

FT61F13X

TIMER2 Application note

目录

1. 定时器(TIMERS)	3
1.1. 定时器 2 (TIMER2).....	5
1.1.1. Timer2 相关寄存器汇总	6
1.1.2. TMR2 寄存器的读/写操作	7
2. 应用范例.....	8
联系信息	11

FT61F13x TIMER2 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"> • LP • XT • HIRC • <u>LIRC</u> 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> • PA2/T0CKI (转变沿计数器) 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> 	<ul style="list-style-type: none"> • LP • XT • HIRC • <u>指令时钟</u> • LIRC • 2x 指令时钟 • 2x HIRC • 2x (EC, LP or XT)

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

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 2 (TIMER2)

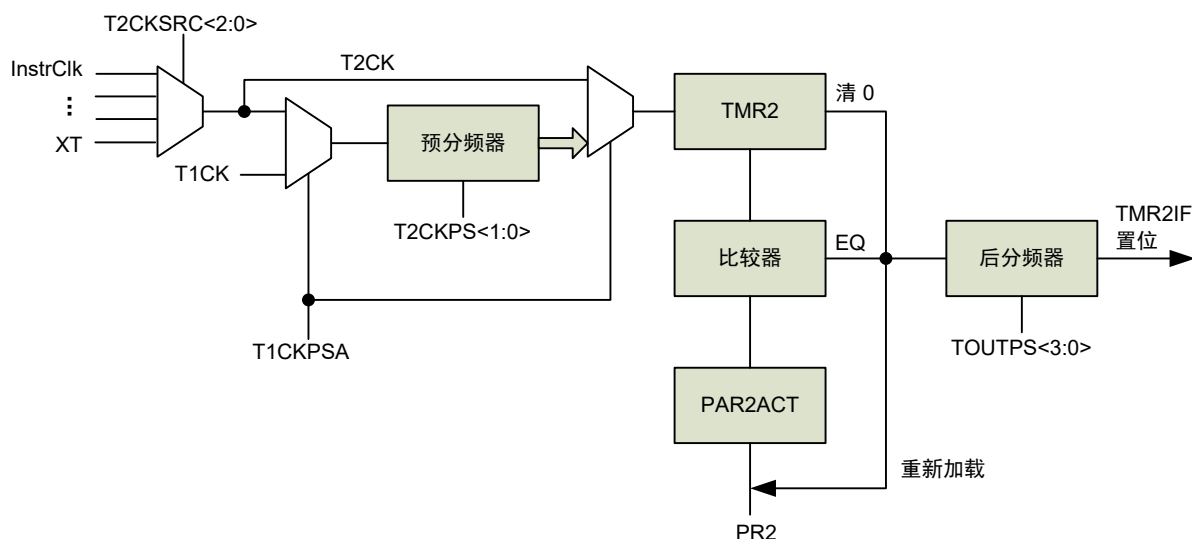


图 1-1 Timer2 结构框图

Timer2 为定时器, 也可用于产生 PWM (无后分频器), 及用于 LIRC 和 HIRC 交叉校准计数(CKCNTI=1)。可同时运用计数匹配和后分频器溢出功能。

Timer2 时钟源由 T2CKSRC 选择 (如果选 LIRC 则由 LFMOD 进一步选定频率), Timer2 时钟被送入 Timer2 预分频器(预分频比为 1, 4 或 16), 预分频器的输出被用于递增 TMR2 寄存器, TMR2 从 0x00 开始递增直至与 PR2 匹配。匹配时:

1. TMR2 在下一个递增周期复位为 0x00;
2. Timer2 后分频器递增;
3. 当 Timer2 后分频器的递增输出值与后分频比设置值 (1, 2 15 或 16) 相等时, Timer2 溢出;
4. 中断标志位 TMR2IF 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控制位(GIE, PEIE 和 TMR2IE);

注:

1. 注意: **当预分频器分配给 Timer1 (T1CKPSA = 1) 时, 不管 TMR2ON 为何值, Timer2 将启动计数**, 与 CPU 是否处于 SLEEP 状态也无关。如需关闭 Timer2 模块, 则需关闭所有 PWM 通道并设置 TMR2IE = 0。
2. 对 T2CON0 进行写操作并不会清零 TMR2 寄存器。
3. TMR2 和 PR2 都是可读/写寄存器。复位时, 其值分别为 0x0000 和 0xFFFF。
4. 当 (“TMR2ON = 1”, “T2CKRUN = 1”, “T2CKSRC ≠ 000”) 时, Timer2 在 SLEEP 模式下将保持运行。

1.1.1. Timer2 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
T2CKRUN	睡眠时T2CK保持工作	1 = Yes (时钟源非指令时钟) 0 = <u>No</u>	MSCON0[0]	0x1B	RW-0
PR2U	<u>更新周期和占空比的即时生效控制位</u> 1 = PR2/P1xDTy 缓冲值立即分别更新到 PR2ACT 和 P1xDTyACT 0 = 周期结束后正常更新		T2CON0[7]	0x12	WO1-0
TOUTPS	<u>Timer2 后分频比</u> 0000 = 1 0100 = 5 1000 = 9 1100 = 13 0001 = 2 0101 = 6 1001 = 10 1101 = 14 0010 = 3 0110 = 7 1010 = 11 1110 = 15 0011 = 4 0111 = 8 1011 = 12 1111 = 16		T2CON0[6:3]		RW-0000
TMR2ON	Timer2 (PWM单脉冲模式下自动清0)	1 = 使能 0 = <u>关闭</u>	T2CON0[2]		RW-0
T2CKPS (T1CKPS)	Timer2/Timer1预分频比	00 = 1 1x = 16 01 = 4	T2CON0[1:0]		RW-00
T2CKSRC	<u>Timer2 时钟源</u> 000 = <u>指令时钟</u> 100 = HIRC 001 = 2 x 指令时钟 101 = LIRC 010 = 2x HIRC 110 = LP ^(*) 011 = 2x LP, XT or EC ^(*) 111 = XT ^(*) ^(*) FOSC 应相应配置成 LP/XT/EC 模式或选择 INTOSCIO 模式, 否则振荡器将不会运行 [*]		T2CON1[2:0]	0x9E	RW-000
LFMOD	1: LIRC = 256kHz 0: <u>LIRC = 32kHz</u>		OSCCON[7]	0x8F	RW-0
PR2L	PR2 周期寄存器低 8 位		PR2L[7:0]	0x91	RW-1111 1111
PR2H	PR2 周期寄存器高 8 位		PR2H[7:0]	0x92	RW-1111 1111
TMR2L	TMR2 计数结果寄存器低 8 位		TMR2L[7:0]	0x11	RW-0000 0000
TMR2H	TMR2 计数结果寄存器高 8 位		TMR2H[7:0]	0x13	RW-0000 0000

表 1-3 Timer2 相关用户控制寄存器

名称	状态	寄存器	地址	复位值
GIE	<u>全局中断</u> 1 = 使能 (PEIE, TMR2IE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B	RW-0
PEIE	外设总中断 1 = 使能 (TMR2IE 适用) 0 = 关闭 (无唤醒)	INTCON[6]	0x18B	RW-0
TMR2IE	Timer2与PR2匹配中断 1 = 使能 0 = 关闭 (无唤醒)	PIE1[1]	0x8C	RW-0
TMR2IF	Timer2与PR2匹配中断标志位 1 = 匹配 (锁存) 0 = 不匹配	PIR1[1]	0x0C	RW-0

表 1-4 Timer2 中断使能和状态位

1.1.2. TMR2 寄存器的读/写操作

TMR2H 和 TMR2L 不能同时读或写。通过 TMR2H 的内部缓存 TMR2H_buf 可解决此问题，必须遵循以下读写顺序：

- 读 TMR2 时，先读 TMR2L，此时 TMR2H 的值将被锁存到 TMR2H_buf，然后读 TMR2H。当 Timer2 的时钟源不是指令时钟时，需设置 “TMR2ON = 0” 以停止计数，然后在读 TMR2 之前执行 1 条 NOP 指令。
- 写 TMR2 时，先写 TMR2H，此时 TMR2H 的值将被储存在 TMR2H_buffer 中。然后写 TMR2L，此时 TMR2H 和 TMR2L 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 “TMR2ON = 0” 以停止计数。

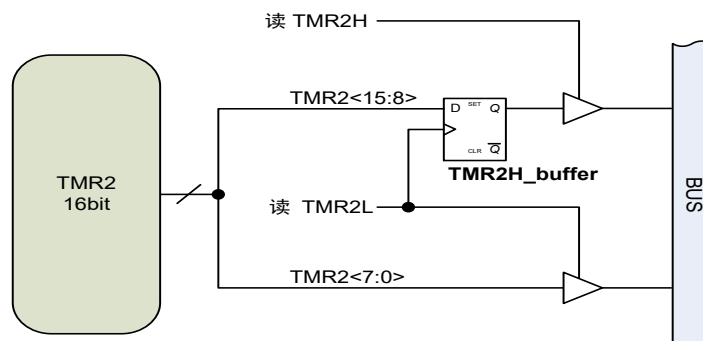


图 1-2 TMR2 读操作结构框图

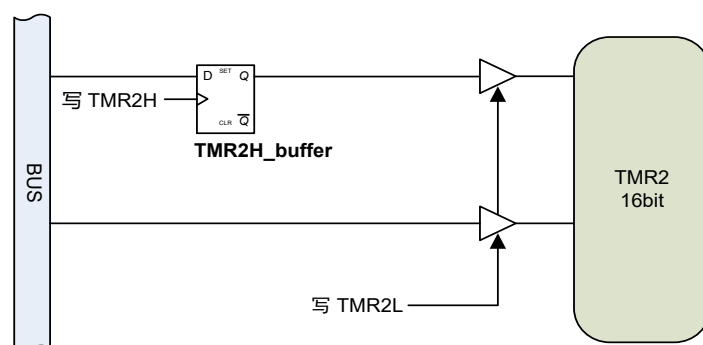


图 1-3 TMR2 写操作结构框图

2. 应用范例

```
//*****
/* 文件名: TEST_61F13x_Timer2.c
* 功能:    FT61F13x-Time2 功能演示
* IC:      FT61F135 SOP20
* 晶振:    16M/2T
* 说明:    当 DemoPortIn 悬空或者高电平时, DemoPortOut 输出 5KHz 占空比 50%的波形
*          当 DemoPortIn 接地时, DemoPortOut 输出高电平-Timer2 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|----DemoPortIn
* 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 unchar      unsigned char
#define DemoPortOut PA4
#define DemoPortIn  PA2
/*-----
* 函数名: interrupt ISR
* 功能:   中断处理程序
* 输入:   无
* 输出:   无
*-----*/
void interrupt ISR(void)
{
    if(TMR2IE && TMR2IF)          //100µs 中断一次 = 5KHz
    {
        TMR2IF = 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
    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 = 0B00000100;             //PA 输入输出 0-输出 1-输入
                                    //PA4-OUT PA2-IN

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

    WPUA = 0B00000100;             //PA 端口上拉控制 1-开上拉 0-关上拉
                                    //开 PA2 上拉
    WPUC = 0B00000000;             //PC 端口上拉控制 1-开上拉 0-关上拉
}

/*-----
 * 函数名:   TIMER2_INITIAL
 * 功能:     初始化设置定时器 2
 * 寄存器:   T2CON TMR2 PR2 PIE1 PIR1 INTCON
 * 设置 TMR2 定时时长=(PR+1)*Tt2ck*TMR2 预分频
 *           =(199+1)*(1/16000000)*2*4=100μs
-----*/

void TIMER2_INITIAL (void)
{
    T2CON0 = 0B00000001;           //TMR2 预分频 1:1, 后分频 1:4
    //Bit7:   更新周期和占空比及时生效位 0-正常更新 1-立即更新
    //Bit[6:3]: 定时器 2 输出后分频比选择 0000-1:1
    //Bit2:    TMR2 使能位 0-关闭定时器 2; 1-打开定时器 2
    //Bit[1:0]: 定时器 2 输出预分频比选择 01-1:4

    T2CON1 = 0B00000000;           //Bit[2:0]=000, TMR2 时钟来自指令时钟

    TMR2H = 0;
    TMR2L = 0;

    PR2H = 0;
    PR2L = 199;

```

```

    TMR2IF = 0;           //清零 TMR2 中断标志
    TMR2IE = 1;           //使能 TMR2 中断
    TMR2ON = 1;           //使能 TMR2 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
    TIMER2_INITIAL();      //初始化 TMR2

    while(1)
    {
        if(DemoPortIn)     //判断输入是否为高电平
        {
            TMR2IE = 1;     //开定时器 2
        }
        else
        {
            TMR2IE = 0;     //关定时器 2
            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.