

FT61F02X

WDT Application note

目录

1. 定时器(TIMERS)	3
1.1. 看门狗定时器 (Watch Dog Timer, WDT).....	4
1.1.1. WDT 相关寄存器汇总.....	5
1.1.2. WDT 的设置和使用	6
1.1.3. 在 Timer0 和 WDT 之间切换分频电路.....	6
2. 应用范例.....	7
联系信息	10

FT61F02x WDT 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"> LIRC 	<ul style="list-style-type: none"> 指令时钟 PA2/T0CKI (转变沿计数器) 	<ul style="list-style-type: none"> 指令时钟 LP PA7/T1CKI (上升沿计数器) 	<ul style="list-style-type: none"> 2x 指令时钟 2x HIRC 	<ul style="list-style-type: none"> HIRC 2x 指令时钟 PA2/T0CKI (转变沿计数器) PA7/T1CKI (上升沿计数器)

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"> 写 TMR0 PSA 切换 	<ul style="list-style-type: none"> TMR1ON = 0 写 TMR1L/H 	<ul style="list-style-type: none"> LIRC 和 HIRC 交叉校准启动 写 T2CON, TMR2L/H 任何复位动作 	<ul style="list-style-type: none"> 写 TMRxL/H 写 TxCKDIV
计数器	<ul style="list-style-type: none"> WDT, OST 溢出 进入/退出 SLEEP CLRWDW 写 WDTCON 	<ul style="list-style-type: none"> Timer0 溢出 	<ul style="list-style-type: none"> TMR1 = PR1 (匹配, 特殊事件触发) ECCP 触发特殊事件 	<ul style="list-style-type: none"> TMR2 = PR2 (匹配) 	<ul style="list-style-type: none"> TMRx = PRx (BUZZER 模式下匹配)
后分频器	<ul style="list-style-type: none"> 除写 WDTCON 外的以上所有条件 PSA 切换 	—	—	<ul style="list-style-type: none"> 写 T2CON, TMR2L/H 任何复位动作 	—

表 1-2 定时器的计数器和分频器的重置事件

1.1. 看门狗定时器 (Watch Dog Timer, WDT)

WDT 用于“从 SLEEP 中唤醒”或“CPU 挂起时产生系统复位”。当 WDT 计数到预设数量的时钟周期数时则产生溢出。

- 在 SLEEP 模式下，WDT 溢出将触发唤醒。CPU 将从其进入 SLEEP 之前的位置恢复操作。唤醒不是中断，也不是系统复位事件。
- 在正常模式(非 SLEEP 模式)下，WDT 溢出将触发系统复位，随后启动初始化配置(参阅 [章节 错误! 未找到引用源。](#)系统复位)。

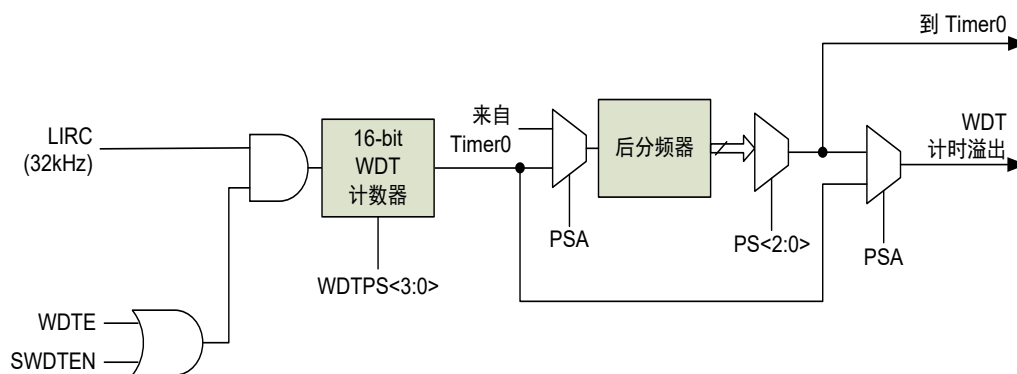


图 1-1 WDT 结构框图

计时超过看门狗定时时间：(WDT-周期 x WDT-后分频比) / WDT 时钟频率，WDT 将溢出。

由于 WDT 后分频器的二进制特性，看门狗定时时间步长呈连续的倍数关系。WDT 溢出前可设置的最长定时时间为：

$$2^{16} \times 2^7 / 32\text{kHz} = \sim 262 \text{ seconds.}$$

1.1.1. WDT 相关寄存器汇总

名称	状态				寄存器	地址	复位值	
WDTPS	WDT 周期				WDTCON[4:1]	0x18	RW-0100	
	0000 = 32		0111 = 4,096					
	0001 = 64		1000 = 8,192					
	0010 = 128		1001 = 16,384					
	0011 = 256		1010 = 32,768					
	0100 = <u>512 (默认)</u>		1011 = 65,536					
	0101 = 1,024		11xx = 65,536					
SWDTEN	1 = WDT使能				WDTCON[0]		RW-0	
	0 = <u>WDT 关闭</u> (当 WDTE = 0 时)							
PSA	1 = <u>分频电路分配给WDT后分频器</u>				OPTION[3]		RW-1	
PS		WDT 后分频比		Timer0 预分频比		OPTION[2:0]	0x81	RW-111
	000		1		2			
	001		2		4			
	010		4		8			
	011	<u>(PSA=1)</u>	8	(PSA=0)	16			
	100		16		32			
	101		32		64			
	110		64		128			
	111		<u>128</u>		<u>256</u>			
	xxx	(PSA = 0)	1	<u>(PSA = 1)</u>	1			

表 1-3 WDT 相关用户寄存器

名称	功能	默认
WDTE	<ul style="list-style-type: none"> 使能 (指令不能禁止) <u>通过指令控制 (SWDTEN)</u> 	SWDTEN 控制

表 1-4 WDT 选项初始化配置寄存器

1.1.2. WDT 的设置和使用

由 WDTE (初始化配置寄存器) 以及 SWDTEN (用户寄存器) 使能 WDT, WDT 触发复位后会产生初始化配置过程。

WDT 时钟源固定使用 32kHz LIRC, 后分频器由 WDTPS, PSA 和 PS 一起设置。

如需阻止 WDT 溢出, 必须在设定的定时时间之前清除 WDT, 具体可参阅 [表 1-2](#) 中的清除 WDT 事件。WDT 被清除后将重新开始计时。

1.1.3. 在 Timer0 和 WDT 之间切换分频电路

共用的硬件分频电路可分配给 Timer0 或 WDT 使用, 当在 Timer0 和 WDT 之间切换分频电路时可能会导致系统误复位。

将分频电路从分配给 Timer0 切换至 WDT 时, 必须遵循以下指令顺序:

```
BANKSEL TMR0           ; Can skip if already in TMR0 bank
CLRWDW                  ; Clear WDT
CLRR TMR0               ; Clear TMR0 and scaler
BANKSEL OPTION
BSR OPTION, PSA         ; Select WDT
LDWI b'11111000'       ; Mask scaler bits (PS2-0)
ANDWR OPTION, W
IORWI b'00000101'       ; Set WDT scaler bits to 32 (or any value desired)
STR OPTION
```

将分频电路从分配给 WDT 切换至 Timer0 时, 必须遵循以下指令顺序:

```
CLRWDW                  ; Clear WDT and scaler
BANKSEL OPTION
LDWI b'11110000'       ; Mask TMR0 select and scaler bits (PSA, PS2-0)
ANDWR OPTION, W
IORWI b'00000011'       ; Set Timer0 scale to 16 (or any value desired)
STR OPTION
```

2. 应用范例

```
//*****
/* 文件名: TEST_61F02x_WDT.c
* 功能:    FT61F02x-WDT 功能演示
* IC:      FT61F023_WDT SOP16
* 晶振:    16M/2T
* 说明:    程序中开启看门狗并将看门狗时间设置为 32ms
*          a. 主函数先在 DemoPortOut 脚输出一个高 3ms 低 3ms 的信号,
*          b. 然后循环输出高 1ms 低 1ms, 500Hz 的信号。
*          如果不在主程序中清看门狗, 则每隔 32ms 单片机复位后, 会重复输出 a+b 信号;
*          如果在主函数中清看门狗, 则一直输出 b 信号
*
*          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
*  NC-----|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
/*-----
* 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/2=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 = 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
```

```
OPTION = 0B00001000;    //Bit3=1,WDT MODE,PS=000=WDT 预分频比 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: DelayUs
* 功能:   短延时函数 --16M-2T--大概快 1%左右.
* 输入:   Time 延时时间长度 延时时长 Time μs
* 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 快 1%
* 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(197);
        }
    }
}
```



```

/*-----
* 函数名: WDT_INITIAL
* 功能:   初始化设置看门狗 1S 时间复位
* 设置看门狗周期=(1/32000)*16 位预分频值*8 位预分频值
*          =(1/32000)*1024*1=32ms
*-----*/
void WDT_INITIAL (void)
{
    //CLRWDWT();           //清看门狗
    PSA=1;                //时钟分频分给 WDT
    WDTCON = 0B00001011;
    //Bit[4:1]: WDTPS=0101-WDT 预分频比 1:1024
    //Bit0:   SWDTEN=1-WDT 使能
}
/*-----
* 函数名: main
* 功能:   主函数
* 输入:   无
* 输出:   无
*-----*/
void main()
{
    POWER_INITIAL();       //系统初始化
    WDT_INITIAL();         //看门狗初始化
    DemoPortOut = 1;
    DelayMs(3);            //3ms
    DemoPortOut = 0;
    DelayMs(3);            //3ms
    while(1)
    {
        CLRWDWT();        //清看门狗
        DemoPortOut = 1;
        DelayMs(1);        //1ms
        DemoPortOut = 0;
        DelayMs(1);        //1ms
    }
}

```

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.