通用定时器（TIM3）波形产生例程实验

实验平台：奋斗版STM32开发板MINI、V2、V2.1、V3、V5
实验内容：板子加电后，LED1（MINI的V2标号）（V2、V2.1、V3、V5板的V6标号）会按照12.5Hz的频率闪烁，该实验学习了特定频率波形产生程序的编制及控制流程。

预先需要掌握的知识

1 复用功能I/O和调试配置(AFIO)

为了优化外设数目，可以把一些复用功能重新映射到其他引脚上。设置复用重映射和调试I/O配置寄存器(AFIO_MAPR)(参见0节)实现引脚的重新映射。这时，复用功能不再映射到它们的原始分配上。

1.1 复用重映射和调试I/O配置寄存器(AFIO_MAPR)

地址偏移：0x04
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					SWJ_CFG[2:0]			保留							
					rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01_REMAP	CAN_REMAP [1:0]		TIM4_REMAP	TIM3_REMAP [1:0]		TIM2_REMAP [1:0]		TIM1_REMAP [1:0]		USART3_REMAP [1:0]		USART2_REMAP	USART1_REMAP	I2C1_REMAP	SPI1_REMAP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位31:27	保留。
位26:24	SWJ_CFG[2:0]：串行线JTAG配置 这些位可由软件读写，用于配置SWJ和跟踪复用功能的I/O口。SWJ(串行线JTAG)支持JTAG或SWD访问Cortex的调试端口。系统复位后的默认状态是启用SWJ但没有跟踪功能，这种状态下可以通过JTMS/JTCK脚上的特定信号选择JTAG或SW(串行线)模式。 000：完全SWJ(JTAG-DP + SW-DP)：复位状态 001：完全SWJ(JTAG-DP + SW-DP)但没有JNTRST 010：关闭JTAG-DP，启用SW-DP 100：关闭JTAG-DP，关闭SW-DP 其它组合：禁用
位23:16	保留。
位15	PD01_REMAP：端口D0/端口D1映像到OSC_IN/OSC_OUT 该位可由软件读写，它控制PD0和PD1的GPIO功能映像。当不使用主振荡器HSE时(系统运行于内部的8MHz阻容振荡器)PD0和PD1可以映像到OSC_IN和OSC_OUT引脚。此功能只能适用于36、48和64管脚的封装(PD0和PD1出现在TQFP100的封装上，不必重映像)。 0：不进行PD0和PD1的重映像 1：PD0映像到OSC_IN，PD1映像到OSC_OUT。
位14:13	CAN_REMAP[1:0]：CAN复用功能重映像 这些位可由软件读写，控制复用功能CANRX和CANTX的重映像 00：CANRX映像到PA11，CANTX映像到PA12 01：未用组合 10：CANRX映像到PB8，CANTX映像到PB9(不能用于36脚的封装) 11：CANRX映像到PD0，CANTX映像到PD1(只适用于100脚的封装)
位12	TIM4_REMAP：定时器4的重映像 该位可由软件读写，只控制100脚封装中定时器4的通道1至4的映像。 0：没有重映像(TIM4_CH1/PB6，TIM4_CH2/PB7，TIM4_CH3/PB8，TIM4_CH4/PB9) 1：完全映像(TIM4_CH1/PD12，TIM4_CH2/PD13，TIM4_CH3/PD14，TIM4_CH4/PD15) 注：重映像不影响在PE0上的TIM4_ETR。

位11:10	TIM4_REMAP[1:0]：定时器3的重映像 这些位可由软件读写，控制定时器3的通道1至4在GPIO端口的映像。 00：没有重映像(CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1) 01：未用组合 10：部分映像(CH1/PB4, CH2/PB5, CH3/PB0, CH4/PB1) 11：完全映像(CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9) 注：重映像不影响在PD2上的TIM3_ETR。
位9:8	TIM2_REMAP[1:0]：定时器2的重映像 这些位可由软件读写，控制定时器2的通道1至4和外部触发(ETR)在GPIO端口的映像。 00：没有重映像(CH1/ETR/PA0, CH2/PA1, CH3/PA2, CH4/PA3) 01：部分映像(CH1/ETR/PA15, CH2/PB3, CH3/PA2, CH4/PA3) 10：部分映像(CH1/ETR/PA0, CH2/PA1, CH3/PB10, CH4/PB11) 11：完全映像(CH1/ETR/PA15, CH2/PB3, CH3/PB10, CH4/PB11)
位7:6	TIM1_REMAP[1:0]：定时器1的重映像 这些位可由软件读写，控制定时器1的通道1至4、1N至3N、外部触发(ETR)和断线输入(BKIN)在GPIO端口的映像。 00：没有重映像(ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15) 01：部分映像(ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1) 10: 未用组合 11: 完全映像(ETR/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BKIN/PE15, CH1N/PE8, CH2N/PE10, CH3N/PE12)
位5:4	USART3_REMAP[1:0]：USART3的重映像 这些位可由软件读写，控制USART3的CTS、RTS、CK、TX和RX复用功能在GPIO端口的映像。 00: 没有重映像(TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14) 01: 部分映像(TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14) 10: 未用组合 11: 完全映像(TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12)
位3	USART2_REMAP：USART2的重映像 该位可由软件读写，控制USART2的CTS、RTS、CK、TX和RX复用功能在GPIO端口的映像。 0: 没有重映像(CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4) 1: 重映像(CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7)
位2	USART1_REMAP：USART1的重映像 该位可由软件读写，控制USART1的TX和RX复用功能在GPIO端口的映像。 0: 没有重映像(TX/PA9, RX/PA10) 1: 重映像(TX/PB6, RX/PB7)
位1	I2C1_REMAP：I2C1的重映像 该位可由软件读写，控制I2C1的SCL和SDA复用功能在GPIO端口的映像。 0: 没有重映像(SCL/PB6, SDA/PB7) 1: 重映像(SCL/PB8, SDA/PB9)

位0	SPI1_REMAP：SPI1的重映像 该位可由软件读写，控制SPI1的NSS、SCK、MISO和MOSI复用功能在GPIO端口的映像。 0: 没有重映像(NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7) 1: 重映像(NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5)

2.1 嵌套向量中断控制器（NVIC）

特性

- 43 个可屏蔽中断通道（不包含16 个Cortex-M3 的中断线）；
- 16 个可编程的优先等级；
- 低延迟的异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；

嵌套向量中断控制器(NVIC)和处理器核的接口紧密相连，可以实现低延迟的中断处理和有效处理地处理晚到的中断。

嵌套向量中断控制器管理着包括核异常等中断。关于更多的异常和NVIC编程的说明请参考ARM《Cortex-M3TM技术参考手册》的第5章的异常和第8章的嵌套向量中断控制器。

2.1.1 系统嘀嗒(SysTick)校准值寄存器

系统嘀嗒校准值固定到9000，当系统嘀嗒时钟设定为9兆赫，产生1ms时基。

2.1.2 中断和异常向量

表1 向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-3	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断 RCC时钟安全系统(CSS)联接到NMI向量	0x0000_0008
	-1	固定	硬件失效	所有类型的失效	0x0000_000C
	0	可设置	存储管理	存储器管理	0x0000_0010
	1	可设置	总线错误	预取指失败，存储器访问失败	0x0000_0014
	2	可设置	错误应用	未定义的指令或非法状态	0x0000_0018
	-	-	-	保留	0x0000_001C ~0x0000_002B
	3	可设置	SVCall	通过SWI指令的系统服务调用	0x0000_002C
	4	可设置	调试监控	调试监控器	0x0000_0030
	-	-	-	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG	窗口定时器中断	0x0000_0040

1	8	可设置	PVD	联到EXTI的电源电压检测(PVD)中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	RTC	实时时钟(RTC)全局中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	RCC	复位和时钟控制(RCC)中断	0x0000_0054
6	13	可设置	EXTI0	EXTI线0中断	0x0000_0058
7	14	可设置	EXTI1	EXTI线1中断	0x0000_005C
8	15	可设置	EXTI2	EXTI线2中断	0x0000_0060
9	16	可设置	EXTI3	EXTI线3中断	0x0000_0064
10	17	可设置	EXTI4	EXTI线4中断	0x0000_0068
11	18	可设置	DMA通道1	DMA通道1全局中断	0x0000_006C
12	19	可设置	DMA通道2	DMA通道2全局中断	0x0000_0070
13	20	可设置	DMA通道3	DMA通道3全局中断	0x0000_0074
14	21	可设置	DMA通道4	DMA通道4全局中断	0x0000_0078
15	22	可设置	DMA通道5	DMA通道5全局中断	0x0000_007C
16	23	可设置	DMA通道6	DMA通道6全局中断	0x0000_0080
17	24	可设置	DMA通道7	DMA通道7全局中断	0x0000_0084
18	25	可设置	ADC	ADC全局中断	0x0000_0088
19	26	可设置	USB_HP_CAN_TX	USB高优先级或CAN发送中断	0x0000_008C
20	27	可设置	USB_LP_CAN_RX0	USB低优先级或CAN接收0中断	0x0000_0090
21	28	可设置	CAN_RX1	CAN接收1中断	0x0000_0094
22	29	可设置	CAN_SCE	CAN SCE中断	0x0000_0098
23	30	可设置	EXTI9_5	EXTI线[9:5]中断	0x0000_009C
24	31	可设置	TIM1_BRK	TIM1断开中断	0x0000_00A0
25	32	可设置	TIM1_UP	TIM1更新中断	0x0000_00A4
26	33	可设置	TIM1_TRG_COM	TIM1触发和通信中断	0x0000_00A8
27	34	可设置	TIM1_CC	TIM1捕获比较中断	0x0000_00AC
28	35	可设置	TIM2	TIM2全局中断	0x0000_00B0
29	36	可设置	TIM3	TIM3全局中断	0x0000_00B4
30	37	可设置	TIM4	TIM4全局中断	0x0000_00B8
31	38	可设置	I2C1_EV	I^2C1事件中断	0x0000_00BC
32	39	可设置	I2C1_ER	I^2C1错误中断	0x0000_00C0
33	40	可设置	I2C2_EV	I^2C2事件中断	0x0000_00C4
34	41	可设置	I2C2_ER	I^2C2错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2全局中断	0x0000_00D0
37	44	可设置	USART1	USART1全局中断	0x0000_00D4

38	45	可设置	USART2	USART2全局中断	0x0000_00D8
39	46	可设置	USART3	USART3全局中断	0x0000_00DC
40	47	可设置	EXTI15_10	EXTI线[15:10]中断	0x0000_00E0
41	48	可设置	RTCAlarm	联到EXTI的RTC闹钟中断	0x0000_00E4
42	49	可设置	USB唤醒	联到EXTI的从USB待机唤醒中断	0x0000_00E8

3. 通用定时器(TIMx)

3.1 概述

通用定时器是一个通过可编程预分频器驱动的16 位自动装载计数器构成。
它适用于多种场合，包括测量输入信号的脉冲长度(输入采集)或者产生输出波形(输出比较和PWM)。使用定时器预分频器和RCC 时钟控制器预分频器，脉冲长度和波形周期可以在几个微秒到几个毫秒间调整。定时器是完全独立的，而且没有互相共享任何资源。它们可以一起同步操作。

3.2 主要特性

通用TIMx 定时器特性包括：

- 16 位向上，向下，向上/向下自动装载计数器
- 16 位可编程预分频器，计数器时钟频率的分频系数为1～65535 之间的任意数值
- 4个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成(边缘或中间对齐模式)
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断/DMA：
  - 更新：计数器向上溢出/向下溢出，计数器初始化(通过软件或者内部/外部触发)
  - 触发事件(计数器启动，停止，初始化或者由内部/外部触发计数)
  - 输入捕获
  - 输出比较

3.3 概述

接口通过三个引脚与其他设备连接在一起(见图1)。任何USART双向通信至少需要两个脚：接收数据输入(RX)和发送数据输出(TX)。
RX：接收数据串行输。通过过采样技术来区别数据和噪音，从而恢复数据。
TX：发送数据输出。当发送器被禁止时，输出引脚恢复到它的I/O 端口配置。当发送器被激活，并且没东西发送时，TX 引脚处于高电平。在单线和智能卡模式里，此I/O 口被用来发送和接收数据。

- 总线在发送或接收前应处于空闲状态
- 一个起始位
- 一个数据字（8 或9 位），最低有效位在前

- 0.5，1.5，2 个的停止位，由此表明数据帧的结束
- 使用分数波特率发生器 —— 12 位整数和4 位小数的表示方法。
- 一个状态寄存器(USART_SR)
- 数据寄存器(USART_DR)
- 一个波特率寄存器(USART_BRR)，12 位的整数和4 位小数
- 一个智能卡模式下的保护时间寄存器(USART_GTPR)

关于以上寄存器中每个位的具体定义，请参考寄存器描述。

在同步模式中需要下列引脚：

SCLK：发送器时钟输出。此引脚输出用于同步传输的 时钟， (在Start 位和Stop位上没有时钟脉冲，软件可选地，可以在最后一个数据位送出一个时钟脉冲)。数据可以在RX 上同步被接收。这可以用来控制带有移位寄存器的外部设备(例如LCD 驱动器)。时钟相位和极性都是软件可编程的。在智能卡模式里，SCLK 可以为智能卡提供时钟。

在IrDA 模式里需要下列引脚：

IrDA_RDI：IrDA 模式下的数据输入。

IrDA_TDO：IrDA 模式下的数据输出。

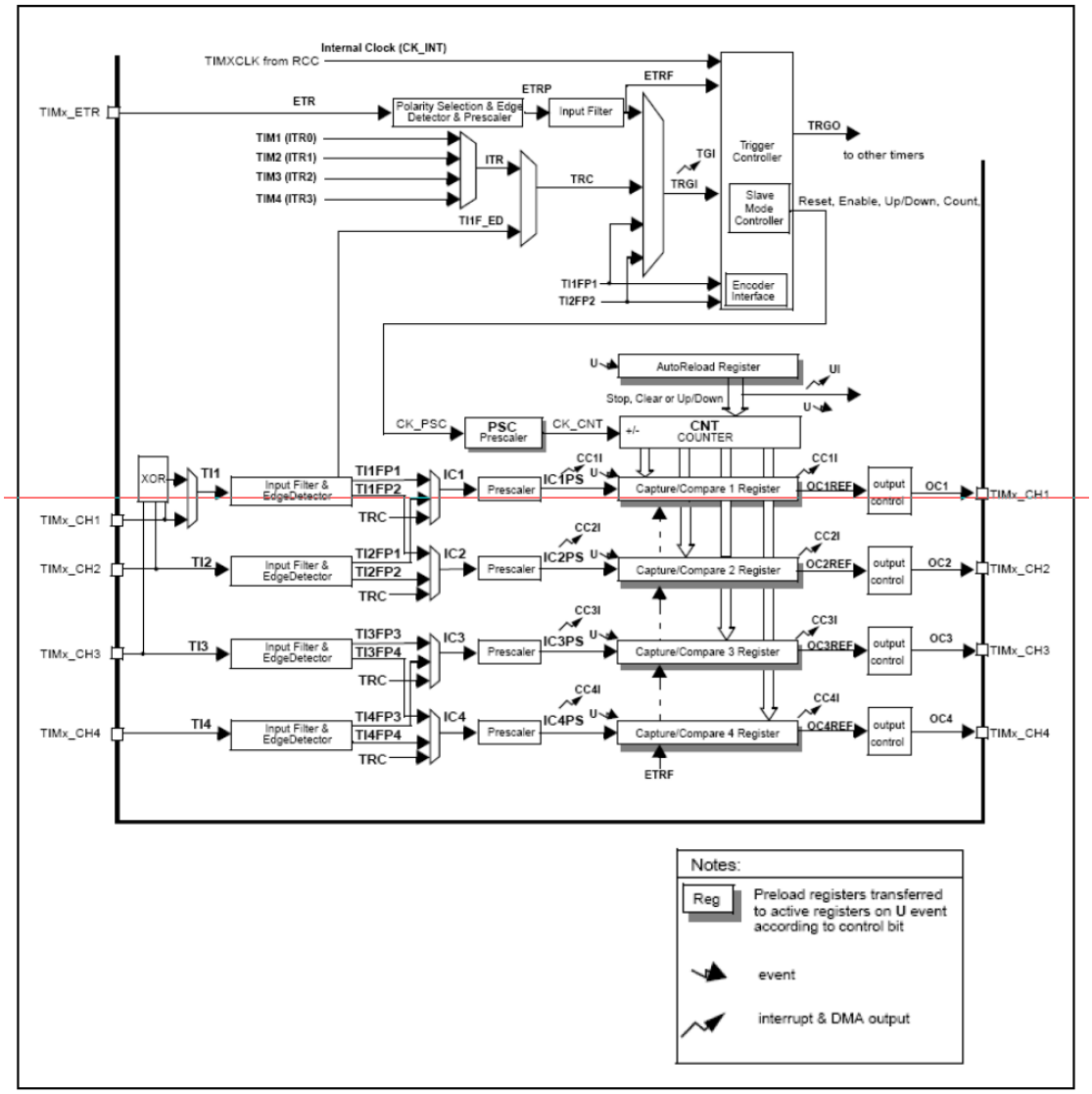
下列引脚在硬件流控模式中需要：

nCTS：清除发送，若是高电平，在当前数据传输结束时阻断下一次的数据发送。

nRTS：发送请求，若是低电平，表明USART 准备好接收数据

3.3.1 框图

图1 通用定时器框图



3.4 功能描述

3.4.1 时基单元

可编程通用定时器的主要部分是一个16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器(TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)

自动装载寄存器是预先装载的。根据在TIMX_CR1 寄存器中的自动装载预装载使能位ARPE)的设置，预装载寄存器的内容被永久地或在每次的更新事件UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当TIMX_CR1 寄存器中的UDIS 位等于0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。计数器由预分频器的时钟输出CK_CNT 驱动，仅当设置了计数器TIMX_CR1

寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(有关更多的计数器使能的细节，请参见控制器的从模式描述)。
注：真正的计数器使能信号CNT_EN 是在CEN 后的一个时钟周期后被设置。

预分频器描述

预分频器可以将计数器的时钟频率按1 到65536 之间的任意值分频。它是基于一个(在TIMx_PSC 寄存器中的)16 位寄存器控制的16 位计数器。因为这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

3.4.2 计数器模式

向上计数模式

在向上计数模式中，计数器从0 计数到自动加载值(TIMx_ARR 计数器的内容)，然后重新从0 开始计数并且产生一个计数器溢出事件。每次计数器溢出时可以产生更新事件，在TIMx_EGR 寄存器中设置UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。通过软件设置TIMx_CR1 寄存器中的UDIS 位，将禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在UDIS 位被清成0 之前，将没有更新事件产生。即使这样，在应该产生更新事件时，计数器仍会被清0，同时预分频器的计数也被请0(但预分频器的数值不变)。此外，如果TIMx_CR1 存器中的URS 位(更新请求选择)被设置，设置UG 位将产生一个更新事件UEV，但硬件不设置UIF 标志(即不产生中断或者DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。当发生一个更新事件，所有的寄存器都被更新，硬件同时(依据URS 位)设置更新标志位(TIMx_SR 寄存器中的UIF 位)。

- 预分频器的缓冲区被置入预装载寄存器的值(TIMx_PSC 寄存器的内容)。
- 自动装载影子寄存器被重新置入预装载寄存器的值(TIMx_ARR)。

向下计数模式

在向下模式中，计数器从自动装入的值(TIMx_ARR 计数器的值)开始向下计数到0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。每次计数器溢出时可以产生更新事件，在TIMx_EGR 寄存器中设置UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。UEV 事件可以通过软件设置TIMx_CR1 寄存器中的UDIS 位被禁止。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。这样UDIS 位被写成0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从0 开始(但预分频器的速率不能被修改)。此外，如果设置了TIMx_CR1 寄存器中的URS 位(更新请求选择) ，设置UG 位将产生一个更新事件UEV 但不设置UIF 标志(因此不产生中断和DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。当发生更新事件时，所有的寄存器都被更新，并且(根据URS 位的设置)更新标志位(TIMx_SR 寄存器中的UIF 位)也被设置。

- 预分频器的缓冲区被置入预装载寄存器的值(TIMx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMx_ARR 寄存器中的内容)。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

中央对齐模式(向上/向下计数)

在中央对齐模式中，计数器从0 开始计数到自动加载的值(TIMx_ARR 寄存器的内

容)-1，产生一个计数器溢出事件，然后向下计数到1 并且产生一个计数器下溢事件；然后再从0 开始重新计数。在这个模式下，不能写入TIMx_CR1 中的DIR 方向位。它由硬件更新并指示当前的计数方向。更新事件可以产生在每一次计数溢出和每一次计数下溢；也可以通过(软件或者使用从模式控制器)设置TIMx_EGR 寄存器中的UG 位来产生更新事件，此时，计数器重新从0 开始计数，预分频器也重新从0 开始计数。UEV 事件可以通过软件设置TIMx_CR1 寄存器中的UDIS 位被禁止。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。这样UDIS 位被写成0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。此外，如果设置了TIMx_CR1 寄存器中的URS 位(更新请求选择) ，设置UG 位将产生一个更新事件UEV 但不设置UIF 标志(因此不产生中断和DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。当发生更新事件时，所有的寄存器都被更新，并且(根据URS 位的设置)更新标志位(TIMx_SR 寄存器中的UIF 位)也被设置。

- 预分频器的缓冲区被置入预装载寄存器的值(TIMx_PSC 寄存器的内容)。
- 当前的自动加载寄存器被更新为预装载值(TIMx_ARR 寄存器中的内容)。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载为新的值)。

3.4.3 时钟选择

计数器时钟可由下列时钟源提供：

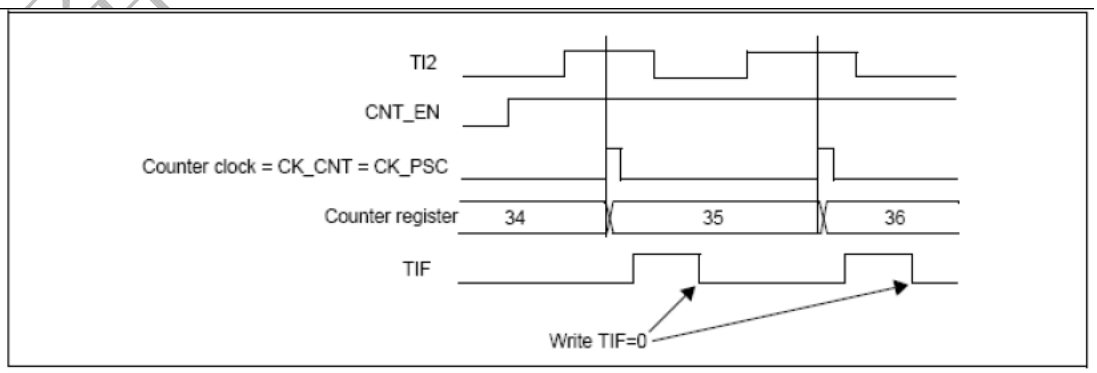
- 内部时钟(CK_INT)
- 外部时钟模式1：外部输入脚(TIx)
- 外部时钟模式2：外部触发输入(ETR)
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器，例如你可以配置一个定时器Timer1 而作为另一个定时器Timer2 的预分频器。

内部时钟源(CK_INT)

如果从模式控制器被禁止(SMS=000)，则CEN、DIR(TIMx_CR1 寄存器中)和UG位(TIMx_EGR 寄存器中)是事实上的控制位同时只能被软件修改(UG 位仍被自动清除)。一旦CEN 位被写成1，预分频器的时钟就由内部时钟CK_INT 提供。

外部时钟源模式1

当TIMx_SMCR 寄存器中的SMS=111 时，此模式被选中。计数器可以在选定的输入上的每个上升沿或下降沿计数。



例如，要配置向上计数器在T12 输入端的上升沿计数，使用下列步骤：

1. 配置TIMx_CCMR1寄存器CC2S=01，配置通道2检测TI2输入的上升沿

2. 配置TIMx_CCMR1寄存器的IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持IC2F=0000)

注： 捕获预分频器不用作触发，所以不需要对它进行配置

3. 配置TIMx_CCER寄存器的CC2P=0，选定上升沿极性
4. 配置TIMx_SMCR寄存器的SMS=111，选择定时器外部时钟模式1
5. 配置TIMx_SMCR寄存器中的TS=110，选定TI2作为触发输入源
6. 设置TIMx_CR1寄存器的CEN=1，启动计数器

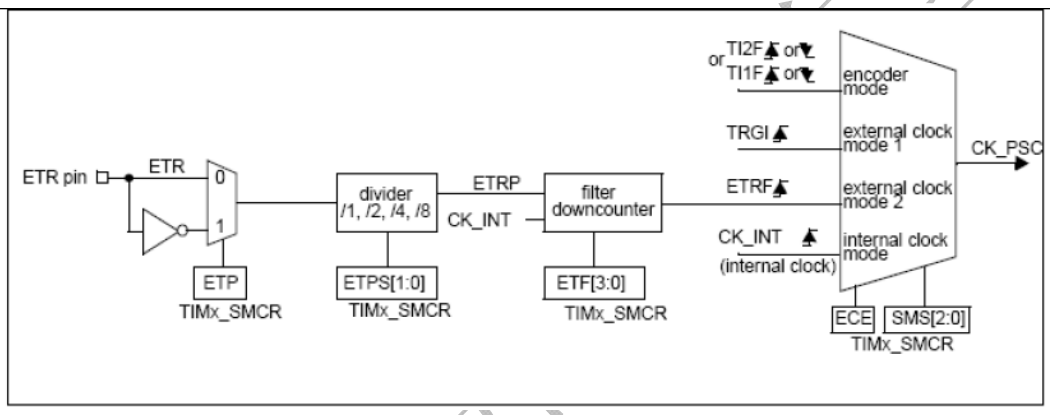
当上升沿出现在TI2，计数器计数一次，且TIF 标志被设置。

在TI2 的上升沿和计数器实际时钟之间的延时取决于在TI2 输入端的重新同步电路。

外部时钟源模式2

选定此模式的方法为：令TIMx_SMCR 寄存器中的ECE=1

计数器能够在外部触发ETR 的每一个上升沿或下降沿计数。



例如，要配置在ETR 下每2 个上升沿计数一次的向上计数器，使用下列步骤：

1. 本例中不需要滤波器，写TIMx_SMCR寄存器中的ETF[3:0]=0000
2. 设置预分频器，写TIMx_SMCR寄存器中的ETPS[1:0]=01
3. 设置在ETR下的上升沿检测，写TIMx_SMCR寄存器中的ETP=0
4. 开启外部时钟模式2，写TIMx_SMCR寄存器中的ECE=1
5. 启动计数器，写TIMx_CR1寄存器中的CEN=1

计数器在每2 个ETR 上升沿计数一次。

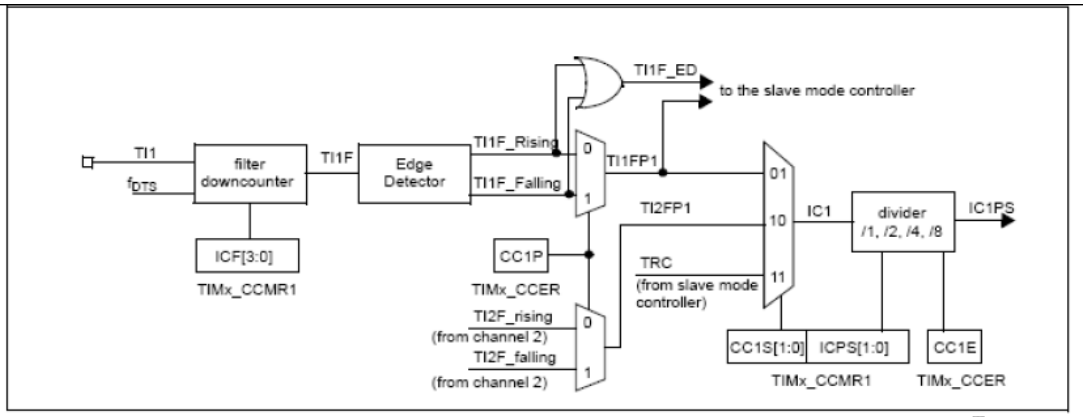
在ETR 的上升沿和计数器实际时钟之间的延时取决于在ETRP 信号端的重新同步电路。

3.4.4 捕获/比较通道

每一个捕获/比较通道是围绕着一个捕获/比较寄存器(包含影子寄存器)，包括捕获的输入部分(包含数字滤波、多路复用和预分频器)，和输出部分(包含比较器和输出控制)。

下图是一个捕获/比较通道概览。

输入部分对相应的TIx 输入信号采样，并产生一个滤波后的信号TIxF。然后，一个带极性选择的边缘监测器产生一个信号(TIxFPx)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号在捕获寄存器(ICxPS)之前已被整形。



输出部分产生一个中间波形OCxRef(高有效)作为基准，链的末端信号决定最终的输出极性。

3.4.5 输入捕获模式

在输入捕获模式下，当检测到ICx 信号上相应的边沿后，捕获/比较寄存器(TIMx_CCRx)被用来锁存计数器的值。当一个捕获事件发生时，相应的CCXIF 标志(TIMx_SR 寄存器)被置1，如果开放了中断或者DMA 操作，则将产生中断或者DMA 操作。如果一个捕获事件发生时CCxIF 标志已经为高，那么重复捕获标志CCxOF(TIMx_SR 寄存器)被置1。写CCxIF=0 可清除CCxIF，或读取存储在TIMx_CCRx 寄存器中的捕获数据也可清除CCxIF。写CCxOF=0 可清除CCxOF。

以下例子说明如何在TI1 输入的上升沿时捕获计数器的值到TIMx_CCR1 寄存器中，步骤如下：

● 选择有效输入端： TIMx_CCR1 必须连接到TI1 输入， 所以写入TIMx_CCR1 寄存器中的CC1S=01，一旦CC1S 不为00 时，通道被配置为输入，并且TM1_CCR1 寄存器变为只读。

● 根据输入信号的特点，配置输入滤波器为所需的带宽(输入为TIx 时，TIMx_CCMRx 寄存器中的ICxF 位)。假设输入信号在最多5 个时钟周期的时间内抖动，我们须配置滤波器的带宽长于5 个时钟周期。因此我们可以(以fDTS 频率)连续采样8 次，以确认在TI1 上一次真实的边沿变换，即在TIMx_CCMR1 寄存器中写入IC1F=0011。

● 选择TI1 通道的有效转换边沿，在TIMx_CCER 寄存器中写入CC1P=0(即上升沿)。

● 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写TIMx_CCMR1 寄存器的IC1PS=00)。

● 设置TIMx_CCER 寄存器的CC1E=1，允许捕获计数器的值到捕获寄存器中。

● 如果需要，通过设置TIMx_DIER 寄存器中的CC1IE 位允许相关中断请求，通过设置TIMx_DIER 寄存器中的CC1DE 位允许DMA 请求。

● 发生当一个输入捕获时：

● 当产生有效的电平转换时，计数器的值被传送到TIMx_CCR1 寄存器。

● CC1IF 标志被设置(中断标志)。当发生至少2 个连续的捕获时，而CC1IF未曾被清除，CC1OF 也被置1。

● 如设置了CC1IE 位，则会产生一个中断。

● 如设置了CC1DE 位，则还会产生一个DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： 设置*TIMx_EGR* 寄存器中相应的*CCxG* 位，可以通过软件产生输入捕获中断和/ 或*DMA* 请求。

3.4.6 PWM输入模式

该模式是输入捕获模式的一个特例，除下列区别外，工作过程与输入捕获模式相同：

- 两个ICx 信号被映射同一个TIx 输入。
- 这2 个ICx 信号为边沿有效，但是极性相反。
- 其中一个TIxFP 信号被作为触发输入信号，并且从模式控制器被配置成复位模式。

例如，你能够测量输入到TI1 上的PWM 信号的长度(TIMx_CCR1 寄存器)和占空比(TIMx_CCR2 寄存器)，具体步骤如下(取决于CK_INT 的频率和预分频器的值)

- 选择TIMx_CCR1 的有效输入端：置TIMx_CCMR1 寄存器的CC1S=01(选择TI1)；
- 选择TI1FP1 的有效极性(用来捕获数据到TIMx_CCR1 中和清除计数器)：置CC1P=0；
- 选择TIMx_CCR2 的有效输入端：置TIMx_CCMR1 寄存器的CC2S=10(选择TI1)；
- 选择TI1FP2 的有效极性(用来捕获数据到TIMx_CCR2)：置CC2P=1(下降沿有效)；
- 选择有效的触发输入信号：置TIMx_SMCR 寄存器中的TS=101(选择TI1FP1)；
- 配置从模式控制器为复位模式：置TIMx_SMCR 中的SMS=100；
- 使能捕获：置TIMx_CCER 寄存器中CC1E=1 且CC2E=1。

3.4.7 强置输出模式

在输出模式(TIMx_CCMRx 寄存器中CCxS=00)下，输出比较信号(OCxREF 和相应的OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置TIMx_CCMRx 寄存器中相应的OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。这样OCxREF 被强置为高电平(OCxREF 始终为高电平有效)，同时OCx 得到CCxP 极性位相反的值。

例如：CCxP=0(OCx 高电平有效)，则OCx 被强置为高电平。置TIMx_CCMRx 寄存器中的OCxM=100，可强置OCxREF 信号为低。该模式下，在TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和DMA 请求。这将会在下面的输出比较模式一节中介绍。

3.4.8 输出比较模式

此项功能是用来控制一个输出波形或者指示何时一段给定的的时间已经到时。当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式(TIMx_CCMRx 寄存器中的OCxM 位) 和输出极性(TIMx_CCER 寄存器中的CCxP 位)定义的值输出到对应的管脚上。在比较匹配时，输出管脚可以保持它的电平(OCxM=011)、被设置成有效电平(OCxM=001) 、被设置成无有效电平(OCxM=010) 或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的CCxIF 位)。
- 如果设置了相应的中断屏蔽(TIMx_DIER 寄存器中的CCXIE 位)，则产生一个中断。
- 若设置了相应的使能位(TIMx_DIER 寄存器中的CCxDE 位，TIMx_CR2 寄存器中的CCDS 位选择DMA 请求功能)，则产生一个DMA 请求。TIMx_CCMRx 中的OCxPE 位用于选择TIMx_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件UEV 对OCxREF 和OCx 输出没有影响。同步的精度到达计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟(内部，外部，预分频器)

2. 将相应的数据写入TIMx_ARR和TIMx_CCRx寄存器中
3. 如果要产生一个中断请求和/或一个DMA请求，设置CCxIE位和/或CCxDE位。
4. 选择输出模式，例如：必须设置OCxM=’011’、OCxPE=’0’、CCxP=’0’和CCxE=’1’，当CNT与CCRx匹配时翻转OCx的输出管脚，CCRx预装载未用，开启OCx输出且高电平有效。
5. 设置TIMx_CR1寄存器的CEN位启动计数器TIMx_CCRx寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器(OCxPE=’0’，否则TIMx_CCRx影子寄存器只能在下一次更新事件发生时被更新)

3.4.9 PWM 模式

脉冲宽度调制模式可以产生一个由TIMx_ARR 寄存器确定频率、由TIMx_CCRx寄存器确定占空比的信号。在TIMx_CCMRx 寄存器中的OCxM 位写入“110”(PWM 模式1)或“111”(PWM 模式2)，能够独立地设置每个通道工作在PWM 模式，每个OCx 输出一路PWM。必须通过设置TIMx_CCMRx 寄存器OCxPE 位使能相应的预装载寄存器，最后还要设置TIMx_CR1 寄存器的ARPE 位使能自动重装载的预装载寄存器(在向上计数或中心对称模式中)。因为仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置TIMx_EGR 寄存器中的UG 位来初始化所有的寄存器。OCx 的极性可以通过软件在TIMx_CCER 寄存器中的CCxP 位设置，它可以设置为高电平有效活和低电平有效。OCx 输出通过TIMx_CCER 寄存器中的CCxE 位使能。详见TIMx_CCERx 寄存器的描述。在PWM 模式(模式1 或模式2)下，TIMx_CNT 和TIM1_CCRx 始终在进行比较，(依据计数器的计数方向)以确定是否符合TIM1_CCRx≤TIM1_CNT 或者TIM1_CNT≤TIM1_CCRx。然而，为了与OCREF_CLR 的功能(在下一个PWM周期之前，OCREF 能够通过ETR 信号被一个外部事件清除)一致，OCREF 信号只能在下述条件下产生：

- 当比较的结果改变，或
- 当输出比较模式(TIMx_CCMRx 寄存器中的OCxM 位)从“冻结”(无比较，OCxM=’000’)切换到某个PWM 模式(OCxM=’110’或’111’)。这样在运行中可以通过软件强置PWM 输出。

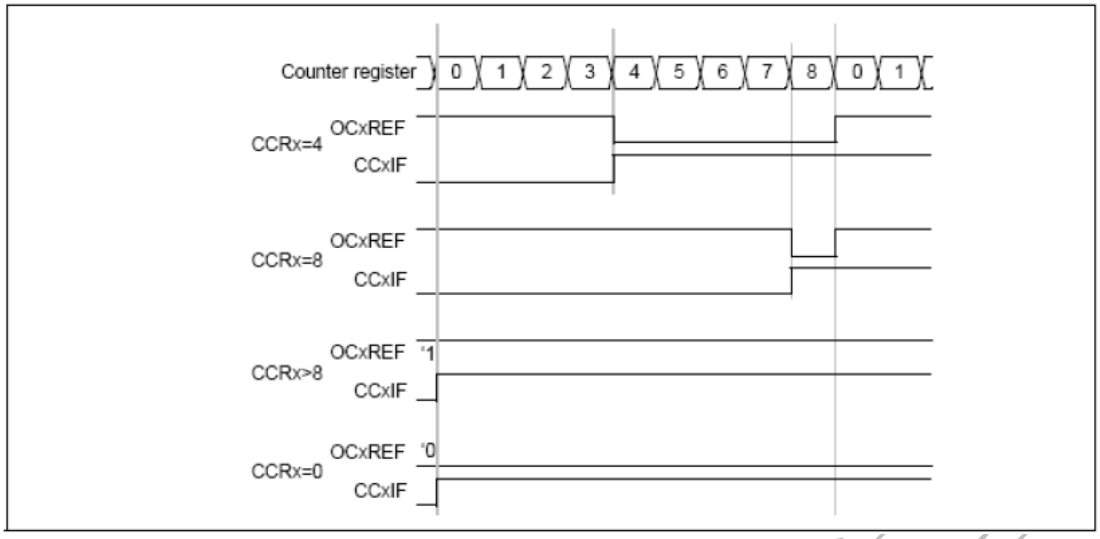
根据TIMx_CR1 寄存器中CMS 位的状态，定时器能够产生边沿对齐的或中央对齐的PWM信号。

PWM 边沿对齐模式

向上计数配置

当TIMx_CR1 寄存器中的DIR位为低的时候执行向上计数。下面是一个PWM模式1 的例子。当TIMx_CNT<TIMx_CCRx时PWM信号OCxREF为高，否则为低。如果TIMx_CCRx中的比较值大于自动重装载值(TIMx_ARR)，则OCxREF保持为“1”。如果比较值为0，则OCxREF保持为“0”。 下图为TIMx_ARR=8 时边沿对齐的PWM波形实例。

奋斗版开发工作室



向下计数的配置

当TIMx_CR1 寄存器的DIR位为高时执行向下计数。在PWM 模式1，当TIMx_CNT>TIMx_CCRx 时OCxREF 为低，否则为高。如果TIMx_CCRx 中的比较值大于TIMx_ARR 中的自动重装载值，则OCxREF 保持为“1”。该模式下不能产生0%的PWM 波形。

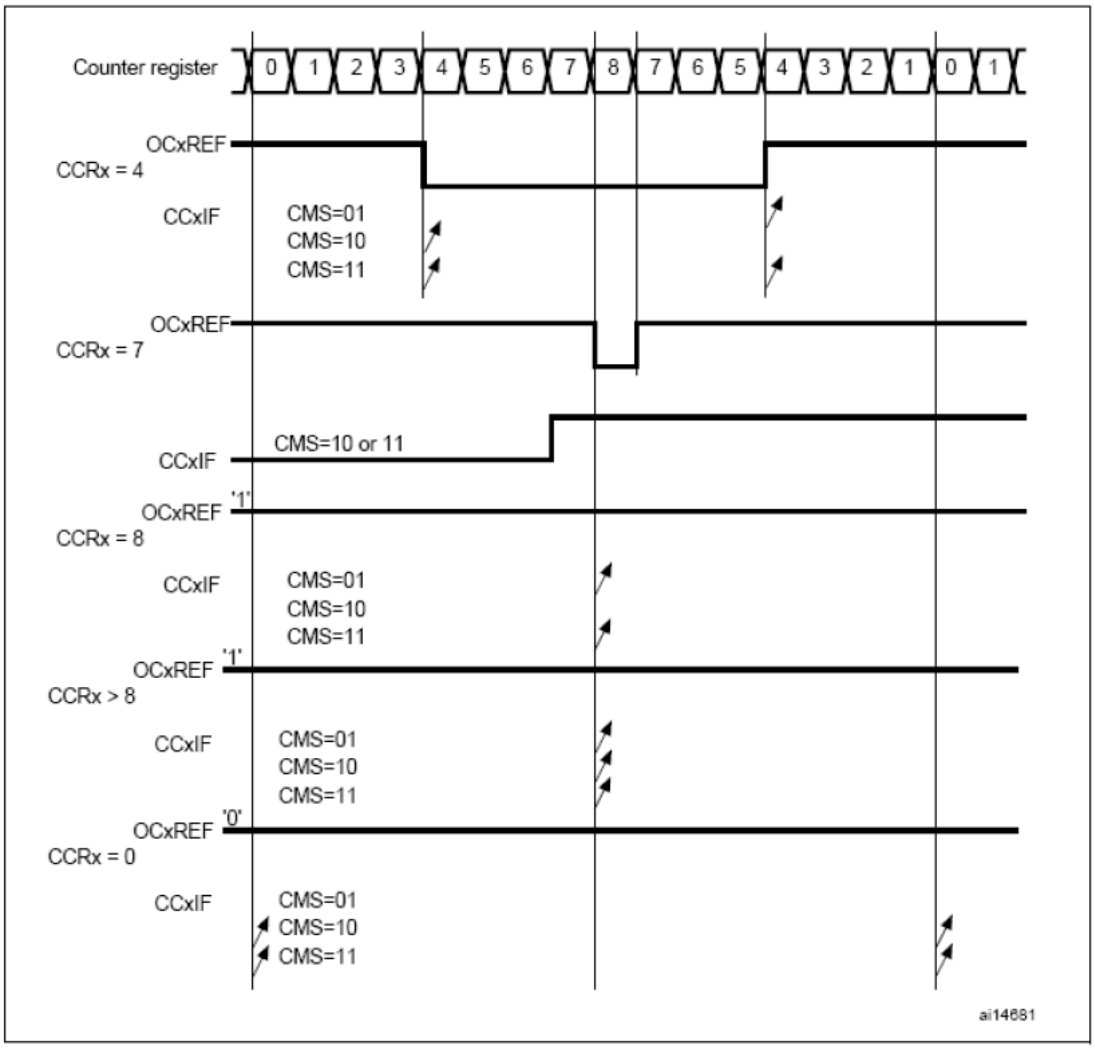
PWM 中央对齐模式

当TIMx_CR1 寄存器中的CMS位不为00 时为中央对齐模式(所有其他的配置对OCxREF/OCx信号都有相同的作用)。根据不同的CMS位的设置，比较标志可能在计数器向上计数时被置1、在计数器向下计数时被置1、或在计数器向上和向下计数时被置1。TIMx_CR1 寄存器中的计数方向位(DIR)由硬件更新，不要用软件修改它。

下图 给出了一些中央对齐的PWM波形的例子

- TIMx_ARR=8
- PWM 模式1
- TIMx_CR1 寄存器中的CMS=01，在中央对齐模式1 时，当计数器向下计数时标志被设置

奋斗嵌入开发工作室



使用中央对齐模式的提示

● 进入中央对齐模式时，当前的上-下配置被使用；这就意味着计数器向上还是向下计数取决于TIMx_CR1 寄存器中DIR 位的当前值。此外，DIR 和CMS 位不能同时被软件修改。

● 不推荐当运行在中央对齐模式时改写计数器，因为会产生不可预知的结果。

特别地：

– 如果写入计数器的值大于自动重加载的值(TIMx_CNT>TIMx_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。

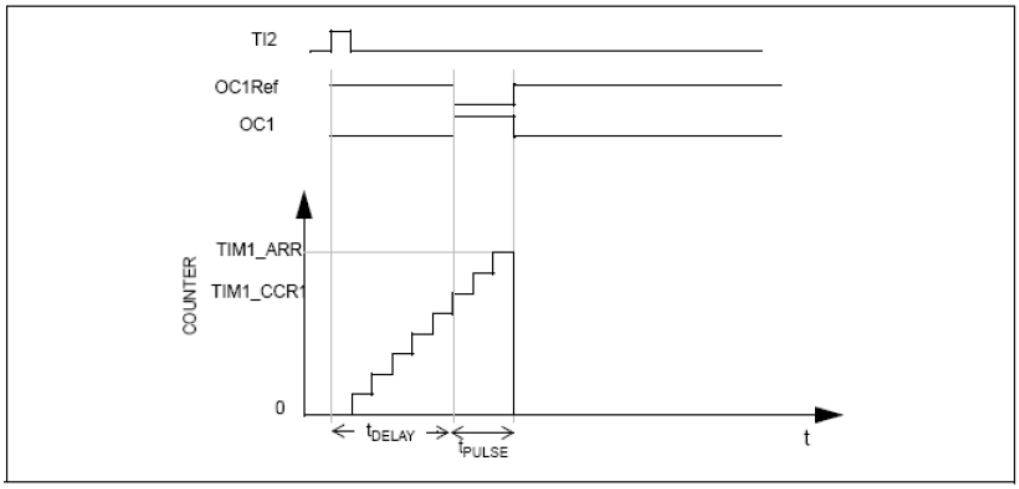
– 如果将0 或者TIMx_ARR 的值写入计数器，方向被更新，但不产生更新事件UEV。

使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新(设置TIMx_EGR 位中的UG 位)，不要在计数进行过程中修改计数器的值。

3.4.10 单脉冲模式

单脉冲模式(OPM)时前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个可编程延时之后产生一个可对脉宽可编程的脉冲。可以通过从模式控制器启动计数器，在输出比较模式或者PWM 模式下产生波形。设置TIMx_CR1 寄存器中的OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件UEV 时停止。仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前(当定时器正在等待触发)，必须如下配置：

向上计数方式：CNT<CCRx≤ARR (特别地, 0<CCRx)，
向下计数方式：CNT>CCRx。



例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定TI2FP2 作为触发1:

- 置TIMx_CCMR1 寄存器中的IC2S=01，把TI2FP2 映像到TI2。
- 置TIMx_CCER 寄存器中的CC2P=0，使TI2FP2 能够检测上升沿。
- 置TIMx_SMCR 寄存器中的TS=110，配置TI2FP2 作为从模式控制器的触发(TRGI)。
- 置TIMx_SMCR 寄存器中的SMS=110(触发模式)，TI2FP2 被用来启动计数器。
- OPM 波形由写入比较寄存器决定(要考虑时钟频率和计数器预分频器)
- t_{DELAY} 由写入TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义(TIMx_ARR - TIMx_CCR1)。
- 假定当发生比较匹配时要产生从0到1的波形，当计数器到达预装载值是要产生一个从1到0的波形；首先要置TIMx_CCMR1寄存器中的OC1M=111进入PWM模式2；有选择的置TIMx_CCMR1中的OC1PE=1和TIMx_CR1寄存器中的ARPE，使能预装载寄存器；然后在TIMx_CCR1寄存器中填写比较值，在TIMx_ARR寄存器中填写自动装载值，修改UG位来产生一个更新事件，然后等待在TI2上的一个外部触发事件。本例中，CC1P=0。在这个例子中，TIMx_CR1 寄存器中的DIR 和CMS 位应该置低。因为只需一个脉冲，所以须在下一个更新事件(当计数器从自动装载值翻转到0)时设置TIMx_CR1 寄存器中的OPM=1，以停止计数。

特殊情况：OCx快速使能

在单脉冲模式下，在TIx 输入脚的边沿检测逻辑设置CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时tDELAY。如果要以最小延时输出波形，可以设置TIMx_CCMRx 寄存器中的OCxFE 位；此时OCxREF(和OCx)被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为PWM1 和PWM2 模式时起作用。

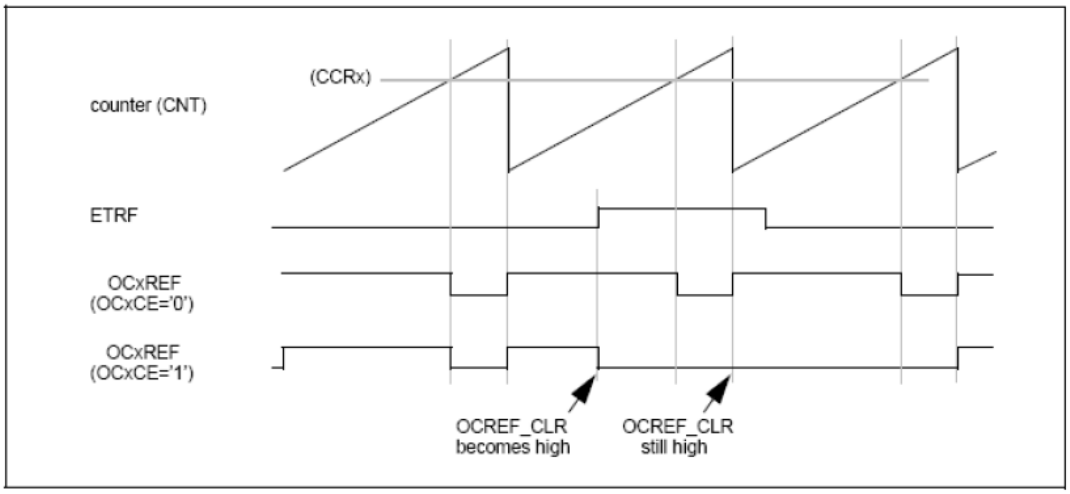
3.4.11 在外部事件时清除OCxREF信号

对于一个给定的通道，在ETRF 输入端(TIMx_CCMRx 寄存器中对应的OCxCE允许位置“1”)的高电平能够把OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件UEV。该功能只能用于输出比较和PWM 模式，而不能用于强置模式。例如，OCxREF

信号可以联到一个比较器的输出，用于处理电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭： TIMx_SMCR 寄存器中的ETPS[1:0]=00。
2. 必须禁止外部时钟模式2：TIMx_SMCR寄存器中的ECE=0。
3. 外部触发极性(ETP)和外部触发滤波器(ETF)可以根据需要配置。

清除TIMx 的OCxREF



3.4.12 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在TI2 的边沿计数，则置TIMx_SMCR 寄存器中的SMS=001；如果只在TI1 边沿计数，则置SMS=010；如果计数器同时在TI1 和TI2 边沿计数，则置SMS=011。通过设置TIMx_CCER 寄存器中的CC1P 和CC2P 位，可以选择TI1 和TI2 极性；如果需要，还可以对输入滤波器编程。两个输入TI1 和TI2 被用来作为增量编码器的接口。参看表39，假定计数器已经启动(TIMx_CR1 寄存器中的CEN=1)，则计数器由每次在TI1FP1 或TI2FP2 上的有效跳变驱动。TI1FP1 和TI2FP2 是TI1 和TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则TI1FP1=TI1；如果没有滤波和变相，则TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，因此TIMx_CR1 寄存器的DIR位由硬件进行相应的设置。不管计数器是依靠TI1 计数、依靠TI2 计数或者同时依靠TI1 和TI2 计数。在任一输入(TI1 或者TI2)跳变时都会重新计算DIR位。编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在0 到TIMx_ARR 寄存器中自动装载值之间连续计数(根据方向，或是0到ARR 计数，或是ARR 到0 计数)。所以在开始计数之前必须配置TIMx_ARR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此，它的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的可能的组合，假设TI1 和TI2 不同时变换。

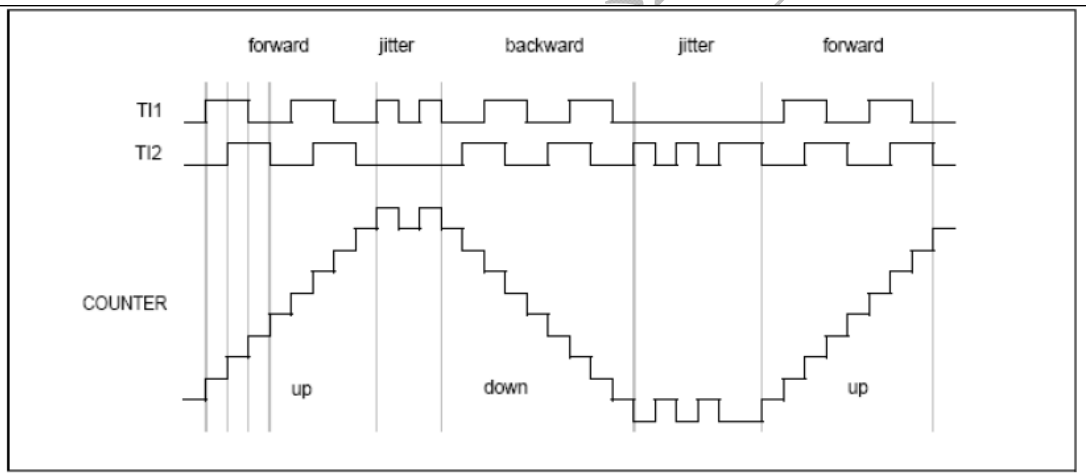
有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数

仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

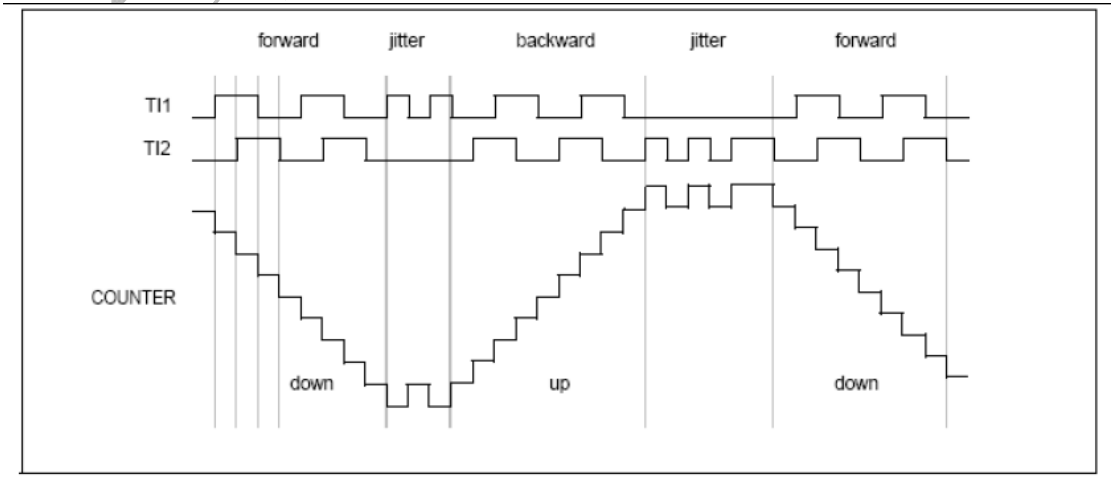
一个外部的增量编码器直接和MCU 连接不需要外部接口逻辑。但是，一般使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以连接到一个外部中断输入和触发一个计数器复位。

下图 给出一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；这种情况可能会在传感器的位置靠近一个转换点事发生。在这个例子中，我们假定配置如下：

- CC1S=’01’ (TIMx_CCMR1 寄存器，IC1FP1 映射到TI1)
- CC2S=’01’ (TIMx_CCMR2 寄存器，IC2FP2 映射到TI2)
- CC1P=’0’ (TIMx_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1)
- CC2P=’0’ (TIMx_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2)
- SMS=’011’ (TIMx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效).
- CEN=’1’ (TIMx_CR1 寄存器，计数器使能)



下图为当IC1FP1 极性反相时计数器的操作实例(CC1P=’1’，其他配置与上例相同)



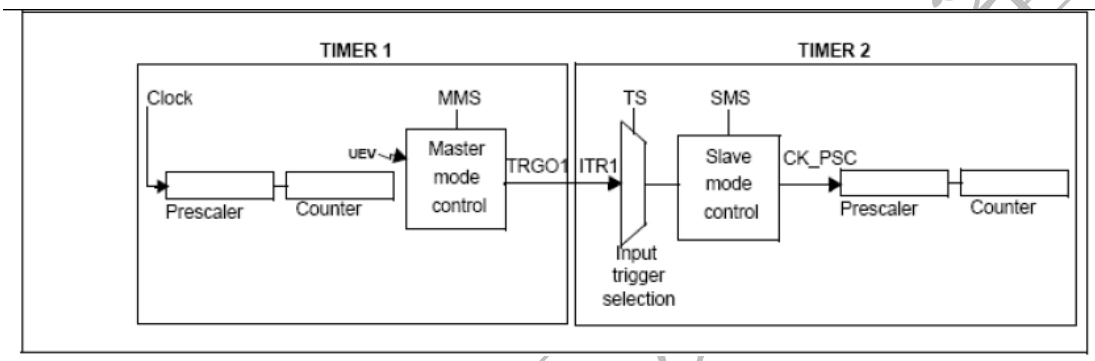
当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模

式定时器测量两个编码器事件的间隔，可以获得动态的信息(速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生)。它也可以通过一个由实时时钟产生的DMA 请求来读取它的值。

3.4.15 定时器同步

所有TIMx 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。
下图显示了触发选择和主模式选择模块的概况。

使用一个定时器作为另一个的预分频器



如：可以配置定时器1 作为定时器2 的预分频器。参考图113，进行下述操作：

● 配置定时器1 为主模式，它可以在每一个更新事件UEV 时输出一个周期的触发信号。在TIM1_CR2 寄存器的MMS='010'时，每次产生一个更新事件时在TRGO1 上产生一个上升沿信号。

● 连接定时器1 的TRGO1 输出至定时器2，定时器2 必须配置为使用ITR1作为内部触发的从模式。设置TIM2_SMCR 寄存器的TS='001'。

● 然后把从模式控制器置于外部时钟模式1(TIM2_SMCR 寄存器的SMS=111)；这样定时器2 即可由定时器1 周期的上升沿(即定时器1 的计数器溢出)信号驱动。

● 最后，必须设置相应(TIMx_CR1 寄存器)的CEN 位分别启动两个定时器。

注： 如果*OCx* 已被选中为定时器*1* 的触发输出*(MMS=1xx)* ，它的上升沿用于驱动定时器*2* 的计数器。

使用一个定时器去使能另一个定时器

在这个例子中，定时器2 的由定时器1 的输出比较启动。定时器2 只当定时器1 的OC1REF 为高时才对分频的内部时钟计数。两个定时器的时钟频率都是由预分频器对CK_INT除以3($f_{CK_CNT}=f_{CK_INT}/3$)。

● 配置定时器1 为主模式，送出它的输出比较参考信号(OC1REF)为触发输出(TIM1_CR2 寄存器的MMS=100)

● 配置定时器1 的OC1REF 波形(TIM1_CCMR1 寄存器)

● 配置定时器2 从定时器1 获得输入触发(TIM2_SMCR 寄存器的TS=001)

● 配置定时器2 为门控模式(TIM2_SMCR 寄存器的SMS=101)

● 置TIM2_CR1 寄存器的CEN=1 以使能定时器2

● 置TIM1_CR1 寄存器的CEN=1 以启动定时器1

注： 定时器*2* 的时钟不与定时器*1* 的时钟同步，这个模式只影响定时器*2* 计数器的启动信号。

使用一个定时器去启动另一个定时器

在这个例子中，使用定时器1 的更新事件使能定时器2。一旦定时器1 产生更新事件，定时器2 即从它当前的数值(可以是非0)依分频的内部时钟开始计数。在收到触发信号时，定时器2 的CEN位被自动地置1，同时计数器开始计数直到写0 到TIM2_CR1 寄存器的CEN位。两个定时器的时钟频率都是由预分频器对CK_INT除以3($f_{CK_CNT}=f_{CK_INT}/3$)。

- 配置定时器1 为主模式，送出它的更新事件(UEV)做为触发输出(TIM1_CR2寄存器的MMS=010)。
- 配置定时器1 的周期(TIM1_ARR 寄存器)。
- 配置定时器2 从定时器1 获得输入触发(TIM2_SMCR 寄存器的TS=001)
- 配置定时器2 为触发模式(TIM2_SMCR 寄存器的SMS=110)
- 置TIM1_CR1 寄存器的CEN=1 以启动定时器1

使用一个定时器作为另一个的预分频器

如：可以配置定时器1 作为定时器2 的预分频器。参考图113，进行下述操作：

- 配置定时器1 为主模式，送出它的更新事件UEV 做为触发输出(TIM1_CR2寄存器的MMS=’010’)。每次计数器溢出时输出一个周期信号。
- 配置定时器1 的周期(TIM1_ARR 寄存器)。
- 配置定时器2 从定时器1 获得输入触发(TIM2_SMCR 寄存器的TS=001)
- 配置定时器2 使用外部时钟模式(TIM2_SMCR 寄存器的SMS=111)
- 置TIM1_CR2 寄存器的CEN=1 以启动定时器2。
- 置TIM1_CR1 寄存器的CEN=1 以启动定时器1。

使用一个外部触发同步地启动2 个定时器

在这个例子中，设置当定时器1 的TI1 输入上升时使能定时器1，使能定时器1 的同时使能定时器2。为保证计数器的对齐，定时器1 必须配置为主/从模式(对应TI1 为从，对应定时器2 为主)：

- 配置定时器1 为主模式，送出它的使能做为触发输出(TIM1_CR2 寄存器的MMS=’001’)。
- 配置定时器1 为从模式，从TI1 获得输入触发(TIM1_SMCR 寄存器的TS=’100’)。
- 配置定时器1 为触发模式(TIM1_SMCR 寄存器的SMS=’110’)。
- 配置定时器1 为主/从模式，TIM1_SMCR 寄存器的MSM=’1’。
- 配置定时器2 从定时器1 获得输入触发(TIM2_SMCR 寄存器的TS=001)
- 配置定时器2 为触发模式(TIM2_SMCR 寄存器的SMS=’110’)。

当定时器1 的TI1 上出现一个上升沿时，两个定时器同步地依内部时钟开始计数，两个TIF标志也同时设置。

3.5 TIMx寄存器描述

3.5.1 控制寄存器1(TIMx_CR1)

偏移地址：0x00
复位值：0x0000

奋斗嵌入开发工作室

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:10	保留，始终读为0。
位9:8	CKD[1:0]: 时钟分频因子 这2位定义在定时器时钟(CK_INT)频率与数字滤波器(ETR,TIx)使用的采样频率之间的分频比例。 00：$t_{DTS} = t_{CK_INT}$ 01：$t_{DTS} = 2 \times t_{CK_INT}$ 10：$t_{DTS} = 4 \times t_{CK_INT}$ 11：保留
位7	ARPE：自动重装载预装载允许位 0：TIMx_ARR寄存器没有缓冲 1：TIMx_ARR寄存器被装入缓冲器
位6:5	CMS[1:0]：选择中央对齐模式 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(TIMx_CCMRx寄存器中CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式2。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx寄存器中CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式3。计数器交替地向上和向下计数。计数器交替地向上和向下计数。配置为输出的通道(TIMx_CCMRx寄存器中CCxS=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设置。 注：在计数器开启时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。
位4	DIR：方向 0：计数器向上计数 1：计数器向下计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
位3	OPM：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。

奋斗版

位2	URS：更新请求源 软件通过该位选择UEV事件的源 0：如果允许产生更新中断或DMA请求，则下述任一事件产生一个更新中断或DMA请求： – 计数器溢出/下溢 – 设置UG位 – 从模式控制器产生的更新 1：如果允许产生更新中断或DMA请求，则只有计数器溢出/下溢产生一个更新中断或DMA请求
位1	UDIS：禁止更新 软件通过该位允许/禁止UEV事件的产生 0：允许UEV。更新(UEV)事件由下述任一事件产生： – 计数器溢出/下溢 – 设置UG位 – 从模式控制器产生的更新 被缓存的寄存器被装入它们的预装载值。 1：禁止UEV。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位或从模式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。
位0	CEN：允许计数器 0：禁止计数器 1：开启计数器 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。触发模式可以自动地通过硬件设置CEN位。 在单脉冲模式下，当发生更新事件时，CEN被自动清除。

3.5.2 控制寄存器2(TIMx_CR2)

偏移地址：0x04
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TI1S	MMS[2:0]			CCDS	保留		
								rw	rw	rw	rw	rw			

位15:8	保留，始终读为0。
位7	TI1S：TI1选择 0：TIMx_CH1管脚连到TI1输入。 1：TIMx_CH1、TIMx_CH2和TIMx_CH3管脚经异或后连到TI1输入。 见上一章的“与霍尔元件的接口”一节。

位6:4	MMS[1:0]：主模式选择 这两位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000：复位 – TIMx_EGR寄存器的UG位被用于作为触发输出(TRGO)。如果触发输入(复位模式下的从模式控制器)产生复位，则TRGO上的信号相对实际的复位会有一个延迟。 001：允许 – 计数器使能信号CNT_EN被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制从定时器的一个窗口。计数器使能信号是通过CEN控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO上会有一个延迟，除非选择了主/从模式(见TIMx_SMCR寄存器中MSM位的描述)。 010：更新 – 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011：比较脉冲 – 一旦发生一次捕获或一次比较成功时，当要设置CC1IF标志时(即是它已经为高)，触发输出送出一个正脉冲(TRGO)。 100：比较 – OC1REF信号被用于作为触发输出(TRGO)。 101：比较 – OC2REF信号被用于作为触发输出(TRGO)。 110：比较 – OC3REF信号被用于作为触发输出(TRGO)。 111：比较 – OC4REF信号被用于作为触发输出(TRGO)。
位3	CCDS：捕获/比较的DMA选择 0：当发生CCx事件时，送出CCx的DMA请求。 1：当发生更新事件时，送出CCx的DMA请求。
位2:0	保留，始终读为0。

3.5.3 从模式控制寄存器(TIMx_SMCR)

偏移地址：0x08

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			保留	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位15	ETP：外部触发极性 该位选择是用ETR还是ETR的反相来作为触发操作 0：ETR不反相，高电平或上升沿有效 1：ETR被反相，低电平或下降沿有效
位14	ECE：外部时钟允许位 该位启用外部时钟模式2 0：禁止外部时钟模式2 1：启用外部时钟模式2。计数器由ETRF信号上的任意有效上升沿驱动。 注1：设置ECE位与选择外部时钟模式1并将TRGI连到ETRF(SMS=111和TS=111)具有相同功效。 注2：下述从模式可以与外部时钟模式2同时使用：复位模式，门控模式和触发模式；但是，这时TRGI不能连到ETRF(TS位不能是111)。

位13:12	ETPS[1:0]：外部触发预分频 外部触发信号ETRP的频率必须最多是CK_INT频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETRP的频率。 00：关闭预分频 01：ETRP频率除以2 10：ETRP频率除以4 11：ETRP频率除以8
位11:8	ETF[3:0]：外部触发滤波 这些位定义了对ETRP信号采样的频率和对ETRP数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到N个事件后会产生一个输出的跳变。 0000：无滤波器，以f_{DTS}采样 0001：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
位7	MSM：主/从模式 0：无作用 1：触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
位6:4	TS[2:0]：触发选择 这3位选择用于同步计数器的触发输入。 000：内部触发器0(ITR0)，TIM1 001：内部触发器0(ITR1)，TIM2 010：内部触发器0(ITR1)，TIM3 011：内部触发器0(ITR1)，TIM4 100：TI1的边沿检测器(TI1F_ED) 101：滤波后的定时器输入1(TI1FP1) 110：滤波后的定时器输入2(TI2FP2) 111：外部触发输入(ETRF) 注：为避免在信号转变时产生错误的边沿检测，必须在未使用这些位(如SMS=000)时修改它们。
位3	保留，始终读为0。

位2:0	SMS：从模式选择 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000：关闭从模式 – 如果CEN=1，则预分频器直接由内部时钟驱动。 001：编码器模式1 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。 010：编码器模式2 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。 011：编码器模式3 – 根据其他输入的电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。 100：复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101：门控模式 – 当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入TRGI的上升沿启动(但不复位)，只有计数器的启动是受控的。 111：外部时钟模式1 – 选中的触发输入(TRGI)的上升沿驱动计数器。 注：如果TI1F_EN被选为触发输入(TS=100)时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。

3.5.4 DMA/中断使能寄存器(TIMx_DIER)

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TDE	保留	CC4DE	CC3DE	CC2DE	CC1DE	UDE	保留	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位15	保留，始终读为0。
位14	TDE：允许触发DMA请求 0：禁止触发DMA请求 1：允许触发DMA请求
位13	保留，始终读为0。
位12	CC4DE：允许捕获/比较4的DMA请求 0：禁止捕获/比较4的DMA请求 1：允许捕获/比较4的DMA请求
位11	CC3DE：允许捕获/比较3的DMA请求 0：禁止捕获/比较3的DMA请求 1：允许捕获/比较3的DMA请求
位10	CC2DE：允许捕获/比较2的DMA请求 0：禁止捕获/比较2的DMA请求 1：允许捕获/比较2的DMA请求
位9	CC1DE：允许捕获/比较1的DMA请求 0：禁止捕获/比较1的DMA请求 1：允许捕获/比较1的DMA请求

位8	UDE：允许更新的DMA请求 0：禁止更新的DMA请求 1：允许更新的DMA请求
位7	保留，始终读为0。
位6	TIE：允许触发中断 0：禁止触发中断 1：允许触发中断
位5	保留，始终读为0。
位4	CC4IE：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
位3	CC3IE：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
位2	CC2IE：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
位1	CC1IE：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
位0	UIE：允许更新中断 0：禁止更新中断 1：允许更新中断

3.5.5 状态寄存器(TIMx_SR)

偏移地址：0x10
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CC4OF	CC3OF	CC2OF	CC1OF	保留		TIF	保留	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc	rc	rc	rc			rc		rc	rc	rc	rc	rc

位15:13	保留，始终读为0。
位12	**CC4OF**：捕获/比较4 过捕获标记 参见CC1OF描述
位11	**CC3OF**：捕获/比较3 过捕获标记 参见CC1OF描述
位10	**CC2OF**：捕获/比较2 过捕获标记 参见CC1OF描述

位9	**CC1OF**：捕获/比较1 过捕获标记 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：无过捕获产生； 1：CC1IF置1时，计数器的值已经被捕获到TIMx_CCR1寄存器。
位8:7	保留，始终读为0。
位6	**TIF**：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：触发器中断等待响应
位5	保留，始终读为0。
位4	**CC4IF**：捕获/比较4 中断标记 参考CC1IF描述
位3	**CC3IF**：捕获/比较3 中断标记 参考CC1IF描述
位2	**CC2IF**：捕获/比较2 中断标记 参考CC1IF描述
位1	**CC1IF**：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TIMx_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。 **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TIMx_CCR1清0。 0：无输入捕获产生； 1：输入捕获产生并且计数器值已装入TIMx_CCR1(在IC1上检测到与所选极性相同的边沿)。
位0	**UIF**：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TIMx_CR1寄存器的UDIS=0，当REP_CNT=0时产生更新事件(重复向下计数器上溢或下溢时)； – 若TIMx_CR1寄存器的UDIS=0、URS=0，当TIMx_EGR寄存器的UG=1时产生更新事件(软件对CNT重新初始化)； – 若TIMx_CR1寄存器的UDIS=0、URS=0，当CNT被触发事件重初始化时产生更新事件。（参考同步控制寄存器的说明）

3.5.6 事件产生寄存器(TIMx_EGR)

偏移地址:0x14

复位值:0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TG	保留	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位15:7	保留，始终读为0。
位6	**TG**：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：TIMx_SR寄存器的TIF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
位5	保留，始终读为0。
位4	**CC4G**：产生捕获/比较4事件 参考CC1G描述
位3	**CC3G**：产生捕获/比较3事件 参考CC1G描述
位2	**CC2G**：产生捕获/比较2事件 参考CC1G描述
位1	**CC1G**：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，若开启对应的中断和DMA，则产生相应的中断和DMA。 **若通道CC1配置为输入：** 当前的计数器值捕获至TIMx_CCR1寄存器，设置CC1IF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。若CC1IF已经为1，则设置CC1OF=1。
位0	**UG**：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TIMx_ARR的值。

3.5.7 捕获/比较模式寄存器1(TIMx_CCMR1)

偏移地址：0x18

复位值：0x0000

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输出模式下的功能。因此你必须注意，同一个位在输出模式和输入模式下的功能是不同的。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位15	OC2CE：输出比较2清0使能
位14:12	0C2M[2:0]：输出比较2模式
位11	OC2PE：输出比较2预装载使能
位10	OC2FE：输出比较2快速使能
位9:8	CC2S[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TI2上； 10：CC2通道被配置为输入，IC2映射在TI1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TIMx_SMCR寄存器的TS位选择）。 注：CC2S仅在通道关闭时(TIMx_CCER寄存器的CC2E=0)才是可写的。
位7	OC1CE：输出比较1清0使能 0：OC1REF 不受ETRF输入的影响； 1：一旦检测到ETRF输入高电平，清除OC1REF=0。

奋斗嵌入开发

位6:4	0C1M[2:0]：输出比较1模式 该位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。OC1REF是高电平有效，而OC1、OC1N的有效电平取决于CC1P、CC1NP位。 000：冻结。输出比较寄存器TIMx_CCR1与计数器TIMx_CNT间的比较对OC1REF不起作用； 001：匹配时设置通道1为有效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时，强制OC1REF为高。 010：匹配时设置通道1为无效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时，强制OC1REF为低。 011：翻转。当TIMx_CCR1=TIMx_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TIMx_CNT<TIMx_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TIMx_CNT>TIMx_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TIMx_CNT<TIMx_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TIMx_CNT>TIMx_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(TIMx_BDTR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。
位3	**OC1PE**：输出比较1预装载使能 0：禁止TIMx_CCR1寄存器的预装载功能，可随时写入TIMx_CCR1寄存器，且新值马上起作用。 1：开启TIMx_CCR1寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1的预装载值在更新事件到来时被载入当前寄存器中。 注1：一旦LOCK级别设为3(TIMx_BDTR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：仅在单脉冲模式下，可以在未确认预装载寄存器情况下使用PWM模式，否则其动作不确定。
位2	**OC1FE**：输出比较1 快速使能 该位用于加快CC输出对触发器输入事件的响应。 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的只在通道被配置成PWM1或PWM2模式时起作用。
位1:0	CC1S[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TI1上； 10：CC1通道被配置为输入，IC1映射在TI2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TIMx_SMCR寄存器的TS位选择）。 注：CC1S仅在通道关闭时(TIMx_CCER寄存器的CC1E=0)才是可写的。

输入捕获模式：

位15:12	IC2F：输入捕获2滤波器
位11:10	IC2PSC[1:0]：输入/捕获2预分频器
位9:8	CC2S[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TI2上； 10：CC2通道被配置为输入，IC2映射在TI1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TIMx_SMCR寄存器的TS位选择）。 注：CC2S仅在通道关闭时(TIMx_CCER寄存器的CC2E=0)才是可写的。
位7:4	IC1F[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，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以f_{DTS}采样 0001：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=2 0010：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=4 0011：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=8 0100：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8 注：在现在的芯片版本中，当ICxF[3:0]=1,2或3时，公式中的fDTS由CK_INT替代。
位3:2	IC1PSC[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(TIMx_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。
位1:0	**CC1S[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TI1上； 10：CC1通道被配置为输入，IC1映射在TI2上；

3.5.8 捕获/比较模式寄存器2(TIMx_CCMR2)

偏移地址：0x1C

复位值：0x0000

参看以上CCMR1 寄存器的描述

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位15	OC4CE：输出比较4清0使能
位14:12	0C4M[2:0]：输出比较4模式
位11	OC4PE：输出比较4预装载使能
位10	OC4FE：输出比较4快速使能
位9:8	CC4S[1:0]：捕获/比较4选择。 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCR寄存器的TS位选择）。 注：CC4S仅在通道关闭时(TIMx_CCER寄存器的CC4E=0)才是可写的。
位7	OC3CE：输出比较3清0使能
位6:4	0C3M[2:0]：输出比较3模式
位3	OC3PE：输出比较3预装载使能
位2	OC3FE：输出比较3快速使能
位1:0	CC3S[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TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRGI上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCR寄存器的TS位选择）。 注：CC3S仅在通道关闭时(TIMx_CCER寄存器的CC3E=0)才是可写的。

奋斗嵌

输入捕获模式：

位15:12	IC4F：输入捕获4滤波器
位11:10	IC4PSC[1:0]：输入/捕获4预分频器
位9:8	CC2S[1:0]：捕获/比较4选择。 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCR寄存器的TS位选择）。 注：CC4S仅在通道关闭时(TIMx_CCER寄存器的CC4E=0)才是可写的。
位7:4	IC3F[3:0]：输入捕获3滤波器
位3:2	IC3PSC[1:0]：输入/捕获3预分频器
位1:0	CC3S[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TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3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TIMx_SMCR寄存器的TS位选择）。 注：CC3S仅在通道关闭时(TIMx_CCER寄存器的CC3E=0)才是可写的。

3.5.9 捕获/比较使能寄存器(TIMx_CCER)

偏移地址：0x20
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CC4P	CC4E	保留		CC3P	CC3E	保留		CC2P	CC2E	保留		CC1P	CC1E
		rw	rw			rw	rw			rw	rw			rw	rw

位15:14	保留，始终读为0。
位13	CC4P：输入/捕获4输出极性。参考CC1P的描述。
位12	CC4E：输入/捕获4输出使能。参考CC1E 的描述。
位11:10	保留，始终读为0。
位9	CC3P：输入/捕获3输出极性。参考CC1P的描述。
位8	CC3E：输入/捕获3输出使能。参考CC1E 的描述。
位7:6	保留，始终读为0。
位5	CC2P：输入/捕获2输出极性。参考CC1P的描述。
位4	CC2E：输入/捕获2输出使能。参考CC1E的描述。
位3:2	保留，始终读为0。

位1	CC1P：输入/捕获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通道配置为输入：** 该位选择是IC1还是IC1的反相信号作为触发或捕获信号。 0：不反相：捕获发生在IC1的上升沿；当用作外部触发器时，IC1不反相。 1：反相：捕获发生在IC1的下降沿；当用作外部触发器时，IC1反相。
位0	CC1E：输入/捕获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TIMx_CCR1寄存器。 0：捕获禁止； 0：捕获使能。

标准OCx 通道的输出控制位

CCxE位	OCx输出状态
0	禁止输出(OCx=0, OCx_EN=0)
1	OCx=OCxREF + 极性，OCx_EN=1

注：　连接到标准OCx通道的外部I/O管脚状态，取决于OCx通道状态和GPIO以及AFIO寄存器。

3.5.10 计数器(TIMx_CNT)

偏移地址：0x24
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	CNT[15:0]：计数器的值

3.5.11 预分频器(TIMx_PSC)

偏移地址：0x28
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	PSC[15:0]：预分频器的值 计数器的时钟频率CK_CNT等于fCK_PSC/(PSC[15:0]+1)。 PSC包含了当更新事件产生时装入当前预分频器寄存器的值。

3.5.12 自动重装载寄存器(TIMx_ARR)

偏移地址:0x2C
复位值:0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	ARR[15:0]: 自动重装载的值 ARR是将要装载入实际的自动重装载寄存器的数值。 详细参考13.4.1：时基单元有关ARR的更新和动作。 当自动重装载的值为空时，计数器不工作。

3.5.13 捕获/比较寄存器1(TIMx_CCR1)

偏移地址：0x34
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	CCR1[15:0]: 捕获/比较1的值 **若CC1通道配置为输出：** CCR1是装入当前捕获/比较1寄存器的值（预装载值）。 如果在TIMx_CCMR1寄存器(OC1PE位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较1寄存器中。 当前捕获/比较寄存器包含了与计数器TIMx_CNT比较的值，并且在OC1端口上输出信号。 **若CC1通道配置为输入：** CCR1包含了由上一次输入捕获1事件（IC1）传输的计数器值。

3.5.14 捕获/比较寄存器2(TIMx_CCR2)

偏移地址：0x38
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	CCR2[15:0]: 捕获/比较2的值 **若CC2通道配置为输出：** CCR2是装入当前捕获/比较2寄存器的值（预装载值）。 如果在TIMx_CCMR2寄存器(OC2PE位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较2寄存器中。 当前捕获/比较寄存器包含了与计数器TIMx_CNT比较的值，并且在OC端口上输出信号。 **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。

3.5.15 捕获/比较寄存器3(TIMx_CCR3)

偏移地址：0x3C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	CCR3[15:0]: 捕获/比较3的值 **若CC3通道配置为输出：** CCR3是装入当前捕获/比较3寄存器的值（预装载值）。 如果在TIMx_CCMR3寄存器(OC3PE位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较3寄存器中。 当前捕获/比较寄存器包含了与计数器TIMx_CNT比较的值，并且在OC端口上输出信号。 **若CC3通道配置为输入：** CCR3包含了由上一次输入捕获3事件（IC3）传输的计数器值。

3.5.16 捕获/比较寄存器4(TIMx_CCR4)

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	CCR4[15:0]: 捕获/比较4的值 **若CC4通道配置为输出：** CCR4是了装入当前捕获/比较4寄存器的值（预装载值）。 如果在TIMx_CCMR4寄存器(OC4PE位)中未选择预装载特性，其始终装入当前寄存器中。否则，只有当更新事件发生时，此预装载值才装入当前捕获/比较4寄存器中。 当前捕获/比较寄存器包含了与计数器TIMx_CNT比较的值，并且在OC端口上输出信号。 **若CC4通道配置为输入：** CCR4包含了由上一次输入捕获4事件（IC4）传输的计数器值。

3.5.17 DMA控制寄存器(TIMx_DCR)

偏移地址：0x48
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			DBL[4:0]					保留			DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位15:13	保留，始终读为0。
位12:8	DBL[4:0]: DMA连续传送长度 这些位定义了DMA在连续模式下的传送长度（当对TIMx_DMAR寄存器的地址进行读或写时，定时器则进行一次连续传送），即：定义被传送的字节数目： 00000：1 字节　　00001：2 字节 00010：3 字节　　…… ……　　10001：18 字节
位7:5	保留，始终读为0。
位4:0	DBA[4:0]: DMA基地址 这些位定义了DMA在连续模式下的基地址（当对TIMx_DMAR寄存器的地址进行读或写时），DBA定义为从TIMx_CR1寄存器所在地址开始的偏移量： 00000：TIMx_CR1， 00001：TIMx_CR2， 00010：TIMx_SMCR， ……

3.5.18 连续模式的DMA地址(TIMx_DMAR)

偏移地址：0x4C
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位15:0	DMAB[15:0]: DMA连续传送寄存器 对TIMx_DMAR寄存器的读或写会导致对以下地址的寄存器的存取操作： TIMx_CR1地址 + DBA + DMA指针，其中： “TIMx_CR1地址”是控制寄存器1的地址； “DBA”是TIMx_DCR寄存器中定义的基地址； “DMA指针”是由DMA自动控制的偏移量，它取决于TIMx_DCR寄存器中定义的DBL。

3.6 TIMx寄存器图

下表中将TIMx 的所有寄存器映射到一个16 位可寻址(编址)空间

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIMx_CR1	保留																						CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	复位值																							0	0	0	0	0	0	0	0	0	0
004h	TIMx_CR2	保留																								TI1S	MMS [2:0]			CCDS	保留		
	复位值																									0	0	0	0	0			
008h	TIMx_SMCR	保留																ETP	ECE	ETPS [1:0]		EFT[3:0]				MSM	TS [2:0]			保留	SMS [2:0]		
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
00Ch	TIMx_DIER	保留																	TDE	保留	CC4DE	CC3DE	CC2DE	CC1DE	UDE	保留	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	复位值																		0		0	0	0	0	0		0		0	0	0	0	0
010h	TIMx_SR	保留																			CC4OF	CC3OF	CC2OF	CC1OF		保留	TIF	保留	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	复位值																				0	0	0	0			0		0	0	0	0	0
014h	TIMx_EGR	保留																								保留	TG	保留	CC4G	CC3G	CC2G	CC1G	UG
	复位值																										0		0	0	0	0	0
018h	TIMx_CCMR1 输出比较模式	保留																OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 输入捕获模式	保留																IC2F [3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	TIMx_CCMR2 输出比较模式	保留																OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 输入捕获模式	保留																IC4F [3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F [3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	TIMx_CCER	保留																		CC4P	CC4E	保留		CC3P	CC3E	保留		CC2P	CC2E	保留		CC1P	CC1E
	复位值																			0	0			0	0			0	0			0	0
024h	TIMx_CNT	保留																CNT[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	TIMx_PSC	保留																PSC[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
02Ch	TIMx_ARR	保留																ARR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	保留																																
034h	TIMx_CCR1	保留																CCR1[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	TIMx_CCR2	保留																CCR2[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	TIMx_CCR3	保留																CCR3[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	TIMx_CCR4	保留																CCR4[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	保留																																
048h	TIMx_DCR	保留																			DBL[4:0]					保留			DBA[4:0]				
	复位值																				0	0	0	0	0				0	0	0	0	0
04Ch	TIMx_DMAR	保留																DMAB[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4. 应用实例

4.1. 设计要求

对TIM3定时器进行控制，使得PB5-TIM3通道2产生频率为12.5Hz的方波，该方波控制LED1的闪烁。

4.2 硬件电路设计

LED1（MINI的V2标号）（V2、V2.1、V3、V5板的V6标号）与CPU的PB5相连。

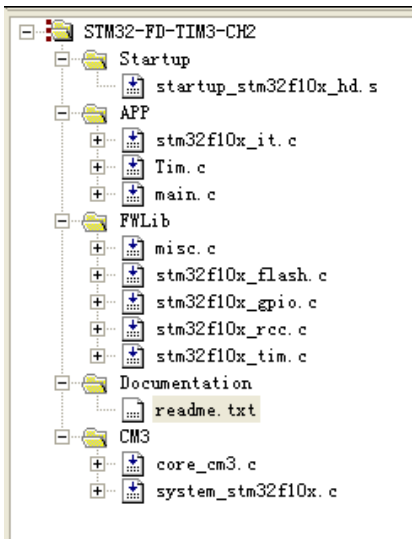
4.3 软件程序设计

根据任务要求，程序内容主要包括：

1. 配置TIM3的通道2为输出比较模式（预分频系数设置为63），TIM3_CCR2=45000。
2. 在相应的TIM3定时器中断服务处理程序中根据定时器的值翻转输出电平。以输出方波。

整个工程包含4类源文件：

奋斗嵌入开发



Startup--startup_stm32f10x_hd.s 由于奋斗板采用的是STM32F103大存储器芯片，因此采用STM32标准库自带的大存储器芯片启动代码，这个文件已经配置好了初始状态，以及中断向量表。可以直接在工程里使用，如果你在以后的应用中采用了中存储器或者小存储器STM32芯片，可以将启动代码换为startup_stm32f10x_md.s 或者 startup_stm32f10x_ld.s。

FWLIB--stm32f10x_gpio.c ST公司的标准库，包含了关于对通用IO口设置的函数。
stm32f10x_rcc.c ST公司的标准库，包含了关于对系统时钟设置的函数。
stm32f10x_USART.c ST公司的标准库，包含了关于对USART设置的函数。
stm32f10x_flash.c ST公司的标准库，包含了关于对flash设置的函数。
Misc.c ST公司的标准库，包含了关于中断设置的函数。

CM3—是关于CORETEX-M3平台的系统函数及定义

App—main.c 例程的主函数。

App—tim.c 定时器3的设置。

App--stm32f10x_it.c 中断服务程序

//_____

```
int main(void)
{
  /* System Clocks Configuration --72M*/
  RCC_Configuration();
  /* NVIC configuration */
  /*嵌套向量中断控制器
     说明了定时器3抢占优先级级别0（最多1位） ，和子优先级级别0（最多7位） */
  NVIC_Configuration();
  /*定时器3的初始化 */
  time_ini();
  while(1);
}


void time_ini(void){
  GPIO_InitTypeDef GPIO_InitStructure;

  RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOB , ENABLE);
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);

  /* GPIOA Configuration: Channel 1 Output */
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;                    //PB5复用为TIM3的通道2
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOB, &GPIO_InitStructure);
```

```
  GPIO_PinRemapConfig(GPIO_PartialRemap_TIM3 , ENABLE);//TIM3局部复用功能开启 在TIM3的局部复用开启时，PB5会被复用为TIM3_CH2

  /* Time Base configuration */
  /*-----------------------------------------------------------------
  TIM3CLK=72MHz  预分频系数Prescaler=63 经过分频 定时器时钟为1.125MHz
  捕获/比较寄存器2 TIM3_CCR2= CCR2_Val
  2通道产生的更新频率是=1.125MHz/CCR2_Val=25Hz

  -----------------------------------------------------------------*/
  TIM3_TimeBaseStructure.TIM_Prescaler = 63;                        //预分频器TIM3_PSC=63
  TIM3_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;      //计数器向上计数模式 TIM3_CR1[4]=0
  TIM3_TimeBaseStructure.TIM_Period =0xffff;                        //自动重装载寄存器TIM3_APR
  TIM3_TimeBaseStructure.TIM_ClockDivision = 0x0;                   //时钟分频因子 TIM3_CR1[9:8]=00

  TIM_TimeBaseInit(TIM3,&TIM3_TimeBaseStructure);                   //写TIM3各寄存器参数

  TIM3_OCInitStructure.TIM_OCMode = TIM_OCMode_Toggle ;  //TIM3_CCMR1[14:12]=011  翻转 当TIM3_CCR2=TIM3_CNT时，翻转OC2REF的电平
  TIM3_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;//输入/捕获2输出允许  OC2信号输出到对应的输出引脚PB5
  TIM3_OCInitStructure.TIM_Pulse =CCR2_Val;  //若CC1通道配置为输出：CCR2是装入当前捕获/比较2 TIM3_CCR2寄存器的值（预装载值）。
                                             //当前捕获/比较寄存器包含了与计数器TIM3_CNT比较的值，并且在OC端口上输出信号
  TIM3_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;                //输出极性  低电平有效 TIM3_CCER[5]=1;

  TIM_OC2Init(TIM3, &TIM3_OCInitStructure);
  TIM_OC2PreloadConfig(TIM3, TIM_OCPreload_Disable);   //TIM3_CCMR1[1]=0  禁止TIM3_CCR2寄存器的预装载功能，可随时写入TIM3_CCR2
                                                       //且新值马上起作用
  /* TIM1 counter enable */
   TIM_Cmd(TIM3,ENABLE);                               //启动定时器3 TIM3_CR1[0]=1;
   TIM_ITConfig(TIM3,TIM_IT_CC2,ENABLE);               //TIM3_DIER[2]=1  允许捕获/比较2中断
 }


/*******************************************************************************
* Function Name  : TIM3_IRQHandler
* Description    : This function handles TIM3 global interrupt request.        产生12.5HZ的频率
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void TIM3_IRQHandler(void)
{

 if (TIM_GetITStatus(TIM3, TIM_IT_CC2) != RESET)       //判断状态寄存器 TIM3_SR[2] 是否发生了捕获/比较2 中断
  {
   TIM_ClearITPendingBit(TIM3, TIM_IT_CC2);            //软件清除状态寄存器 TIM3_SR[2] 捕获/比较2 中断中断标志

   capture1 = TIM_GetCapture2(TIM3);                   //获取TIM3_CCR2的预装值
   TIM_SetCompare2(TIM3, capture1+ CCR2_Val)    //和CCR2_Val累加后装入TIM3_CCR2 这是为了配合计数器值的递增。以在下一次作出比较
  }
}
```

4.4 程序设计思路

为了实现特定频率的产生， 为了方便看到效果，将LED1的控制脚作为了频率输出引脚，产生的频率不能过高，会造成看不到闪烁变换，所以，将设计频率设定为12.5Hz，PB5引脚默认是作为通用IO口使用，为了产生特定频率，利用了该引脚可以复用为TIM3_CH2，根据手册，需要用到局部复用映射，才能将TIM3_CH2映射到PB5。说明如下图。在定时器3的初始化中，用到了输出比较2模式的翻转设置。也就是当TIM3_CCR2=TIM3_CNT时，翻转输出的电平。以产生需要的频率，TIM3_CCR2的值决定了翻转的频率。当TIM3_CCR2=TIM3_CNT时，产生一次电平翻转，并在中断服务程序里重新完成对TIM3_CCR2的装载。依次往复，产生了所需要的12.5Hz的频率。

复用功能	TIM3_REMAP[1:0] = 00 (没有重映像)	TIM3_REMAP[1:0] = 10 (部分重映像)	TIM3_REMAP[1:0] = 11 (完全重映像)(1)
TIM3_CH1	PA6	PB4	PC6
TIM3_CH2	PA7	PB5	PC7
TIM3_CH3	PB0		PC8
TIM3_CH4	PB1		PC9

5 运行过程

按 编译工程， 完成后会提示如下。

```
Build target 'STM32-FD-TIM3-CH2'
compiling stm32f10x_it.c...
compiling Tim.c...
compiling main.c...
linking...
Program Size: Code=2056 RO-data=336 RW-data=8 ZI-data=640
FromELF: creating hex file...
"..\ObjFlash\STM32-FD-TIM3-CH2.axf" - 0 Error(s), 0 Warning(s).
```

1. 通过JLINK V8或者串口将代码写入板子，具体的烧写步骤，参考奋斗板文档目录下的《奋斗版STM32开发板JTAG下载步骤》或者《奋斗版STM32开发板串口下载步骤》。

2. 通过USB线给板子加电。LED1受PB5引脚的12.5HZ频率影响，闪烁起来。
该例程实验完成。用户可根据例程编写自己的应用例程。

奋斗嵌入开发工作室