

FT62F08X

Application note

目录

1. 通用定时器 TIM2	3
1.1. 特性	3
1.2. 原理框图	3
1.3. 功能描述	4
1.3.1. 计数基本单元	4
1.3.1.1. 时钟源选择	4
1.3.1.2. 向上计数器	5
1.3.1.3. 预分频器	5
1.3.2. 捕捉比较通道	6
1.3.2.1. 捕捉输入通道	6
1.3.2.2. 输出比较通道	6
1.3.3. TIM2 中断	7
1.4. 与 Timer2 相关寄存器汇总	7
1.4.1. TIM2CR1, 地址 0x30C	8
1.4.2. TIM2IER, 地址 0x30D	8
1.4.3. TIM2SR1, 地址 0x30E	9
1.4.4. TIM2SR2, 地址 0x30F	10
1.4.5. TIM2EGR, 地址 0x310	10
1.4.6. TIM2CCMR1, 地址 0x311	11
1.4.7. TIM2CCMR2, 地址 0x312	12
1.4.8. TIM2CCMR3, 地址 0x313	13
1.4.9. TIM2CCER1, 地址 0x314	14
1.4.10. TIM2CCER2, 地址 0x315	15
1.4.11. TIM2CNTRH, 地址 0x316	15
1.4.12. TIM2CNTRL, 地址 0x317	15
1.4.13. TIM2PSCR, 地址 0x318	15
1.4.14. TIM2ARRH, 地址 0x319	16
1.4.15. TIM2ARRL, 地址 0x31A	16
1.4.16. TIM2CCR1H, 地址 0x31B	16
1.4.17. TIM2CCR1L, 地址 0x31C	17
1.4.18. TIM2CCR2H, 地址 0x31D	17
1.4.19. TIM2CCR2L, 地址 0x31E	17
1.4.20. TIM2CCR3H, 地址 0x29E	17
1.4.21. TIM2CCR3L, 地址 0x29F	18
2 应用范例	18

62F08x_TIM2_CAPTURE 的应用

1. 通用定时器 TIM2

1.1. 特性

tiemr2 的功能除捕捉比较通道数量不同以外，其他相同：

- 16bit 的向上计数，支持自动重载；
- 计数时钟预分频；
- 支持 1/2 个独立的捕捉比较通道，通道可支持：
 - 输入捕捉
 - 输出比较
 - PWM 产生
- 中断事件：
 - 更新事件：计数器溢出，计数器初始化
 - 输入捕捉事件
 - 输出比较事件

1.2. 原理框图

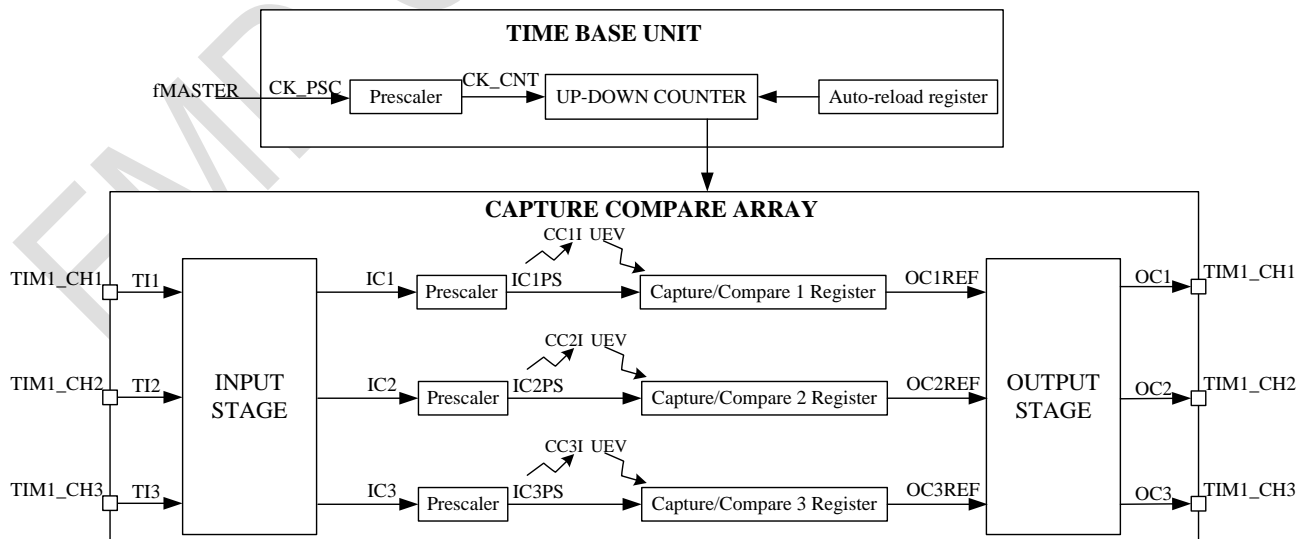


图 11.1 Timer2 原理框图

1.3. 功能描述

整个 Timer2 可以分为两个大的功能部分：计数基本单元和捕捉比较通道。计数基本单元分为向上计数器、自动加载寄存器、预分频器；捕捉比较通道分为捕捉输入通道，输出比较通道和输出控制。

1.3.1. 计数基本单元

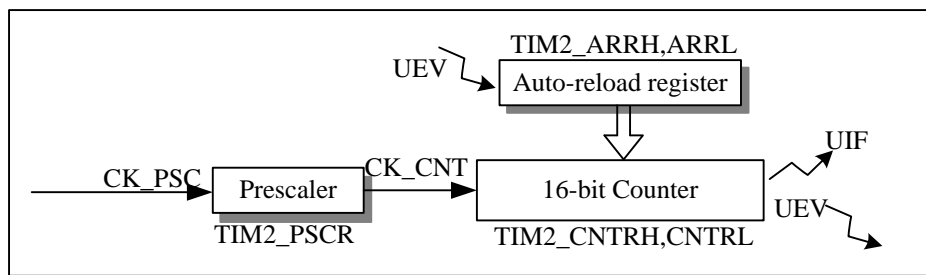


图 11.2 计数基本单元框图

计数基本单元包括：

- 16 位向上计数器
- 16 位自动重加载寄存器
- 4 位可编程预分频器

Timer2 没有重复计数器

1.3.1.1. 时钟源选择

时钟源可由 TCKSRC 寄存器进行配置：

- T2CKSRC[2:0]=000 时，系统时钟/主时钟为 Timer2 时钟
- T2CKSRC[2:0]=001 时，HIRC 为 Timer2 时钟
- T2CKSRC[2:0]=010 时，XT 时钟/外部时钟为 Timer2 时钟
- T2CKSRC[2:0]=011 时，HIRC 的 2 倍频为 Timer2 时钟
- T2CKSRC[2:0]=100 时，XT 时钟/外部时钟的 2 倍频为 Timer2 时钟
- T2CKSRC[2:0]=101 时，LIRC 为 Timer2 时钟
- T2CKSRC[2:0]=110 时，LP 时钟/外部时钟为 Timer2 时钟
- T2CKSRC[2:0]=111 时，LP 时钟/外部时钟的 2 倍频为 Timer2 时钟

1.3.1.2. 向上计数器

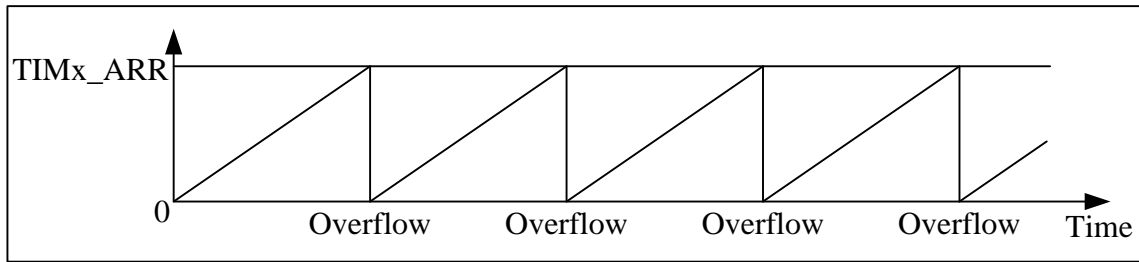


图 11.3 向上计数器

Timer2 计数器只能向上计数。计数器从 0 开始计数向上计数，计到 TIM1_ARR 寄存器所设数值。然后重新从 0 开始计数并产生一个计数器上溢事件；如果 T2UDIS 设为 0，那么还会产生一个更新事件 UEV。

1.3.1.3. 预分频器

计数时钟可以进行 4bit 的时钟预分频：

$$f_{CK_CNT} = f_{CK_PSC} / 2^{(PSCR[3:0])}$$

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 T2CEN 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。

1.3.2. 捕捉比较通道

1.3.2.1. 捕捉输入通道

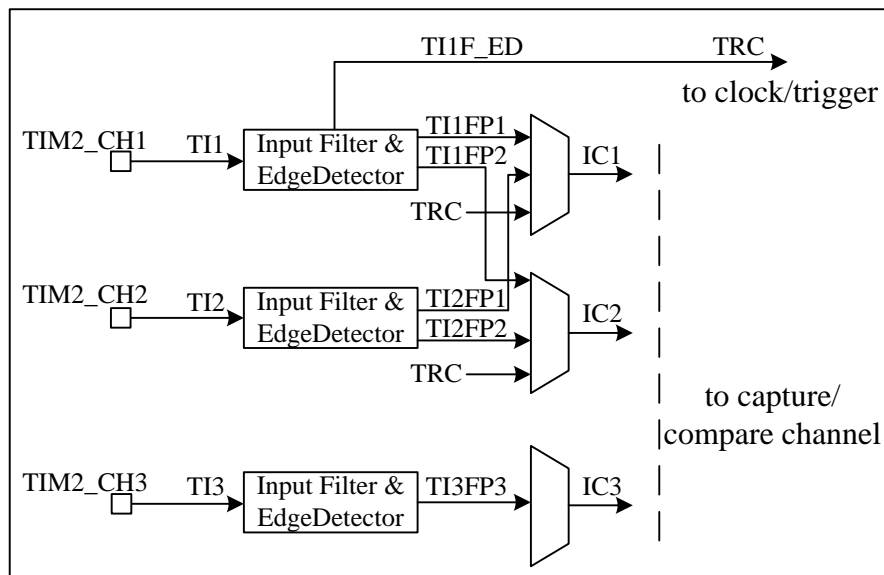


图 11.4 输入通道框图

1.3.2.2. 输出比较通道

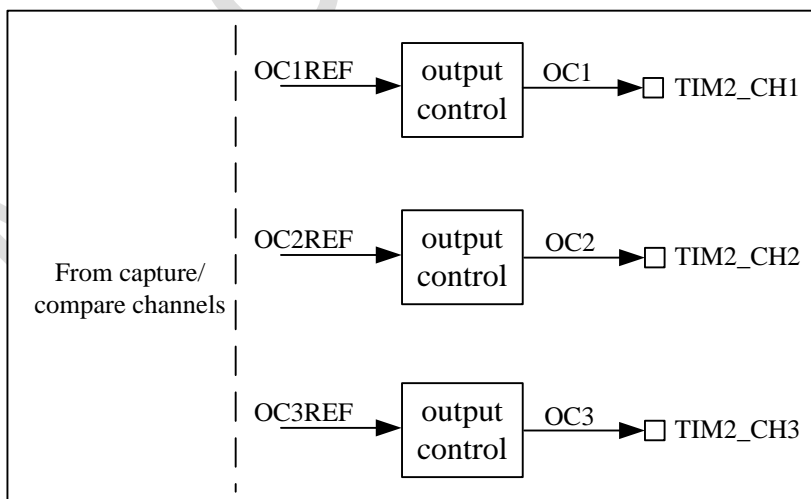


图 11.5 输出通道框图

Timer2 的输出没有死区功能，没有互补输出功能，也没有刹车功能。

1.3.3. TIM2 中断

Timer2 有以下 4 个中断请求源：

- 捕捉/比较 3 中断
- 捕捉/比较 2 中断
- 捕捉/比较 1 中断
- 更新中断

在用这些中断之前需要提前打开 TIM2_IER 寄存器中的中断使能位(T2CCiIE 和 T2UIE)。

不同的中断源还可以配置通过 TIM2_EGR 寄存器来产生(软件产生中断)。

1.4. 与 Timer2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TIM2CR1	0x30C	T2ARPE	—	—	—	T2OPM	T2URS	T2UDIS	T2CEN	0--- 0000
TIM2IER	0x30D	—	—	—	—	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
TIM2SR1	0x30E	—	—	—	—	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
TIM2SR2	0x30F	—	—	—	—	T2CC3OF	T2CC2OF	T2CC1OF	—	---- 0000
TIM2EGR	0x310	—	—	—	—	T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
TIM2CCMR1 (output mode)	0x311	—	T2OC1M[2:0]			T2OC1PE	—	T2CC1S[1:0]		-000 0-00
TIM2CCMR1 (input mode)		T2IC1F[3:0]			T2IC1PSC[1:0]		T2CC1S[1:0]		0000 0000	
TIM2CCMR2 (output mode)	0x312	—	T2OC2M[2:0]			T2OC2PE	—	T2CC2S[1:0]		-000 0-00
TIM2CCMR2 (input mode)		T2IC2F[3:0]			T2IC2PSC[1:0]		T2CC2S[1:0]		0000 0000	
TIM2CCMR3 (output mode)	0x313	—	T2OC2M[2:0]			OC3PE	—	T2CC3S[1:0]		-000 0-00
TIM2_CCMR3 (input mode)		T2IC3F[3:0]			T2IC3PSC[1:0]		T2CC3S[1:0]		0000 0000	
TIM2CCER1	0x314	—	—	T2CC2P	T2CC2E	—	—	T2CC1P	T2CC1E	--00 --00
TIM2CCER2	0x315	—	—	—	—	—	—	T2CC3P	T2CC3E	---- --00
TIM2CNTRH	0x316	T2CNT[15:8]								0000 0000
TIM2CNTRL	0x317	T2CNT[7:0]								0000 0000
TIM2PSCR	0x318	—	—	—	—	T2PSC[3:0]				---- 0000
TIM2ARRH	0x319	T2ARR[15:8]								1111 1111
TIM2ARRL	0x31A	T2ARR[7:0]								1111 1111
TIM2CCR1H	0x31B	T2CCR1[15:8]								0000 0000

TIM2CCR1L	0x31C	T2CCR1[7:0]	0000 0000
TIM2CCR2H	0x31D	T2CCR2[15:8]	0000 0000
TIM2CCR2L	0x31E	T2CCR2[7:0]	0000 0000
TIM2CCR3H	0x29E	T2CCR3[15:8]	0000 0000
TIM2CCR3L	0x29F	T2CCR3[7:0]	0000 0000

1.4.1. TIM2CR1, 地址 0x30C

Bit	7	6	5	4	3	2	1	0
Name	T2ARPE	reversed			T2OPM	T2URS	T2UDIS	T2CEN
Reset	0	—	—	—	0	0	0	0
Type	RW	RO-0	RO-0	RO-0	RW	RW	RW	RW
7	T2ARPE: 自动预装载允许位 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。							
3	T2OPM: 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。							
2	T2URS: 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)							
1	T2UDIS: 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、CCR _x)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。							
0	T2CEN: 允许计数器 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了CEN位后, 外部时钟、门控模式和编码器模式才能工作。然而触发模式可以自动地通过硬件设置CEN位。							

1.4.2. TIM2IER, 地址 0x30D

Bit	7	6	5	4	3	2	1	0
Name	reversed				T2CC3IE	T2CC2IE	T2CC1IE	T2UIE

Reset	—	—	—	—	0	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	R-W0	R-W0	R-W0	R-W0
3	T2CC3IE : 允许捕获/比较3中断 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。							
2	T2CC2IE : 允许捕获/比较2中断 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。							
1	T2CC1IE : 允许捕获/比较1中断 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。							
0	T2UIE : 允许更新中断 0: 禁止更新中断; 1: 允许更新中断。							

1.4.3. TIM2SR1, 地址 0x30E

Bit	7	6	5	4	3	2	1	0
Name	reversed				T2CC3IF	T2CC2IF	T2CC1IF	T2UIF
Reset	—	—	—	—	0	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	R-W0	R-W0	R-W0	R-W0
3	T2CC3IF : 捕获/比较3中断标记 参考CC1IF描述。							
2	T2CC2IF : 捕获/比较2中断标记 参考CC1IF描述。							
1	T2CC1IF : 捕获/比较1中断标记 如果通道CC1配置为输出模式: 当计数器值与比较值匹配时该位由硬件置1, 但在中心对称模式下除外(参考TIM1_CR1寄存器的CMS位)。它由软件清0。 0: 无匹配发生; 1: TIMx_CNT的值与TIMx_CCR1的值匹配。 注: 在中心对称模式下, 当计数器值为0时, 向上计数, 当计数器值为ARR时, 向下计数(它从0向上计数到ARR-1, 再由ARR向下计数到1)。因此, 对所有的SMS位值, 这两个值都不置标记。但是, 如果CCR1>ARR, 则当CNT达到ARR值时, CC1IF置1。 如果通道CC1配置为输入模式: 当捕获事件发生时该位由硬件置1, 它由软件清0或通过读TIM1_CCR1L清0。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至TIM1_CCR1(在IC1上检测到与所选极性相同的边沿)。							
0	T2UIF : 更新中断标记 当产生更新事件时该位由硬件置1。它由软件清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TIM1_CR1寄存器的UDIS=0, 当计数器上溢或下溢时; 若TIM1_CR1寄存器的UDIS=0、URS=0, 当设置TIM1_EGR寄存器的UG位软件对计数器CNT重新初始化时; 若TIM1_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 (参考0从模式控制寄存器TIM1_SMCR)。							

1.4.4. TIM2SR2, 地址 0x30F

Bit	7	6	5	4	3	2	1	0
Name	reversed				T2CC3OF	T2CC2OF	T2CC1OF	reserved
Reset	—	—	—	—	0	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	R-W0	R-W0	R-W0	R-W0
3	T2CC3OF: 捕获/比较3重复捕获标记 参见CC1OF描述。							
2	T2CC2OF: 捕获/比较2重复捕获标记 参见CC1OF描述。							
1	T2CC1OF: 捕获/比较1重复捕获标记 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置1。写0可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到TIM1_CCR1寄存器时, CC1IF的状态已经为1。							

1.4.5. TIM2EGR, 地址 0x310

Bit	7	6	5	4	3	2	1	0
Name	reversed				T2CC3G	T2CC2G	T2CC1G	T2UG
Reset	—	—	—	—	0	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	RW	RW	RW	RW
3	T2CC3G: 产生捕获/比较3事件 参考CC1G描述。							
2	T2CC2G: 产生捕获/比较2事件 参考CC1G描述。							
1	T2CC1G: 产生捕获/比较1事件 该位由软件置1, 用于产生一个捕获/比较事件, 由硬件自动清0。 0: 无动作; 1: 在通道CC1上产生一个捕获/比较事件: 若通道CC1配置为输出: 设置CC1IF=1, 若开启对应的中断, 则产生相应的中断。若通道CC1配置为输入: 当前的计数器值被捕获至TIM1_CCR1寄存器, 设置CC1IF=1, 若开启对应的中断, 则产生相应的中断。若CC1IF已经为1, 则设置CC1OF=1。							
0	T2UG: 产生更新事件 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。							

1.4.6. TIM2CCMR1, 地址 0x311

Bit	7	6	5	4	3	2	1	0
Name	reversed	T2OC1M[2:0]			T2OC1PE	T2OC1FE	T2CC1S[1:0]	
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW
6:4	<p>T2OC1M[2:0]: 输出比较1模式</p> <p>该3位定义了输出参考信号OC1REF的动作, 而OC1REF决定了OC1的值。OC1REF是高电平有效, 而OC1的有效电平取决于CC1P位。</p> <p>000: 冻结。输出比较寄存器TIM1_CCR1与计数器TIM1_CNT间的比较对OC1REF不起作用;</p> <p>001: 匹配时设置通道1的输出为有效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为高。</p> <p>010: 匹配时设置通道1的输出为无效电平。当计数器TIM1_CNT的值与捕获/比较寄存器1(TIM1_CCR1)相同时, 强制OC1REF为低。</p> <p>011: 翻转。当TIM1_CCR1=TIM1_CNT时, 翻转OC1REF的电平。</p> <p>100: 强制为无效电平。强制OC1REF为低。</p> <p>101: 强制为有效电平。强制OC1REF为高。</p> <p>110: PWM模式1— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为有效电平, 否则为无效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM模式2— 在向上计数时, 一旦TIM1_CNT<TIM1_CCR1时通道1为无效电平, 否则为有效电平; 在向下计数时, 一旦TIM1_CNT>TIM1_CCR1时通道1为有效电平, 否则为无效电平。</p> <p>注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出) 则该位不能被修改。</p> <p>注2: 在PWM模式1或PWM模式2中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时, OC1REF电平才改变。(参考17.5.7PWM模式)</p> <p>注3: 在有互补输出的通道上, 这些位是预装载的。如果TIM1_CR2寄存器的CCPC=1, OCM 位只有在COM事件发生时, 才从预装载位取新值。</p>							
3	<p>T2OC1PE: 输出比较1预装载使能</p> <p>0: 禁止TIM1_CCR1寄存器的预装载功能, 可随时写入TIM1_CCR1寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启TIM1_CCR1寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注1: 一旦LOCK级别设为3(TIM1_BKR寄存器中的LOCK位)并且CC1S=00(该通道配置成输出) 则该位不能被修改。</p> <p>注2: 为了操作正确, 在PWM模式下必须使能预装载功能。但在单脉冲模式下(TIM1_CR1寄存器的OPM=1), 它不是必须的。</p>							
2	<p>T2OC1FE: 输出比较1 快速使能</p> <p>该位用于加快CC输出对触发输入事件的响应。</p> <p>0: 根据计数器与CCR1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。</p> <p>1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。</p> <p>OCFE只在通道被配置成PWM1或PWM2模式时起作用。</p>							
1:0	<p>T2CC1S[1:0]: 捕获/比较1 选择。</p> <p>这2位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1FP1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2FP1上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。</p>							

注：CC1S仅在通道关闭时(TIM1_CCER1寄存器的CC1E=0)才是可写的。

Name	T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T2IC1F[3:0]: 输入捕获1滤波器 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，只有发生了N个事件后输出的跳变才被认为有效。 0000: 无滤波器, fSAMPLING=fMASTER 1000: 采样频率fSAMPLING=fMASTER/8, N=6 0001: 采样频率fSAMPLING=fMASTER, N=2 1001: 采样频率fSAMPLING=fMASTER/8, N=8 0010: 采样频率fSAMPLING=fMASTER, N=4 1010: 采样频率fSAMPLING=fMASTER/16, N=5 0011: 采样频率fSAMPLING=fMASTER, N=8 1011: 采样频率fSAMPLING=fMASTER/16, N=6 0100: 采样频率fSAMPLING=fMASTER/2, N=6 1100: 采样频率fSAMPLING=fMASTER/16, N=8 0101: 采样频率fSAMPLING=fMASTER/2, N=8 1101: 采样频率fSAMPLING=fMASTER/32, N=5 0110: 采样频率fSAMPLING=fMASTER/4, N=6 1110: 采样频率fSAMPLING=fMASTER/32, N=6 0111: 采样频率fSAMPLING=fMASTER/4, N=8 1111: 采样频率fSAMPLING=fMASTER/32, N=8 注：即使对于带互补输出的通道，该位域也是非预装载的，并且不会考虑CCPC (TIM1_CR2 寄存器) 的值。							
3:2	T2IC1PSC[1:0]: 输入/捕获1预分频器 这2位定义了CC1输入(IC1)的预分频系数。 一旦CC1E=0(TIM1_CCER寄存器中)，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01: 每2个事件触发一次捕获； 10: 每4个事件触发一次捕获； 11: 每8个事件触发一次捕获。							
1:0	T2CC1S[1:0]: 捕获/比较1 选择。 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC1通道被配置为输出； 01: CC1通道被配置为输入，IC1映射在TI1FP1上； 10: CC1通道被配置为输入，IC1映射在TI2FP1上； 11: CC1通道被配置为输入，IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。 注：CC1S仅在通道关闭时(TIM1_CCER1寄存器的CC1E=0)才是可写的。							

1.4.7. TIM2CCMR2, 地址 0x312

Bit	7	6	5	4	3	2	1	0
Name	reversed	T2OC2M[2:0]			T2OC2PE	T2OC2FE	T2CC2S[1:0]	
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW
6:4	T2OC2M[2:0]: 输出比较2模式							
3	T2OC2PE: 输出比较2预装载使能							
2	T2OC1FE: 输出比较2快速使能							
1:0	T2CC2S[1:0]: 捕获/比较2选择。							

	该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2FP2上; 10: CC2通道被配置为输入, IC2映射在TI1FP2上; 11: 预留 注: CC2S仅在通道关闭时(TIM1_CCER1寄存器的CC2E=0, CC2NE=0且已被更新)才是可写的。
--	--

Name	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T2IC2F[3:0]: 输入捕获2滤波器							
3:2	T2IC2PSC[1:0]: 输入/捕获2预分频器							
1:0	T2CC2S[1:0]: 捕获/比较2选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2FP2上; 10: CC2通道被配置为输入, IC2映射在TI1FP2上; 11: CC2通道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIM1_SMCR寄存器的TS位选择)。 注: CC2S仅在通道关闭时(TIM1_CCER1寄存器的CC2E=0, CC2NE=0且已被更新)才是可写的。							

1.4.8. TIM2CCMR3, 地址 0x313

Bit	7	6	5	4	3	2	1	0
Name	reversed	T2OC3M[2:0]			T2OC3PE	T2OC3FE	T2CC3S[1:0]	
Reset	—	0	0	0	0	0	0	0
Type	RO-0	RW	RW	RW	RW	RW	RW	RW
6:4	T2OC3M[2:0]: 输出比较3模式							
3	T2OC3PE: 输出比较3预装载使能							
2	T2OC3FE: 输出比较3快速使能							
1:0	T2CC3S[1:0]: 捕获/比较3选择。 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3通道被配置为输出; 01: CC3通道被配置为输入, IC3映射在TI3FP3上; 10: CC3通道被配置为输入, IC3映射在TI4FP3上; 11: 预留 注: CC3S仅在通道关闭时(TIM1_CCER2寄存器的CC3E=0, CC3NE=0且已被更新)才是可写的。							

Name	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]	
Reset	0	0	0	0	0	0	0	0

Type	RW	RW	RW	RW	RW	RW	RW	RW
7:4	T2IC3F[3:0]: 输入捕获3滤波器							
3:2	T2IC3PSC[1:0]: 输入/捕获3预分频器							
1:0	T2CC3S[1:0]: 捕获/比较3选择。 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3通道被配置为输出; 01: CC3通道被配置为输入, IC3映射在TI3FP3上; 10: CC3通道被配置为输入, IC3映射在TI4FP3上; 11: 预留 注: CC3S仅在通道关闭时(TIM1_CCER2寄存器的CC3E=0, CC3NE=0且已被更新)才是可写的。							

1.4.9. TIM2CCER1, 地址 0x314

Bit	7	6	5	4	3	2	1	0
Name	reversed		T2CC2P	T2CC2E	reversed		T2CC1P	T2CC1E
Reset	—	—	0	0	—	—	0	0
Type	RO-0	RO-0	RW	RW	RO-0	RO-0	RW	RW
5	T2CC2P: 输入捕获/比较2输出极性。参考CC1P的描述。							
4	T2CC2E: 输入捕获/比较2输出使能。参考CC1E的描述。							
1	T2CC1P: 输入捕获/比较1输出极性CC1通道配置为输出: 0: OC1高电平有效; 1: OC1低电平有效。 CC1通道配置为触发(参考图61): 0: 触发发生在TI1F的高电平或上升沿; 1: 触发发生在TI1F的低电平或下降沿。 CC1通道配置为输入(参考图61): 0: 捕捉发生在TI1F的高电平或上升沿; 1: 捕捉发生在TI1F的低电平或下降沿。 注1: 一旦LOCK级别(TIM1_BKR寄存器中的LCKCK位)设为3或2, 则该位不能被修改。 注2: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1 (TIM1_CR2寄存器), 只有在COM事件发生时, CC1P位才从预装载位中取新值。							
0	T2CC1E: 输入捕获/比较1输出使能 CC1通道配置为输出: 0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 CC1通道配置为输入: 该位决定了计数器的值是否能捕获入TIM1_CCR1寄存器。 0: 捕获禁止; 0: 捕获使能。 注: 对于有互补输出的通道, 该位是预装载的。如果CCPC=1(TIM1_CR2寄存器), 只有在COM事件发生时, CC1E位才从预装载位中取新值。							

1.4.10. TIM2CCER2, 地址 0x315

Bit	7	6	5	4	3	2	1	0
Name	reserved		T2CC4P	T2CC4E	reserved		T2CC3P	T2CC3E
Reset	—	—	0	0	—	—	0	0
Type	RO-0	RO-0	RW	RW	RO-0	RO-0	RW	RW
5	T2CC4P : 输入捕获/比较4输出极性。参考CC1P的描述。							
4	T2CC4E : 输入捕获/比较4输出使能。参考CC1E 的描述。							
1	T2CC3P : 输入捕获/比较3输出极性。参考CC1P的描述。							
0	T2CC3E : 输入捕获/比较3输出使能。参考CC1E 的描述。							

1.4.11. TIM2CNTRH, 地址 0x316

Bit	7	6	5	4	3	2	1	0
Name	T2CNT[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CNT[15:8] : 计数器的高8位值							

1.4.12. TIM2CNTRL, 地址 0x317

Bit	7	6	5	4	3	2	1	0
Name	T2CNT[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CNT[7:0] : 计数器的低8位值							

1.4.13. TIM2PSCR, 地址 0x318

Bit	7	6	5	4	3	2	1	0
Name	reversed				T2PSC[7:0]			
Reset	—	—	—	—	0	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	RW	RW	RW	RW
3:0	T2PSC[3:0] : 预分频器的值 预分频器对输入的CK_PSC时钟进行分频。 计数器的时钟频率 f_{CK_CNT} 等于 $f_{CK_PSC}/2^{(PSC[3:0])}$ 。PSC[7:4]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。
--	---

1.4.14. TIM2ARRH, 地址 0x319

Bit	7	6	5	4	3	2	1	0
Name	T2ARR[15:8]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2ARR[15:8]: 自动重装载的高8位值 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。							

1.4.15. TIM2ARRL, 地址 0x31A

Bit	7	6	5	4	3	2	1	0
Name	T2ARR[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2ARR[7: 0]: 自动重装载的低8位值 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。							

1.4.16. TIM2CCR1H, 地址 0x31B

Bit	7	6	5	4	3	2	1	0
Name	T2CCR1[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CCR1[15:8]: 捕获/比较1的高8位值 若CC1通道配置为输出(TIM1_CCMR1的CC1S位): CCR1包含了装入当前捕获/比较1寄存器的值(预装载值)。 如果在TIM1_CCMR1寄存器(OC1PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC1端口上产生输出信号。若CC1通道配置为输入: CCR1包含了上一次输入捕获1事件(IC1)发生时的计数器值(此时该寄存器为只读)。							

1.4.17. TIM2CCR1L, 地址 0x31C

Bit	7	6	5	4	3	2	1	0
Name	T2CCR1[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CCR1[7:0]: 捕获/比较1的低8位值							

1.4.18. TIM2CCR2H, 地址 0x31D

Bit	7	6	5	4	3	2	1	0
Name	T2CCR2[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CCR2[15:8]: 捕获/比较2的高8位值 若CC2通道配置为输出(TIM1_CCMR2的CC2S位): CCR2包含了装入当前捕获/比较2寄存器的值(预装载值)。 如果在TIM1_CCMR2寄存器(OC2PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。 当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC2端口上产生输出信号。若CC2通道配置为输入: CCR2包含了由上一次输入捕获2事件(IC2)传输的计数器值(此时该寄存器为只读)。							

1.4.19. TIM2CCR2L, 地址 0x31E

Bit	7	6	5	4	3	2	1	0
Name	T2CCR2[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CCR2[7:0]: 捕获/比较1的低8位值							

1.4.20. TIM2CCR3H, 地址 0x29E

Bit	7	6	5	4	3	2	1	0
Name	T2CCR3[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

7:0	<p>T2CCR3[15:8]: 捕获/比较3的高8位值</p> <p>若CC3通道配置为输出(TIM1_CCMR3的CC3S位):</p> <p>CCR3包含了装入当前捕获/比较3寄存器的值(预装载值)。</p> <p>如果在TIM1_CCMR3寄存器(OC3PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。</p> <p>当前捕获/比较寄存器的值同计数器TIM1_CNT的值相比较, 并在OC3端口上产生输出信号。若CC3通道配置为输入:</p> <p>CCR3包含了由上一次输入捕获3事件(IC3)传输的计数器值(此时该寄存器为只读)。</p>
-----	---

1.4.21. TIM2CCR3L, 地址 0x29F

Bit	7	6	5	4	3	2	1	0
Name	T2CCR3[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW
7:0	T2CCR3[7:0]: 捕获/比较3的低8位值							

2 应用范例

```

//*****
/* 文件名: TEST_62F08x_TIM2_CAPTURE.c
* 功能:   FT62F08x-TIM2_CAPTURE 功能演示
* IC:     FT62F088 LQFP32
* 内部:   16M
* empno: 500
* 说明:   例程通过 TIM2 的捕获通道 1 测量 TIM4 输出到 RB3 的波形的周期 (4K)
*         TIM2 计数器捕获的数据放在 testdata 数组里, 一共 20 个。
*         程序只用到一个数据(testdata[1])来计算周期。
*
*
*
* 参考原理图 TEST_62F08x_sch.pdf
*/
//*****
#include "SYSCFG.h"
//*****
//*****宏定义*****
#define unchar    unsigned char
#define uint      unsigned int
#define ulong     unsigned long

#define DemoPortOut RB3

```

```
#define DemoPortOut1    RB4

//volatile unchar mydata; //全局查看变量定义
volatile uint testdata[20];
volatile unchar y=0;
volatile unchar dataH;
volatile unchar dataL;
volatile uint period=0;
/*-----
 * 函数名: 中断
 * 功能:
 * 输入: 无
 * 输出: 无
-----*/
void interrupt ISR(void)
{

    //定时器 4 的中断处理*****
    if(T4UIE && T4UIF)
    {
        T4UIF = 1;                //写 1 清零标志位
        DemoPortOut = ~DemoPortOut; //翻转电平
    }

    //定时器 2 的中断处理*****
    if(T2CC1IE && T2CC1IF)        //捕获中断
    {
        T2CC1IF = 1;            //写 1 清零标志位

        TIM2CNTRH =0x00;
        TIM2CNTRL =0x00;

        DemoPortOut1 = ~DemoPortOut1;    //翻转电平

        dataH = TIM2CCR1H;
        dataL = TIM2CCR1L;

        testdata[y++] =(uint)(dataH<<8 | dataL);
        NOP();

        if(y>=20)
        {
            y=0;
        }
    }
}
```

```
//计算周期(us)
period = testdata[1]/32;    //周期约等于 250us
    }
}
}
/*-----
* 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;    //WDT 32KHZ IRCF=111=16MHZ
                           //Bit0=1,系统时钟为内部振荡器
                           //Bit0=0,时钟源由 FOSC<2: 0>决定即编译选项时选择

    INTCON = 0;             //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B11111111;    //PA 输入输出 0-输出 1-输入
    PORTB = 0B00000000;
    TRISB = 0B11100111;    //PB 输入输出 0-输出 1-输入
    PORTC = 0B00000000;
    TRISC = 0B11111111;    //PC 输入输出 0-输出 1-输入
    PORTD = 0B00000000;
    TRISD = 0B11111111;    //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 端口上拉控制 1-开下拉 0-关下拉
    WPDB = 0B00000000;    //PB 端口上拉控制 1-开下拉 0-关下拉
    WPDC = 0B00000000;    //PC 端口上拉控制 1-开下拉 0-关下拉
    WPDD = 0B00000000;    //PD 端口上拉控制 1-开下拉 0-关下拉

    PSRC0 = 0B11111111;    //PORTA,PORTB 源电流设置最大
    //BIT7~BIT6:PORTB[7:4]源电流能力控制,BIT5~BIT4:PORTB[3:0]源电流能力控制
    //BIT3~BIT2:PORTA[7:4]源电流能力控制,BIT1~BIT0:PORTA[3:0]源电流能力控制
```

```
PSRC1 = 0B11111111;    //PORTC,PORTD 源电流设置最大
//BIT7~BIT6:PORTD[7:4]源电流能力控制,BIT5~BIT4:PORTD[3:0]源电流能力控制
//BIT3~BIT2:PORTC[7:4]源电流能力控制,BIT1~BIT0:PORTC[3:0]源电流能力控制
```

```
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: DelayUs
* 功能:    短延时函数 --16M-2T--大概快 1%左右.
* 输入参数: Time 延时时间长度 延时时长 Time Us
* 返回参数: 无
```

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

```
void DelayUs(unsigned char Time)
```

```
{
```

```
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
```

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

```
* 函数名称: DelayMs
* 功能:    短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time ms
* 返回参数: 无
```

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

```
void DelayMs(unsigned char Time)
```

```
{
```

```
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<5;b++)
        {
            DelayUs(197); //快 1%
        }
    }
}
```

```
}
```

```

/*-----
* 函数名称: DelayS
* 功能: 短延时函数
* 输入参数: Time 延时时间长度 延时时长 Time S
* 返回参数: 无
-----*/

```

```
void DelayS(unsigned char Time)
```

```

{
    unsigned char a,b;
    for(a=0;a<Time;a++)
    {
        for(b=0;b<10;b++)
        {
            DelayMs(100);
        }
    }
}

```

```

/*-----
* 函数名称: Time4Initial
* 功能:
* 输入参数:
* 返回参数: 无
-----*/

```

```
void Time4Initial(void)
```

```

{
    PCKEN |= 0B00001000;    //TIME4 模块时钟使能

    TIM4CR1 = 0B00000001;
    //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
    //BIT5~BIT4: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:
    //      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:
    //      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:
//      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。

TIM4ISR=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

}

/*-----

* 函数名称: Time2Initial

* 功能:

* 输入参数:

* 返回参数: 无

-----*/

void Time2Initial(void)

{

PCKEN |=0B00000100; //使能 timer2 时钟模块

CKOCON=0B00100000;

TCKSRC=0B00110000; //TIM2 时钟为 HIRC 的 2 倍频

//BIT7 低频内振模式: 1 = 256K 振荡频率模式,0 = 32K 振荡频率模式

//BIT6~BIT4TIM2 时钟源选择位

//值 时钟源

//0 系统时钟/主时钟

//1 HIRC

//2 XT 时钟/外部时钟

//3 HIRC 的 2 倍频

//4 XT 时钟/外部时钟的 2 倍频

//5 LIRC

//6 LP 时钟/外部时钟

//7 LP 时钟/外部时钟的 2 倍频

//BIT3:保留位

//BIT2~BIT1:TIM2 时钟源选择位

//值 时钟源

//0 系统时钟/主时钟

//1 HIRC

//2 XT 时钟/外部时钟

//3 HIRC 的 2 倍频

//4 XT 时钟/外部时钟的 2 倍频

//5 LIRC

//6 LP 时钟/外部时钟
//7 LP 时钟/外部时钟的 2 位频

TIM2CR1 = 0B10000110; //预载允许, 边沿对齐向上计数器, 计数器使能

//BIT7:自动预装载允许位

//0: TIM2_ARR 寄存器没有缓冲, 它可以被直接写入;

//1: TIM2_ARR 寄存器由预装载缓冲器缓冲。

//BIT6~BIT5:选择对齐模式

//00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。

//01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIM2_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 只在计数器向下计数时被置 1。

//10:中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIM2_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 只在计数器向上计数时被置 1。

//11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIM2_CCMRx 寄存器中 CciS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被置 1。

//BIT4:方向

//0: 计数器向上计数;

//1: 计数器向下计数。

//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、CCR_x)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 允许计数器

//0: 禁止计数器;

//1: 使能计数器。

TIM2IER = 0B00000010; //允许捕获/比较 1 中断

//BIT7: 允许刹车中断

//0: 禁止刹车中断;

//1: 允许刹车中断。

//BIT6: 触发中断使能

//0: 禁止触发中断;

//1: 使能触发中断。

//BIT5: 允许 COM 中断

//0: 禁止 COM 中断;

//1: 允许 COM 中断。

//BIT4: 允许捕获/比较 4 中断

//0: 禁止捕获/比较 4 中断;

//1: 允许捕获/比较 4 中断。

//BIT3: 允许捕获/比较 3 中断

//0: 禁止捕获/比较 3 中断;

//1: 允许捕获/比较 3 中断。

//BIT2: 允许捕获/比较 2 中断

//0: 禁止捕获/比较 2 中断;

//1: 允许捕获/比较 2 中断。

//BIT1: 允许捕获/比较 1 中断

//0: 禁止捕获/比较 1 中断;

//1: 允许捕获/比较 1 中断。

//BIT0: 允许更新中断

//0: 禁止更新中断;

//1: 允许更新中断。

TIM2SR1 = 0B00000000;

//BIT7: 刹车中断标记,一旦刹车输入有效,由硬件对该位置 1。如果刹车输入无效,则该位可由软件清 0。

//0: 无刹车事件产生;

//1: 刹车输入上检测到有效电平。

//BIT6: 触发器中断标记,当发生触发事件(当从模式控制器处于除门控模式外的其它模式时,在 TRGI 输入端检测到有效边沿,或门控模式下的任一边沿)时由硬件对该位置 1。它由软件清 0。

//0: 无触发器事件产生;

//1: 触发中断等待响应。

//BIT5: COM 中断标记,一旦产生 COM 事件(当捕获/比较控制位: CciE、CciNE、OciM 已被更新)该位由硬件置 1。它由软件清 0。

//0: 无 COM 事件产生;

//1: COM 中断等待响应。

//BIT4: 捕获/比较 4 中断标记

//参考 CC1IF 描述。

//BIT3: 捕获/比较 3 中断标记

//参考 CC1IF 描述。

//BIT2: 捕获/比较 2 中断标记

//参考 CC1IF 描述。

//BIT1: 捕获/比较 1 中断标记 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置 1,但在中心对称模式下除外(参考 TIM2_CR1 寄存器的 CMS 位)。它由软件清 0。

//0: 无匹配发生;

//1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配。

//注: 在中心对称模式下,当计数器值为 0 时,向上计数,当计数器值为 ARR 时,向下计数(它从 0 向上计数到 ARR-1,再由 ARR 向下计数到 1)。因此,对所有的 SMS 位值,这两个值都不置标记。但是,如果 CCR1>ARR,则当 CNT 达到 ARR 值时,CC1IF 置 1。

//如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置 1,它由软件清 0 或通过读 TIM2_CCR1L 清 0。

//0: 无输入捕获产生;

//1: 计数器值已被捕获(拷贝)至 TIM2_CCR1(在 IC1 上检测到与所选极性相同的边沿)。

//BIT0: 更新中断标记,当产生更新事件时该位由硬件置 1。它由软件清 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 TIM2_CR1 寄存器的 UDIS=0,当计数器上溢或下溢时;

//若 TIM2_CR1 寄存器的 UDIS=0、URS=0,当设置 TIM2_EGR 寄存器的 UG 位软件对计数器

//CNT 重新初始化时;

//若 TIM2_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(参考 0

//从模式控制寄存器 TIM2_SMCR)。

TIM2SR2=0B00000000;

TIM2EGR=0B00000000;

//BIT7: 产生刹车事件,该位由软件置 1,用于产生一个刹车事件,由硬件自动清 0。

//0: 无动作;

//1: 产生一个刹车事件。此时 MOE=0、BIF=1,若开启对应的中断(BIE=1),则产生相应的中断。

//BIT6: 产生触发事件,该位由软件置 1,用于产生一个触发事件,由硬件自动清 0。

//0: 无动作;

//1: TIM2_SR 寄存器的 TIF=1,若开启对应的中断(TIE=1),则产生相应的中断。

//BIT5: 捕获/比较事件,产生控制更新该位由软件置 1,由硬件自动清 0。

//0: 无动作;

//1: 当 CCPC=1,允许更新 CCIE、CCINE、CciP, CciNP, OCIM 位。

//注: 该位只对拥有互补输出的通道有效。

//BIT4: 产生捕获/比较 4 事件

//参考 CC1G 描述。

//BIT3: 产生捕获/比较 3 事件

//参考 CC1G 描述。

//BIT2: 产生捕获/比较 2 事件

//参考 CC1G 描述。

//BIT1: 产生捕获/比较 1 事件

//该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0。

//0: 无动作;

//1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出:

//设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。若通道 CC1 配置为输入:

//当前的计数器值被捕获至 TIM2_CCR1 寄存器, 设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。若 CC1IF 已经为 1, 则设置 CC1OF=1。

//BIT0: 产生更新事件

//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 TIM2_ARR 的值。

TIM2CCMR1 = 0B00000001; //CC1 通道被配置为输出

//BIT7: 输出比较 1 清零使能

//该位用于使能使用 TIM2_TRIG 引脚上的外部事件来清通道 1 的输出信号 (OC1REF), 参考 17.5.9 在外部事件发生时清除 OCREF 信号

//0: OC1REF 不受 ETRF 输入 (来自 TIM2_TRIG 引脚) 的影响;

//1: 一旦检测到 ETRF 输入高电平, OC1REF=0。

//BIG6~BIT4: 输出比较 1 模式, 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。

//000: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用;

//001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为高。

//010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为低。

//011: 翻转。当 TIM2_CCR1=TIM2_CNT 时, 翻转 OC1REF 的电平。

//100: 强制为无效电平。强制 OC1REF 为低。

//101: 强制为有效电平。强制 OC1REF 为高。

//110: PWM 模式 1— 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。

//111: PWM 模式 2— 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为有

效电平，否则为无效电平。

//注 1: 一旦 LOCK 级别设为 3(TIM2_BKR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

//注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。(参考 17.5.7PWM 模式)

//注 3: 在有互补输出的通道上, 这些位是预装载的。如果 TIM2_CR2 寄存器的 CCPC=1, OCM 位只有在 COM 事件发生时, 才从预装载位取新值。

//BIT3: 输出比较 1 预装载使能

//0: 禁止 TIM2_CCR1 寄存器的预装载功能, 可随时写入 TIM2_CCR1 寄存器, 并且新写入的数值立即起作用。

//1: 开启 TIM2_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。

//注 1: 一旦 LOCK 级别设为 3(TIM2_BKR 寄存器中的 LOCK 位)并且 CC1S=00(该通道配置成输出) 则该位不能被修改。

//注 2: 为了操作正确, 在 PWM 模式下必须使能预装载功能。但在单脉冲模式下(TIM2_CR1 寄存器的 OPM=1), 它不是必须的。

//BIT2: 输出比较 1 快速使能, 该位用于加快 CC 输出对触发输入事件的响应。

//0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。

//1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。

//OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。

//BIT1~BIT0: 捕获/比较 1 选择。这 2 位定义通道的方向(输入/输出), 及输入脚的选择:

//00: CC1 通道被配置为输出;

//01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上;

//10: CC1 通道被配置为输入, IC1 映射在 TI2FP1 上;

//11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由

//TIM2_SMCR 寄存器的 TS 位选择)。

//注: CC1S 仅在通道关闭时(TIM2_CCER1 寄存器的 CC1E=0)才是可写的。

//TIM2CCMR2 =0B01101000;//CC2 通道被配置为输出

//TIM2CCMR3 =0B00000000;

TIM2CCER1 =0B00000001; //比较 2 互补输出使能, 低电平有效; 比较器 1 输出使能, 低电平有效

//BIT5: 输入捕获/比较 2 输出极性。参考 CC1P 的描述。

//BIT4: 输入捕获/比较 2 输出使能。参考 CC1E 的描述。

//BIT1: 输入捕获/比较 1 输出极性 CC1 通道配置为输出:

//0: OC1 高电平有效;
 //1: OC1 低电平有效。
 //CC1 通道配置为触发(参考图 61):
 //0: 触发发生在 TI1F 的高电平或上升沿;
 //1: 触发发生在 TI1F 的低电平或下降沿。
 //CC1 通道配置为输入(参考图 61):
 //0: 捕捉发生在 TI1F 的高电平或上升沿;
 //1: 捕捉发生在 TI1F 的低电平或下降沿。
 //注 1: 一旦 LOCK 级别(TIM2_BKR 寄存器中的 LCCK 位)设为 3 或 2, 则该位不能被修改。
 //注 2: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM2_CR2 寄存器), 只有在
 //COM 事件发生时, CC1P 位才从预装载位中取新值。
 //BIT0: 输入捕获/比较 1 输出使能
 //CC1 通道配置为输出:
 //0: 关闭— OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
 //1: 开启— OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
 CC1 通道配置为输入:
 //该位决定了计数器的值是否能捕获入 TIM2_CCR1 寄存器。
 //0: 捕获禁止;
 //0: 捕获使能。
 //注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1(TIM2_CR2 寄存器), 只有在
 //COM 事件发生时, CC1E 位才从预装载位中取新值。

```
//TIM2CCER2=0B00000000;
```

```
TIM2CNTRH=0B00000000;//TIM2 计数器
```

```
TIM2CNTRL=0B00000000;
```

```
TIM2PSCR=0B00000000;
```

```
//TIM2ARRH=0x03;           //自动重载, 周期
```

```
//TIM2ARRL=0xe8;
```

```
//TIM2CCR1H=0x01;           //PWM 脉宽
```

```
//TIM2CCR1L=0xf4;
```

```
//TIM2CCR2H=0x01;
```

```
//TIM2CCR2L=0xf4;
```

```
//TIM2CCR3H =0B00000000;  
//TIM2CCR3L =0B00000000;  
T2CEN =1;  
}  
/*-----  
* 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  
    Time2Initial();  
    Time4Initial();  
    while(1)  
    {  
        NOP();  
    }  
}
```


Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong 518057

Tel: (86 755) 86117811

Fax: (86 755) 86117810

Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong

Tel: (852) 27811186

Fax: (852) 27811144

Fremont Micro Devices (USA), Inc.

42982 Osgood Road Fremont, CA 94539

Tel: (1-510) 668-1321

Fax: (1-510) 226-9918

Web Site: <http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) 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, Incorporated (BVI). Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.