIM4 中断

TIM4 只有一个中断请求源：

- 更新中断 (计数器上溢或计数器初始化)

在用这些中断之前需要提前打开 TIM4IER 寄存器中的中断使能位 (T4UIE)。不同的中断源还可以配置通过 TIM4EGR 寄存器来产生 (软件产生中断 T4UG)。

¹ 软件置 1，硬件自动清 0。

2. 应用范例

```
//*****
/* 文件名: TEST_62F08x_TIM4.c
* 功能:    FT62F08x-TIM4 功能演示
* IC:      FT62F088 LQFP32
* 内部:    16M
* empno:   500
* 说明:    程序通过 TIM4 中断在 PB3 输出频率为 4K 的方波
*
* 参考原理图 TEST_62F08x_sch.pdf
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define DemoPortOut PB3
/*-----
* 函数名: interrupt ISR
* 功能:   定时器 4 的中断处理
* 输入:   无
* 输出:   无
-----*/
void interrupt ISR(void)
{
    if(T4UIE && T4UIF)
    {
        T4UIF = 1;                //写 1 清零标志位
        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;    //16MHz 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
    PORTB = 0B00000000;
    TRISB = 0B00000000;    //PB 输入输出 0-输出 1-输入 PB3 输出
}
```

```

PORTC = 0B00000000;
TRISC = 0B00000000;    //PC 输入输出 0-输出 1-输入
PORTD = 0B00000000;
TRISD = 0B00000000;    //PD 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
WPUB = 0B00000000;    //PB 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
WPUD = 0B00000000;    //PD 端口上拉控制 1-开上拉 0-关上拉

WPDA = 0B00000000;    //PA 端口下拉控制 1-开下拉 0-关下拉
WPDB = 0B00000000;    //PB 端口下拉控制 1-开下拉 0-关下拉
WPDC = 0B00000000;    //PC 端口下拉控制 1-开下拉 0-关下拉
WPDD = 0B00000000;    //PD 端口下拉控制 1-开下拉 0-关下拉

PSRC0 = 0B11111111;    //PORTA,PORTB 源电流设置最大
PSRC1 = 0B11111111;    //PORTC,PORTD 源电流设置最大

PSINK0 = 0B11111111;    //PORTA 灌电流设置最大 0:最小, 1:最大
PSINK1 = 0B11111111;    //PORTB 灌电流设置最大 0:最小, 1:最大
PSINK2 = 0B11111111;    //PORTC 灌电流设置最大 0:最小, 1:最大
PSINK3 = 0B11111111;    //PORTD 灌电流设置最大 0:最小, 1:最大

ANSELA = 0B00000000;    //全为数字管脚
}
/*-----
* 函数名称: Time4Initial
* 功能:      初始化计时器 4
* 输入参数: 无
* 返回参数: 无
-----*/
void Time4Initial(void)
{
    PCKEN |= 0B00001000;    //TIME4 模块时钟使能

    TIM4CR1 = 0B000000101;
    //Bit7: 0: TIM1_ARR 寄存器没有缓冲, 它可以被直接写入;
    //1: TIM1_ARR 寄存器由预装载缓冲器缓冲。
    //Bit6: 保留
    //Bit[5:4]: timer4 时钟选择位。
    //00: 系统时钟/主时钟
    //01: 内部快时钟 HIRC
    //10: LP 时钟, 只有当 FOSC 选择 LP 模式时才有意义

```

//11: XT 时钟, 只有当 FOSC 选择 XT 模式时才有意义

//Bit3:单脉冲模式

//0: 在发生更新事件时, 计数器不停止;

//1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。

//Bit2:更新请求源

//0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断:

//寄存器被更新(计数器上溢/下溢)

//软件设置 UG 位

//时钟/触发控制器产生的更新

//1: 如果 UDIS 允许产生更新事件, 则只有当下列事件发生时才产生更新中断,

//并 UIF 置 1:

//寄存器被更新(计数器上溢/下溢)

//Bit1:禁止更新

//0: 一旦下列事件发生, 产生更新(UEV)事件:

//计数器溢出/下溢

//产生软件更新事件

//时钟/触发模式控制器产生的硬件复位被缓存的寄存器被装入它们的预装载值。

//1: 不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。

//Bit0: 0: 禁止计数器; 1: 使能计数器。

TIM4IER= 0B00000001;

//BIT0: 0: 禁止更新中断; 1: 允许更新中断。

TIM4SR = 0B00000000;

//Bit0: 当产生更新事件时该位由硬件置 1。它由软件写 1 清 0

//0: 无更新事件产生;

//1: 更新事件等待响应。当寄存器被更新时该位由硬件置 1:

//若 TIM4_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时;

//若 TIM4_CR1 寄存器的 UDIS=0、URS=0, 当设置 TIM4_EGR 寄存器的 UG 位软件对计数器

//CNT 重新初始化时;

//若 TIM4_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始化时。

TIM4EGR = 0B00000000;

//BIT0:该位由软件置 1, 由硬件自动清 0。

//0: 无动作;

//1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清 0(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清 0; 若 DIR=1(向下计数)则计数器取 TIM1_ARR 的值。


```
TIM4CNTR=0;           //TIM4 8 位计数器
```

```
TIM4PSCR=0B00000100;
```

//预分频器对输入的 CK_PSC 时钟进行分频。

//计数器的时钟频率 fCK_CNT 等于 fCK_PSC/2(PSC[2:0])。PSC[7:3]由硬件清 0。

//PSCR 包含了当更新事件产生时装入当前预分频器寄存器的值(包括由于清除 TIMx_EGR 寄存器的 UG 位产生的计数器清除事件)。这意味着如要新的预分频值生效，必须产生更新事件或者 CEN=0。

```
TIM4ARR = 124;
```

//ARR 包含了将要装载入实际的自动重装载寄存器的值。

//当自动重装载的值为空时，计数器不工作。

```
INTCON |= 0B11000000;    //开总中断和外设中断
```

```
}
```

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

```
* 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
```

```
    Time4Initial();
```

```
    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.