
**STM32F1xx、STM32F2xx、STM32F4xx、STM32L1xx、
STM32F30/31/37/38x 定时器概览**

前言

本文档：

- 概述了 STM32F1xx、STM32F2xx、STM32F4xx、STM32L1xx 和 STM32F30/31/37/38x 微控制器系列的定时器外设；
- 介绍了定时器的多种模式和特殊功能，如时钟源；
- 介绍了如何使用各种可用模式与功能；
- 介绍了如何计算各种配置下的时基；
- 除基本定时器模式外，还介绍了定时器同步序列以及用于电机控制应用的高级功能。

本文档针对各模式提供了典型配置以及使用方法的示例。

除非另外说明，本文档余下部分中的术语 **STM32xx** 均用于指代 **STM32F1xx**、**STM32F2xx**、**STM32F4xx**、**STM32L1xx** 和 **STM32F30/31/37/38x** 微控制器系列。

表 1. 适用的产品

类型	适用的产品
微控制器	STM32F1xx、STM32F2xx、STM32F4xx、STM32L1xx、STM32F30x、STM32F31x、STM32F37x、STM32F38x

目录

1	概述	6
2	基本定时器模式	9
2.1	时钟输入源	9
2.1.1	内部时钟	9
2.1.2	外部时钟	9
2.2	时基发生器	10
2.3	定时器输入捕捉模式	12
2.4	定时器输出比较模式	13
2.5	定时器 PWM 模式	14
2.6	定时器单脉冲模式	15
2.7	STM32F30x/STM32F31x 系列专用的定时器不对称 PWM 模式	15
2.8	STM32F30x/STM32F31x 系列专用的定时器混合 PWM 模式	17
2.9	STM32F30x/STM32F31x 系列专用的可重复触发单脉冲模式	18
3	定时器同步	19
3.1	定时器系统链接	19
3.2	主配置	19
3.3	从配置	21
4	用于电机控制的高级功能	22
4.1	信号生成	22
4.2	STM32F30x/STM32F31x 系列专用的混合 3 相 PWM 模式	24
4.3	用于电机控制应用的特殊功能	25
4.3.1	互补信号和死区功能	25
4.3.2	断路输入	26
4.3.3	锁定机制	27
4.3.4	用于反馈测量的特殊功能	27

5 特殊应用 31

5.1 红外应用 31

5.2 3 相交流和 PMSM 控制电机 31

5.3 六步模式 31

6 版本历史 32

表格索引

表 1. 适用的产品 1

表 2. STM32 系列定时器 7

表 3. 定时器特性概览 8

表 4. 高级定时器配置 23

表 5. 定时器输出与 Break1 和 Break2 输入的特性 26

表 6. 锁定级别 27

表 7. 文档版本历史 32



图片索引

图 1. 不对称 PWM 模式与中心对齐 PWM 模式..... 16

图 2. 混合 PWM 模式..... 17

图 3. 可重复触发 OMP 模式 18

图 4. 定时器系统链接..... 19

图 5. 混合 3 相 PWM..... 24

图 6. 生成两个带死区插入的信号..... 25

图 7. X4 分辨率下的位置..... 28

图 8. X2 分辨率下的位置..... 28

图 9. 典型霍尔传感器的输出波形..... 29

图 10. 换向序列..... 30

1 概述

STM 32 器件使用多种类型的定时器，各种定时器特点如下：

- **通用定时器：**可用于输出比较（时序和延迟生成）、单脉冲模式、输入捕捉（用于测量外部信号频率）、传感器接口（编码器和霍尔传感器）等各种场合。
- **高级定时器：**此类定时器的功能最多。除通用功能外，它们还包含一些与电机控制和数字能量转换应用相关的功能：三个带死区控制的互补信号以及紧急关断输入。
- **单通道或双通道定时器：**用作通用定时器，通道数有限。
- **带互补输出的单通道或双通道定时器：**与上一类型相同，只是其中一个通道上具有死区发生器。这样可得到时基与高级定时器无关的互补信号。
- **基本定时器：**没有任何输入/输出，既可用作时基定时器，也可用于触发 DAC 外设。

表 2 汇总了 STM32 系列定时器。

表 3 给出了定时器特性的总览。

表 2. STM32 系列定时器

定时器类型		STM32F101/ 102/103/ 105/107 系列	STM32F100 系列	STM32L1 系列	STM32F2 和 STM32F4 系列	STM32F30x STM32F31x 系列	STM32F37x STM32F38x 系列
高级		TIM1	TIM1		TIM1	TIM1	
		TIM8			TIM8	TIM8	
通用	16 位	TIM2	TIM2	TIM2		TIM2	TIM2
		TIM3	TIM3	TIM3	TIM3	TIM3	TIM3
		TIM4	TIM4	TIM4	TIM4	TIM4	TIM4
		TIM5	TIM5				TIM5
	32 位						TIM19
					TIM2	TIM2	TIM2
基本					TIM5		TIM5
		TIM6	TIM6	TIM6	TIM6	TIM6	TIM6
		TIM7	TIM7	TIM7	TIM7	TIM7	TIM7
单通道							TIM18
		TIM10		TIM10	TIM10		
		TIM11		TIM11	TIM11		
		TIM13	TIM13		TIM13		TIM13
双通道		TIM14	TIM14		TIM14		TIM14
		TIM9		TIM9	TIM9		
单通道，带一个 互补输出		TIM12	TIM12		TIM12		TIM12
			TIM15			TIM15	TIM15
双通道，带一个 互补输出							
			TIM16			TIM16	TIM16
			TIM17			TIM17	TIM17

注：STM32F30x 和 STM32F31x 定时器还具有一些新模式：不对称模式、混合模式、单脉冲可重复触发模式、混合 3 相 PWM 模式以及次级断路输入，这些模式仅适用于这两种系列。

表 3. 定时器特性概览

定时器类型	计数器分辨率	计数器类型	DMA	通道	比较通道	同步	
						主配置	从配置
高级	16 位	递增、递减和中心对齐	有	4	3	有	有
通用	16 位 32 位 ⁽¹⁾	递增、递减和中心对齐	有	4	0	有	有
基本	16 位	递增	有	0	0	有	无
单通道	16 位	递增	无	1	0	有 (OC 信号)	无
双通道	16 位	递增	无	2	0	有	有
单通道, 带一个互补输出	16 位	递增	有	1	1	有 (OC 信号)	无
双通道, 带一个互补输出	16 位	递增	有	2	1	无	有

1. 在 STM32F2、STM32F4 和 STM32F30x/STM32F31x 系列中, TIM2 和 TIM5 的计数器分辨率为 32 位。



2 基本定时器模式

基本定时器经过编程可在以下其中一种配置下工作。

2.1 时钟输入源

定时器可同时与多个时钟同步：

- 内部时钟
- 外部时钟
 - 外部时钟模式 1 (TI1 或 TI2 引脚)
 - 外部时钟模式 2 (ETR 引脚)
 - 内部触发时钟 (ITRx)

2.1.1 内部时钟

默认情况下，定时器由 RCC 提供的内部时钟驱动。要选择该时钟源，应该将 SMCR_SMS (如果存在) 位复位。

2.1.2 外部时钟

外部时钟定时器分为两类：

- 连接到 TI1 或 TI2 引脚的外部时钟
- 连接到 ETR 引脚的外部时钟

在上述情况下，通过与 TIx 引脚或 ETR 引脚连接的外部信号提供时钟。并且应该验证最大外部时钟频率。

- 注：
- 1 除以上所有时钟源外，定时器还应该通过 APBx 时钟驱动。
 - 2 外部时钟并非直接馈送预分频器，而是首先通过专用逻辑块与 APBx 时钟同步。

外部时钟模式 1 (TI1 或 TI2 引脚)

在此模式下，外部时钟将施加到定时器的输入引脚 TI1 或 TI2 上。为此：

1. 配置定时器将 TIx 引脚用作输入：
 - a) 通过对 TIMx_CCMR1 寄存器中的 CCxS 位执行写操作，选择要使用的引脚。
 - b) 选择输入的极性：
对于 STM32F10x 系列：通过对 TIMx_CCER 寄存器中的 CCxP 位执行写操作，选择上升沿触发或下降沿触发；
对于其它系列：通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作，选择上升/下降沿触发，或者边沿触发^{a)}。

a. 对于 STM32F10x 系列，可使用 TI1F_ED 选择边沿触发的极性，但仅限于 TI1 输入。

- c) 根据需要，通过对 TIMx_CCMR1 寄存器中的 ICxF[3:0] 位执行写操作配置滤波器和预分频器：
通过对 TIMx_SMCR 寄存器的 TS 位执行写操作，将定时器 Tlx 选为触发输入源。
通过对 TIMx_SMCR 寄存器写入 SMS=111 选择外部时钟模式 1。通过将 TIMx_CCER 寄存器中的 CCEx 位置 1 使能相应的通道。
2. 通过对 TIMx_SMCR 寄存器中的 TS 位执行写操作，将定时器 Tlx 选为触发输入源。
3. 通过对 TIMx_SMCR 寄存器写入 SMS=111 选择外部时钟模式 1。

外部时钟模式 2 (ETR 引脚)

外部时钟模式 2 将 ETR 引脚用作定时器输入时钟。要使用该功能：

1. 通过对 TIMx_SMCR 寄存器写入 ECE = 1 选择外部时钟模式 2。
2. 根据需要，通过对 TIMx_SMCR 寄存器中的 ETPS [1:0]、ETF [3:0] 和 ETP 位执行写操作配置预分频器、滤波器和极性。

内部触发时钟 (ITRx)

这是一个特殊的定时器同步模式。当将一个定时器用作另一个定时器的预分频器时，第一个定时器的更新事件或输出比较信号将用作第二个定时器的时钟。

2.2 时基发生器

定时器可用作时基发生器。根据时钟、预分频器和自动重载、以及重复计数器（如果存在）的参数，16 位定时器可生成 1 纳秒到数分钟的更新事件。对于 32 位定时器，时间范围更大。

更新事件周期示例

更新事件周期按如下公式计算：

$$\text{Update_event} = \text{TIM_CLK} / ((\text{PSC} + 1) * (\text{ARR} + 1) * (\text{RCR} + 1))$$

其中： TIM_CLK = 定时器时钟输入

PSC = 16 位预分频器寄存器

ARR = 16/32 位自动重载寄存器

RCR = 16 位重复计数器

TIM_CLK = 72 MHz

预分频比 = 1

自动重载值 = 65535

无重复计数器 RCR = 0

$$\text{Update_event} = 72 * 106 / ((1 + 1) * (65535 + 1) * (1))$$

$$\text{Update_event} = 549.3 \text{ Hz}$$

外部时钟模式 2 示例

在此模式下，更新事件周期按如下公式计算：

$$\text{Update_event} = \text{ETR_CLK} / ((\text{ETR_PSC}) * (\text{PSC} + 1) * (\text{ARR} + 1) * (\text{RCR} + 1))$$

其中：ETR_CLK = 连接到 ETR 引脚的外部时钟频率。

ETR_CLK = 100 kHz

预分频比 = 1

ETR_PSC = 2

自动重载值 = 255

重复计数器 = 2

$$\text{Update_event} = 100 * 103 / ((2 + 1) * (1 + 1) * ((255 + 1) * (2 + 1)))$$

$$\text{Update_event} = 21.7 \text{ Hz}$$

外部时钟模式 1 示例

在此模式下，更新事件周期按如下公式计算：

$$\text{Update_event} = \text{TIx_CLK} / ((\text{PSC} + 1) * (\text{ARR} + 1) * (\text{RCR} + 1))$$

其中：TIx_CLK = 连接到 TI1 引脚或 TI2 引脚的外部时钟频率。

TIx_CLK = 50 kHz

预分频比 = 1

自动重载值 = 255

重复计数器 = 2

$$\text{Update_event} = 50\,000 / ((1 + 1) * ((255 + 1) * (2 + 1)))$$

$$\text{Update_event} = 32.55 \text{ Hz}$$

内部触发时钟 (ITRx) 模式 1 示例

在此模式下，更新事件周期按如下公式计算：

$$\text{Update_event} = \text{ITRx_CLK} / ((\text{PSC} + 1) * (\text{ARR} + 1) * (\text{RCR} + 1))$$

其中 ITRx_CLK = 映射到定时器触发输入 (TRGI) 的内部触发频率

ITRx_CLK = 8 kHz

预分频比 = 1

自动重载值 = 255

重复计数器 = 1

$$\text{Update_event} = 8000 / ((1 + 1) * ((255 + 1) * (1 + 1)))$$

$$\text{Update_event} = 7.8 \text{ Hz}$$

根据计数器模式的不同，更新事件通常在以下情况产生：

- 每次上溢（如果使用递增计数模式）：TIMx_CR1 寄存器中的 DIR 位复位
- 每次下溢（如果使用递减计数模式）：TIMx_CR1 寄存器中的 DIR 位置 1
- 每次上溢和下溢（如果使用中心对齐模式）：CMS 位不为零。

还可以通过以下方式生成更新事件：

- 通过软件生成更新事件（如果 TIM_EGR 寄存器中的 UG（更新生成）位置 1）。
- 通过从模式控制器生成更新事件

因为缓冲的寄存器（ARR、PSC 和 CCRx）需要一个装载预装载值的更新事件，因此将 URS（更新请求源）置 1 可避免每次装载这些值时触发更新标志位。这样，只有当计数器发生上溢/下溢时才会生成更新事件。

此外，可以通过将 CR1 寄存器中的 UDIS（更新禁止）位置 1 以禁止更新事件。这样，便不会生成更新事件，而各影子寄存器（ARR、PSC 和 CCRx）的值会保持不变。但如果将 UG 位置 1，或者从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

当 DIER 寄存器中的 UIE 位或/和 UDE 位置 1 时，会产生中断或/和 DMA 请求。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/TimeBase 文件夹中 STM32xx 标准外设库提供的示例。

2.3 定时器输入捕捉模式

在输入捕捉模式下，定时器可用于测量外部信号。根据定时器时钟、预分频器和定时器分辨率，可推导出最大测量周期。

要在此模式下使用定时器：

1. 通过将 CCMRx 寄存器中的 CCxS 位置 1 选择有效输入。这些位应该不为零，否则 CCRx 寄存器将是只读的。
2. 根据需要，通过对 CCMRx 寄存器中的 IC1F[3:0] 位执行写操作以编程滤波器，并通过对 IC1PSC[1:0] 位执行写操作以编程预分频器。
3. 通过对 CCxNP/CCxP 位执行写操作以编程极性，即选择上升沿触发、下降沿触发或边沿触发。

当输入通道检测到相应的计数器的值发生转变时，可使用输入捕捉模块进行捕捉。要获取外部信号周期，需连续进行两次捕捉。通过将两次捕捉到的值相减可计算得到周期。

$$\text{周期} = \text{捕捉}(1) / (\text{TIMx_CLK} * (\text{PSC} + 1) * (\text{ICxPSC}) * \text{polarity_index}(2))$$

两次连续捕捉值 CCRx_tn 和 CCRx_tn+1 的捕捉差值：

- 如果 $\text{CCRx_tn} < \text{CCRx_tn+1}$ ：捕捉 = $\text{CCRx_tn+1} - \text{CCRx_tn}$
- 如果 $\text{CCRx_tn} > \text{CCRx_tn+1}$ ：捕捉 = $(\text{ARR_max} - \text{CCRx_tn}) + \text{CCRx_tn+1}$

如果使用上升沿触发或下降沿触发，则极性指数为 1；而如果选择边沿触发，则极性指数为 2。

特例

要简化输入捕捉测量，可在每次定时器输入通道检测到上升沿后，通过以下方式复位定时器计数器：

- 通过将 SMCR 寄存器中的 TS 位置 1，将 TlxFPx 选为输入触发
- 通过配置 SMCR 寄存器中的 SMS 位，将复位模式选为从模式

如果使用此配置，则当检测到边沿时，会复位计数器，并且会通过 CCRx 寄存器上的值自动给出外部信号的周期。此方法只能单独与通道 1 或通道 2 配合使用。

这样，计算周期时便不需要考虑输入捕捉预分频器 (ICPSC)。

周期可以按如下公式计算：

$$\text{周期} = \text{CCR}_x / (\text{TIM}_x_CLK * (\text{PSC} + 1) * \text{polarity_index}(1))$$

如果使用上升沿触发或下降沿触发，则极性指数为 1；而如果使用边沿触发，则极性指数为 2。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/InputCapture 文件夹中 STM32xx 标准外设库提供的示例。

2.4 定时器输出比较模式

要控制输出波形，或者指示何时一段给定的时间已经到时，可使用以下其中一种输出比较模式的定时器。这些模式的主要区别在于输出信号波形。

- **输出比较定时：**输出比较寄存器 CCRx 与计数器 CNT 之间的比较对输出没有影响。该模式用于生成时基。
- **输出比较有效：**将通道输出设置为匹配时输出有效电平。当计数器 (CNT) 与捕捉/比较寄存器 (CCRx) 匹配时，OCxRef 信号强制变为高电平。
- **输出比较无效：**将通道输出设置为匹配时输出无效电平。当计数器 (CNT) 与捕捉/比较寄存器 (CCRx) 匹配时，OCxRef 信号强制变为低电平。
- **输出比较切换：**当计数器 (CNT) 与捕捉/比较寄存器 (CCRx) 匹配时，OCxRef 信号发生切换。
- **输出比较强制有效/无效：**无论计数器值为何，OCREF 信号都强制变为高电平（有效模式）或低电平（无效模式）。

要将定时器配置为以上其中一种模式：

1. 选择时钟源。
2. 在 ARR 和 CCRx 寄存器中写入所需数据。
3. 配置输出模式：
 - a) 选择输出比较模式：时间/有效/无效/切换。
 - b) 如果选择有效、无效和切换模式，则需通过对 CCER 寄存器中的 CCxP 位执行写操作以选择极性。
 - c) 通过对 CCMRx 寄存器中的 OCxPE 位执行写操作以禁止 CCx 的预装载功能。
 - d) 通过对 CCMRx 寄存器中的 CCxE 位执行写操作使能捕捉/比较输出。
4. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 使能计数器。
5. 如果要生成中断/DMA 请求，则将 CCxIE/CCxDE 位置 1。

定时器输出比较定时/切换更新率计算

$$\text{CCx 更新率} = \text{TIM}_x_Counter_CLK / \text{CCR}_x$$

- 如果为内部时钟：TIMx_Counter_CLK = TIM_CLK / (PSC + 1)
- 如果为外部时钟模式 2：TIMx_Counter_CLK = ETR_CLK / ((ETR_PSC)*(PSC + 1))
- 如果为外部时钟模式 1：TIMx_Counter_CLK = ETR_CLK / ((ETR_PSC)*(PSC + 1))

定时器输出比较有效/无效延迟计算

$$CCx_delay = CCRx / TIMx_Counter_CLK$$

- 如果为内部时钟: $TIMx_Counter_CLK = TIM_CLK / (PSC + 1)$
- 如果为外部时钟模式 2: $TIMx_Counter_CLK = ETR_CLK / ((ETR_PSC) * (PSC + 1))$
- 如果为外部时钟模式 1: $TIMx_Counter_CLK = ETR_CLK / ((ETR_PSC) * (PSC + 1))$
- 如果为内部触发时钟: $TIMx_Counter_CLK = ITRx_CLK / (PSC + 1)$

有关在此模式下使用定时器的详细信息, 请参见 /Project/STM32xx_StdPeriph_Examples/TIM/OC_Toggle、/OCActive 和 /OCInactive 文件夹中 STM32xx 标准外设库提供的示例。

2.5 定时器 PWM 模式

在边沿对齐模式或中心对齐模式下, 这种定时器可在每个通道上单独生成 PWM。其中: 频率由 TIMx_ARR 寄存器的值确定, 占空比由 TIMx_CCRx 寄存器的值确定。

PWM 模式 1

- 递增计数时, 只要 $CNT < CCRx$, 便激活通道 x, 否则禁止通道 x。
- 递减计数时, 只要 $CNT > CCRx$, 便禁止通道 x, 否则激活通道 x。
- 递增计数时, 只要 $CNT < CCRx$, 便禁止通道 x, 否则激活通道 x。
- 递减计数时, 只要 $CNT > CCRx$, 便激活通道 x, 否则禁止通道 x。

注: 当 $OCREF = 1$ 时激活, $OCREF = 0$ 时禁止。

要将定时器配置为该模式:

1. 配置输出引脚:
 - a) 通过对 CCMRx 寄存器中的 CCS 位执行写操作以选择输出模式。
 - b) 通过对 CCER 寄存器中的 CCxP 位执行写操作以选择极性。
2. 通过对 CCMRx 寄存器中的 OCxM 位执行写操作以选择 PWM 模式 (PWM1 或 PWM2)。
3. 分别在 ARR 和 CCRx 寄存器中编程周期和占空比。
4. 将 CCMRx 寄存器中的预装载位置 1, 然后将 CR1 寄存器中的 ARPE 位置 1。
5. 选择计数模式:
 - a) PWM 边沿对齐模式: 计数器必须配置为递增计数或递减计数模式。
 - b) PWM 中心对齐模式: 定时器模式必须为中心对齐计数模式 (CMS 位不为 “00”)。
6. 使能捕捉比较。
7. 使能计数器。

有关在此模式下使用定时器的详细信, 请参见 /Project/STM32xx_StdPeriph_Examples/TIM/PWM_Output 和 /7PWM_Output 文件夹中 STM32xx 标准外设库提供的示例。

2.6 定时器单脉冲模式

单脉冲模式 (OPM) 是输入捕捉模式和输出比较模式的一个特例。该模式下可以通过响应某个激励信号来启动计数器，并在可编程的延时后生成长度可编程的脉冲。

要将定时器配置为该模式：

1. 配置输入引脚和模式：
 - a) 通过对 CCMRx 寄存器中的 CCxS 位执行写操作以选择要使用的 TlxFPx 触发。
 - b) 通过对 CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作以选择输入引脚的极性。
 - c) 通过对 SMCR 寄存器中的 TS 位执行写操作，将 TlxFPx 触发配置为从模式触发。
 - d) 通过对 SMCR 寄存器写入 SMS = 110，将触发模式选为从模式。
2. 配置输出引脚和模式：
 - a) 通过对 CCER 寄存器中的 CCyP 位执行写操作以选择输出极性。
 - b) 通过对 CCMRy 寄存器中的 OCyM 位执行写操作以选择输出比较模式（PWM1 或 PWM2 模式）。
 - c) 通过对 CCRy 寄存器执行写操作以设置延迟值。
 - d) 设置自动重装载值以获得所需脉冲：脉冲 = TIMy_ARR - TIMy_CCRy。
3. 如果只需要生成一个脉冲，则可通过将 CR1 寄存器中的 OPM 位置 1 以选择单脉冲模式。否则，应该复位此位。
 延迟 = $CCRy / (TIMx_CLK / (PSC + 1))$
 Pulse-Length = $(ARR - CCRy) / (TIMx_CLK / (PSC + 1))$

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/OnePulse 文件夹中 STM32xx 标准外设库提供的示例。

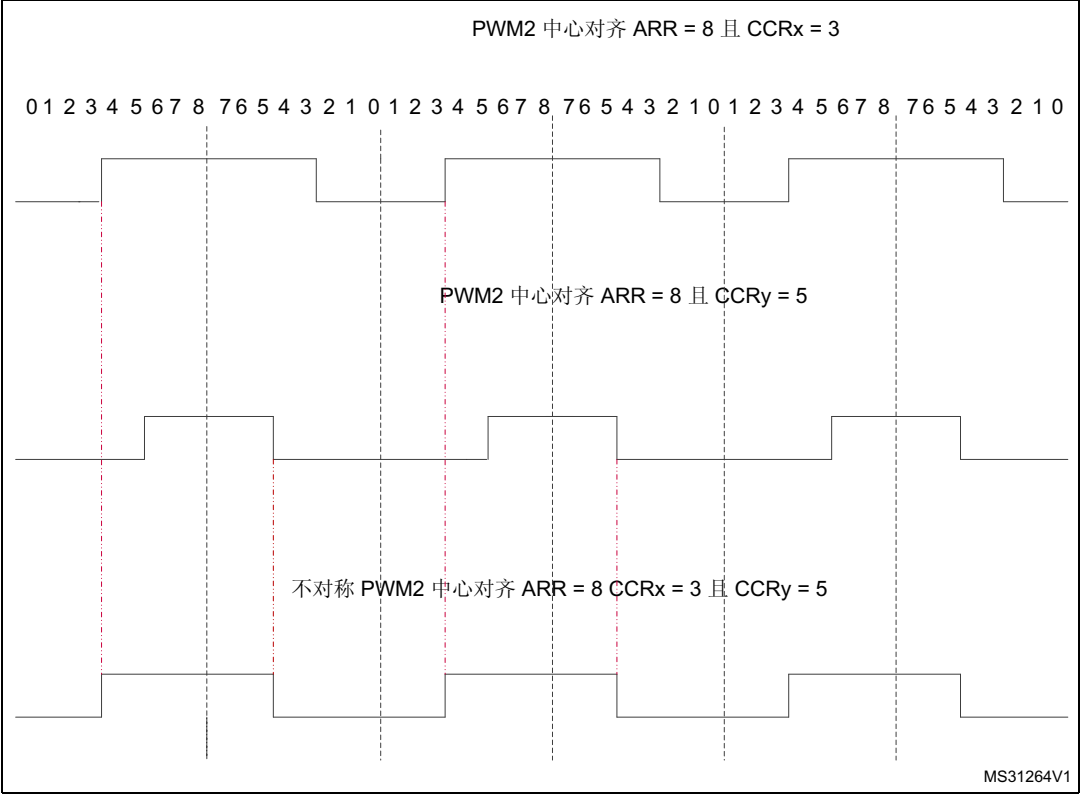
2.7 STM32F30x/STM32F31x 系列专用的定时器不对称 PWM 模式

在不对称模式下，可生成相移可编程的中心对齐的 PWM 信号。对于专用通道，可使用两个 TIMx_CCRx 寄存器（TIMx_CCR1 和 TIMx_CCR2 或者 TIMx_CCR3 和 TIMx_CCR4）编程相移和脉宽，频率由 TIMx_ARR 寄存器的值确定。因此，可单独在两个通道上通过编程 TIMx_CCMRx 寄存器中的 OCxM 位来选择不对称 PWM 模式：

- OCxM = 1110 时使用不对称 PWM1，此模式下的输出参考特性与 PWM1 模式相同。当计数器递增计数时，输出参考与 OC1/3REF 相同；当计数器递减计数时，输出参考与 OC2/4REF 相同
- OCxM = 1111 时使用不对称 PWM2，此模式下的输出参考特性与 PWM2 模式相同。当计数器递增计数时，输出参考与 OC1/3REF 相同；当计数器递减计数时，输出参考与 OC2/4REF 相同

下图显示了不对称 PWM 模式与中心对齐 PWM 模式的特性

图 1. 不对称 PWM 模式与中心对齐 PWM 模式



要将定时器配置为该模式：

1. 配置输出引脚：
 - a) 通过对 CCMRx 寄存器中的 CCS 位执行写操作以选择输出模式。
 - b) 通过对 CCER 寄存器中的 CCxP 位执行写操作以选择极性。
2. 通过对 CCMRx 寄存器中的 OCxM 位执行写操作以选择不对称 PWM 模式（不对称 PWM1 或不对称 PWM2）。
3. 在 ARR、CCRx 和 CCRy 寄存器中分别编程周期、脉宽和相移。
4. 选择计数模式：不对称 PWM 模式只能与中心对齐模式配合使用：定时器模式必须为中心对齐计数模式（CMS 位不为“00”）。
5. 使能捕捉比较。
6. 使能计数器。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/Asymetric 文件夹中 STM32xx 标准外设库提供的示例。

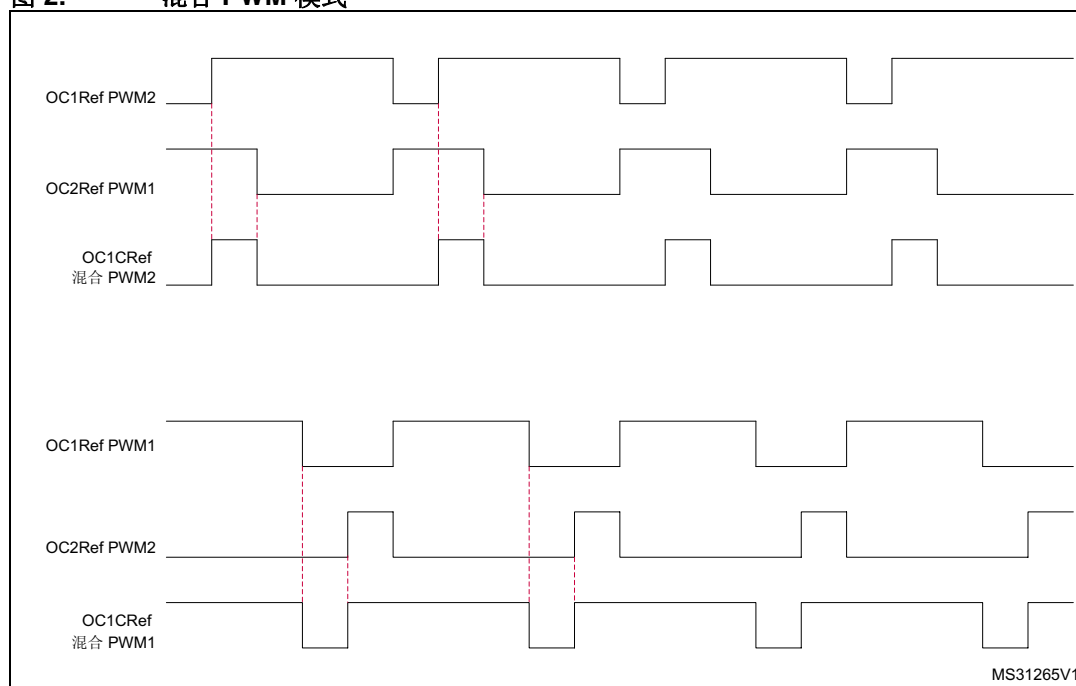
2.8 STM32F30x/STM32F31x 系列专用的定时器混合 PWM 模式

在混合模式下，既可以生成边沿对齐的 PWM 信号，也可以生成中心对齐的 PWM 信号，并且各脉冲之间的延迟与相移可编程。要生成混合信号，必须使用 TIMx_CCRx 和 TIMx_CCRy 编程延迟和相移，频率由 TIMx_ARR 寄存器的值确定。结果信号（混合信号）由两个参考 PWM 的“或”或“与”逻辑组合构成。因此，可单独在两个通道上通过编程 TIMx_CCMRx 寄存器中的 OCxM 位来选择混合 PWM 模式：

- OCxM = 1100 时使用混合 PWM1，此时混合输出参考特性与 PWM 模式 1 相同。混合输出参考是 OC1/3REF 与 OC2/4REF 之间的逻辑或运算。
- OCxM = 1101 时使用混合 PWM2 模式，此时混合输出参考特性与 PWM 模式 2 相同。混合输出参考是 OC1/2REF 与 OC2/4REF 之间的逻辑与运算

下图显示了混合模式

图 2. 混合 PWM 模式



要将定时器配置为该模式：

- 配置输出引脚：
 - 通过对 CCMRx 寄存器中的 CCS 位执行写操作以选择输出模式。
 - 通过对 CCER 寄存器中的 CCxP 位执行写操作以选择极性。
- 通过对 CCMRx 寄存器中的 OCxM 位执行写操作以选择混合 PWM 模式（混合 PWM1 或混合 PWM2）。
- 在 ARR、CCRx 和 CCRy 寄存器中分别编程周期、延迟和相移。
- 选择计数模式：
 - 边沿对齐模式：计数器必须配置为递增计数或递减计数模式。
 - 中心对齐模式：定时器模式必须为中心对齐计数模式（CMS 位不为“00”）
- 使能捕捉比较。
- 使能计数器。

2.9 STM32F30x/STM32F31x 系列专用的可重复触发单脉冲模式

可重复触发单脉冲模式是具有以下特性的单脉冲模式：

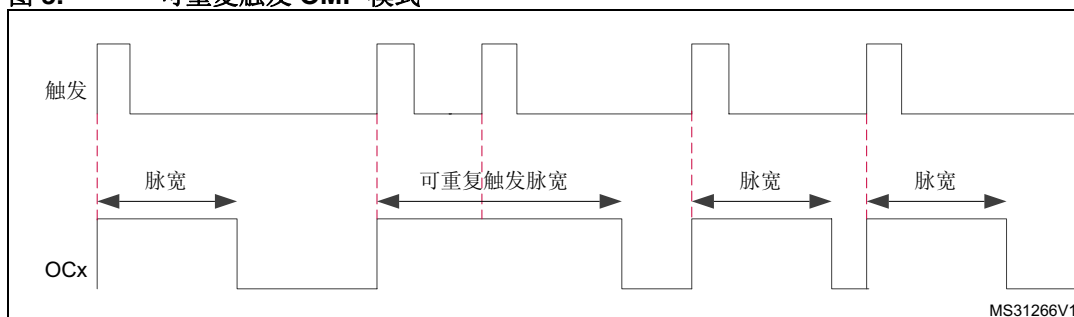
- 一旦产生触发便生成脉冲（无可编程延迟）
- 如果完成上一触发之前又产生了新触发，则脉冲会得到延长

如果计数器配置为递增计数模式，则必须将相应的 **CCRx** 置 0，此时脉宽由 **ARR** 寄存器确定。如果定时器配置为递减计数模式，则必须将 **ARR** 置 0，此时脉宽由 **CCRx** 寄存器确定。对于 **OPM** 模式，则存在两种可重复触发单脉冲模式，可重复触发 **OPM** 模式 1 与可重复触发 **OPM** 模式 2：

- 通过将 **OCXM** 位置为 1000 来选择可重复触发 **OPM** 模式 1：
 - 在递增计数模式下，通道始终处于无效状态，直到检测到触发事件（**TRGI** 信号），随后像在 **PWM** 模式 1 中那样执行比较，并在下一次更新时再次禁止通道。
 - 在递减计数模式下，通道始终处于有效状态，直到检测到触发事件（**TRGI** 信号），随后像在 **PWM** 模式 1 中那样执行比较，并在下一次更新时再次激活通道。
- 通过将 **OCXM** 位设置为 1001 来选择可重复触发 **OPM** 模式 2：
 - 在递增计数模式下，通道始终处于无效状态，直到检测到触发事件（**TRGI** 信号），随后像在 **PWM** 模式 2 中那样执行比较，并在下一次更新时再次禁止通道。在递减计数模式下，通道始终处于无效状态，直到检测到触发事件（**TRGI** 信号），随后像在 **PWM** 模式 2 中那样执行比较，并在下一次更新时再次禁止通道。

下图给出了可重复触发 **OPM** 模式的示例

图 3. 可重复触发 **OPM** 模式



要将定时器配置为该模式：

1. 配置输入引脚和模式：
 - a) 通过对 **CCMRx** 寄存器中的 **CCxS** 位执行写操作以选择要使用的 **TlxFPx** 触发。
 - b) 通过对 **CCER** 寄存器中的 **CCxP** 和 **CCxNP** 位执行写操作以选择输入引脚的极性。
 - c) 通过对 **SMCR** 寄存器中的 **TS** 位执行写操作将 **TlxFPx** 触发配置为从模式触发。
 - d) 通过对 **SMCR** 寄存器写入 **SMS = 1000**，将混合复位 + 触发模式选为从模式。
2. 配置输出引脚和模式：
 - a) 通过对 **CCER** 寄存器中的 **CCyP** 位执行写操作以选择输出极性。
 - b) 通过对 **CCMRy** 寄存器中的 **OCyM** 位执行写操作以选择输出比较模式（可重复触发 **OPM** 模式 1 或可重复触发 **OPM** 模式 2）。
 - c) 如果计数器为递减计数，则通过对 **CCRy** 寄存器执行写操作以设置脉宽值；如果计数器为递增计数，则通过对 **ARR** 执行写操作以设置脉宽值。

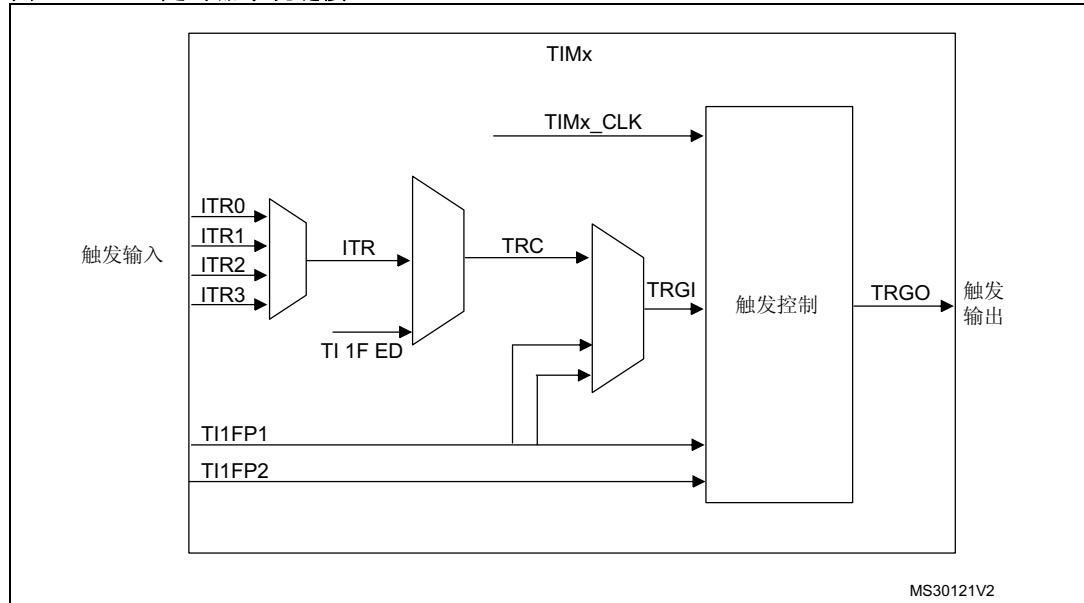
有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/Retrigerrable OPM 文件夹中 STM32xx 标准外设库提供的示例。

3 定时器同步

3.1 定时器系统链接

STM32 定时器从内部连接在一起，以实现定时器同步或链接。每个定时器都有多个内部触发输入和输出。这些信号允许定时器之间的连接。

图 4. 定时器系统链接



3.2 主配置

当选择一个定时器作为主定时器时，其相应的触发输出信号可用作从定时器内部触发信号（配置时）。可以根据以下各项选择触发输出：

- **复位：** EGR 寄存器中的 UG 位用作触发输出 (TRGO)。
- **使能：** 计数器使能信号用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制用于使能从定时器的窗口。
- **更新：** 选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。
- **比较脉冲：** CC1IF 标志置 1 时（即使已为高电平），只要发生捕捉或比较匹配，触发输出都会发送一个正脉冲。
- **OC1Ref：** OC1REF 信号用作触发输出 (TRGO)。
- **OC2Ref：** OC2REF 信号用作触发输出 (TRGO)。
- **OC3Ref：** OC3REF 信号用作触发输出 (TRGO)。
- **OC4Ref：** OC4REF 信号用作触发输出 (TRGO)。

要将定时器配置为主模式：

1. 配置定时器。
2. 通过对 CR2 寄存器中的 MSM（主模式选择）位执行写操作以选择要使用的触发输出。
3. 使能 SMCR 寄存器中的 MSM（主/从模式）位以实现当前定时器与其从定时器的完美同步（通过 TRGO 实现）。

对于 STM32F30x/31x 系列，定时器可生成两个触发输出，即 TRGO（如上所述）和 TRGO2（用于同步 TIM 和 ADC），其中 TRGO2 可从以下各项中选择：

- 复位——EGR 寄存器中的 UG 位用作触发输出 (TRGO2)。
- 使能——计数器使能信号 CNT_EN 用作触发输出 (TRGO2)。该触发输出可用于同时启动多个定时器，或者控制用于使能从定时器的窗口。在门控模式下，计数器使能信号由 CEN 控制位与触发输入的逻辑或运算组合而成。
- 更新——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。
- 比较脉冲——CC1IF 标志置 1 时（即使已为高电平），只要发生捕捉或比较匹配，触发输出都会发送一个正脉冲。
- 比较——OC1REF 信号用作触发输出 (TRGO2)
- 比较——OC2REF 信号用作触发输出 (TRGO2)
- 比较——OC3REF 信号用作触发输出 (TRGO2)
- 比较——OC4REF 信号用作触发输出 (TRGO2)
- 比较——OC5REF 信号用作触发输出 (TRGO2)
- 比较——OC6REF 信号用作触发输出 (TRGO2)
- 比较脉冲——OC4REF 上升沿或下降沿时在 TRGO2 上生成脉冲
- 比较脉冲——OC6REF 上升沿或下降沿时在 TRGO2 上生成脉冲
- 比较脉冲——OC4REF 上升沿或 OC6REF 上升沿时在 TRGO2 上生成脉冲
- 比较脉冲——OC4REF 上升沿或 OC6REF 下降沿时在 TRGO2 上生成脉冲
- 比较脉冲——OC5REF 上升沿或 OC6REF 上升沿时在 TRGO2 上生成脉冲
- 比较脉冲——OC5REF 上升沿或 OC6REF 下降沿时在 TRGO2 上生成脉冲

3.3 从配置

从定时器通过输入触发与主定时器连接。每个 ITRx 从内部与另一个定时器连接，这种连接是因每个 STM32 产品而异的。

从模式可以是：

- **复位模式：**在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
- **门控模式：**触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。同时控制计数器的启动和停止。
- **触发模式：**触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
- **外部时钟模式 1：**在所选触发 (TRGI) 的上升沿驱动计数器。
- **混合复位 + 触发模式**——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，并生成一个寄存器更新事件，然后启动计数器。仅 STM32F30x/31x 系列具有该模式。

要将定时器配置为从模式：

1. 通过对 SMCR 寄存器中的 SMS（从模式选择）位执行写操作以选择要使用的从模式。
2. 通过对 SMCR 寄存器中的 TS（触发选择）位执行写操作以选择要使用的内部触发。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/Cascade_Synchro、/ExtTrigger_Synchro 和 /Parallel_Synchro 文件夹中 STM32xx 标准外设库提供的示例。

4 用于电机控制的高级功能

4.1 信号生成

STM32 定时器可输出两路互补信号，并在输出关断与接通的瞬间进行管理。

互补信号 OCx 和 OCxN 通过多个控制位的组合激活：CCxE 和 CCxNE 以及 MOE、OISx、OISxN、OSSI 和 OSSR 位。

当断路输入为活动状态时，主输出使能 (MOE) 位将复位。该位既可以通过软件置 1，也可以自动置 1，具体取决于自动输出使能 (AOE) 位。当此位 (MOE) 复位时，OCx 和 OCxN 输出将禁止或强制进入空闲状态 (OISx OISxN)，具体取决于 OSSI 位是否置 1。

注： MOE 位只在配置为输出的通道上有效。

当在互补输出配置为输出的通道上 MOE=1 时，会使用运行模式 (OSSR) 位的关闭状态选项。当该位置 1 时，一旦 CCxE=1 或 CCxNE=1，便将 OCx 和 OCxN 输出使能为相应的无效电平。输出仍受定时器控制。

当配置为输出的通道上由于断路事件或软件写操作导致 MOE=1 时，会使用空闲模式 (OSSI) 位的关闭状态选项。当该位置 1 时，首先将 OCx 和 OCxN 输出强制为相应的无效电平，经过一段死区后，再强制为相应的空闲电平。定时器会保持对输出的控制。

表 4 给出了高级定时器的可能配置。

表 4. 高级定时器配置

控制位					输出状态		典型应用
MOE	OSSI	OSSR	OCxE	OCxNE	OCx 输出状态	OCxN 输出状态	
1	x	0	0	0	禁止输出	禁止输出	通用
	x	0	0	1	禁止输出	OCxREF + 极性	
	x	0	1	0	OCxREF + 极性	禁止输出	
	x	0	1	1	OCxREF + 极性 + 死区	(无 OCxREF) + 极性 + 死区	电机控制 (正弦波)
	x	1	0	0	禁止输出	禁止输出	电机控制 (6 步)
	x	1	0	1	关闭状态	OCxREF + 极性	
	x	1	1	0	OCxREF + 极性	关闭状态	
	x	1	1	1	OCxREF + 极性 + 死区	(无 OCxREF) + 极性 + 死区	电机控制 (正弦波)
MOE	OSSI	OSSR	OCxE	OCxNE	OCx 输出状态	OCxN 输出状态	注释
0	0	x	0	0	禁止输出		输出与 I/O 端口 断开连接
	0	x	0	1			
	0	x	1	0			
	0	x	1	1			
	1	x	0	0	关闭状态 (首先将输出强制为相应的无效电 平, 经过一段死区后, 再强制为相 应的空闲电平。)		所有 PWM 均关 闭 (低阻抗安全 停止)
	1	x	0	1			
	1	x	1	0			
	1	x	1	1			

- 注:
- 1 通过将 **CCxE** 和 **CCxNE** 位以及 **MOE** 位置 1 以使能死区插入。
 - 2 如果仅使能 **OCxN** (**CCxE=0**, **CCxNE=1**), 两者不互补, 一旦 **OCxREF** 为高电平, **OCxN** 即变为有效。例如, 如果 **CCxNP=0**, 则 **OCxN=OCxRef**。另一方面, 如果同时使能 **OCx** 和 **OCxN** (**CCxE=CCxNE=1**), **OCx** 在 **OCxREF** 为高电平时变为有效, 而 **OCxN** 则与之互补, 在 **OCxREF** 为低电平时变为有效。

4.2 STM32F30x/STM32F31x 系列专用的混合 3 相 PWM 模式

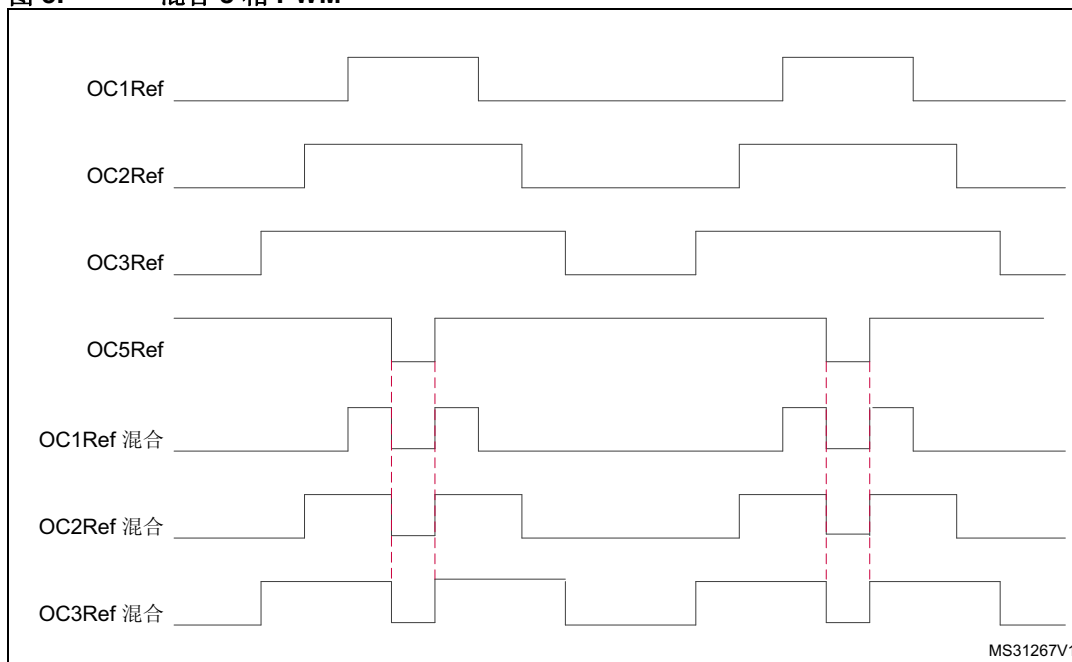
在混合 3 相模式下，可生成一到三个中心对齐的 PWM 脉冲信号，这些 PWM 脉冲信号的中央部分将与一个可编程信号进行与运算。

使用 TIMx_CCR5 中的 3 位 GC5C[3:1]，TIM 的每个通道都可以是原始信号与 OC5Ref 信号的组合：

- 如果 GC5C1 置 1，则 OC1 输出由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2 输出由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3 输出由 TIMx_CCR3 和 TIMx_CCR5 控制

下图给出了该模式的示例

图 5. 混合 3 相 PWM



要将定时器配置为该模式：

1. 配置输出引脚：
 - a) 通过对 CCMRx 寄存器中的 CCS 位执行写操作以选择输出模式。
 - b) 通过对 CCER 寄存器中的 CCxP 位执行写操作以选择极性。
2. 配置 PWM 模式下使用的通道（1、2 或/和 3）
 - a) 配置频率、占空比和极性。
 - b) 选择 PWM 1 或 2。
3. 使用所需参数（占空比）配置 PWM 模式下的通道 5
4. 通过编程 GC5Cx 位选择混合 PWM 模式。
5. 将中心对齐模式选为计数模式
6. 使能捕捉比较。
7. 使能计数器。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/Combined 文件夹中 STM32xx 标准外设库提供的示例。

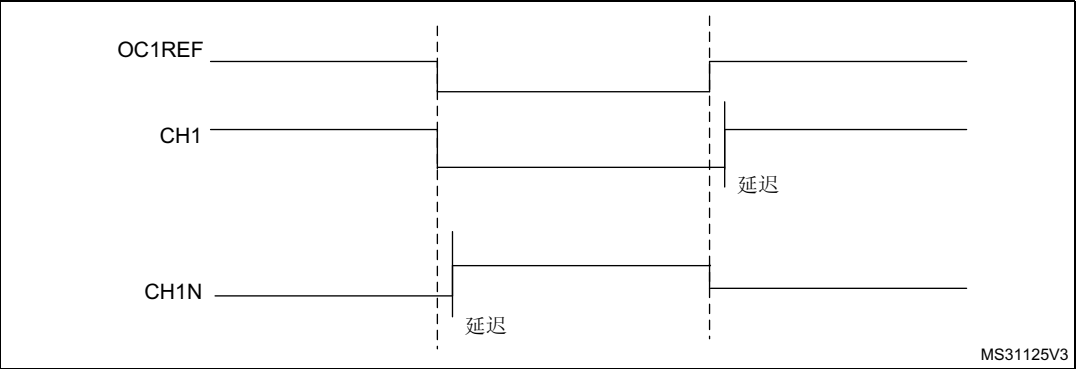
4.3 用于电机控制应用的特殊功能

4.3.1 互补信号和死区功能

STM32 高级定时器可产生多达三个带死区插入的互补输出。

要对一个通道使用互补信号，需将该通道及其互补通道对应的两个输出比较使能位（OCxE 和 OCxNE）置 1。如果死区位不为零，则会生成两个带死区插入的信号，如图 6: 生成两个带死区插入的信号所示：

图 6. 生成两个带死区插入的信号



使用 DTG[7:0] 位与死区时钟 (Tdtg) 计算死区参数。

死区时钟按如下方式计算：

- Tdtg = TDTS, 如果 DTG[7] = 0
- Tdtg = 2 x TDTS, 如果 DTG[6] = 0
- Tdtg = 8 x TDTS, 如果 DTG[5] = 0
- Tdtg = 16 x TDTS, 如果 DTG[7:5] = 111

- 其中：
- TDTS = TCK_INT, 如果 CKD[1:0] = 00
 - TDTS = 2 x TCK_INT, 如果 CKD[1:0] = 01
 - TDTS = 4 x TCK_INT, 如果 CKD[1:0] = 10

注：TCK_INT 为定时器内部时钟。

使用如下公式计算死区延迟：

$$\begin{aligned} \text{死区时间} &= \text{DTG}[7:0] \times \text{Tdtg}, \text{ 如果 } \text{DTG}[7] = 0 \\ \text{死区时间} &= (64 + \text{DTG}[5:0]) \times \text{Tdtg}, \text{ 如果 } \text{DTG}[6] = 0 \\ \text{死区时间} &= (32 + \text{DTG}[4:0]) \times \text{Tdtg}, \text{ 如果 } \text{DTG}[5] = 0 \\ \text{死区时间} &= (32 + \text{DTG}[4:0]) \times \text{Tdtg}, \text{ 如果 } \text{DTG}[7:5] = 111 \end{aligned}$$

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/ComplementarySignals 文件夹中 STM32xx 标准外设库提供的示例。

注：该示例不适用于 STM32L1xx 器件。
该示例也不适用于 STM32F37x/38x 器件。

4.3.2 断路输入

断路输入是电机控制应用中的一种紧急输入。断路功能可保护 PWM 信号所驱动的电源开关，PWM 信号通常由高级定时器产生。断路输入通常与电源级和 3 相逆变器的故障输出连接。当激活断路输入时，断路电路会将 TIM 输出关断，并强制这些输出进入预定义的安全状态。

可以通过以下方式生成断路事件：

- BRK 输入（具有一个可编程极性和一个使能位 BKE）。
- CSS（时钟安全系统）。
- 软件（将 EGR 寄存器中的 BG 位置 1）。

发生断路事件时：

- MOE（主输出使能）位将清零。
- 断路状态标志将置 1，并可产生中断请求。
- 每个输出通道将由在 OISx 位中编程的电平驱动。

STM32F30x/31x 系列的断路输入

在此系列中，存在两个断路输入 Break1 和 Break2。断路事件可通过这两个 BRK 输入中的任意一个生成，其中 BRK 输入具有：

- 可编程的极性（TIMx_BDTR 寄存器中的 BKPx 位）
- 可编程的使能位（TIMx_BDTR 寄存器中的 BKEx 位）
- 可编程的滤波器（TIMx_BDTR 寄存器中的 BKxF[3:0] 位）。该滤波器可避免意外事件。

下表给出了两个断路输入的优先级。

表 5. 定时器输出与 Break1 和 Break2 输入的特性

断路输入 1	断路输入 2	OCxN 输出	OCx 输出
有效	无效	死区插入之后开启	关闭
无效	有效	关闭	关闭
有效	-	死区插入之后开启	关闭

4.3.3 锁定机制

通过锁定机制（即编程 **BDTR** 寄存器中的 **LOCK** 位），可以对高级定时器的寄存器和位进行保护或锁定，进而保护应用的安全性。该锁定机制有以下三种锁定级别。

表 6. 锁定级别

级别 1		级别 2		级别 3	
寄存器	位	寄存器	位	寄存器	位
CR2	OISx	CR2	OISx	CR2	OISx
	OISxN		OISxN		OISxN
BDTR	DTG[7:0]	BDTR	DTG[7:0]	BDTR	DTG[7:0]
	BKE		BKE		BKE
	BKP		BKP		BKP
	AOE		AOE		AOE
	BK2E ⁽¹⁾		OSSR		OSSR
	BK2P ⁽¹⁾		OSSI		OSSI
	BKF[3:0] ⁽¹⁾	CCER	CCxP	CCER	CCxP
	BK2F[3:0] ⁽¹⁾		CCxNP		CCxNP
				CCMRx	OCxM
					OCxPE

1. 在 STM32F30x/31x 系列中存在的位。

注：复位后只能对 **LOCK** 位执行一次写操作。一旦 **BDTR** 寄存器已写入，其中的数据就会冻结，直到下一次复位。

4.3.4 用于反馈测量的特殊功能

编码模式

增量编码器是一种传感器，用于在电机控制应用中测量角位置。

通常情况下，增量编码器可产生三种信号：**A** 相、**B** 相和索引。

电机的转向取决于 **A** 相超前 **B** 相还是 **B** 相超前 **A** 相。第三个通道称为索引脉冲，每转一圈产生一个脉冲，其为测量绝对位置的基准。

A 相和 **B** 相输出信号与编码器接口连接，用于计算频率，进而推导出转速和位置。转速和位置信息可在 **X2** 或 **X4** 分辨率下测得。下图显示了编码器的接口功能。

每次输入 **TI1** 和 **TI2** 上发生电平转变时，定时器的计数器都会进行递增或递减计数。

编码器模式下的 TIM 配置

1. 选择并配置定时器输入：
 - 输入选项：
 - TI1 连接到 TI1FP1，CCMR1 寄存器中 CC1S= “01”
 - TI2 连接到 TI2FP2，CCMR1 寄存器中 CC2S= “01”
 - 输入极性：
 - CC1P= “0” 且 CC1NP= “0”（CCER 寄存器，TI1FP1 非反相，TI1FP1=TI1）。
 - CC2P= “0” 且 CC2NP= “0”（CCER 寄存器，TI1FP2 非反相，TI1FP2=TI2）。
2. 选择编码器模式
 - 编码器模式 1（TI2 上分辨率 X2）：SMCR 寄存器中 SMS= “001”。
 - 编码器模式 2（TI1 上分辨率 X2）：SMCR 寄存器中 SMS= “010”。
 - 编码器模式 3（TI1 和 TI2 上分辨率 X4）：SMCR 寄存器中 SMS= “011”。
3. 使能定时器计数器
 - 将计数器使能位置 1，CR1 寄存器中 CEN= “1”。

霍尔传感器

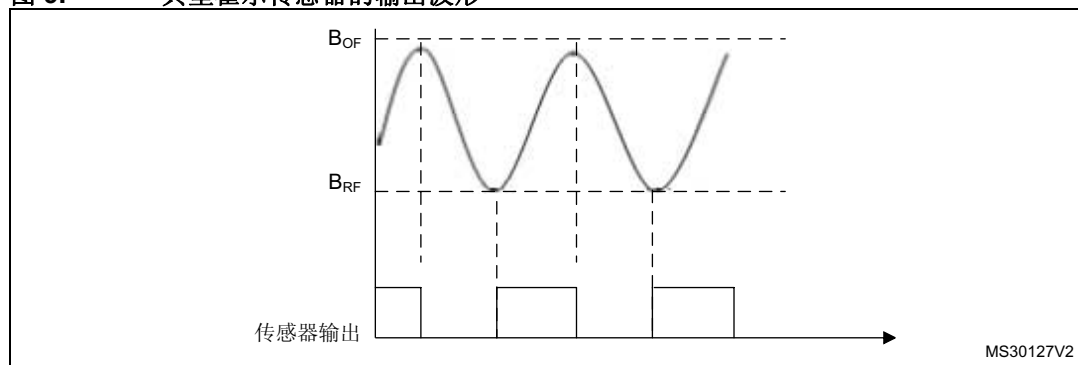
霍尔传感器是一种基于霍尔效应的传感器：将一个导体置于磁场中时，在与电流和磁场都垂直的方向上会产生一个电压。

可提供数字输出的四种霍尔传感器集成电路器件分别是：单极开关、双极开关、全极开关和锁存器。它们的主要区别在于输出波形（脉冲持续时间）。

数字霍尔传感器可提供与其所在磁场相关的数字输出。当磁场增大并超过 **BRP**（磁场释放点值）时，将开启输出。当磁场减小并低于 **BOP**（磁场动作点值）时，将关闭输出。

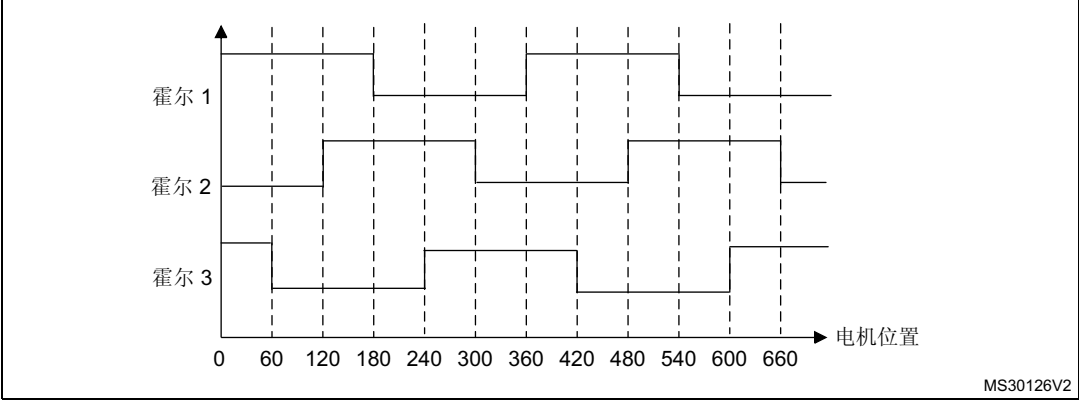
图 9 给出了典型霍尔传感器的输出波形。

图 9. 典型霍尔传感器的输出波形



霍尔传感器通常用于 3 相电机控制。图 10: 换向序列 给出了换向序列。

图 10. 换向序列



如何使用霍尔传感器接口

STM32 定时器可通过标准输入（CH1、CH2 和 CH3）与霍尔效应传感器连接。通过 CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

将从模式控制器配置为复位模式：从输入为 TI1F_ED。这样，每次 3 个输入中有一个发生切换时，计数器便会从 0 重新开始计数。这样将产生由霍尔输入的任何变化而触发的时基。

将通道 1 配置为输入捕捉模式，捕捉信号为 TRC。捕捉值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

霍尔传感器接口模式下的 TIM 配置

1. 向 CR2 寄存器中的 TI1S 位写入“1”，使 3 个定时器输入经过或运算后进入 TI1 输入通道。
2. 编程时基：向 ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
3. 将通道 1 编程为捕捉模式（选择 TRC）：向 CCMR1 寄存器中的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。

5 特殊应用

5.1 红外应用

STM32 通用定时器可用于仿真多个红外协议。应用笔记 AN3174 “通过 STM32F10xx 微控制器实现 RC5 红外遥控接收器”中介绍了该应用的示例。

该应用笔记介绍了使用 STM32 通用定时器实现 RC5 接收器的软件解决方案。

5.2 3 相交流和 PMSM 控制电机

STM32 带 ADC 和 DAC 的高级定时器和通用定时器可用于控制两种 3 相电机：交流感应电机与 PMSM，并采用下列不同的电流检测方法：

- 独立电流检测
- 三个分流电阻
- 单个分流电阻（ST 获专利的解决方案）

STM32 定时器还可以用于反馈环路，以与转子位置反馈中使用的以下各种传感器进行连接：

- 测速传感器
- 正交编码器
- 霍尔传感器：60° 和 120° 放置

有关详细信息，请参见：stm32_pmsm_foc_motorcontrol_fmlib。

5.3 六步模式

六步模式是 STM32 高级定时器的特殊模式。当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

该模式的应用示例为步进电机控制。

将定时器配置为生成六步信号来控制步进电机：

- 时基配置：预分频器、周期、时钟源
- 将通道 1、2、3 和 4 配置为 PWM 模式
- 将捕捉比较预装载控制位 CCPC 置 1
- 使能换向中断源
- 使用系统时钟生成时基
- 每次发生换向事件后，将针对下一个换向事件更新 TIM 配置。

有关在此模式下使用定时器的详细信息，请参见 /Project/STM32xx_StdPeriph_Examples/TIM/6Steps 文件夹中 STM32xx 标准外设库提供的示例。

注：该示例不适用于 STM32L1xx 器件。

6 版本历史

表 7. 文档版本历史

日期	版本	变更
2012 年 02 月 21 日	1	初始版本。
2012 年 10 月 22 日	2	增加了对 STM32F30x、STM32F31x、STM32F37x 和 STM32F38x 的支持。



请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与ST产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有ST产品均根据ST的销售条款出售。

买方自行负责对本文所述ST产品和服务的选择和使用，ST概不承担与选择或使用本文所述ST产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为ST授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在ST的销售条款中另有说明，否则，ST对ST产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且/或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的ST产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致ST针对本文所述ST产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大ST的任何责任。

ST和ST徽标是ST在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2014 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com

