

# **FT64F0AX**

## **TIM2\_INTERRUPT 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. 应用范例.....	15
联系信息 .....	18

## FT64F0Ax TIM2\_INTERRUPT 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

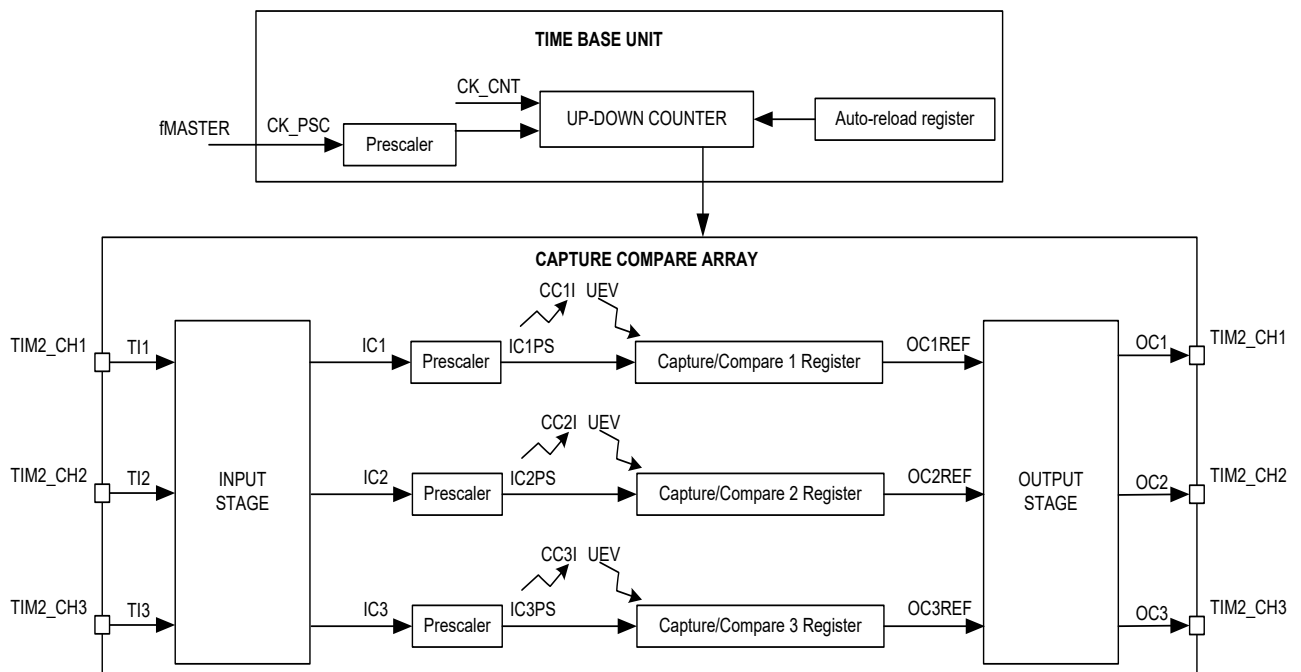


图 1-1 TIM2 原理框图

## 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	UART2EN	I2CEN	UART1EN	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]			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

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: 装入当前捕获/比较 1 寄存器的值(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: 装入当前捕获/比较 2 寄存器的值(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: 装入当前捕获/比较 3 寄存器的值(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
DTYSEL	<u>TIM1/TIM2 倍频时钟占空比调节位</u> 00 = 2ns 延迟      10 = 4ns 延迟 01 = 3ns 延迟      11 = 7ns 延迟		CKOCON[5:4]	0x95	RW-10
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[7:4] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW-0000
	Value	采样频率 (f <sub>SAMPLING</sub> )	数字滤波器长度(N)			
	0000	Fmaster / 2	0			
	0001	Fmaster	2			
	0010	Fmaster	4			
	0011	Fmaster	8			
	0100	Fmaster / 2	6			
	0101	Fmaster / 2	8			
	0110	Fmaster / 4	6			
	0111	Fmaster / 4	8			
	1000	Fmaster / 8	6			
	1001	Fmaster / 8	8			
	1010	Fmaster / 16	5			
	1011	Fmaster / 16	6			
	1100	Fmaster / 16	8			
	1101	Fmaster / 32	5			
	1110	Fmaster / 32	6			
	1111	Fmaster / 32	8			
T2ICxPSC	通道 x 输入捕获预分频器 (几个事件触发一次捕获)			TIM2CCMRx[3:2]		RW-00
	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					
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

<sup>1</sup> 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 <sup>2</sup>	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 <sup>2</sup>	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)		禁止
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	PWM 模式 1	TIM2_CNT < CCRx_SHAD	1
		TIM2_CNT > CCRx_SHAD	0
111	PWM 模式 2	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. PWM 模式下比较结果改变时, 或输出比较模式下从冻结模式切换到 PWM 模式时, OCxREF 电平才会改变;

表 1-5 T2OCxM 配置为输出比较模式

<sup>2</sup> 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被加载) 0 = 禁止 (T2CCRx 立即被加载)		TIM2CCMRx[3] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW-0
T2CC1S <sup>3</sup>	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 <sup>3</sup>	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 <sup>3</sup>	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 TIxF 低电平或下降沿 0 = <u>捕获 / 触发发生在 TIxF 高电平或上升沿</u>
T2CCxE	通道 x 引脚 使能	1 = 使能 (OCx 输出到对应的引脚) 0 = <u>禁止</u>	1 = 使能 (捕获计数器的值到 TIM2CCRx 寄存器中) 0 = <u>禁止</u>

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>	AFP0[4]	0x19E	RW-0
TIM2_CH3	TIM2 通道 3 管脚功能重映射 1 = PA3 0 = <u>PB5</u>	AFP0[3]	0x19E	RW-0

表 1-8 Timer2 通道管脚功能重映射

<sup>3</sup> 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 <sup>4</sup>	通道 3 捕获/比较软件中断		TIM2EGR[3]	0x310	WO-0
T2CC2G <sup>4</sup>	通道 2 捕获/比较软件中断		TIM2EGR[2]		WO-0
T2CC1G <sup>4</sup>	通道 1 捕获/比较软件中断		TIM2EGR[1]		WO-0
T2CC3IF <sup>5</sup>	<u>通道 x 匹配/捕获中断标志位</u> • 输出模式： 1 = CNT 值与 T2CCR <sub>x</sub> 值匹配 0 = <u>不匹配</u> • 输入模式： 1 = 计数值已被捕获至 TIM2CCR 0 = <u>无捕获产生</u> 注：软件清 0 或读 TIM2CCR <sub>xL</sub> 清 0		TIM2SR1[3]	0x30E	R_W1C-0
T2CC2IF <sup>5</sup>			TIM2SR1[2]		R_W1C-0
T2CC1IF <sup>5</sup>			TIM2SR1[1]		R_W1C-0
T2CC3OF <sup>5</sup>	<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 TIM2CCR <sub>x</sub> 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 <sup>5</sup>			TIM2SR2[2]		R_W1C-0
T2CC1OF <sup>5</sup>			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 <sup>4</sup>	<u>允许更新软件中断</u>		TIM2EGR[0]	0x310	WO-0
T2UIF <sup>5</sup>	<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

<sup>4</sup> 软件置 1, 硬件自动清 0。

<sup>5</sup> 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

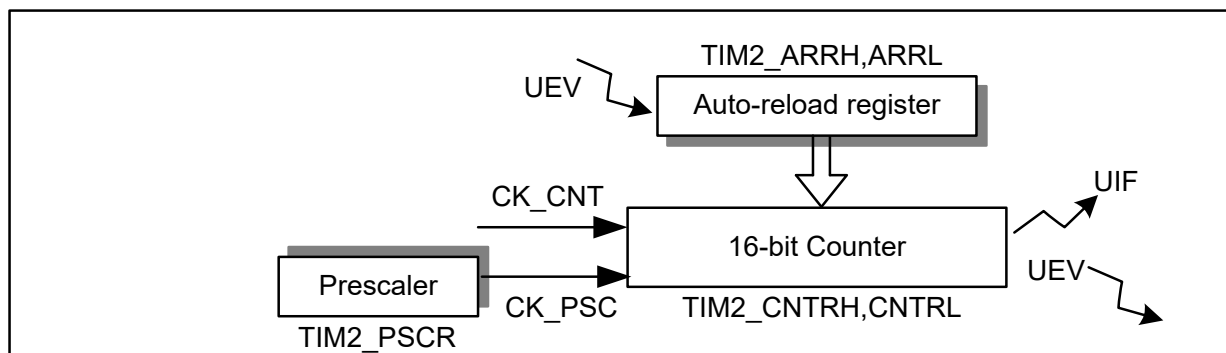


图 1-2 计数基本单元框图

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

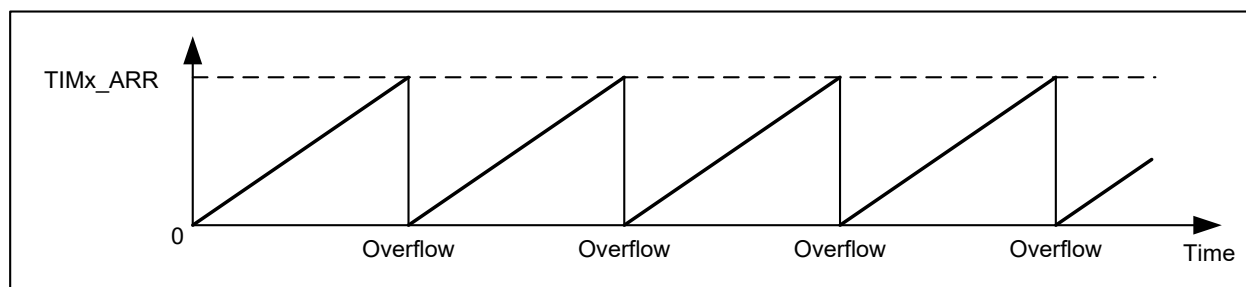


图 1-3 向上计数器

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：

$$f_{CK\_CNT} = f_{CK\_PSC} / 2^{(PSCR[3:0])}$$

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

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

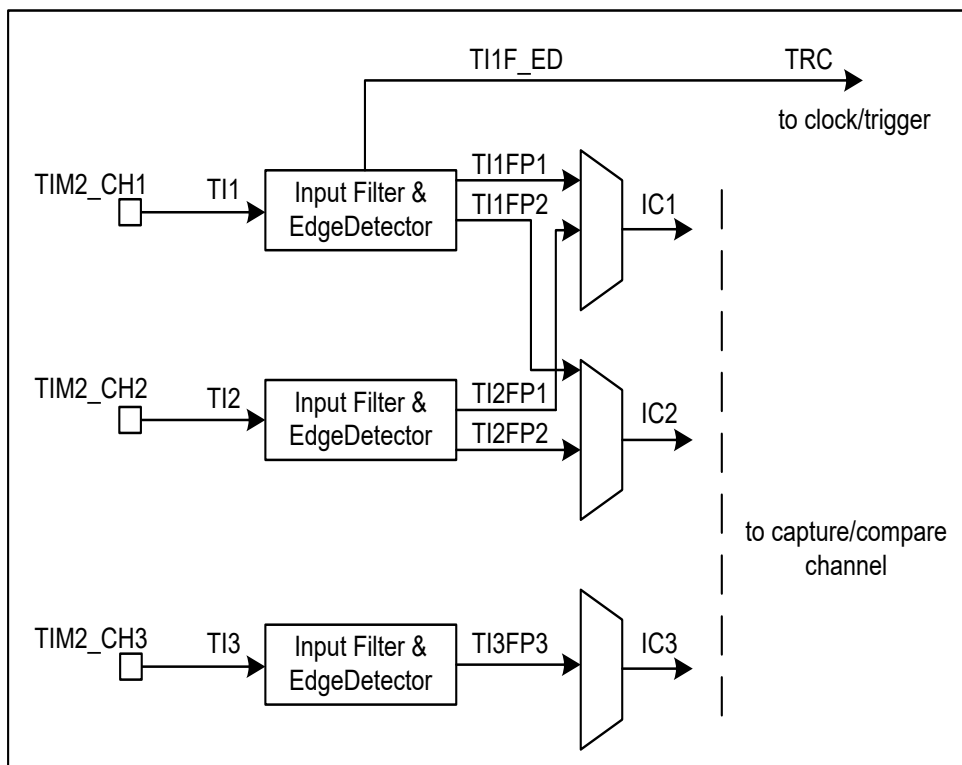


图 1-4 输入通道框图

### 1.3.2.2. 输出比较通道

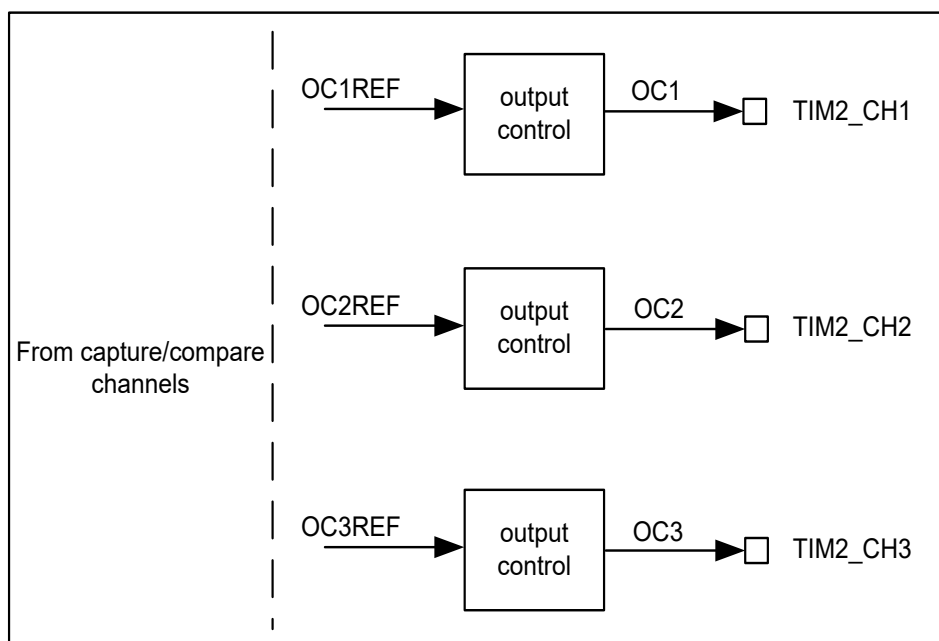


图 1-5 输出通道框图

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. 应用范例

```
//*****
/* 文件名: TEST_64F0Ax_TIM2_INTERRUPT.c
* 功能:   FT64F0Ax_TIM2_INTERRUPT 功能演示
* IC:     FT64F0A5 TSSOP20
* 内部:   16M
* 说明:   TIM2 通过中断在 RB3 输出频率为 16kHz 的方波
*
*
*           FT64F0A5 TSSOP20
*           -----
* NC-----|1(PA5)      (PA4)20|-----NC
* NC-----|2(PA6)      (PA3)19|-----NC
* NC-----|3(PA7)      (PA2)18|-----NC
* NC-----|4(PC0)      (PA1)17|-----NC
* NC-----|5(PC1)      (PA0)16|-----NC
* NC-----|6(PB7)      (PB0)15|-----NC
* GND-----|7(GND)     (PB1)14|-----NC
* NC-----|8(PB6)      (PB2)13|-----NC
* VDD-----|9(VDD)     (PB3)12|--DemoPortOut
* NC-----|10(PB5)     (PB4)11|-----NC
*
*           -----
*/
//*****
#include "SYSCFG.h";
#include "FT64F0AX.h";
//*****宏定义*****
#define DemoPortOut PB3
/*-----
* 函数名: interrupt ISR
* 功能:   定时器 2 中断
* 输入:   无
* 输出:   无
*-----*/
void interrupt ISR(void)
{
    //定时器 2 的中断处理程序
    if(T2UIE&&T2UIF)
    {
        T2UIF=1;                //写 1 清零标志位
        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; //系统时钟选择为内部振荡器 16MHz,分频比为 1:1

INTCON=0; //禁止所有中断

PORTA=0B00000000;

PORTB=0B00000000;

PORTC=0B00000000;

WPUA=0B00000000; //弱上拉的开关, 0-关, 1-开

WPUB=0B00000000;

WPUC=0B00000000;

WPDA=0B00000000; //弱下拉的开关, 0-关, 1-开

WPDB=0B00000000;

WPDC=0B00000000;

TRISA=0B00000000; //输入输出设置, 0-输出, 1-输入

TRISB=0B00000000; //RB3-输出

TRISC=0B00000000;

PSRC0=0B11111111; //源电流设置最大

PSRC1=0B11111111;

PSRC2=0B00001111;

PSINK0=0B11111111; //灌电流设置最大

PSINK1=0B11111111;

PSINK2=0B00000011;

ANSELA=0B00000000; //设置对应的 IO 为数字 IO

}

/\*-----

\* 函数名: TIM2\_INITIAL

\* 功能： 初始化 TIM2

\* 输入： 无

\* 输出： 无

-----\*/

void TIM2\_INITIAL(void)

{



```
PCKEN|=0B00000100;           //使能 TIMER2 模块时钟
CKOCON=0B00100000;           //Timer2 倍频时钟占空比调节位 4ns 延迟
TCKSRC=0B00110000;           //Timer2 时钟源为 HIRC 的 2 倍频

TIM2CR1=0B10000101;          //允许自动装载，使能计数器

TIM2IER=0B00000001;          //允许更新中断

TIM2ARRH=0x03;                //自动装载高 8 位 03H
TIM2ARRL=0xe8;                //自动装载低 8 位 e8H

INTCON=0B11000000;           //使能总中断和外设中断
}
/*-----
* 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
    TIM2_INITIAL();

    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.