FT61F02X

TIMER0 Application note

目录

FT61F02x TIMER0 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	• LIRC	• 指令时钟 • PA2/T0CKI (转变沿计数器)	• 指令时钟 • LP • PA7/T1CKI (上升沿计数器)	• 2x 指令时钟 • 2x HIRC	• 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	–	• 写 TMR0 • PSA 切换	• TMR1ON = 0 • 写 TMR1L/H	• LIRC 和 HIRC 交叉校准启动 • 写 T2CON, TMR2L/H • 任何复位动作	• 写 TMRxL/H • 写 TxCKDIV
计数器	• WDT, OST 溢出 • 进入/退出 SLEEP • CLRWDT • 写 WDTCON	• Timer0 溢出	• TMR1 = PR1 (匹配，特殊事件触发) • ECCP 触发特殊事件	• TMR2 = PR2 (匹配)	• TMRx = PRx (BUZZER 模式下匹配)
后分频器	• 除写 WDTCON 外的以上所有条件 • PSA 切换	–		• 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. 定时器 0 (TIMER0)

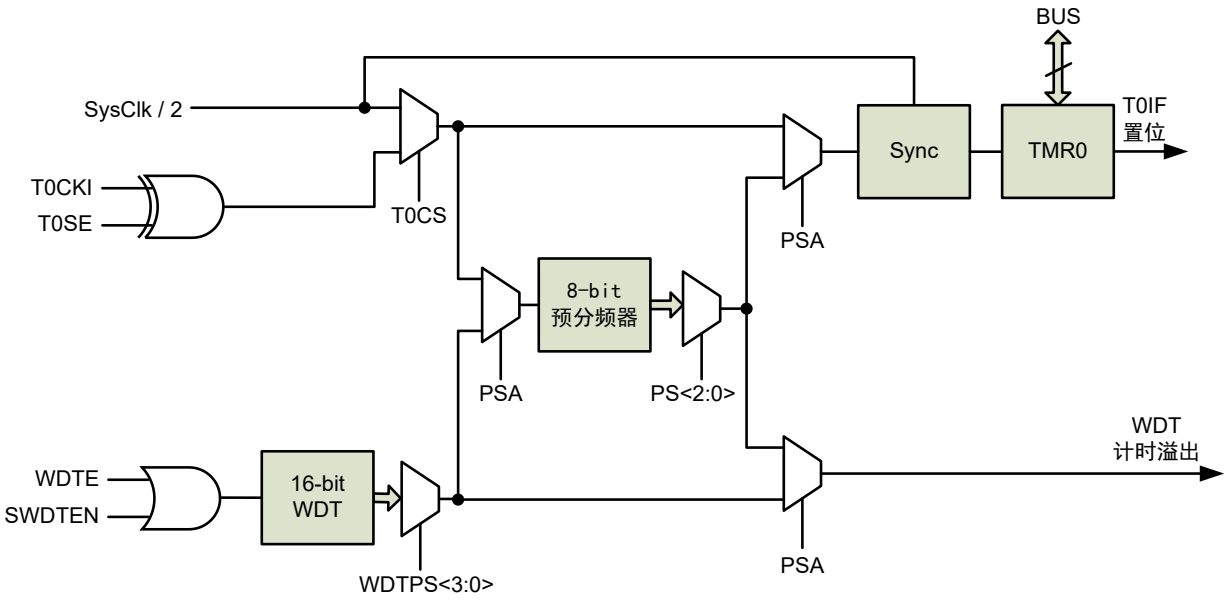

图 1-1 Timer0 结构框图

Timer0 可用作 I/O “PA2−T0CKI” 的上升沿/下降沿计数器，或计时的定时器 (时钟源为指令时钟)。

Timer0 计数和定时溢出时间 = *TMR0[7:0] * Timer0_预分频*

Timer0 溢出将置位中断标志位(T0IF)，是否触发中断取决于相应的使能控制位(T0IE 和 GIE)。

注：

1. 对 TMR0 进行写操作后的 2 个指令周期内，Timer0 停止递增；
2. 在 SLEEP 模式下，Timer0 将停止计数，维持其进入睡眠前的计数值；
3. 如果 Timer0 用于对 T0CKI 进行计数，那么相对于 Timer0，对 T0CKI 有最小周期、高/低脉冲宽度的要求。除非 T0CKI 非常快且 T_{T0CK} 非常慢，否则通常都满足这些限制条件；

T0CKI	最小值	单位	条件
高/低脉冲宽度	0.5 * T_{T0CK} + 20	ns	无预分频
	10	ns	有预分频
周期	20 和 (T_{T0CK}+40)/N 中的较大者	ns	N = 1, 2, 4, …, 256 (有预分频) N = 1 (无预分频)

4. 关于 “在 Timer0 和 WDT 之间切换分频电路” 请参阅 **章节 1.1.1**；

1.1.1. Timer0 相关寄存器汇总

名称	状态				寄存器	地址	复位值
T0CS	Timer0时钟源	1 = PA2/T0CKI (计数器) 0 = 指令时钟 (定时器)			OPTION[5]	0x81	RW-1
T0SE	计数器触发沿	1 = 下降沿 0 = 上升沿			OPTON[4]		RW-1
PSA	1 = 分频电路分配给 WDT 后分频器 0 = 分频电路分配给 Timer0 预分频器				OPTION[3]		RW-1
PS		WDT 后分频比		TIMER0 预分频比	OPTION[2:0]		RW-111
	000		1		2		
	001		2		4		
	010		4		8		
	011	(PSA=1)	18	(PSA=0)	16		
	100		16		32		
	101		32		64		
	110		64		128		
	111		128		256		
	xxx	(PSA =0)	1	(PSA =1)	1		
TMR0[7:0]	Timer0 计数值				TMR0[7:0]	0x01	RW-xxxx xxxx

表 1-3 Timer0 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 (T0IE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B	RW-0
T0IE	Timer0溢出中断控制位	1 = 使能 0 = 关闭 (无唤醒)	INTCON[5]		RW-0
T0IF	Timer0溢出中断标志位	1 = 已经溢出 (锁存) 0 = 未溢出	INTCON[2]		RW-0

表 1-4 Timer0 中断使能和状态位

1.1.2. 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
CLRWDT                      ; 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：

```
CLRWDT                      ; 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_Timer0.c
* 功能：    FT61F02x-Timer0 功能演示
*  IC：     FT61F023 SOP16
* 晶振：    16M/2T
* 说明：    DemoPortOut 输出 60Hz 占空比 50%的波形-Timer0 实现
*
*                 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：interrupt ISR
*   功能：    定时器 0 的中断处理
*   输入：    无
*   输出：    无
 ---------------------------------------------*/
void interrupt ISR(void)
{
    if(T0IE && T0IF)                        //8.16ms 翻转一次≈60Hz
    {
        T0IF = 0;
        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 = 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 = 0B0000000;               //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：TIMER0_INITIAL
 *   功能：   初始化设置定时器 0
 *   设置 TMR0 定时时长=(1/系统时钟频率)*指令周期*预分频值*255
 *                      =(1/16000000)*2*256*255=8.16ms
 ---------------------------------------------*/
void TIMER0_INITIAL (void)
{
    OPTION = 0B00000111;
    //Bit5:    T0CS Timer0 时钟源选择
    //       1-外部引脚电平变化 T0CKI   0-内部时钟(FOSC/2)
    //Bit4:    T0CKI 引脚触发方式  1-下降沿   0-上升沿
    //Bit3:    PSA  预分频器分配位  0-Timer0   1-WDT
    //Bit[2:0]:PS 8 个预分频比  111 - 1:256

    T0IF = 0;                       //清空 T0 软件中断
}
/*---------------------------------------------
 *   函数名：main
 *   功能：   主函数
 *  输入：   无
 *  输出：   无
 ---------------------------------------------*/
void main()
```

```
{
    POWER_INITIAL();                    //系统初始化
    TIMER0_INITIAL();
    GIE = 1;                            //开中断
    T0IE = 1;                           //开定时器/计数器 0 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.