SR5 E1 line – cross series timer overview

Introduction

The purpose of this document is to present an overview of the timer peripherals for the SR5 E1 line products listed in the following table.

It describes the various modes and specific timer features, such as clock sources and it explains how to use the available modes and features. It explains how to compute the time base in each configuration.

This technical note describes the timer synchronization sequences and the advanced features for motor control applications, in addition to the general-purpose timer modes.

For each mode, the document provides typical configurations and implementation examples.

Table 1. Device list

Device	Part number
SR5E1	SR5E1E3, SR5E1E7

1 Overview

The SR5 E1 line devices, based on the Arm® cores, have various built-in timers outlined as follows:

- **General-purpose timers**: these timers may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare and PWM), one-pulse mode, and sensor interface (encoder, hall sensor).
- **Advanced-control timers**: these timers, in addition to the general-purpose features, include several features related to motor control and digital power conversion applications, for example, complementary signals with deadtime insertion and emergency shutdown inputs.
- **Basic timers** consist of a 16-bit or 32-bit auto-reload counter driven by a programmable prescaler and may be used for time-base generation. They can also be used for triggering the digital-to-analog converter.
- **High-resolution timers** are specialized timer peripherals designed to drive power conversion in lighting and power source applications. They can also be used in other fields that require very fine timing resolution.

Note: Arm is a registered trademark of Arm limited (or its subsidiaries) in the US and/or elsewhere.

Table 2. Overview of timers' availability summarizes the timers' availability in the Stellar E product series and Table 3. Overview of timers' features presents a general overview of timers' features.

Table 2. Overview of timers' availability

Timer type	SR5E1
Advanced-control timers	TIM1 / TIM8
General-purpose timers	TIM2 / TIM3 / TIM4 / TIM15 / TIM16
Basic timers	TIM6 / TIM7 / TIM_TS
High-resolution timers	HRTIM1 / HRTIM2

Table 3. Overview of timers' features

Parameter	TIM1	TIM8	TIM2	TIM5	TIM3	TIM4	TIM15	TIM16	TIM6	TIM7	TIM_TS
	Advanced		General-purpose						Basic		
Resolution [bits]	16		32		16						32
Prescaler [bits]	16										
APB bus	APB2		APB1	APB2	APB1	APB2			APB1		
Max clock frequency [MHz](1)	306.70		153.35	306.70	153.35	306.70			153.35		
Counter direction	Up, down, up and down						Up				
Number of channels	6(2)		4(3)				2(4)	1(5)	0		
DMA	Yes										
Input capture mode	Yes								No		
PWM input mode	Yes							No			
Output compares mode	Yes								No		
PWM modes	Standard Asymmetric Combined Combined three-phase Six-step PWM		Standard Asymmetric Combined					Standard	No		
Programmable dead-time	Yes(6)		No				Yes(7)	No			
PWM dithering	Yes								No		

Parameter	TIM1	TIM8	TIM2	TIM5	TIM3	TIM4	TIM15	TIM16	TIM6	TIM7	TIM_TS
	Advanced		General-purpose						Basic		
Break inputs	2 (bidirectional)		No				1 (bidirectional)		No		
One-Pulse mode	Yes								No		
Retrig. One-Pulse mode	Yes							No			
Encoder modes	Quadrature Clock plus direction Directional clock Index management Error management						No				
Timer input XOR function	Yes		No				Yes		No		

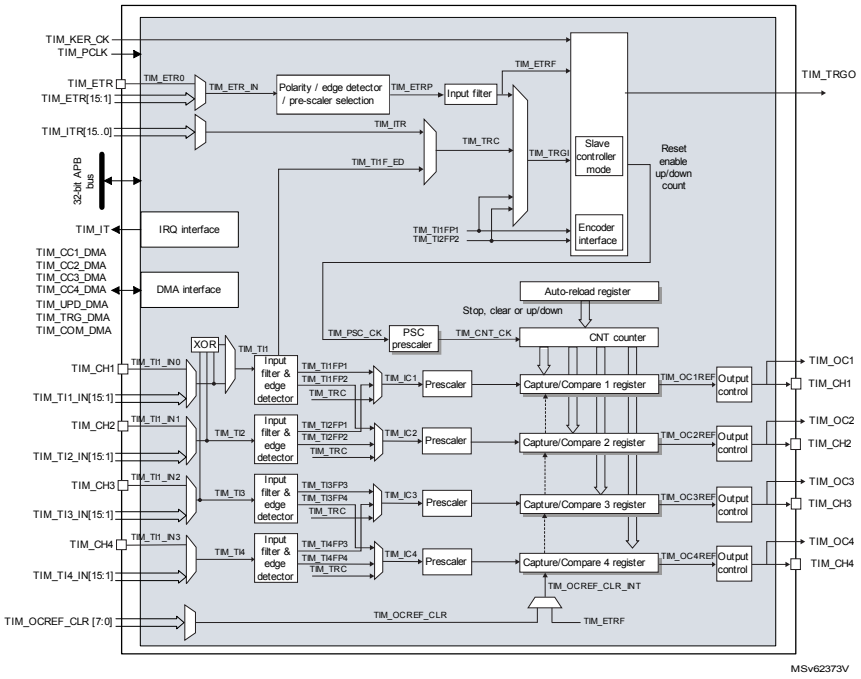
1. *This is TIM_KER_CK.*
2. *Channel 1/1N, …, 4/4N, 5 & 6 (output only, not available externally on pads).*
3. *Channel 1, …, 4*
4. *Channel 1/1N, 2*
5. *Channel 1/1N*
6. *Only Channel 1, …, 4*
7. *Only Channel 1*

2 General-purpose features

2.1 Block diagram

The figure below shows the block diagram of a general-purpose timer. Additional features are available for advanced-control timers. For further details on the block diagram of the advanced-control timers, refer to “SR5 E1 line of Stellar electrification MCUs” (RM0483).

Figure 1. General-purpose timer block diagram



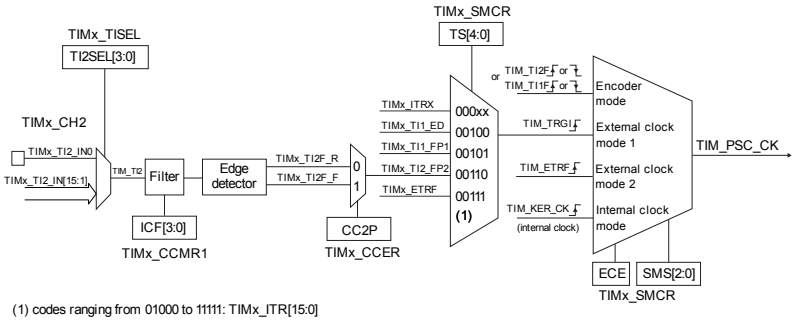
2.2 Clock input sources

The following clock sources can provide the TIM counter clock:

- Internal clock (by internal signal TIM_KER_CK).
- Internal trigger clock (by internal signal TIM_ITRx).
- External clock:
  - External clock mode1 (by TIMx_TI1 or TIMx_TI2 input pins).
  - External clock mode2 (by TIMx_ETR input pin).

The following table shows the timer clock circuitry. TIMx_TI1_FP1 is selected by a similar structure.

Figure 2. **TIM_TI2 external clock connection example**



2.2.1 Internal clock

Internal clock (TIM_KER_CK)

The timer counter receives the clock through the TIM_KER_CK (TIM kernel clock).

The TIM_KER_CK, synchronous with the TIM_PCLK clock (derived from the same source), is twice the relevant APB clock provided by the RCC (reset and clock control).

The internal clock TIM_KER_CK clocks the timer counter if the SMS[3:0] in the TIMx_SMCR register is set to 0000b (default value). By the TIMx_PSC register is selected the prescaler between the TIM_KER_CK clock and the timer counter clock TIMx_CNT_CK and, by the DIR bit in the TIMx_CR1 register is selected the up-counting or down-counting mode. By the CEN bit in the TIMx_CR1 register, the counter is enabled. Note that the counter starts counting one clock cycle after setting the CEN bit in the TIMx_CR1 register.

Internal trigger clock (ITRx)

This is a particular timer synchronization mode, using one timer as a prescaler for another timer.

The first timer update event or output compare signal is used as a clock for the second one.

See the section “Timer synchronization” for further details.

2.2.2 External clock

The external clock is provided in two modes:

1. By the TI1 or TI2 input pins (mode1).
2. By the ETR input pin (mode2).

In these cases, the counter clock is provided by an external signal connected to the TIy pins or the ETR pin and the external signal frequency cannot be greater than the TIM_PCLK clock frequency, that always must be provided to the timer. Moreover, the external clocks are not directly feeding the prescaler, but they are first synchronized with the TIM_PCLK clock through dedicated logic.

External clock mode1 (TI1 or TI2 pins)

For example, to configure the timer to up-count on the rising edge of the TI2 input, use the following procedure:

1. In the TIMx_CCMR1 register:
   a. Configure the capture/compare 2 to detect the edges on the TIMx_TI2 input by setting CC2S[1:0] to 01b.
   b. Configure the input filter duration by setting IC2F[3:0] (if no filter is needed, keep it to the default value).
2. In the TIMx_CCER register:
   a. Select rising edge polarity by setting CC2P[1:0] to 00b and CC2NP[1:0] to 00b.

Note: b. *Both edges can be selected only for TI1 input.*

   c. Enable capture by setting the CC2E bit to 1b.
3. In the TIMx_SMCR register:
   a. Configure the timer in external clock mode1 by setting SMS[3:0] to 0111b.
   b. Select TI2 as the trigger input source by setting the TS[3:0] to 00110b.
   c. In the TIMx_CR1 register, enable the counter by setting the CEN bit to 1b.

External clock mode2 (ETR pin)

For example, to configure the timer to up-count each two rising edges on the ETR pin, use the following procedure:

1. In the TIMx_AF1 register, select the proper source (internal or external) setting ETRSEL[3:0].
2. In the TIMx_SMCR register:
   a. As no filter is needed in this example, setting ETF[3:0] to 0000b.
   b. Configure the prescaler by setting ETPS[1:0] to 01b.
   c. Select rising edge detection by setting the ETP bit to 0b.
   d. Enable external clock mode2 by setting the ECE bit to 1b.
3. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

Note: A delay between the rising edge on the ETR signal and the actual clock of the counter can occur and it is due to the resynchronization circuit on the prescaled signal ETRP. Consequently, the maximum frequency that can be correctly captured by the counter is at most ¼ of the TIM clock frequency. When the prescaled signal ETRP is faster, the user must apply a division of the external signal by a proper ETPS prescaler setting.

2.3 Time base generator

The main block of a TIM is a 16-bit/32-bit counter (TIMx_CNT) with its related prescaler register (TIMx_PSC), auto reload register (TIMx_ARR) and repetition counter register (TIMx_RCR), if present. The counter in general can count up, down, or both up and down. The counter, prescaler, the auto reload and repetition counter register can be written or read even when the counter is running.

The auto reload register is preloaded. Writing to or reading from the auto reload register accesses the preload register. The content of the preload register is transferred into the shadow register permanently or at each update event (UEV), depending on the auto reload preload enable bit (ARPE) in the TIMx_CR1 register. The update event is sent when the counter reaches the overflow (or underflow when down-counting) and if the UDIS bit equals 0 in the TIMx_CR1 register. It can also be generated by software. Depending on the clock, prescaler, auto reload and repetition counter (if present) parameters, the 16-bit timer can generate an UEV from a few nanoseconds to minutes. The 32-bit timers provide a wider range.

The prescaler divides the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (TIMx_PSC). It can be changed on the fly as this control register is buffered by a shadow register. The new prescaler ratio takes effect at the next UEV.

Counter modes

The counter mode is selected by setting the DIR bit and CMS[1:0] in the TIMx_CR1 register:

1. Up-counting mode (DIR = 0b and CMS = 00b):
   a. The counter counts from 0 to the auto reload value (stored in the TIMx_ARR register), then restarts from 0 and generates a counter overflow event. If the repetition counter is used, the UEV is generated after the occurrence of (TIMx_RCR + 1) overflow events, else the UEV is generated at each overflow event.
2. Down-counting mode (DIR = 1b and CMS = 00b):
   a. The counter counts from the auto reload value (stored in the TIMx_ARR register) down to 0, then restarts from the auto reload value and generates a counter underflow event. If the repetition counter is used, the UEV is generated after the occurrence of (TIMx_RCR + 1) underflow events, else the UEV is generated at each underflow event.
3. Up/Down-counting mode (center-aligned mode):
   a. The counter counts from 0 up to the auto reload value – 1 and generates a counter overflow event, then counts from the auto reload value down to 1, generates a counter underflow event and restarts counting from 0. Center-aligned mode is active when the CMS[1:0] in TIMx_CR1 register is not equal to 00b. In this mode, the DIR direction bit in the TIMx_CR1 register cannot be written. It is updated by hardware and gives the current direction of the counter. If the repetition counter is used, the UEV is generated after the occurrence of (TIMx_RCR + 1) over/underflow events, else the UEV is generated at each event.

Update event UEV

The UEV frequency is calculated by the formula TIMx_CLK/((TIMx_PSC + 1) * (TIMx_ARR + 1) * (TIMx_RCR + 1)) where TIMx_CLK is the selected timer clock period (or frequency).

For example, if the timer clock frequency is 306.7 MHz, prescaler is set to 1, auto reload is set to 65535 and no repetition counter, the UEV frequency is 306.7 * (10^6))/((1 + 1) * (65535 + 1) * (1) = 2.34 kHz.

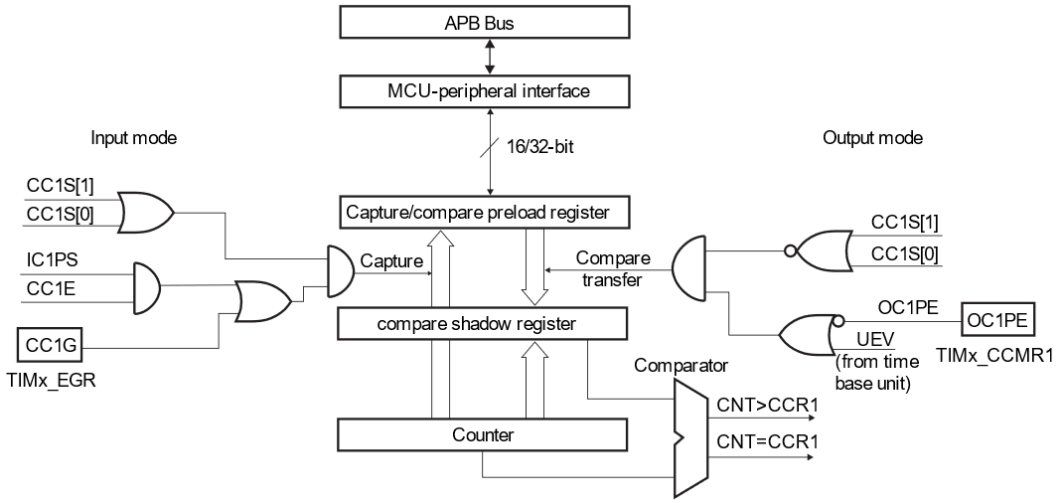
The UEV can be generated also by software (setting the UG bit in the TIMx_EGR register) or by hardware through the subordinate mode controller and it can be disabled by setting the UDIS bit in the TIMx_CR1 register. This is to avoid updating the shadow registers while writing new values in the preload registers. Then no update event occurs until the UDIS bit has been written to 0. However, the counter restarts according to the selected counter mode and the prescaler counter restarts from 0 (but the prescale rate does not change).

An interrupt or/and a DMA request can be generated when the UIE bit or/and UDE bit are set in the TIMx_DIER register and the events generating an update interrupt or DMA request can be selected by the URS bit (update request selection) in the TIMx_CR1 register. For example, with URS = 1b, setting the UG bit generates an UEV but without setting the UIF flag (thus no interrupt or DMA request is sent). This is to avoid generating both update and capture interruptions when clearing the counter on the capture event.

2.4 Capture/compare channels

Each Capture/Compare channel is built around a capture/compare register (including a shadow register), an input stage for capture (with a digital filter, multiplexing, and prescaler), and an output stage (with comparator and output control). The figure below shows the timer capture/compare circuitry for timer channel 1. The circuitry is the same for the other timer channels.

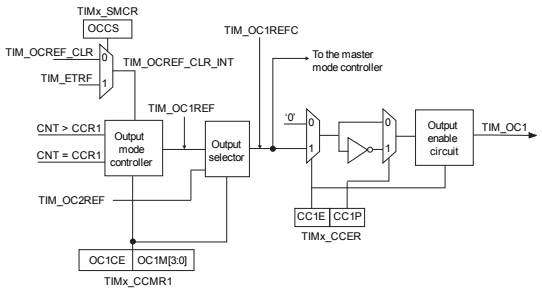
Figure 3. Capture/compare channel one main circuit



The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register. In capture mode, captures are actually done in the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register, which is compared to the counter.

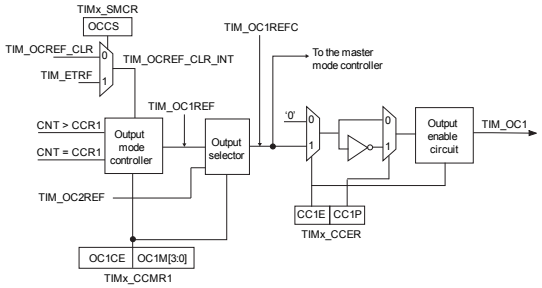
The input stage samples the corresponding TIMx_TIy input to generate a filtered signal TIMx_TIyF. Then, an edge detector with polarity selection generates a signal (TIMx_TIyFPy) which can be used as trigger input by the subordinate mode controller or as the capture command. The selected trigger input TIMx_ICy is prescaled before sent to the capture register. The following figure shows the input stage of the capture/compare channel 1. The circuitry is the same for the other timer capture/compare channels.

Figure 4. Capture/compare channel (example: channel 1 input stage)



The output stage generates an intermediate waveform, which is then used for reference TIMx_OCyREF (active high) and the polarity acts at the end of the chain. The figure below shows the output stage of a capture/compare channel 1. The circuitry is the same for the other timer capture/compare channels.

Figure 5. Output stage of capture/compare channel 1



2.5 Input capture mode

The input capture mode can be used to measure the period of an external signal.

The maximum measurable period depends on the timer clock, prescaler and resolution.

The Capture/Compare registers TIMx_CCRy are used to latch the value of the counter after a transition detected on the selected input signal. When a capture occurs, the corresponding CCyIF flag in the TIMx_SR register is set and an interrupt, or a DMA request can be sent if they are enabled. If a capture occurs while the CCyIF flag was already high, then the overcapture flag CCyOF in the TIMx_SR register is set. CCyIF can be cleared by writing it to 0 or by reading the captured data stored in the TIMx_CCRy register, CCyOF is cleared by writing it to 0.

It is recommended to read the data before the overcapture flag to avoid missing an overcapture which could happen after reading the flag and before reading the data. Note that IC interrupt and/or DMA requests can be generated by software by setting the corresponding CCyG bit in the TIMx_EGR register.

For example, TIMx_CCR1 can capture the counter value on the rising edge of the signal on TIMx_TI1:

1. Select TI1 external pin as TIMx_TI1 input by setting TI1SEL[3:0] in the TIMx_TISEL register to 0000b.
2. Configure the CC1 channel as input on TIMx_TI1 by setting CC1S[1:0] in the TIMx_CCMR1 register to 01b. As soon as CC1S[1:0] is set different from 00b, the channel is configured in input, and the TIMx_CCR1 register becomes read only.
3. Program the needed input filter duration. Assuming the input signal is not stable during the first five internal clock cycles, the filter duration must be longer than these five clock cycles. By setting IC1F[3:0] in the TIMx_CCMR1 register to 0011b, a valid transition is detected on the input when eight consecutive samples with the same new level are acquired at TIM_KER_CK frequency.
4. Select the rising edge of the transition by setting the CC1P and CC1NP bits to 0b in the TIMx_CCER register.

5. Program the input prescaler. In our example, the capture is performed at each valid transition, so the prescaler is disabled and IC1PS[1:0] is set to 00b (default value) in the TIMx_CCMR1 register.
6. Enable capture by setting the CC1E bit in the TIMx_CCER register to 1b.
7. If needed, enable the related interrupt request by setting the CC1IE bit in the TIMx_DIER register, and/or the DMA request by setting the CC1DE bit in the TIMx_DIER register.

To get the external signal period, two consecutive captures are needed. The period is calculated by the formula CAPT_DIFF / (TIM_CLK * (PSC + 1) * (ICxPSC) * POL_INDEX) where CAPT_DIFF = CCR1(t_{n+1}) - CCR1(t_n) if CCR1(t_{n+1}) >= CCR1(t_n) and CAPT_DIFF = (ARR - CCR1(t_n)) + CCR1(t_{n+1}) if CCR1(t_{n+1}) < CCR1(t_n) and the POL_INDEX is 1 if the rising or falling edge is used, and 2 if both edges are used assuming a 50% duty cycle.

The timer counter can reset after each rising edge detected on the timer input channel by:

1. Selecting TI1FP1 as the input trigger by setting TS[4:0] in the TIMx_SMCR register to 00101b;
2. Selecting the reset mode as the subordinate mode by setting SMS[3:0] in the TIMx_SMCR register to 0100b.

Using this configuration, when an edge is detected, the counter is reset and the period of the external signal is automatically given by the value on the TIMx_CCR1 register. This configuration is available only for channels 1 and 2 and, in this case, the input capture prescaler ICPSC is not considered in the period computation. The period is calculated by the formula CCR1 / (TIM_CLK * (PSC + 1) * POL_INDEX) where the POL_INDEX is 1 if the rising or falling edge is used, and 2 if both edges are used assuming a 50% duty cycle.

2.6 PWM input mode

This mode is a particular case of the input capture mode, and it allows to measure both the period and the duty cycle of a PWM signal connected to an input TIMx_TIy_INy:

- The TIMx_CCRy register holds the period value (interval between two consecutive rising edges).
- The TIMx_CCRy + 1 register holds the duty cycle (interval between two consecutive rising and falling edges).

The configuration is similar to one of the input capture modes with the following differences:

- Two CCy (Capture/Compare) channels are connected to the same TIMx_TIy_INy input.
- These two CCy channels are active on edges with opposite polarity.
- One of the related TIMx_TIy_FPy signals is selected as trigger input with the subordinate mode controller in reset mode.

For example, the period and the duty cycle of a PWM signal applied on TIMx_TI1 external pin can be measured using the following procedure:

1. Select the proper source TIMx_TI1 external pin by setting TI1SEL[3:0] to 0000b in the TIMx_TISEL register.
2. Program the needed input filter duration by setting IC1F[3:0] in the TIMx_CCMR1.
3. Select the active polarity for TIMx_TI1_FP1 (used both for capture in TIMx_CCR1 and counter clear) by setting CC1P and CC1NP both to 0b in the TIMx_CCER register (active on rising edge).
4. Select the active polarity for TIMx_TI1_FP2 (used for capture in TIMx_CCR2) by setting CC2P bit to 1b and CC2NP to 0b in the TIMx_CCER register (active on falling edge).
5. Select TIMx_TI1 as the active input for the CC1 channel by setting CC1S[1:0] to 01b in the TIMx_CCMR1 register.
6. Select TIMx_TI1 as the active input for the CC2 channel by setting CC2S[1:0] to 10b in the TIMx_CCMR1 register.
7. Select TIMx_TI1_FP1 as the valid trigger input by setting TS[4:0] to 00101b in the TIMx_SMCR register.
8. Configure the subordinate mode controller in reset mode by setting SMS[3:0] to 0100b in the TIMx_SMCR register.
9. Enable the capture by setting CC1E and CC2E both to 1b in the TIMx_CCER register.

2.7 Output compare modes

This mode can be used to control the output waveform or to indicate when a period has elapsed.

The following options are available by setting OCyM[3:0] in the TIMx_CCMRy register:

- **Timing** by setting OCyM[3:0] to 0000b. The comparison between the capture/compare register TIMx_CCRy and the counter TIMx_CNT has no effect on the output. This mode is used to generate a timing base.

- **Active** by setting OCyM[3:0] to 0001b. The OCyREF signal is forced to the active level when the counter TIMx_CNT matches the capture/compare register TIMx_CCRy.
- **Inactive** by setting OCyM[3:0] to 0010b. The OCyREF signal is forced to the inactive level when the counter TIMx_CNT matches the capture/compare register TIMx_CCRy.
- **Toggle** by setting OCyM[3:0] to 0011b. The OCyREF signal toggles when the counter TIMx_CNT matches the capture/compare register TIMx_CCRy.
- **Forced** inactive or active by setting OCyM[3:0] to 0100b or 0110b. In output mode (CCyS[1:0] set to 00b in the TIMx_CCMRy register), each output compare signal (OCyREF and then OCy) can be forced to the inactive or active level directly by the software, independently of any comparison between the output compare register and the counter. Anyway, the comparison between the TIMx_CCRy shadow register and the counter TIMx_CNT is still performed and allows the flag to be set. Interrupt and DMA requests can be sent accordingly.

To configure the timer in one of these modes:

1. Select the clock source (internal, external, prescaler).
2. Configure the TIMx_ARR and TIMx_CCRy registers.
3. Configure the output mode:
   a. Select the output compare mode: timing/active/inactive/toggle;
   b. In the case of active, inactive and toggle modes, select the polarity by writing CCyP in TIMx_CCER register;
   c. Disable the preload feature for CCy by writing OCyPE in the TIMx_CCMRy register;
   d. Enable the capture/compare output by writing CCyE in the TIMx_CCERy register.
4. Enable the counter by setting the CEN bit in the TIMx_CR1 register.
5. Set the CCyIE/CCyDE bit if an interrupt/DMA request is to be generated.

2.8 PWM modes

PWM mode allows to generate a signal with a frequency determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRy register and edge-aligned or center-aligned PWM can be generated depending on the value of CMS[1:0] in the TIMx_CR1 register.

The PWM mode can be selected independently on each channel y by setting OCyM[3:0] to 0110b (PWM mode 1) or 0111b (PWM mode 2) in the TIMx_CCMRx register.

For PWM mode 1:

- in up-counting, channel y is active as long as TIMx_CNT < TIMx_CCRy, otherwise it is inactive;
- in down-counting, channel y is inactive as long as TIMx_CNT > TIMx_CCRy, otherwise it is active.

For PWM mode 2:

- in up-counting, channel y is inactive as long as TIMx_CNT < TIMx_CCRy, otherwise it is active:
- in down-counting, channel y is active as long as TIMx_CNT > TIMx_CCRy, otherwise it is inactive.

The output polarity (active high or active low) is software programmable using the CCyP bit in the TIMx_CCER register and the output is enabled by setting the CCxE bit in the TIMx_CCER register.

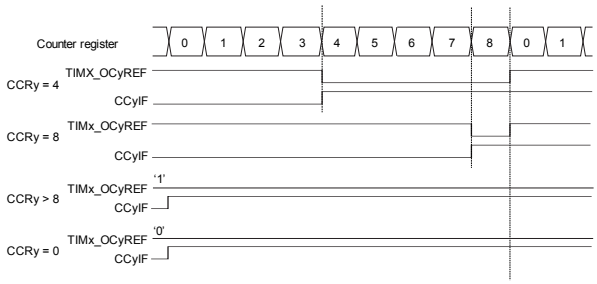
The corresponding preload register must be enabled by setting the OCyPE bit in the TIMx_CCMRy register to allow the synchronous of the signal duty cycle, and eventually the auto reload preload register by setting the ARPE bit in the TIMx_CR1 register to allow the synchronous change of the signal period.

PWM edge-aligned mode (standard)

The following figure shows some edge-aligned PWM waveforms for PWM mode 1 with TIMx_ARR = 1000b and DIR = 0b (up-counting). The reference PWM signal TIMx_OCyREF is high as long as TIMx_CNT < TIMx_CCRy else it becomes low. If the compare value in TIMx_CCRy is greater than the autoreload value (in TIMx_ARR), then TIMx_OCyREF is held at 1. If the comparable value is 0, then TIMx_OCyREF is held at 0.

The following figure shows some edge-aligned PWM waveforms where TIMx_ARR = 8.

Figure 6. Output compares mode, toggle on TIM_OC1

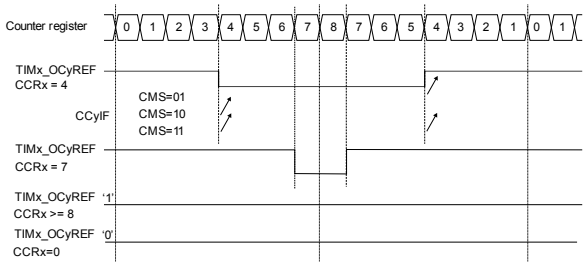


PWM center-aligned mode (standard)

The following figure shows some center-aligned PWM waveforms for PWM mode 1 with TIMx_ARR = 1000b, DIR = 0b (up-counting), and CMS = 01b (the counter counts up and down alternatively). Output compare interrupt flags of channels configured in output are set based on the value of CCyS[1:0] in the TIMx_CCMRy register.

The following figure shows some center-aligned PWM waveforms.

Figure 7. Edge-aligned PWM waveforms (ARR = 8)



Asymmetric PWM mode

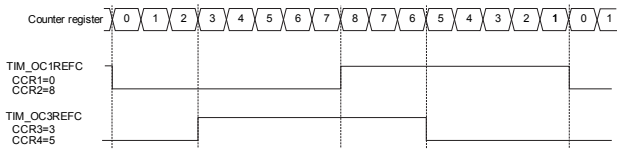
Asymmetric mode allows center-aligned PWM signals to be generated with a programmable phase shift.

For each channel, the phase shift and the pulse length are programmed using the two TIMx_CCRy registers (TIMx_CCR1 and TIMx_CCR2 or TIMx_CCR3 and TIMx_CCR4), the frequency is determined by the value of the TIMx_ARR register. Asymmetric PWM mode can be selected independently on two channels by programming the OCyM bits in the TIMx_CCMRy register:

- OCyM = 1110b to use the asymmetric PWM mode 1, in this mode, the output reference has the same behavior as in PWM mode 1. When the counter is counting up the output reference is identical to OC1REF (OC3REF), when the counter is counting down, the output reference is identical to OC2REF (OC4REF).
- OCyM = 1111b to use the asymmetric PWM mode 2, in this mode, the output reference has the same behavior as in PWM mode 2. When the counter is counting up the output reference is identical to OC1REF (OC3REF), when the counter is counting down, the output reference is identical to OC2REF (OC4REF).

The following figure shows an example of signals that can be generated using asymmetric PWM mode (channels 1 to 4 are configured in asymmetric PWM mode 2).

Figure 8. Generation of two phase-shifted PWM signals with 50% duty cycle



Combined PWM mode

Combined mode allows edge or center-aligned PWM signals to be generated with programmable delay and phase shift between respective pulses. To generate a combined signal, the two TIMx_CCRy registers (TIMx_CCR1 and TIMx_CCR2 or TIMx_CCR3 and TIMx_CCR4) must be used to program the delay and the phase shift. The frequency is determined from the value of the TIMx_ARR register, and the combined signal is made of an OR or an AND logical combination of two reference PWMs.

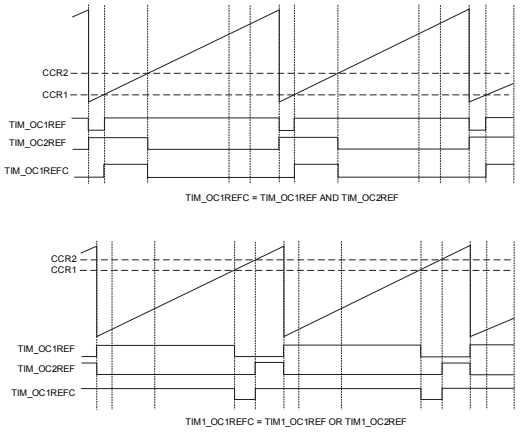
Combined PWM mode can be selected independently on two channels by programming the OCyM bits in the TIMx_CCMRy register:

- OCyM = 1100b to use the combined PWM mode 1, in this case, the combined output reference has the same behavior as in PWM mode 1. The combined output reference is the logical OR between OC1REF (OC3REF) and OC2REF (OC4REF).
- OCyM = 1101b to use the combined PWM2, in this case, the combined output reference has the same behavior as in PWM mode 2. The combined output reference is the logical AND between OC1REF (OC3REF) and OC2REF (OC4REF).

The following figure shows an example of signals that can be generated using combined PWM mode, obtained with the following configuration:

- Channel one is configured in combined PWM mode 2.
- Channel two is configured in PWM mode 1.
- Channel three is configured in combined PWM mode 2.
- Channel four is configured in PWM mode 1.

Figure 9. Combined PWM mode on channel 1 and 3



2.9 Timer in one-pulse mode

One pulse mode (OPM) is a particular case of the input capture mode and the output compare mode. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

For example, to generate a positive pulse on TIMx_OC1 with a length of t_{PULSE} and after a delay of t_{DELAY} as soon as a positive edge is detected on the TIMx_TI2 external input pin:

1. Select the proper source the TIMx_TI2 external pin by setting TI2SEL[3:0] to 0000b in the TIMx_TISEL register.

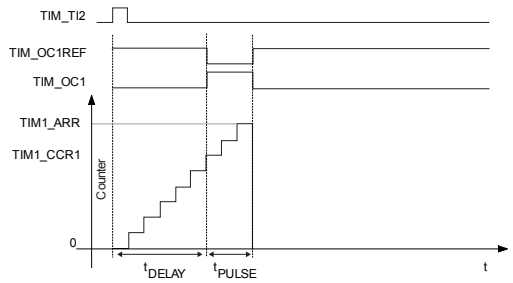
2. Program the needed input filter duration by setting IC2F[3:0] in the TIMx_CCMR1.
3. Configure the CC2 channel as input on TIM_TI2 by setting CC2S[1:0] in the TIM_CCMR1 register to 01b.
4. Select the active polarity for TIM_TI2_FP2 (used both for capture in TIMx_CCR1 and counter clear) by setting CC1P and CC1NP both to 0b in the TIMx_CCER register (active on rising edge).
5. Select TIM_TI2_FP2 as the valid trigger input by setting TS[4:0] to 00110b in the TIMx_SMCR register.
6. Configure TIM_TI2_FP2 to start the counter by setting SMS to 110b in the TIMx_SMCR register.
7. Define the OPM waveform by setting the TIMx_ARR, the TIMx_CCR1, and the TIMx_PSC registers:

- t_{DELAY} = TIMx_CCR1/(TIMx_CLK/(TIMx_PSC + 1));
- t_{PULSE} = (TIMx_ARR + 1–TIMx_CCR1)/(TIMx_CLK/(TIMx_PSC + 1));
- a pulse can be correctly generated only if TIMx_CNT < TIMx_CCRy ≤ TIMx_ARR and TIMx_CCRy > 0.

1. To build a waveform with a transition from 0 to 1 on a compare match and a transition from 1 to 0 on the auto reload match, select the PWM mode 2 by setting OC1M[3:0] to 0111b in the TIMx_CCMR1 register.
2. Optionally the preload registers can be enabled by setting OC1PE to 1b in the TIMx_CCMR1 register and by setting ARPE to 1b in the TIMx_CR1 register. In this case, written the compare value in the TIMx_CCR1 register, the auto reload value in the TIMx_ARR register, by setting the UG bit an UEV is generated and the timer starts waiting for the external trigger event on TIMx_TI2 external input pin.
3. In the TIMx_CCER, CC1P is set to 0b, in the TIMx_CR1 register, DIR is set to 0b and CMS[1:0] is set to 00b.
4. Finally, since only one pulse (single mode) is needed, OPM in the TIMx_CR1 register is set to 1b to stop the counter at the UEV (when the counter rolls over from the auto reload value back to 0).

The figure below shows an example of one-pulse mode.

Figure 10. Example of one pulse mode



Particular cases: TIMx_OCy fast enable.

In one-pulse mode, the edge detection on the TIMx_TIy input sets the CEN bit, which enables the counter. Then the comparison between the counter and the compared value makes the output toggle. But several clock cycles are needed for these operations, and it limits the minimum delay t_{DELAY_MIN}. If a waveform with the minimum delay is required, the OCyFE bit can be set in the TIMx_CCMRy register. Then TIMx_OCyREF (and TIMx_OCy) is forced in response to the stimulus, without taking into account the comparison. Its new level is the same as if a compared match had occurred. OCyFE acts only if the channel is configured in PWM1 or PWM2 mode.

2.10 Retriggerable one-pulse mode

The retriggerable one-pulse mode is a one-pulse mode with the following differences:

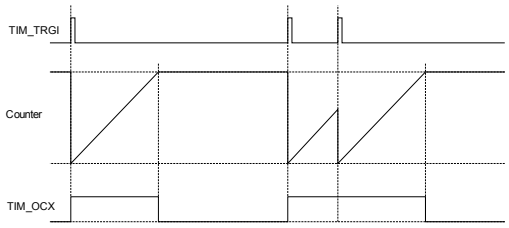
- The pulse starts as soon as the trigger occurs (no programmable delay);
- The pulse is extended if a new trigger occurs before the previous one is completed.

If the counter is configured in up-counting mode, the corresponding CCRy must be set to 0, in this case, the pulse length is determined by the value stored in the ARR register. If the timer is configured in down-counting mode, the ARR register must be set to 0 and, in this case, the pulse length is determined by the value stored in the CCRy register. As for the OPM mode, there are two retriggerable one-pulse modes:

1. Retriggerable OPM mode 1 is selected by setting OCyM[3:0] in the TIMx_CCMRy register to 1000b:
   – In up-counting mode, the channel is inactive until a trigger event is detected, the comparison is performed like in PWM mode 1, then the channel becomes inactive again at the next update;
   – In down-counting mode, the channel is active until a trigger event is detected, the comparison is performed like in PWM mode 1, then the channel becomes active again at the next update.
2. Retriggerable OPM mode 2 is selected by setting OCyM[3:0] in the TIMx_CCMRy register to 1001b:
   – In up-counting mode, the channel is active until a trigger event is detected, the comparison is performed like in PWM mode 2, then the channel becomes inactive again at the next update;
   – In down-counting mode, the channel is inactive until a trigger event is detected, the comparison is performed like in PWM mode 1, then the channel becomes inactive again at the next update.

The figure below shows an example of retriggerable one-pulse mode.

Figure 11. Retriggerable one pulse mode



3 Timer synchronization

3.1 Timers internal connections

The timers are linked together internally for timer synchronization or chaining. When one timer is configured in initiator mode, it can reset, start, stop, or clock the counter of another timer configured in subordinate mode.

The following figures show examples of controller/target timer connections.

Figure 12. Controller/target timer example

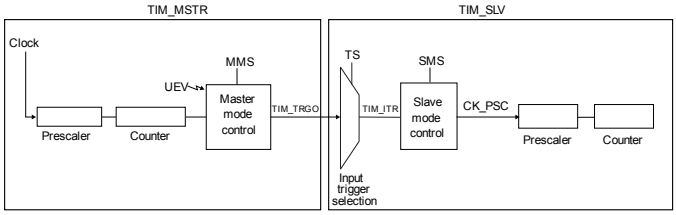
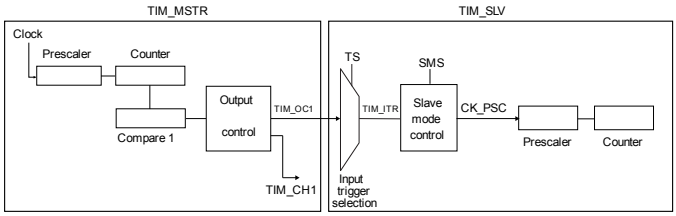


Figure 13. Controller/target connection example with one channel only timer



3.2 Controller configuration

A timer is configured as a controller timer by default (MSM bit in the TIMx_SMCR register set to 0b).

By setting MMS[3:0] in the TIMx_CR2 register, the following options are available:

1. **Reset**: the UG bit from the TIMx_EGR register is used as trigger output TIMx_TRGO.
2. **Enable**: the counter enable signal, CNT_EN, is used as the trigger output TIMx_TRGO. It is useful to start several timers at the same time or to control a window in which a subordinate timer is enabled.
3. **Update**: the UEV is selected as trigger output TIMx_TRGO. For example, a controller timer can be used as a prescaler for a subordinate timer.
4. **Compare pulse**: the trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high) as soon as a capture or a compare match occurs.
5. **Compare**: TIMx_OCyREF signal is used as trigger output TIMx_TRGO.
6. **Encoder clock output**: the encoder clock signal is used as the trigger output TIMx_TRGO.

The advanced control timer can generate two trigger outputs: TIMx_TRGO and TIMx_TRGO2. The source for the TIMx_TRGO2 can be selected by setting MMS2[3:0] in the TIMx_CR2 register.

3.3 Subordinate configuration

Each timer configured as a subordinate (by setting the MSM bit in the TIMx_SMCR register to 1b) can be connected to a timer configured as a controller through one of the inputs triggers ITRx. The proper input trigger can be configured by setting TS[4:0] in the TIMx_SMCR register and the available connections are device-specific.

By setting SMS[3:0] in the TIMx_SMCR register, the following options are available:

1. **Reset**: The rising edge of the selected trigger input TIMx_TRGI reinitializes the counter and generates a UEV.
2. **Gated**: the counter clock is enabled when the selected trigger input TIMx_TRGI is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both counter starts and stops are controlled.
3. **Trigger**: the counter starts at a rising edge of the selected trigger input TIMx_TRGI (but it is not reset). Only the counter start is controlled.
4. **External clock**: rising edges of the selected trigger input TIMx_TRGI clock the counter.
5. **Combined reset + trigger**: rising edge of the selected trigger input TIMx_TRGI reinitializes the counter, generates a UEV and starts the counter.
6. **Combined gate + reset**: the counter clock is enabled when the selected trigger input TIMx_TRGI is high. The counter stops and is reset as soon as the trigger becomes low. Both start and stop of the counter are controlled.

For example, a TIM controller can be configured to act as a prescaler for a TIM subordinate:

1. Configure the TIM controller in controller mode so that it outputs a periodic trigger signal on each update event UEV. By setting MMS[2:0] in the relevant TIM_CR2 register to 010b, a rising edge is output on TIM_TRGO each time an update event is generated
2. To connect the TIM_TRGO of TIM controller to TIM subordinate, TIM subordinate must be configured in subordinate mode using ITR2 as an internal trigger by setting TS[4:0] in the TIM_SMCR register to 00010b
3. Then the subordinate mode controller must be put in external clock mode1 by setting SMS[3:0] to 0111b in the TIM_SMCR reg. With this configuration, the TIM subordinate is clocked by the rising edge of the periodic TIM controller trigger signal (which corresponds to the TIM controller counter overflow)
4. Finally, both timers must be enabled by setting their respective CEN bits in the TIMx_CR1 register

4 Advanced features for electric motor control

4.1 Output signal generation

The timers can output two complementary signals (if present) and manage the dead-time between the outputs.

The complementary signals TIMx_OCy and TIMx_OCyN are activated by a combination of several control bits:

- CCyE and CCyNE bits in the TIMx_CCER register.
- MOE, OSSR, and OSSI bits in the TIMx_BDTR register.
- OISy, OISyN, and TIMx_CR2 registers.

The main output enable (MOE) bit is reset once a break input becomes active. It is set by software or automatically based on the automatic output enable (AOE) bit. When the MOE bit is reset, the OCy and OCyN outputs are disabled (the timer releases the output control, which is taken over by the GPIO logic and which imposes a high-Z state) or forced to idle state defined by OISy (output idle state channel y) and OISyN (output idle state channel yN) bit, depending on whether the OSSI (off-state selection for idle mode) bit is set or not.

When the MOE bit is set, the OSSR (off-state selection for run mode) bit determines the pin output when the channel output is not enabled. When this bit is set, OCx and OCxN outputs are set to their inactive level as soon as their complementary bits CCxE or CCxNE are set. The output is still controlled by the timer.

The OSSI bit is used when MOE is reset due to a break event or by software, on channels configured as outputs. When this bit is set, OCy and OCyN outputs are first forced to their inactive level, then forced to their idle level after the deadtime. The timer maintains its control over the output.

The table below summarizes the possible configurations of the timer.

Table 4. Timer configurations

Control bits					Output state		Typical use
MOE	OSSI	OSSR	OCyE	OCyNE	OCy	OCyN	
1	x	0	0	0	Output disabled	Output disabled	General purpose
	x	0	0	1	Output disabled	OCyREF (polarity)	
	x	0	1	0	OCyREF (polarity)	Output disabled	
	x	0	1	1	OCyREF (polarity) with dead-time	(polarity) with dead-time	Motor control (sine wave)
	x	1	0	0	Output disabled	Output disabled	Motor control (6-steps)
	x	1	0	1	Off-state	OCyREF (polarity)	
	x	1	1	0	OCyREF (polarity)	Off-state	
	x	1	1	1	OCyREF (polarity) with dead-time	(polarity) with dead-time	Motor control (sine wave)
0	0	x	0	0	Output disabled		Outputs disconnected from I/O ports
	0	x	0	1			
	0	x	1	0			
	0	x	1	1			
	1	x	0	0	Off-state (outputs are first forced with their inactive level then forced to their idle level after the deadtime)		All PWMs OFF (low Z for safe stop)
	1	x	0	1			
	1	x	1	0			
	1	x	1	1			

Note: Deadtime insertion is enabled by setting both CCyE and CCyNE bits, and the MOE bit. When only OCyN is enabled (CCyE = 0, CCyNE = 1), it is not complemented and becomes active as soon as OCyREF is high. For example, if CCyNP = 0 then OCyN = OCyREF. On the other hand, when both OCy and OCyN are enabled (CCyE = CCyNE = 1) OCy becomes active when OCyREF is high, whereas OCyN is complemented and becomes active when OCyREF is low.

4.2 Combined three-phase PWM mode

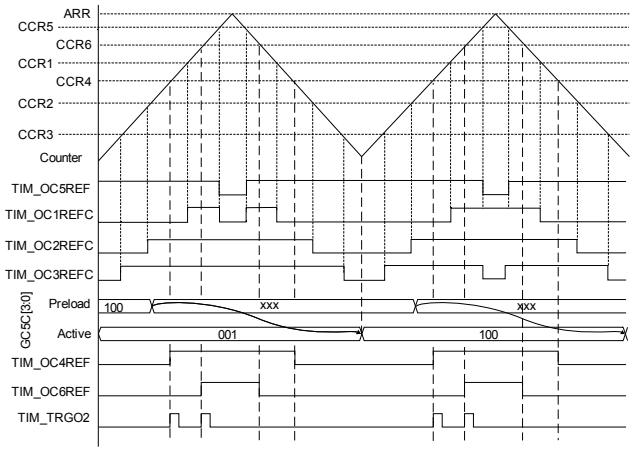
Combined three-phase PWM mode allows one to three center-aligned PWM signals to be generated with a single programmable signal ANDed in the middle of the pulses. The TIM_OC5REF signal is used to define the resulting combined signal. The reference signal to be combined with the TIM_OC5REF is selected by setting GC5C[3:1] in the TIMx_CCR5. A combined 3-phase PWM mode can be selected independently on channels 1 to 3 by setting at least one of the 3-bits GC5C[3:1].

The resulting signal TIM_OCyREFC is the AND logical combination of two reference signals:

- If GC5C1 is set, OC1 output is controlled by TIMx_CCR1 and TIMx_CCR5
- If GC5C2 is set, OC1 output is controlled by TIMx_CCR2 and TIMx_CCR5
- If GC5C3 is set, OC1 output is controlled by TIMx_CCR3 and TIMx_CCR5

The following figure shows 3-phase combined PWM signals with multiple trigger pulses per period.

Figure 14. 3-phase combined PWM signals with multiple trigger pulses per period



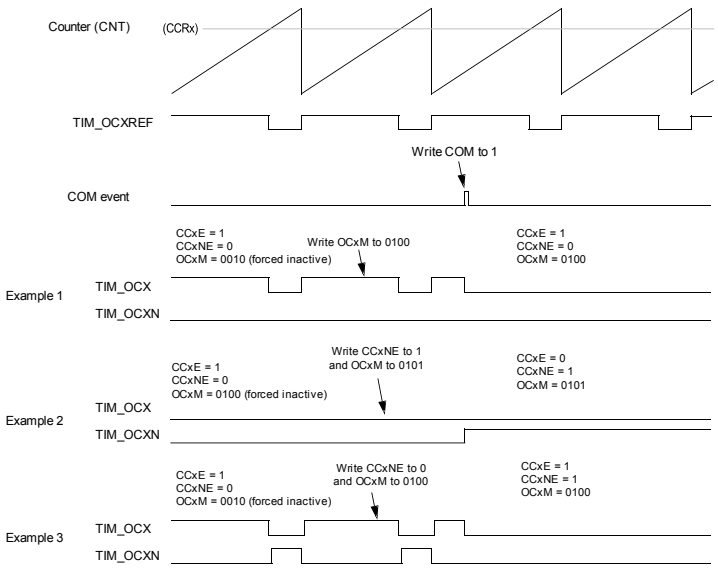
4.3 Six-step PWM mode

When complementary outputs are used on a channel, preload bits are available on the OCyM, CCyE, and CCyNE bits. The preload bits are transferred to the shadow bits at the so-called COM event. As a result, the configuration for the next step can be programmed in advance and applied to all the channels at the same time.

A COM event can be generated by software by setting the COM bit in the TIMx_EGR register or by hardware (on TIMx_TRGI rising edge). A flag is set when the COM event occurs (COMIF bit in the TIMx_SR register), which can generate an interrupt (if the COMIE bit is set in the TIMx_DIER register) or a DMA request (if the COMDE bit is set in the TIMx_DIER register).

The following figure describes the behavior of the TIMx_OCy and TIMx_OCyN outputs when a COM event occurs, in three different examples of programmed configurations.

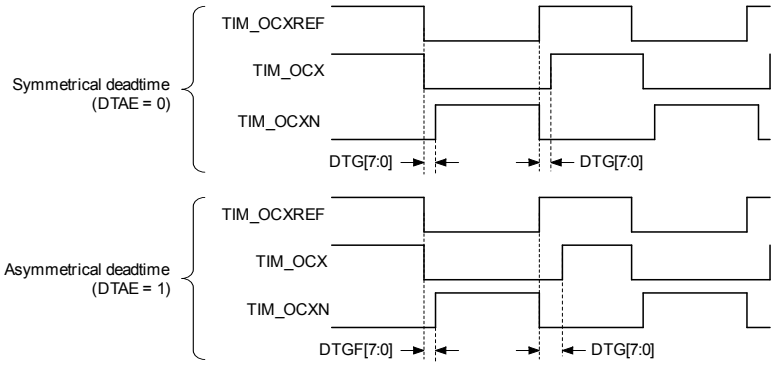
Figure 15. **6-step generation, COM example (OSSR = 1)**



4.4 Complementary outputs and dead-time insertion

To use the complementary signal for one channel, set the two outputs to compare the enable bits of this channel and its complementary (OCyE and OCyNE) channel. If the dead-time bits are not zero, the two signals are generated with the insertion of a dead-time as illustrated in the following figure.

Figure 16. **Asymmetrical deadtime**



The dead-time DT is computed by setting DTG[7:0] in the TIMx_BDTR register as follows:

- If DTG[7:5] = 0xxb, DT = DTG[7:0] x t_{dtg} with t_{dtg} = t_{DTS}
- If DTG[7:5] = 10xb, DT = (64 + DTG[5:0]) x t_{dtg} with t_{dtg} =2 x t_{DTS}
- If DTG[7:5] = 110b, DT = (32 + DTG[4:0]) x t_{dtg} with t_{dtg} =8 x t_{DTS}
- If DTG[7:5] = 111b, DT = (32 + DTG[4:0]) x t_{dtg} with t_{dtg} =16 x t_{DTS}

Where t_{DTS} is the sampling clock configured by setting CKD[1:0] in the TIMx_CR1 register.

For example, if t_{DTS} = 3.26 ns (306.7 MHz), the dead-time possible values are:

- 0 to 414.02 ns by 3.26 ns steps
- 417.28 ns to 828.04 ns by 6.52 ns steps
- 834.56 ns to 1643.04 ns by 26.08 ns steps

- 1669.12 ns to 3286.08 ns by 13.04 ns steps

The DTAE bit in the TIMx_DTR2 allows to differentiate the dead-time values for a rising and falling edges of the reference signal. In asymmetrical mode (DTAE = 1b), the rising edge-referred dead-time is defined by DTG[7:0] in the TIMx_BDTR register, while the falling edge-referred is defined by DTGF[7:0] in the TIMx_DTR2 register. The DTAE bit must be written before enabling the counter and must not be modified while the counter is enabled (CEN = 1b). It is possible to have the dead-time value updated on-the-fly during PWM operation, using a preload mechanism. DTG[7:0] and DTGF[7:0] are preloaded when the DTPE bit in the TIMX_DTR2 register is set. The preload value is loaded in the active register on the UEV.

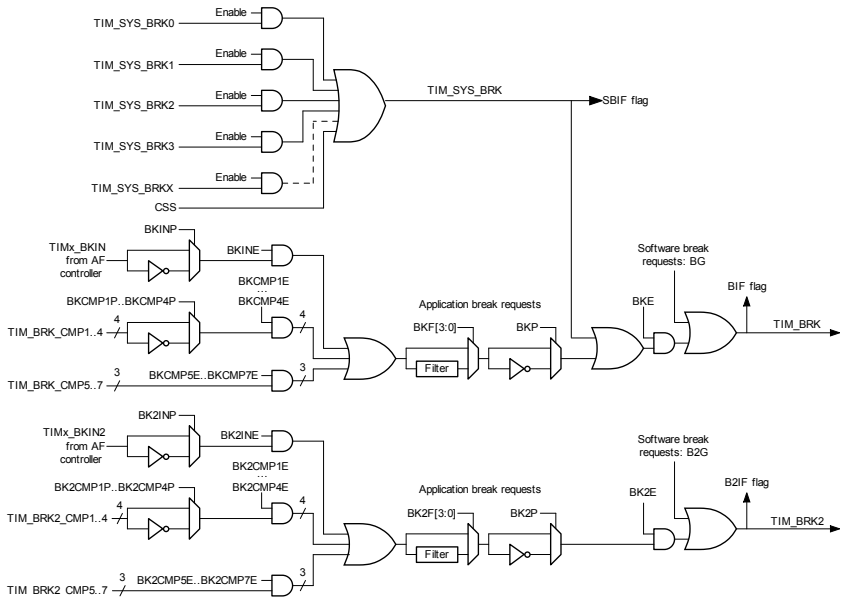
4.5 Break input

The purpose of the break function is to protect power switches driven by PWM signals generated with the timers.

The break function generally has two channels, BRK and BRK2, and the break inputs are usually connected to fault outputs of power stages and 3-phase inverters. If activated, the break function circuitry shuts down the PWM outputs and forces them to a predefined safe state. Internal events can also be selected to trigger a break event.

The following shows the break function circuitry.

Figure 17. Break and Break 2 circuitry overview



The break function is configurable depending on several control bits:

- The MOE bit in the TIMx_BDTR register allows to enable the timer output and it is reset in case of a break event.
- The OSSI bit in the TIMx_BDTR register defines whether the timer controls the output in inactive state or releases the control to the GPIO controller (typically to have it in high-Z mode).
- The OISx and OISxN bits in the TIMx_CR2 register, which are setting the output shut-down level, either active or inactive. The TIMx_OCy and TIMx_OCyN outputs cannot be set both to active level at a given time, whatever the OISx and OISxN values.

The break event is generated by the selected source (for example, by setting the relevant BKINE bit in the TIMx_AF1 register for the TIMx_BKIN input connected to the TIM1x_BKIN external pin) and the break function is enabled by setting the BKE and/or BK2E bits in the TIMx_BDTR register. The user has to configure also the polarity (BKP and/or BKP2 bits in the TIMx_BDTR Register) and, if needed, the programmable filter (BKF[3:0] and/or BK2F[3:0] in the TIMx_BDTR Register) to avoid spurious events. The break event can also be generated by software by setting the BG and/or B2G bits in the TIMx_EGR register.

Moreover, bidirectional mode is available for both break input and requires the I/O to be configured in open-drain mode with active low polarity (using BKINP, BKP, BK2INP and BK2P bits). Any break request coming either from the system, from on-chip peripherals, or from break inputs forces a low level on the break input to signal the fault event. The bidirectional mode is inhibited if the polarity bits are not correctly set (active high polarity), for safety purposes. A safe disarming mechanism prevents the system from being definitively locked-up (a low level on the break input triggers a break, which enforces a low level on the same input).

In addition to the break input and the output management, write protection has been implemented inside the break circuit to safeguard the application. It allows to freeze the configuration of several parameters (dead-time duration, TIMx_OCy/ TIMx_OCyN polarities and state when disabled, OCyM configurations, break enable and polarity). There are three locking levels illustrated in the following table.

Table 5. Locking levels

	LOCK Level 1 [01b]	LOCK Level 2 [10b]	LOCK Level 3 [11b]
Register	**Bits**	**Bits**	**Bits**
TIMx_CR2	OISy / OISyN	OISy / OISyN	OISy / OISyN
TIMx_BDTR	AOE	AOE	AOE
	DTG[7:0]	DTG[7:0]	DTG[7:0]
	BKE / BK2E / BKP / BK2P	BKE / BK2E / BKP / BK2P	BKE / BK2E / BKP / BK2P
	BKBID / BK2BID	BKBID / BK2BID	BKBID / BK2BID
	BKF[3:0] / BK2F[3:0]	BKF[3:0] / BK2F[3:0]	BKF[3:0] / BK2F[3:0]
	-	OSSR / OSSI	OSSR / OSSI
TIMx_CCER	-	CCyP / CCyNP	CCyP / CCyNP
TIMx_CCMRy	-	-	OCyM / OCyPE

Note: The LOCK bit can only be written once after the reset and its content is frozen until the next reset.

4.6 Specific features for feedback measurement

4.6.1 Incremental quadrature encoder

The incremental quadrature encoder is a type of position sensor used in electric motor control applications to measure the angular position and the rotation direction.

In general, the incremental quadrature encoder generates three signals: phase A, phase B and index.

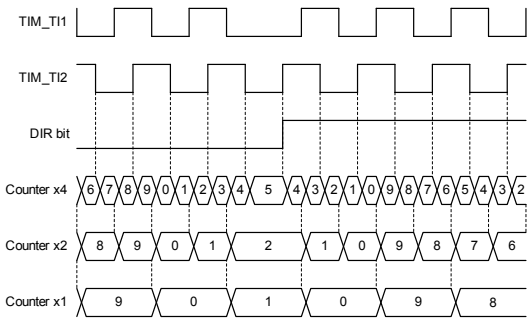
Phase A and phase B are square waves in quadrature (there is a 90 degree phase difference), the direction of the motor depends on whether phase A leads phase B, or phase B leads phase A, and the third channel index pulse occurs once per revolution and is used as a reference to measure an absolute position.

Quadrature encoder

The TIMx_TI1 and TIMx_TI2 external pins are used to interface a quadrature encoder. The timer counter is clocked by each valid transition on these signals. The sequence of transitions of the two inputs is evaluated and generates count pulses as well as the direction signal. Depending on the sequence the counter counts up or down, the DIR bit in the TIMx_CR1 register is accordingly hardware modified. Encoder interface mode basically acts as an external clock with direction selection and the counter counts continuously between 0 and the auto reload value stored in the TIMx_ARR register (from 0 up to ARR value or from ARR value down to zero depending on the direction). The TIMx_ARR register must be configured before starting.

The following figure shows the timer counter value during a speed reversal, for various counting modes.

Figure 18. **Quadrature encoder counting modes**



For example, for counter mode x4, the timer counter is incremented or decremented for each transition on the selected input TI1 or TI2. To configure a timer in this encoder mode:

- Select the proper source TIMx_TI1 and TIMx_TI2 external pin by setting TI1SEL[3:0] and TI2SEL[3:0] to 0000b in the TIMx_TISEL register.
- Program the needed input filter duration by setting IC1F[3:0] and IC2F[3:0] in the TIMx_CCMR1.
- Configure the CC1 channel as input on TIM_TI1 by setting CC1S[1:0] in the TIM_CCMR1 register to 01b and the CC2 channel as input on TIM_TI2 by setting CC2S[1:0] in the TIM_CCMR2 register to 01b.
- Select the active polarity for TIM_TI1_FP1 and TIM_TI2_FP2 by setting CC1P, CC1NP, CC2P, and CC2NP to 0b in the TIMx_CCER register (TIM_TI1_FP1 noninverted equal to TIM_TI1 and TIM_TI2_FP2 noninverted equal to TIM_TI2).
- Select the quadrature encoder mode 3 (x4 mode - counter counts up/down on both TIM_TI1FP1 and TIM_TI2FP2 edges depending on the level of the other input) by setting SMS[3:0] to 0011b in the register TIMx_SMCR.
- Enable the timer counter by setting the CEN bit to 1b in TIMx_CR1 register.

The table below shows the counting direction versus encoder signals for different modes.

Table 6. **Counting direction versus encoder signals**

Active edge	SMS[3:0]	Level on the other signal (TIMx_TI1FP1 for TIMx_TI2 and TIMx_TI2FP2 for TIMx_TI2)	TIMx_TI1FP1 signal		TIMx_TI2FP2 signal	
			Rising	Falling	Rising	Falling
x1 mode on TIMx_TI1 only	1110b	High	Down	Up	No count	No count
		Low	No count	No count	No count	No count
x1 mode on TIMx_TI2 only	1111b	High	No count	No count	Up	Down
		Low	No count	No count	No count	No count
x2 mode on TIMx_TI1 only	0001b	High	Down	Up	No count	No count
		Low	Up	Down	No count	No count
x2 mode on TIMx_TI2 only	0010b	High	No count	No count	Up	Down
		Low	No count	No count	Down	Up
x4 mode on both TIMx_TI1 and TIMx_TI2	0011b	High	Down	Up	Up	Down
		Low	Up	Down	Down	Up

Index management

The third encoder output, the index, indicating the mechanical zero position, can be connected to the TIMx_ETR external pin to trigger a timer counter reset.

It can be filtered using the digital input filter and the index management functionality is enabled by setting the IE bit in the TIMX_ECR register. The IE bit must be set only in encoder mode.

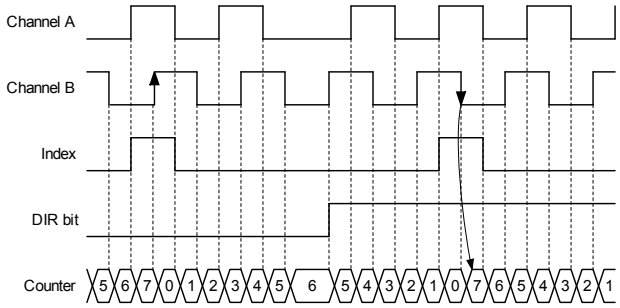
Commercially, available encoders are proposed with several options for index pulse conditioning:

- Gated with A and B, the pulse width is 1/4 of one channel period, aligned with both A and B edges.
- Gated with A (B), the pulse width is 1/2 of one channel period, aligned with the two edges of channel A (B).
- Ungated, the pulse width is up to one channel period, without any alignment to the edges.

The timer supports the three gating options identically, without any specific programming needed, but on which encoder state (that are channel A and channel B state combination) the index must be synchronized must be defined by setting IPOS[1:0] in the TIMx_ECR register.

For example, the following figure shows the counter with index gated on channel A.

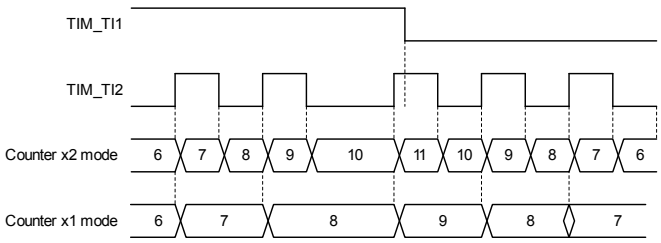
Figure 19. Counter reading with index gated on channel A (IPOS[1:0] = 11b)



Clock plus direction

In this mode shown in the following figure, the clock is provided on a single line, on TIMx_TI2 input, while the direction is forced using the TIMx_TI1 input. The figure below shows the clock provided on a single line TIMx_TI2, while the direction is forced using the TIMx_TI1 input.

Figure 20. Direction plus clock encoder mode



This mode is enabled by setting SMS[3:0] in the TIMx_SMCR register:

- 1011b for x1 mode, the counter is updated on a single clock edge, selected by setting CC2P.
- 1010b for x2 mode, the counter is updated on both rising and falling edges of the clock.

The polarity of the direction signal on TIMx_TI1 input is selected by setting CC1P.

Directional clock

In this mode, the clocks are provided on two lines, with a single one at once, depending on the direction, so as to have one up-counting clock line and one down-counting clock line.

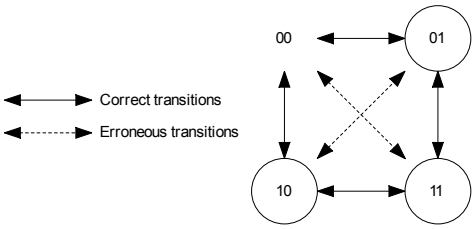
This mode is selected by setting SMS[3:0] in the TIMx_SMCR register while the clock idle state and the counting edge are selected by setting CC1P and CC2P in the TIMx_CCER register, as follows:

- 1100b for x2 mode, the counter is updated on both rising and falling edges of any of the two clock lines.
- 1101b for x1 mode, the counter is updated on a single clock edge.

Error management

For encoder configurations where two quadrature signals are available, it is possible to detect transition errors. The reading on the two inputs corresponds to a 2-bits gray code, which can be represented as a state diagram, in the following figure. A single bit is expected to change at once. An erroneous transition sets the TERRF error flag in the TIMx_SR register and an error interrupt is generated if the TERRIE bit is set in the TIMx_DIER register.

Figure 21. State diagram for quadrature encoded signals



4.6.1.1 *Hall sensors*

A Hall effect sensor (or simply Hall sensor) is a type of sensor, which detects the presence and magnitude of a magnetic field using the Hall effect. The output voltage of a Hall sensor is directly proportional to the strength of the field. Frequently, the Hall sensors are combined with threshold detection to act as a binary switch and are used for proximity sensing, positioning, speed detection, and current sensing.

They are used with three-phases BLDC electric motors to detect the position of the rotor.

Timer input XOR function

The timers can interface three Hall sensors using the TIMx_TI1, TIMx_TI2 and TIMx_TI3 external pins, connected through an XOR function to the TIMx_TI1 input by setting the TI1S bit in the TIMx_CR2 register to 1b.

The subordinate mode controller is configured in reset mode and the subordinate input is TIMx_TI1F_ED. Thus, each time one of the three inputs toggles, the counter restarts counting from zero creating a time base triggered by any change on each Hall sensor output. The timer channel 1 is configured in capture mode and the capture signal is TIMx_TRC. The captured value, which corresponds to the time elapsed between two changes on the inputs, can be used to calculate the motor speed. The TIMx_ARR register must be set to the max value because the timer counter must be cleared by any TIMx_TI1 input change and select the prescaler to have a period longer than the maximum time between two consecutive input changes.

5 Advanced features for power conversion

The HRTIM (High-Resolution TIMer) is designed specifically to control power conversion systems in lighting systems switch mode power supply. Even though it really excels in this role, it can of course be used in other applications with high-resolution timer requirements.

The HRTIM features up to 12 outputs which can be configured in various coupled and autonomous modes using five timing units tied to a common master for synchronization purposes. The synchronization with other timers is also facilitated. The HRTIM is strongly tied to ADCs and fault inputs for feedback purposes.

Application related information can be found in the following documents:

“SR5 E1 line: getting started with the HRTIM” (TN1439) (see Section 6.1 Reference documents).

“SR5 E1 line of Stellar electrification MCUs” (RM0483) (see Section 6.1 Reference documents).

6 Other information

6.1 Reference documents

Table 7. **Reference documents**

Doc name	Title
DS13808	SR5 E1 line of Stellar electrification MCUs — 32-bit Arm® Cortex®-M7 automotive MCU 2x cores, 300 MHz, 2 MB flash, rich analog, 104 ps 24 ch high-resolution timer, HSM, ASIL-D
RM0483	SR5E1x 32-bit Arm® Cortex®-M7 architecture microcontroller for electrical vehicle applications
TN1439	SR5 E1 line: getting started with the HRTIM

Revision history

Table 8. **Document revision history**

Date	Version	Changes
07-Nov-2023	1	Initial release.

Contents

List of tables

List of figures