

FT62F08X

TIM2_CAPTURE Application note

目录

1. 通用定时器 TIM2.....	3
1.1. 特性	3
1.2. Timer2 相关寄存器汇总	4
1.3. 功能描述	11
2. 基本定时器 TIM4.....	15
2.1. 特性	15
2.2. TIM4 相关寄存器汇总	15
2.3. TIM4 时钟源.....	17
2.4. 预分频器	17
2.5. TIM4 中断	17
3. 应用范例.....	18
联系信息	28

FT62F08x TIM2_CAPTURE 应用

1. 通用定时器 TIM2

1.1. 特性

Timer2 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

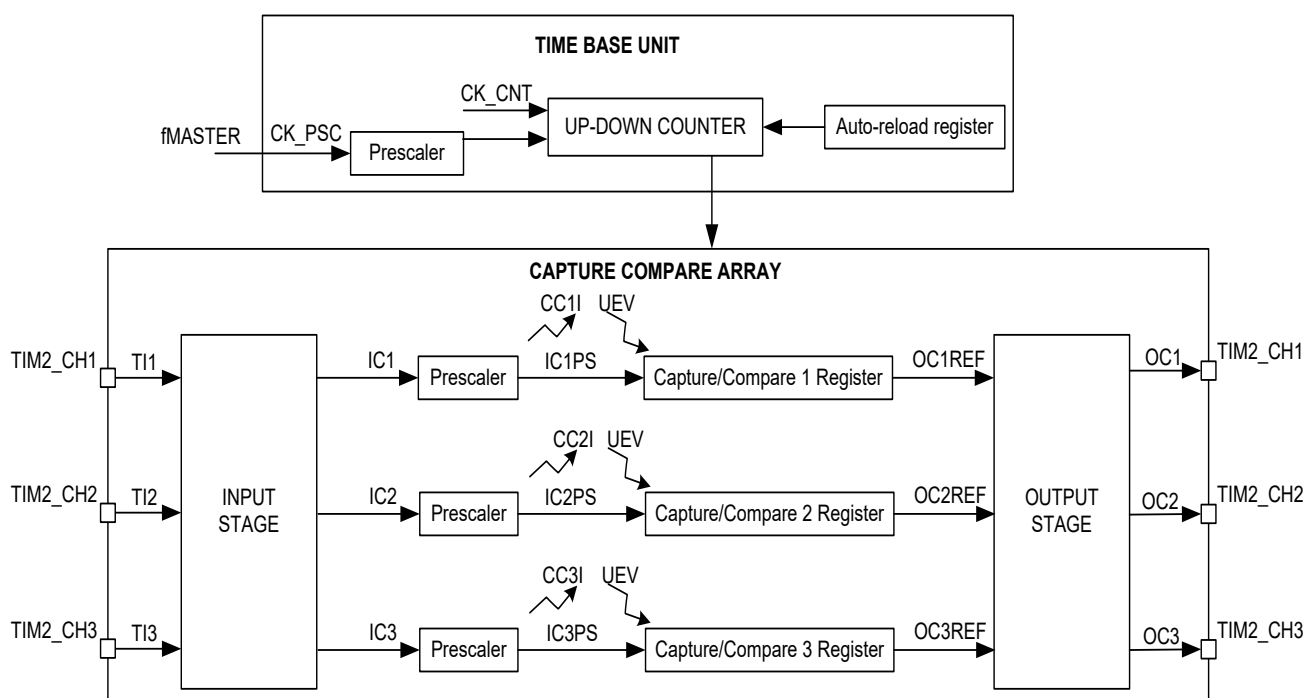


图 1-1 TIM2 原理框图

1.2. Timer2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
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	—	---- 000-
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	—	T2OC3M[2:0]			T2OC3PE	—	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

表 1-1 Timer2 相关用户寄存器汇总

名称	状态		寄存器	地址	复位值
T2CNT	TIM2 计数值	高 8 位	TIM2CNTRH[7:0]	0x316	RW-0000 0000
		低 8 位	TIM2CNTRL[7:0]	0x317	RW-0000 0000
T2PSC	TIM2 预分频器		TIM2PSCR[3:0]	0x318	RW-0000
T2ARR	输出比较模式：PWM 周期的自动重装载寄存器(预装载值) 注：此值为 0 时，计数器不工作；	高 8 位	TIM2ARRH[7:0]	0x319	RW-1111 1111
		低 8 位	TIM2ARRL[7:0]	0x31A	RW-1111 1111
T2CCR1	输出比较模式：TIM2_CH1 占空比(预装载值)	高 8 位	TIM2CCR1H[7:0]	0x31B	RW-0000 0000
		低 8 位	TIM2CCR1L[7:0]	0x31C	RW-0000 0000
	输入捕获模式：上一次捕获事件(IC1)捕获的计数值	高 8 位	TIM2CCR1H[7:0]	0x31B	RO-0000 0000
		低 8 位	TIM2CCR1L[7:0]	0x31C	RO-0000 0000
T2CCR2	输出比较模式：TIM2_CH2 占空比(预装载值)	高 8 位	TIM2CCR2H[7:0]	0x31D	RW-0000 0000
		低 8 位	TIM2CCR2L[7:0]	0x31E	RW-0000 0000
	输入捕获模式：上一次捕获事件(IC2)捕获的计数值	高 8 位	TIM2CCR2H[7:0]	0x31D	RO-0000 0000
		低 8 位	TIM2CCR2L[7:0]	0x31E	RO-0000 0000
T2CCR3	输出比较模式：TIM2_CH3 占空比(预装载值)	高 8 位	TIM2CCR3H[7:0]	0x29E	RW-0000 0000
		低 8 位	TIM2CCR3L[7:0]	0x29F	RW-0000 0000
	输入捕获模式：上一次捕获事件(IC3)捕获的计数值	高 8 位	TIM2CCR3H[7:0]	0x29E	RO-0000 0000
		低 8 位	TIM2CCR3L[7:0]	0x29F	RO-0000 0000

表 1-2 Timer2 周期相关寄存器

名称	状态		寄存器	地址	复位值
TIM2EN	<u>TIM2 模块时钟</u>	1 = 使能 0 = 关闭	PCKEN[2]	0x9A	RW-0
SYSON	<u>睡眠模式下，系统时钟控制</u>	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T2CKSRC	<u>Timer2 时钟源 (Fmaster)</u> 000 = Sysclk 100 = 2x (XT or EC) ^(*) 001 = HIRC 101 = LIRC 010 = XT or EC ^(*) 110 = LP or EC ^(*) 011 = 2x HIRC 111 = 2x (LP or EC) ^(*) ^(*) 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。		TCKSRC[6:4]	0x31F	RW-000
T2ARPE	<u>PWM 周期的自动预装载</u> 1 = 使能 (T2ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T2ARR 立即被加载)		TIM2CR1[7]	0x30C	RW-0
T2OPM	<u>单脉冲模式</u> 1 = 使能 (下一次更新事件到来时，T2CEN 自动清零，计数器停止) 0 = 关闭 (发生更新事件时，计数器不停止)		TIM2CR1[3]		RW-0
T2URS	<u>当 T2UDIS=0 时，更新事件中断源</u> 1 / 0 = 计数器上溢/下溢		TIM2CR1[2]		RW-0
T2UDIS	<u>产生更新事件控制</u> 1 = 禁止 0 = 允许		TIM2CR1[1]		RW-0
T2CEN	<u>TIM2 计数器</u>	1 = 使能 0 = 关闭	TIM2CR1[0]		RW-0

表 1-3 Timer2 相关用户控制寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM2CCMR1	0x311	T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		RW-0000 0000
TIM2CCMR2	0x312	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		RW-0000 0000
TIM2CCMR3	0x313	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		RW-0000 0000

名称	状态			寄存器	地址	复位值
T2ICxF	通道 x 输入捕获采样频率和数字滤波器长度			TIM2CCMRx[6:4] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW-0000
	Value	采样频率 (f_{SAMPLING})	数字滤波器 长度 (N)			
	0000	$F_{\text{master}}/2$	0			
	0001	F_{master}	2			
	0010	F_{master}	4			
	0011	F_{master}	8			
	0100	$F_{\text{master}} / 2$	6			
	0101	$F_{\text{master}} / 2$	8			
	0110	$F_{\text{master}} / 4$	6			
	0111	$F_{\text{master}} / 4$	8			
	1000	$F_{\text{master}} / 8$	6			
	1001	$F_{\text{master}} / 8$	8			
	1010	$F_{\text{master}} / 16$	5			
	1011	$F_{\text{master}} / 16$	6			
	1100	$F_{\text{master}} / 16$	8			
	1101	$F_{\text{master}} / 32$	5			
	1110	$F_{\text{master}} / 32$	6			
	1111	$F_{\text{master}} / 32$	8			
T2ICxPSC	通道 x 输入捕获预分频器 (几个事件触发一次捕获) 00 = 1 个 01 = 2 个 10 = 4 个 11 = 8 个 注: 当 T2CCxE = 0 时, 该预分频器复位为 00			TIM2CCMRx[3:2]		RW-00
T2CC1S ¹	通道 1 模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI1FP1 10 = 输入, 输入脚映射在 TI2FP1 11 = 保留		TIM2CCMR1[1:0]	0x311	RW-00

¹ 仅在通道 x 关闭时(即 T2CCxE = 0)可写, x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
T2CC2S ²	通道2模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI2FP2 10 = 输入, 输入脚映射在 TI1FP2 11 = 保留	TIM2CCMR2[1:0]	0x312	RW-00
T2CC3S ²	通道3模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI3FP3 1x = 保留	TIM2CCMR3[1:0]	0x313	RW-00

表 1-4 TIM2CCMRx 作为输入配置寄存器

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	复位值
TIM2CCMR1	0x311	-	T2OC1M[2:0]		T2OC1PE		-	T2CC1S[1:0]		RW--000 0-00
TIM2CCMR2	0x312	-	T2OC2M[2:0]		T2OC2PE		-	T2CC2S[1:0]		RW--000 0-00
TIM2CCMR3	0x313	-	T2OC3M[2:0]		T2OC3PE		-	T2CC3S[1:0]		RW--000 0-00

T2OCxM	输出模式描述		OCxREF (输出参考信号)
000	冻结 (不比较)		<u>禁止</u>
001	当 TIM2_CNT = CCRx_SHAD 时		1
010	当 TIM2_CNT = CCRx_SHAD 时		0
011	当 TIM2_CNT = CCRx_SHAD 时		电平翻转
100	强制为无效电平		0
101	强制为有效电平		1
110	PWM1 模式	TIM2_CNT < CCRx_SHAD	1
		TIM2_CNT > CCRx_SHAD	0
111	PWM2 模式	TIM2_CNT < CCRx_SHAD	0
		TIM2_CNT > CCRx_SHAD	1

1. OCxREF 与 T2CCxP 共同决定输出引脚 OCx 的值;
2. 在 PWM 模式仅当比较结果改变, 或在输出比较模式从冻结模式切换到 PWM 模式时, OCxREF 电平才改变;

表 1-5 T2OCxM 配置为输出比较模式

² 仅在通道 x 关闭时(即 T2CCxE = 0)可写, x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
T2OCxPE	通道 x 输出比较占空比的自动预装载 1 = 使能 (T2CCRx 预装载值在更新事件到来时加载) 0 = 禁止 (T2CCRx 立即被加载)		TIM2CCMRx[3] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW-0
T2CC1S ³	通道 1 模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI1FP1 10 = 输入, 输入脚映射在 TI2FP1 11 = 保留	TIM2CCMR1[1:0]	0x311	RW-00
T2CC2S ³	通道 2 模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI2FP2 10 = 输入, 输入脚映射在 TI1FP2 11 = 保留	TIM2CCMR2[1:0]	0x312	RW-00
T2CC3S ³	通道 3 模式选择	00 = 输出 01 = 输入, 输入脚映射在 TI3FP3 1x = 保留	TIM2CCMR3[1:0]	0x313	RW-00

表 1-6 TIM2CCMRx 作为输出配置寄存器

名称	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	地址	复位值
TIM2CCER1	-	-	T2CC2P	T2CC2E	-	-	T2CC1P	T2CC1E	0x314	RW---00 --00
TIM2CCER2	-	-	-	-	-	-	T1CC3P	T1CC3E	0x315	RW-----00

名称	功能	输出比较模式	输入捕获/触发模式
T2CCxP	通道 x 引脚 输出极性选择	1 = OCx 低电平有效 0 = <u>OCx 高电平有效</u>	1 = 捕获 / 触发发生在 TlxF 低电平或下降沿 0 = <u>捕获 / 触发发生在 TlxF</u> 高电平或上升沿
T2CCxE	通道 x 引脚 使能	1 = 使能 (OCx 输出到对应的引脚) 0 = 禁止	1 = 使能 (捕获计数器的值到 TIM2CCRx 寄存器中) 0 = 禁止

注: 通道输出电平由 T2OISx 和 T2CCxE 位的值共同决定

表 1-7 Timer2 通道输出和极性选择

名称	状态	寄存器	地址	复位值
TIM2_CH1	TIM2 通道 1 管脚功能重映射 1 = PB0 0 = <u>PA5</u>	AFP1[2]	0x19F	RW-0

表 1-8 Timer2 通道管脚功能重映射

³ 仅在通道 x 关闭时(即 T2CCxE = 0)可写, x = 1, 2, 3。

名称	状态		寄存器	地址	复位值
GIE	<u>全局中断</u> 1 = 使能 (PEIE, T2CCxIE, T2CCxG, T2UIE 适用) 0 = <u>全局关闭</u> (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	<u>外设总中断</u> 1 = 使能 (T2CCxIE, T2CCxG, T2UIE 适用) 0 = <u>关闭</u> (无唤醒)		INTCON[6]		RW-0
T2CC3IE	通道 3 捕获/比较中断	1 = 使能 0 = <u>关闭</u>	TIM2IER[3]	0x30D	RW-0
T2CC2IE	通道 2 捕获或比较中断		TIM2IER[2]		RW-0
T2CC1IE	通道 1 捕获或比较中断		TIM2IER[1]		RW-0
T2CC3G ⁴	通道 3 捕获/比较软件中断		TIM2EGR[3]	0x310	WO-0
T2CC2G ⁴	通道 2 捕获/比较软件中断		TIM2EGR[2]		WO-0
T2CC1G ⁴	通道 1 捕获/比较软件中断		TIM2EGR[1]		WO-0
T2CC3IF ⁵	<u>通道 x 匹配/捕获中断标志位</u> • 输出模式： 1 = CNT 值与 T2CCRx 值匹配 0 = <u>不匹配</u> • 输入模式： 1 = 计数器值已被捕获至 TIM2CCR 0 = <u>无捕获产生</u> 注：软件清 0 或读 TIM2CCRxL 清 0		TIM2SR1[3]	0x30E	R_W1C-0
T2CC2IF ⁵			TIM2SR1[2]		R_W1C-0
T2CC1IF ⁵			TIM2SR1[1]		R_W1C-0
T2CC3OF ⁵	<u>通道 x 重复捕获中断标志位</u> 1 = 发生重复捕获 (计数器的值被捕获到 TIM2CCRx 寄存器时, T2CCxIF 的状态已经为 1) 0 = <u>无重复捕获</u> 注：仅通道配置位捕获输入时有效		TIM2SR2[3]	0x30F	R_W1C-0
T2CC2OF ⁵			TIM2SR2[2]		R_W1C-0
T2CC1OF ⁵			TIM2SR2[1]		R_W1C-0
T2UIE	<u>允许更新中断</u>	1 = 使能 0 = 关闭	TIM2IER[0]	0x30D	RW-0
T2UG ⁴	<u>允许更新软件中断</u>		TIM2EGR[0]	0x310	WO-0
T2UIF ⁵	<u>更新中断标志位</u> 1 = 更新事件等待响应 0 = <u>无更新事件</u>		TIM2SR1[0]	0x30E	R_W1C-0

表 1-9 Timer2 中断使能和状态位

⁴ 软件置 1, 硬件自动清 0。

⁵ 写 1 清 0, 写 0 无效。建议只使用 STR、MOVWI 指令进行写操作, 而不要用 BSR 或 IOR 指令。

1.3. 功能描述

整个 TIM2 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

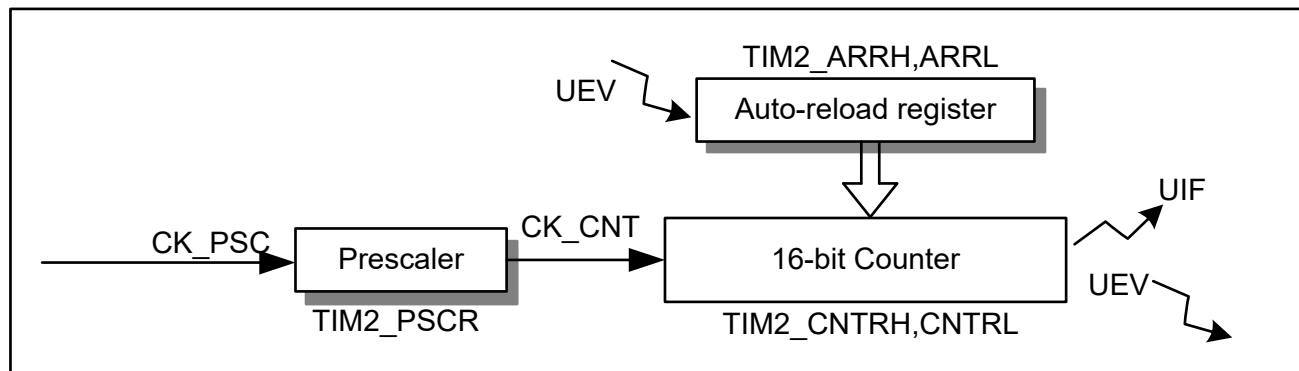


图 1-2 计数基本单元框图

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

TIM2 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 TIM2 时钟
- T2CKSRC[2:0] = 001 时，HIRC 为 TIM2 时钟
- T2CKSRC[2:0] = 010 时，XT 时钟/外部时钟为 TIM2 时钟
- T2CKSRC[2:0] = 011 时，HIRC 的 2 倍频为 TIM2 时钟
- T2CKSRC[2:0] = 100 时，XT 时钟/外部时钟的 2 倍频为 TIM2 时钟
- T2CKSRC[2:0] = 101 时，LIRC 为 TIM2 时钟
- T2CKSRC[2:0] = 110 时，LP 时钟/外部时钟为 TIM2 时钟
- T2CKSRC[2:0] = 111 时，LP 时钟/外部时钟的 2 倍频为 TIM2 时钟

1.3.1.2. 向上计数器

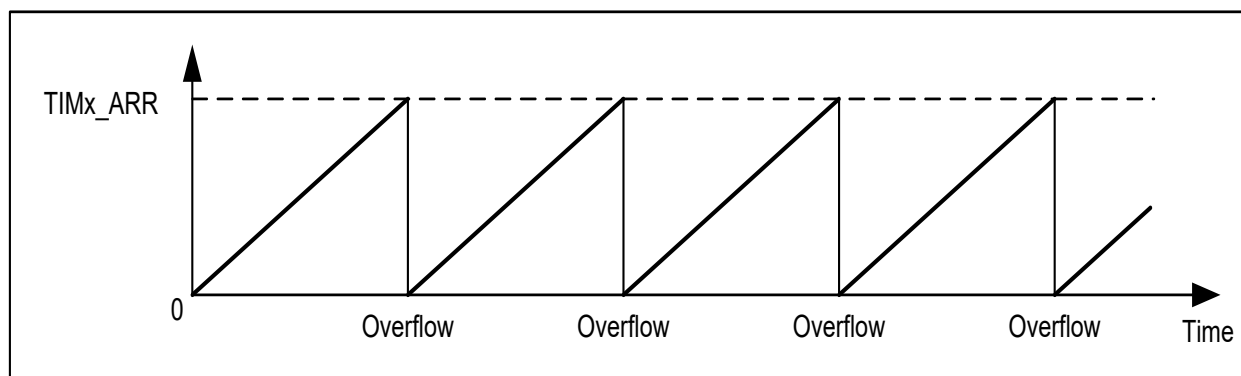


图 1-3 向上计数器

TIM2 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：

$$f_{CK_CNT} = F_{master} / 2^{(PSCR[3:0])} \cdot 2^{(PSCR[3:0])}$$

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

TIM2CCMRx 寄存器是复用寄存器。

当作为输出比较通道时，TIM2CCMRx 寄存器作为输出配置寄存器，并且第 7 位和第 2 位禁止配置，保持为默认值；

当作为输入捕捉通道时，TIM2CCMRx 寄存器作为输入配置寄存器；

1.3.2.1. 捕捉输入通道

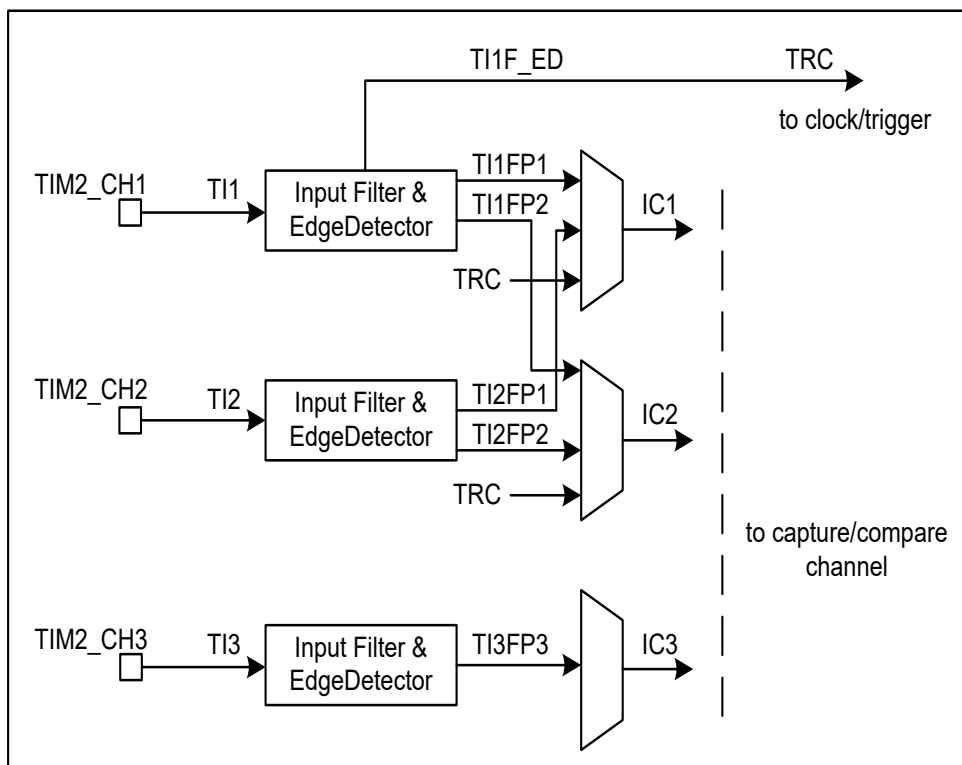


图 1-4 输入通道框图

1.3.2.2. 输出比较通道

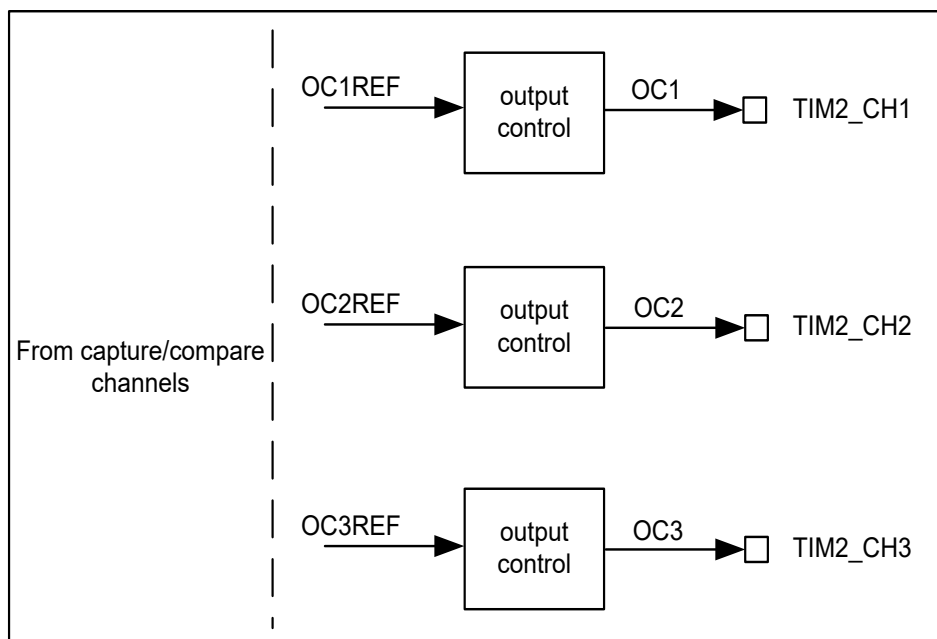


图 1-5 输出通道框图

TIM2 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

TIM2 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 TIM2IER 寄存器中的中断使能位 (T2CCxIE 和 T2UIE)。

不同的中断源还可以配置通过 TIM2EGR 寄存器来产生 (软件产生中断)。

2. 基本定时器 TIM4

2.1. 特性

- 8bit 自动重载向上计数器
- 计数时钟可编程预分频
- 计数器溢出中断

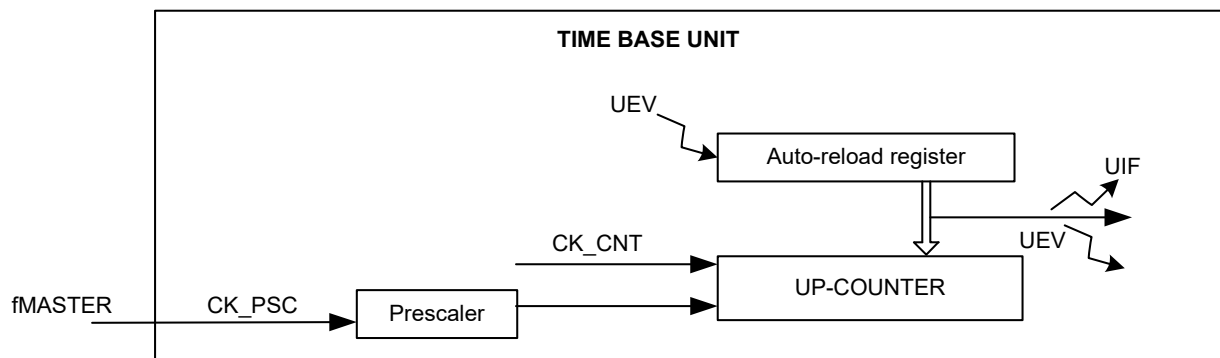


图 2-6 TIM4 原理框图

2.2. TIM4 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM4CR1	0x111	T4ARPE	—	T4CKS[1:0]		T4OPM	T4URS	T4UDIS	T4CEN	0-00 0000
TIM4IER	0x112	—	—	—	—	—	—	—	T4UIE	---- ---0
TIM4SR	0x113	—	—	—	—	—	—	—	T4UIF	---- ---0
TIM4EGR	0x114	—	—	—	—	—	—	—	T4UG	---- ---0
TIM4CNTR	0x115	T4CNT[7:0]								0000 0000
TIM4PSCR	0x116	—	—	—	—	—	T4PSC[2:0]			---- -000
TIM4ARR	0x117	T4ARR[7:0]								1111 1111

表 2-10 Timer4 相关用户寄存器汇总

名称	状态		寄存器	地址	复位值
TIM4EN	<u>TIM4 模块时钟</u>	1 = 使能 0 = 关闭	PCKEN[3]	0x9A	RW-0
SYSON	睡眠模式下, 系统时钟控制	1 = 使能 0 = 关闭	CKOCON[7]	0x95	RW-0
T4ARPE	<u>周期的自动预装载</u> 1 = 使能 (T4ARR 预装载值在更新事件到来时被加载) 0 = 禁止 (T4ARR 立即被加载)		TIM4CR1[7]	0x111	RW-0
T4CKS	<u>Timer4 时钟源</u> 00 = Sysclk 10 = LP ^(*) 01 = HIRC 11 = XT ^(*) ^(*) FOSC 应相应配置成 LP/XT 或选择 INTOSCIO 模式, 否则振荡器将不会运行		TIM4CR1[5:4]		RW-00
T4OPM	<u>单脉冲模式</u> 1 = 使能 (下一次更新事件到来时, 计数器停止) 0 = 关闭 (发生更新事件时, 计数器不停止)		TIM4CR1[3]		RW-0
T4URS	<u>当 T4UDIS = 0 时, 更新事件中断源</u> 1 = 计数器上溢 0 = 软件设置 T4UG 位或计数器上溢		TIM4CR1[2]		RW-0
T4UDIS	<u>产生更新事件控制</u> 1 = 禁止 0 = 允许		TIM4CR1[1]		RW-0
T4CEN	TIM4 计数器	1 = 使能 0 = 关闭	TIM4CR1[0]		RW-0
T4PSC	<u>Timer4 预分频器</u> 000 = 1 100 = 16 001 = 2 101 = 32 010 = 4 110 = 64 011 = 8 111 = 128 注: 必须产生更新事件或 T4CEN=0, 更新的预分频值才生效		TIM4PSCR[2:0]	0x116	RW-000
T4CNT	Timer4 计数值		TIM4CNTR[7:0]	0x115	RW-0000 0000
T4ARR	<u>周期的自动重载寄存器(预装载值)</u> 注: 此值为 0 时, 计数器不工作		TIM4ARR[7:0]	0x117	RW-1111 1111

表 2-11 Timer4 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 (PEIE, T4UIE, T4UG 适用) 0 = 全局关闭 (唤醒不受影响)		INTCON[7]	Bank 首地址 +0x0B	RW-0
PEIE	外设总中断	1 = 使能 (T4UIE, T4UG 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
T4UIE	允许更新中断	1 = 使能	TIM4IER[0]	0x112	RW-0
T4UG ⁶	允许更新软件中断	0 = 关闭	TIM4EGR[0]	0x114	WO-0
T4UIF	更新中断标志位	1 = 更新事件等待响应 0 = 无更新事件	TIM4SR[0]	0x113	R_W1C-0

表 2-12 Timer4 中断使能和状态位

2.3. TIM4 时钟源

TIM4 有 4 种时钟源可选，由寄存器位 T4CKS 设置。在 TIM4 的被使能 (PCKEN.TIM4EN=1) 的情况下，所选择的时钟源被自动使能。

注意：

1. 如果要选择 LP 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 LP 模式，否则对应的时钟源将不被使能；
2. 同理，如果要选择 XT 晶体时钟，系统时钟配置寄存器位 FOSC 必须选择 XT 模式，否则对应的时钟源将不被使能；

SLEEP 模式下，如果 SYSON 为 1，且 TIM4EN=1，则所选择的时钟源将保持振荡，TIM4 将继续工作；否则，所选的时钟源取决于其他模块的设置情况。

2.4. 预分频器

计数时钟可以进行 3bit 的时钟预分频：

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

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 T4CEN 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。

2.5. TIM4 中断

TIM4 只有一个中断请求源：

- 更新中断 (计数器上溢或计数器初始化)

在用这些中断之前需要提前打开 TIM4IER 寄存器中的中断使能位 (T4UIE)。不同的中断源还可以配置通过 TIM4EGR 寄存器来产生 (软件产生中断 T4UG)。

⁶ 软件置 1，硬件自动清 0。

3. 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 PB3 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 DemoPortOut PB3
#define DemoPortOut1 PB4

volatile uint testdata[20];
volatile unchar y=0;
volatile unchar dataH;
volatile unchar dataL;
volatile uint period=0;
/*-----
* 函数名: interrupt ISR
* 功能:    中断处理函数, 包括定时器 2 与 4
* 输入:    无
* 输出:    无
-----*/
void interrupt ISR(void)
{
    //定时器 4 的中断处
    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(μs)
        period = testdata[1]/32;    //周期约等于 250μs
    }
}
}
/*-----
* 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 1:1
    INTCON = 0;                 //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B00100000;         //PA 输入输出 0-输出 1-输入  PA5-IN
    PORTB = 0B00000000;
    TRISB = 0B00000000;         //PB 输入输出 0-输出 1-输入  PB3,PB4-OUT

    PORTC = 0B00000000;
    TRISC = 0B00000000;         //PC 输入输出 0-输出 1-输入
    PORTD = 0B00000000;
    TRISD = 0B00000000;         //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 端口下拉控制 1-开下拉 0-关下拉
WPDB = 0B00000000;          //PB 端口下拉控制 1-开下拉 0-关下拉
WPDC = 0B00000000;          //PC 端口下拉控制 1-开下拉 0-关下拉
WPDD = 0B00000000;          //PD 端口下拉控制 1-开下拉 0-关下拉

PSRC0 = 0B11111111;          //PORTA,PORTB 源电流设置最大
PSRC1 = 0B11111111;          //PORTC,PORTD 源电流设置最大

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: Time4Initial
* 功能:      初始化定时器 4
* 输入参数: 无
* 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
    //Bit[5:4]: 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: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: 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。

TIM4SR =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 TIM_x_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: 初始化计时器 2

* 输入参数: 无

* 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

//Bit[6:4]: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 倍频

//Bit3:保留位

//Bit[2:1]:TIM1 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

```

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。

//Bit[6:4]:保留位

//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_SHAD、PSC_SHAD、CCRx_SHAD)保持它们的值。

//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
//Bit[7:4]:保留位
//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;

//Bit[7:4]:保留位

//Bit3: 捕获/比较 3 中断标记(写 1 清 0, 写 0 无效)

//参考 CC1IF 描述。

//Bit2: 捕获/比较 2 中断标记(写 1 清 0, 写 0 无效)

//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: (写 1 清 0, 写 0 无效)

//如果通道 1 配置为输出比较模式: 当计数器值与比较值匹配时该位由硬件置 1; 它由软件清 0。

//0: 无匹配发生;

//1: TIM2_CNT 的值与 TIM2CCR1H/L 的值匹配。

//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(写 1 清 0, 写 0 无效)

//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。

TIM2SR2 =0B00000000;

TIM2EGR =0B00000000;

//Bit[7:4]:保留位

//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 TIM2CCR1H/L 寄存器, 设置 T2CC1IF=1, 若开启对应的中断, 则产生相应的中断。

//若 T2CC1IF 已经为 1，则设置 T2CC1OF=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 T2DIR=0(向上计数)则计数器被清 0；若 T2DIR=1(向下计数)则计数器初始化为 TIM1ARRH/L 的值。

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

//Bit7:保留位

//Bit[6:4]：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 T2CC1P 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：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。

//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。

//Bit[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 //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。

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 T2CC1P 的描述。

//Bit4: 输入捕获/比较 2 输出使能。参考 T2CC1E 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 配置为触发输入:

//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:

//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。

//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。

//1: 开启— OC1 信号输出到对应的输出引脚。

//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。

TIM2CNTRH = 0B00000000; //TIM2 计数器

TIM2CNTRL = 0B00000000;

TIM2PSCR = 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();
```

```
    }
```

```
}
```

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.