

FT61F02X

PWM345 Application note

目录

1. PWMX 和 TIMERX.....	3
Timer3/4/5 溢出周期	3
Timer3/4/5 相关寄存器汇总	4
1.1. TMRxH/L 的读写	6
1.2. BUZZER 工作模式	6
BUZZER 周期	6
1.3. PWM 工作模式	6
PWMx 周期	7
PWMx 占空比	7
PWMx 配置	7
2. 应用范例.....	8
联系信息	12

FT61F02x PWM345 应用

1. PWMx 和 TIMERx

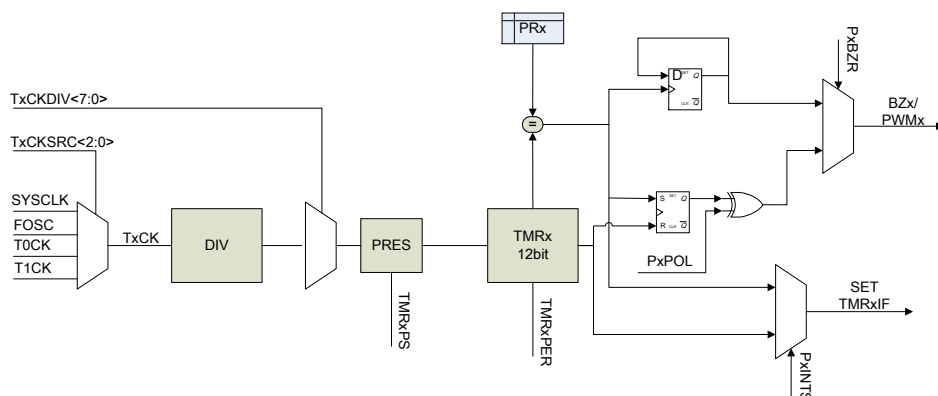


图 1-1 PWM3/4/5 结构框图

除了 ECCP 模块所提供的 1 路增强型 PWM 之外,片内还集成了 3 路时基和占空比相互独立的 12bit PWM, 它们都具备以下特性:

- 递增计数器
- BUZZER / PWM 输出可选
- BUZZER / PWM 输出极性可选
- 8 种 PWM 分辨率
- 4 种时钟源可选
- 1~256 分频可选
- 7 位预分频
- 溢出中断或者匹配中断
- 时钟调制输出

TIMER3/4/5 都是 12 位的递增计数器, 可以通过寄存器 TMRxH:TMRxL (x 可以是 3/4/5) 对计数值进行访问, 软件对 TMRxH:TMRxL 的写操作将直接更新计数值。当 PxEN (PWMxCR1.7) 为 0 时, 工作在定时器模式。当选择内部时钟作为计数源, 则为定时器, 当选择外部时钟作为计数源, 则为计数器。

Timer3/4/5 溢出周期

TIMERx 最大位数为 12bit, 通过对 PxPER[2:0]的配置可以选择不同的溢出周期。

- 在 BUZZER 模式下, 当计数值和 PR 寄存器相等时, 再来一个计数时钟 TIMERx 便会自动清 0;
- 在 TIMERx 已经开启的情况下改写 PxPER 的值可能会把 TMRxIF 置 1, 建议先配置好 PxPER 后才开启 TMRxON。

当 TIMERx 计数值 TMRxH:TMRxL 递增到最大计数值(由 PxPER 指定)后, 此时 TIMERx 发生溢出, 并置位中断标志位(TMRxIF), 而 TMRxH:TMRxL 寄存器将在下一个递增周期复位为 0x00。溢出后是否触发中断和/或从睡眠中唤醒则取决于相应的使能/关闭控制位 (GIE, PEIE 和 TMRxIE)。在中断服务退出

前需将标志位 TMRxIF 清 0，以免循环进入中断。

如需从睡眠中唤醒，需配置为使用外部时钟 T0CK/T1CK (PxCKSRC=010 或者 011)，否则 Timer1 将停止计数，维持其进入睡眠前的计数值。

Timer3/4/5 相关寄存器汇总

名称	状态		寄存器	地址	复位值
PxPER	<u>PWMx 周期选择</u>		PWMxCR0 [6:4] x = 3,4,5	0x10F/ 0x115/ 0x11B	RW- 000
	000 = 4位	100 = 9位			
	001 = 5位	101 = 10位			
	010 = 6位	110 = 11位			
	011 = 8位	111 = 12位			
PxCKSRC	<u>TIMERx/PWMx 时钟选择</u> ¹		PWMxCR0 [3:1] x = 3,4,5	0x10F/ 0x115/ 0x11B	RW-000
	000 = 系统时钟/(TxCKDIV+1)				
	001 = HIRC/(TxCKDIV +1)				
	010 = T0CK/(TxCKDIV +1)				
	011 = T1CK/(TxCKDIV +1)				
	100 = HIRC/(TxCKDIV +1), PWM3 输出低电平				
	101 = HIRC/(TxCKDIV +1), PWM3 输出高电平				
	110 = HIRC/(TxCKDIV +1), PWM3 根据高脉冲调制 PxCK				
PxBZR	<u>输出选择</u>	1 = BUZZER输出 0 = PWM输出	PWMxCR0[0] x = 3,4,5		RW-0
PxEN	<u>TIMERx/PWMx 工作模式</u>	1 = PWM/BUZZER模式 0 = 定时器模式	PWMxCR1 [7] x = 3,4,5	0x110/ 0x116/ 0x11C	RW-0
PxPOL	<u>PWMx 输出极性</u>	1 = 低电平有效 0 = 高电平有效	PWMxCR1[6] x = 3,4,5		RW-0
TMRxPS	<u>PWMx预分频比</u> ²		PWMxCR1 [5:3] x = 3,4,5		0x110/ 0x116/ 0x11C
	000 = 1:1	100 = 1:16			
	001 = 1:2	101 = 1:32			
	010 = 1:4	110 = 1:64			
	011 = 1:8	111 = 1:128			
TMRxON	<u>TIMERx 使能位</u>	1 = 使能 0 = 关闭	PWMxCR1[2] x = 3,4,5		RW-0
T3CKDIV	TMR3时钟频率为 $F_{T3CK} / (T3CKDIV + 1)$ ³		T3CKDIV[7:0]	0x111	RW-0000 0000
T4CKDIV	TMR4时钟频率为 $F_{T4CK} / (T4CKDIV + 1)$ ³		T4CKDIV[7:0]	0x117	RW-0000 0000

¹ 注: 选择为内部 RC 快时钟(HIRC)时, 不管系统时钟选择慢时钟、外部时钟或者晶体时钟, 内部 HIRC 都会打开, 除非进入睡眠模式

² 预分频计数器 TMRxPS 不能直接读写, 当对 TMRxH/xL 寄存器进行写操作时, 预分频计数器会自动清 0。

³ 注: 当对 TxCKDIV 寄存器进行写操作时, 除数分频器将自动清 0。

名称	状态	寄存器	地址	复位值
T5CKDIV	TMR5时钟频率为 $F_{T5CK} / (T5CKDIV + 1)$ 3	T5CKDIV[7:0]	0x11D	RW-0000 0000
TMR3L	TMR3 计数结果寄存器低 8 位	TMR3L[7:0]	0x10C	RW-xxxx xxxx
TMR3H	TMR3 计数结果寄存器高 4 位	TMR3H[7:4]	0x10D	RW-xxxx
PR3H	PR3 周期寄存器高 4 位	TMR3H[3:0]		RW-1111
PR3L	PR3 周期寄存器低 8 位	PR3L[7:0]	0x10E	RW-1111 1111
TMR4L	TMR4 计数结果寄存器低 8 位	TMR4L[7:0]	0x112	RW-xxxx xxxx
TMR4H	TMR4 计数结果寄存器高 4 位	TMR4H[7:4]	0x113	RW-xxxx
PR4H	PR4 周期寄存器高 4 位	TMR4H[3:0]		RW-1111
PR4L	PR4 周期寄存器低 8 位	PR4L[7:0]	0x114	RW-1111 1111
TMR5L	TMR5 计数结果寄存器低 8 位	TMR5L[7:0]	0x118	RW-xxxx xxxx
TMR5H	TMR5 计数结果寄存器高 4 位	TMR5H[7:4]	0x119	RW-xxxx
PR5H	PR5 周期寄存器高 4 位	TMR5H[3:0]		RW-1111
PR5L	PR5 周期寄存器低 8 位	PR5L[7:0]	0x11A	RW-1111 1111

表 1-1 Timer3/4/5 相关用户控制寄存器

名称	状态	寄存器	地址	复位值
<u>全局中断</u>				
GIE	1 = 使能 (PEIE, TMR2IE 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 (TMR2IE 适用) 0 = <u>关闭</u> (无唤醒)	INTCON[6]		RW-0
P3INTS	Timer3中断选择位 1 = TMR3 与 PR3 匹配中断 0 = TMR3 溢出中断	PWM3CR0[7]	0x10F	RW-0
TMR3IE	Timer3中断使能位 1 = 使能 0 = <u>关闭</u> (无唤醒)	PWM3CR1[1]	0x110	RW-0
TMR3IF	Timer3中断标志位 1 = 匹配/溢出 (锁存) 0 = <u>不匹配/无溢出</u>	PWM3CR1[0]		RW-0
P4INTS	Timer4中断选择位 1 = TMR4 与 PR4 匹配中断 0 = TMR4 溢出中断	PWM4CR0[7]	0x115	RW-0
TMR4IE	Timer4中断使能位 1 = 使能 0 = <u>关闭</u> (无唤醒)	PWM4CR1[1]	0x116	RW-0
TMR4IF	Timer4中断标志位 1 = 匹配/溢出 (锁存) 0 = <u>不匹配/无溢出</u>	PWM4CR1[0]		RW-0
P5INTS	Timer5中断选择位 1 = TMR5 与 PR5 匹配中断 0 = TMR5 溢出中断	PWM5CR0[7]	0x11B	RW-0
TMR5IE	Timer5中断使能位 1 = 使能 0 = <u>关闭</u> (无唤醒)	PWM5CR1[1]	0x11C	RW-0
TMR5IF	Timer5中断标志位 1 = 匹配/溢出 (锁存) 0 = <u>不匹配/无溢出</u>	PWM5CR1[0]		RW-0

表 1-2 Timer3/4/5 中断使能和状态位

1.1. TMRxH/L 的读写

TIMERx 为异步时钟，其在运行状态下软件读取 12bit 的计数值时，可能在读取低 8bit 之后，TIMERx 产生了溢出导致计数值复位，此时若再读高 4bit 则为 0。

而对于写操作，建议先把 TIMERx 停止 (TMRxON=0)，再把目标值写入 TMRxH/TMRxL。

1.2. BUZZER 工作模式

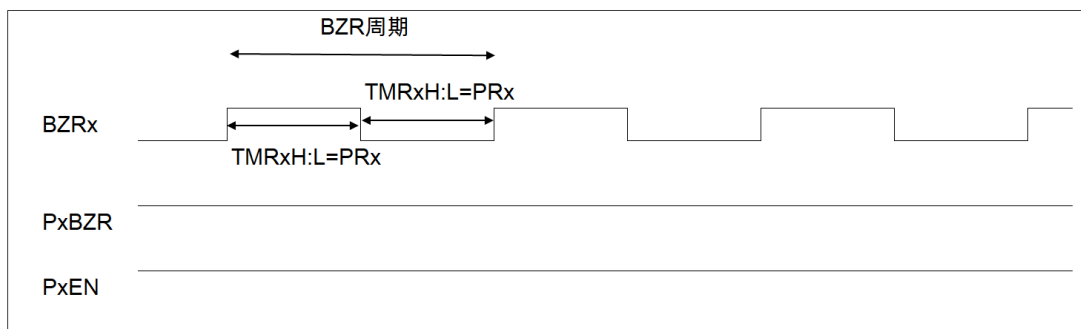


图 1-2 50%占空比的 BUZZER 方波

BUZZER 周期

$$\text{公式 1.1} \quad T_{BUZ} = 2 * 2^{TMRxPS} * PRx * T_{PxCK}$$

注意：

1. 工作在 BUZZER 模式下时，TIMERx 自动工作在 12bit 模式，与 PxPER 的值无关
2. 当 TMRxH:L 等于 PRx 时，TMRxH:L 将自动清 0
3. BUZZER 模式下：
 - 如果 PRx=0x000，则 BZRx 管脚固定输出 0
 - 12bit 的 TIMERx 溢出时 TMRxIF 才会置 1
4. 同 TIMER 模式一样，通过配置 (PxEN、PxBZR、TMRxON、TMRxIE、PEIE) 和“PxCKSRC = 010 或 011”，BUZZER 可在睡眠模式工作

1.3. PWM 工作模式

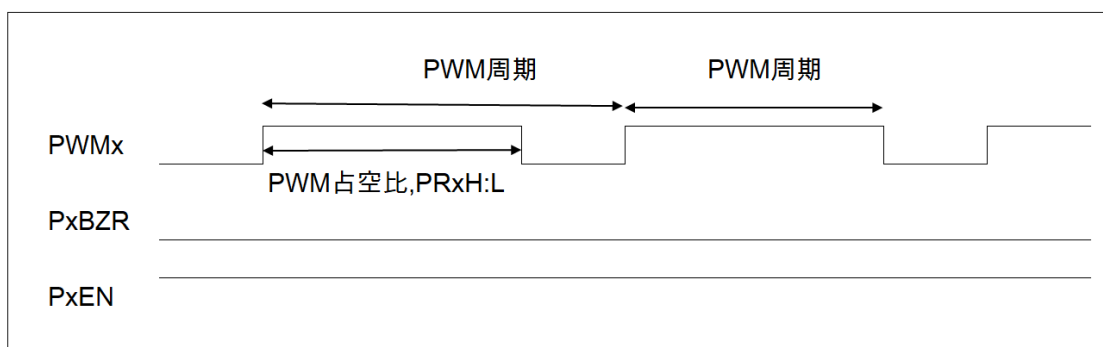


图 1-3 PWMx 工作模式(正向输出)

注意：

如果想同时使用 ECCP 的 PWM (P1A 管脚输出 PWM 调制) 和 PWM3/4/5, 则寄存器 CCP1CON 的 P1M<1:0>要配置为 00 (单输出模式 = P1A 调制, P1B/P1C/P1D 为端口)。即, ECCP 优先级比 PWM3/PWM4/PWM5 高。

PWMx 周期

PWMx 周期由 TIMERx 的预分频比设置寄存器 TMRxPS 和 PxPER 共同决定。

$$\text{公式 1.2} \quad \text{PWM 周期} = 2^{TMRxPS} * 2^{Nbit} * T_{PxCK}$$

- Nbit 为 PxPER 设定的定时器位数。

PWMx 占空比

PWM 的占空比由寄存器 PRxH:L 决定。PRxH:L 一共 12bit, 软件需要分开两次写。软件写入 PRxH:L 的值将立即生效, 将直接影响当前 PWMx 占空比, 建议在启动 TIMERx 之前把目标值写入 PRx。

$$\text{公式 1.3} \quad T_{pwm} = 2^{TMRxPS} * (PRx) * T_{PxCK}$$

PWMx 配置

1. 将相关的 TRIS 位置 1, 禁止 PWMx 引脚的输出驱动器
2. 写 PWMxCR0 寄存器值, 设置 PWMx 周期, 中断产生方式以及选择时钟源
3. 写 PWMxCR1 寄存器值, 配置为 PWM 模式, 和预分频比
4. 装载 PRx 寄存器, 设置 PWM 占空比
5. 配置并启动 TIMERx:
 - 将 PWMxCR1 寄存器的 TMRXIF 中断标志位清零
 - 将 PWMxCR1 寄存器的 TMRxON 位置 1 启动 TIMERx
6. 重新开始一个 PWM 周期后, 使能 PWM 输出:
 - 等待 TIMERx 溢出 (PWMxCR1 寄存器的 TMRxIF 位置 1)
 - 将相关的 TRIS 位清零使能 PWMx 引脚的输出驱动器
7. 同 TIMER 模式一样, 通过配置 (PxEN、PxBZR、TMRxON、TMRxIE、PEIE) 和“PxCKSRC = 010 或 011”, PWM 可在睡眠模式工作

注意: 如果 PWM 时钟设置为系统时钟 (PxCKSRC=000), 则系统时钟频率的任何改变将导致 PWM 频率的改变。

2. 应用范例

```

/* 文件名: TEST_61F02x_PWM345.C
* 功能:    FT61F02x-增强型 PWM 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/2T
* 说明:    此程序用来演示 PWM345 功能
*          演示程序中在 PWM345 脚分别输出 2kHz,1kHz,500Hz
*          占空比的为 50%的信号
*
*          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
* PWM4-----|5(PC3) (PA3)12|-----NC
* PWM5-----|6(PC2) (PC0)11|-----NC
* NC-----|7(PA4)  (PC1)10|-----NC
* NC-----|8(PC5)  (PC4)09|----PWM3
*          -----
*/
#include "SYSCFG.h"
//*****宏定义*****
//PWM 引脚输入输出控制
#define PWM3Dir    TRISC4
#define PWM4Dir    TRISC3
#define PWM5Dir    TRISC2
/*-----
* 函数名: interrupt
* 功能:    中断处理
* 输入:    无
* 输出:    无
*-----*/
void interrupt ISR(void)
{
    NOP();
}
/*-----
* 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 1-输入 0-输出
    PORTC = 0B00000000;
    TRISC = 0B00000000;      //PC 输入输出 1-输入 0-输出

    WPUA = 0B00000000;       //禁止所有 PA 口上拉
    WPUC = 0B00000000;       //禁止所有 PC 口上拉
    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: PWM_INITIAL
* 功能:   PWM3,4,5 初始化
* 设置 PWM3 周期=2^TMRXPS*2^PXPERR*[(T3CKDIV+1)/PWM 时钟源]
*          =2^0*2^8*[(30+1)/16000000]=496μs
*   PWM4 周期=2^TMRXPS*2^PXPERR*[(T4CKDIV+1)/PWM 时钟源]
*          =2^0*2^8*[(62+1)/16000000]=1.008ms
*   PWM5 周期=2^TMRXPS*2^PXPERR*[(T5CKDIV+1)/PWM 时钟源]
*          =2^0*2^8*[(124+1)/16000000]=2ms
*-----*/
void PWM_INITIAL (void)
{
    PWM3Dir = 1;              //PWM3 输出 PIN 暂为输入模式
    PWM4Dir = 1;              //PWM4 输出 PIN 暂为输入模式
    PWM5Dir = 1;              //PWM5 输出 PIN 暂为输入模式
    //-----PWM3-----
    PWM3CR0 = 0B00110010;
    //Bit7:   禁止中断
    //Bit[6:4]: 周期位选择 011-8 位
    //Bit[3:1]: 时钟选择 001-内部 RC 快时钟/(T3CKDIV+1)
    //Bit0:    PWM 输出

    PWM3CR1 = 0B10000000;
    //Bit7:    1-TMR3 为 PWM/BUZZER 模式
    //Bit6:    0-PWM3 为高电平有效
    //Bit[5:3]: 000-PWM3 预分频比设置为 1:1
}
```

```
//Bit2:    0-暂关 TMR3
//Bit1:    0-禁止 TMR3 中断
//Bit0:    0-TMR3 中断标志位只读
```

```
TMR3H=0;
T3CKDIV = 30;          //赋值 T3CKDIV
PR3L = 128;
//-----PWM4-----
PWM4CR0 = 0B00110010;
//Bit7:    禁止中断
//Bit[6:4]: 周期位选择 011-8 位
//Bit[3:1]: 时钟选择 001-内部 RC 快时钟/(T4CKDIV+1)
//Bit0:    PWM 输出
```

```
PWM4CR1 = 0B10000000;
//Bit7:    1-TMR4 为 PWM/BUZZER 模式
//Bit6:    0-PWM4 为高电平有效
//Bit[5:3]: 000-PWM4 预分频比设置为 1:1
//Bit2:    0-暂关 TMR4
//Bit1:    0-禁止 TMR4 中断
//Bit0:    0-TMR4 中断标志位只读
```

```
TMR4H=0;
T4CKDIV = 62;          //赋值 T4CKDIV
PR4L = 128;
//-----PWM5-----
PWM5CR0 = 0B00110010;
//Bit7:    禁止中断
//Bit[6:4]: 周期位选择 011-8 位
//Bit[3:1]: 时钟选择 001-内部 RC 快时钟/(T5CKDIV+1)
//Bit0:    PWM 输出
```

```
PWM5CR1 = 0B10000000;
//Bit7:    1-TMR5 为 PWM/BUZZER 模式
//Bit6:    0-PWM5 为高电平有效
//Bit[5:3]: 000-PWM5 预分频比设置为 1:1
//Bit2:    0-暂关 TMR5
//Bit1:    0-禁止 TMR5 中断
//Bit0:    0-TMR5 中断标志位只读
```

```
TMR5H=0;
T5CKDIV = 124;         //赋值 T5CKDIV
PR5L = 128;
```

```
}
```

```
/*-----  
* 函数名: main  
* 功能: 主函数  
* 输入: 无  
* 输出: 无  
-----*/  
  
void main(void)  
{  
    POWER_INITIAL();  
    PWM_INITIAL();  
  
    TMR3ON=1;  
    TMR4ON=1;  
    TMR5ON=1;  
    PWM3Dir = 0;           //PWM3 PIN 设为输出模式允许 PWM 输出  
    PWM4Dir = 0;           //PWM4 PIN 设为输出模式允许 PWM 输出  
    PWM5Dir = 0;           //PWM5 PIN 设为输出模式允许 PWM 输出  
    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 or 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.