

29.4.3 Базовая единица времени

Основным блоком программируемого таймера является 16-битный/32-битный счетчик с соответствующим регистром автоперезагрузки. Счетчик может считать вверх, вниз или одновременно вверх и вниз. Часы счетчика могут быть разделены предварительным делителем.

Счетчик, регистр автоматической перезагрузки и регистр предварительного делителя могут быть записаны или прочитаны программным обеспечением. Это верно даже при работающем счетчике.

Базовая единица времени включает в себя:

- Регистр счетчика (TIMx_CNT)
- Регистр предделителя (TIMx_PSC):
- Регистр автоматической перезагрузки (TIMx_ARR)

Регистр автоматической перезагрузки предварительно загружен. Запись или чтение из регистра автоматической перезагрузки обеспечивает доступ к регистру предварительной загрузки. Содержимое регистра предварительной загрузки передается в теневой регистр постоянно или при каждом событии обновления (UEV), в зависимости от бита разрешения предварительной загрузки автоматической перезагрузки (ARPE) в регистре TIMx_CR1. Событие обновления отправляется, когда счетчик достигает переполнения (или потери значимости при обратном счете) и если бит UDIS равен 0 в регистре TIMx_CR1. Он также может быть сгенерирован программным обеспечением. Генерация события обновления подробно описана для каждой конфигурации.

Счетчик тактируется выходом предварительного делителя tim_cnt_ck, который включается только тогда, когда бит разрешения счетчика (CEN) в регистре TIMx_CR1 установлен (см. также описание контроллера ведомого режима, чтобы получить более подробную информацию о включении счетчика).

Обратите внимание, что фактический сигнал включения счетчика CNT_EN устанавливается через 1 такт после CEN.

29.4.10 Режим сравнения выхода

Эта функция используется для управления формой выходного сигнала или индикации истечения периода времени.

Когда найдено совпадение между регистром захвата/сравнения и счетчиком, функция сравнения вывода:

- Назначает соответствующему выходному контакту программируемое значение, определяемое режимом сравнения выходов (биты OCxM в регистре TIMx_CCMRx) и полярностью выхода (бит CCxP в регистре TIMx_CCER). Выходной контакт может сохранять свой уровень (OCxM=000), быть установлен активным (OCxM=001), быть установлен неактивным (OCxM=010) или может переключаться (OCxM=011) при совпадении.

- Устанавливает флаг в регистре состояния прерывания (бит CCxIF в регистре TIMx_SR).

- Генерирует прерывание, если установлена соответствующая маска прерывания (бит CCxIE в регистре TIMx_DIER).

- Отправляет запрос DMA, если установлен соответствующий бит разрешения (бит CCxDE в регистре TIMx_DIER, бит CCDS в регистре TIMx_CR2 для выбора запроса DMA).

Регистры TIMx_CCRx можно запрограммировать с регистрами предварительной загрузки или без них, используя бит OSxPE в регистре TIMx_CCMRx.

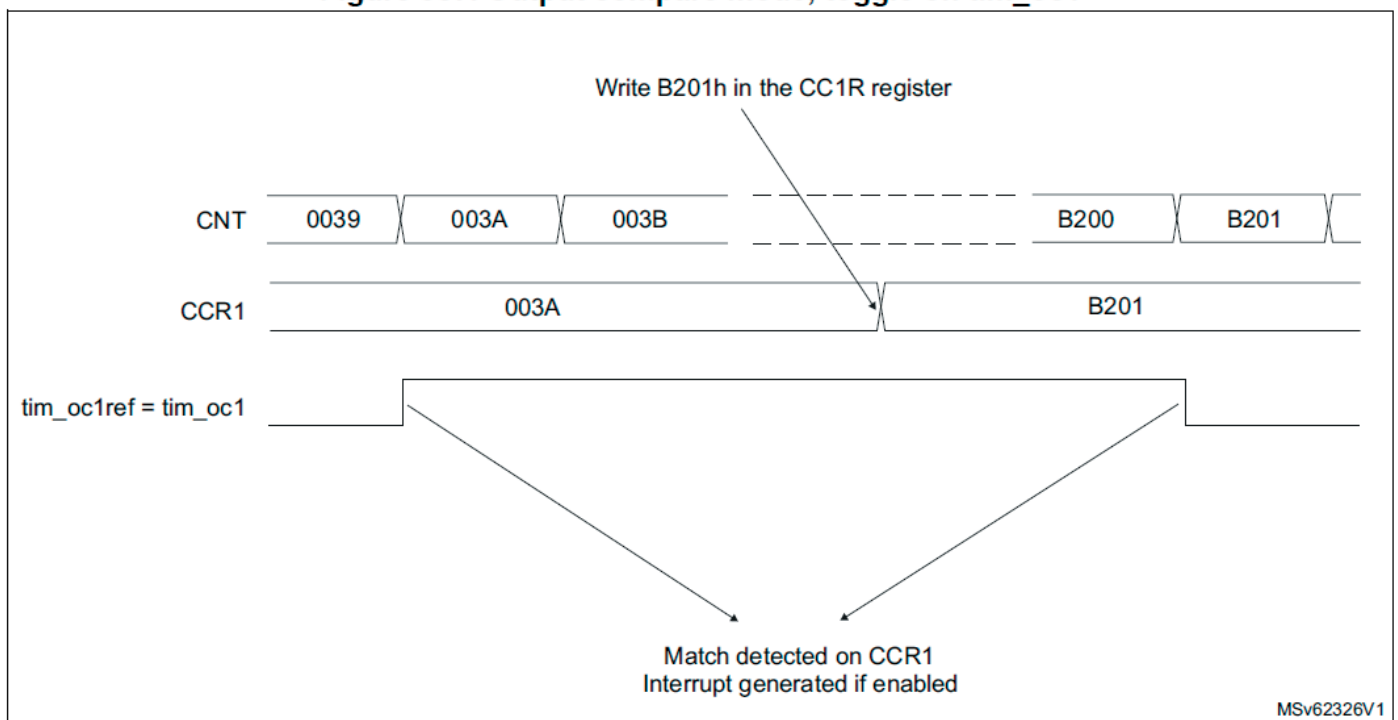
В режиме сравнения вывод события обновления UEV не влияет на вывод tim_oscxref и tim_oscx. Разрешение по времени составляет один отсчет счетчика. Режим сравнения вывода также можно использовать для вывода одиночного импульса (в одноимпульсном режиме).

Процедура

1. Выберите часы счетчика (внутренние, внешние, предварительный делитель).
2. Запишите нужные данные в регистры TIMx_ARR и TIMx_CCRx.
3. Установите биты CSxIE и/или CSxDE, если необходимо сгенерировать прерывание и/или запрос DMA.
4. Выберите режим вывода. Например:
 - a) Напишите OSxM = 0011, чтобы переключить выходной контакт tim_oscx, когда CNT соответствует CCRx
 - b) Напишите OSxPE = 0, чтобы отключить предварительную загрузку регистра
 - c) Запишите CSxP = 0, чтобы выбрать активную высокую полярность.
 - d) Напишите CSxE = 1, чтобы включить вывод
5. Включите счетчик, установив бит CEN в регистре TIMx_CR1.

Регистр TIMx_CCRx может быть обновлен в любое время программным обеспечением для управления формой выходного сигнала, при условии, что регистр предварительной загрузки не включен (OSxPE=0, в противном случае теневой регистр TIMx_CCRx обновляется только при следующем событии обновления UEV). Пример приведен на рисунке 387.

Figure 387. Output compare mode, toggle on tim_oc1



** Совпадение обнаружено на CCR1. Генерируется прерывание, если включено*

29.4.11 ШИМ-режим

Режим широтно-импульсной модуляции позволяет генерировать сигнал с частотой, определяемой значением регистра TIMx_ARR, и скважностью, определяемой значением регистра TIMx_CCRx.

Режим ШИМ можно выбрать независимо для каждого канала (один ШИМ на каждый выход tim_osc), записав 110 (режим ШИМ 1) или '111 (режим ШИМ 2) в битах ОСхМ в регистре TIMx_CCMRx. Соответствующий регистр предварительной загрузки необходимо включить, установив бит ОСхРЕ в регистре TIMx_CCMRx, и, в конечном итоге, регистр предварительной загрузки с автоматической перезагрузкой (в режимах восходящего счета или выравнивания по центру), установив бит ARPE в регистре TIMx_CR1.

Поскольку регистры предварительной загрузки передаются в теньевые регистры только при возникновении события обновления, перед запуском счетчика все регистры должны быть инициализированы установкой бита UG в регистре TIMx_EGR.

Полярность tim_osc программируется с помощью бита ССхР в регистре TIMx_CCER. Его можно запрограммировать как активный высокий или активный низкий. Выход tim_osc включается битом ССхЕ в регистре TIMx_CCER. Дополнительные сведения см. в описании регистра TIMx_CCERx.

В режиме PWM (1 или 2) TIMx_CNT и TIMx_CCRx всегда сравниваются, чтобы определить, является ли $TIMx_CCRx \leq TIMx_CNT$ или $TIMx_CNT \leq TIMx_CCRx$ (в зависимости от направления счетчика). tim_ocref_clr может быть очищен внешним событием через сигналы tim_etr_in или tim_oceref_clr. В этом случае утверждается только сигнал tim_ocref_clr:

- После события сравнения совпадений
- Когда выходной режим сравнения (биты ОСхМ в регистре TIMx_CCMRx) переключается с «замороженной» конфигурации (без сравнения, ОСхМ='000) на один из режимов ШИМ (ОСхМ='110 или '111). Это вызывает ШИМ с помощью программного обеспечения во время работы таймера.

Таймер может генерировать ШИМ в режиме с выравниванием по краю или с выравниванием по центру в зависимости от битов CMS в регистре TIMx_CR1.

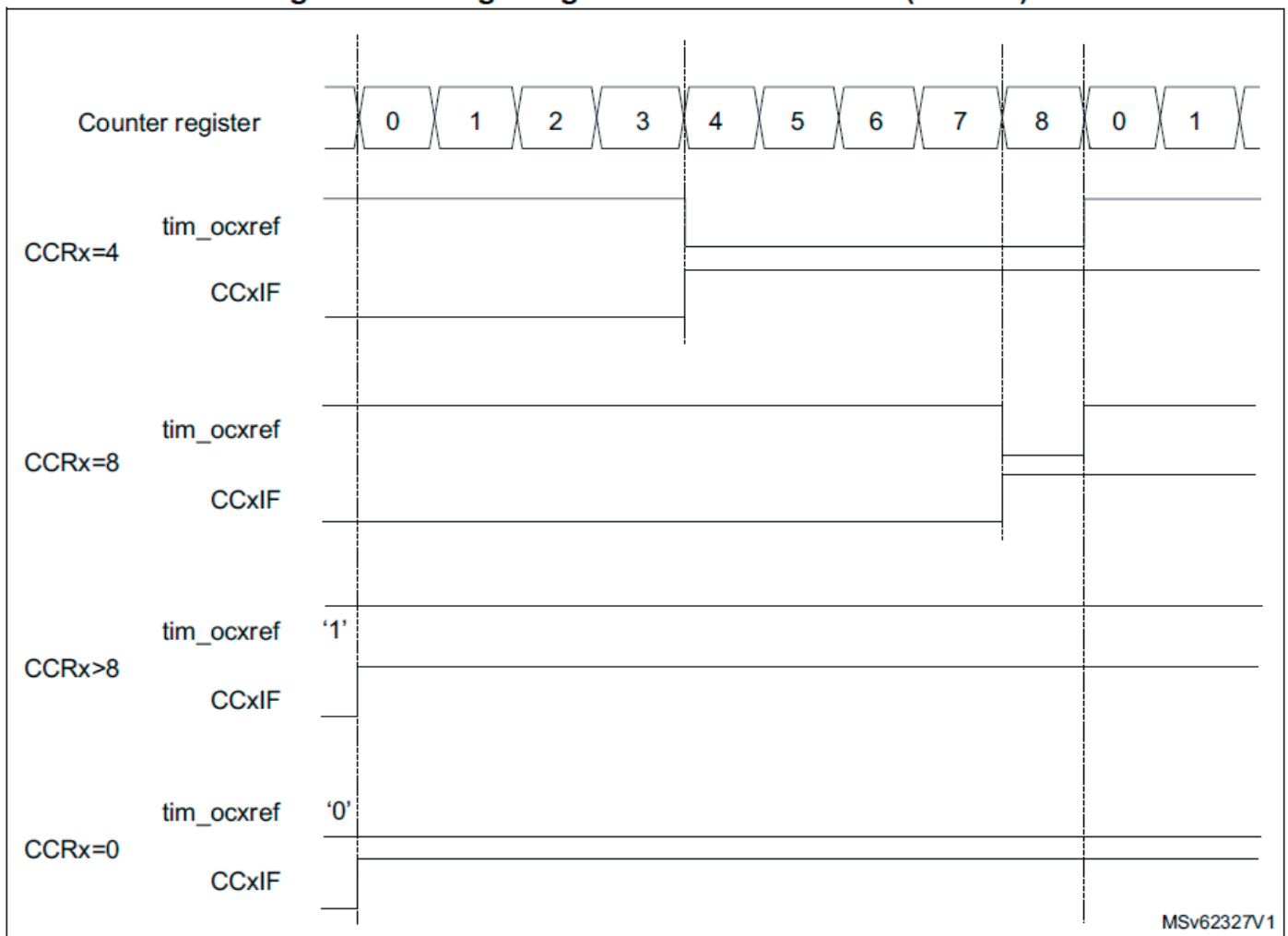
Режим ШИМ с выравниванием по фронту

Конфигурация прямого подсчета

Счет вверх активен, когда бит DIR в регистре TIMx_CR1 низкий. См. раздел «Режим прямого счета» на стр. 1230.

В следующем примере мы рассматриваем режим ШИМ 1. Опорный сигнал ШИМ tim_oscref имеет высокий уровень, пока $TIMx_CNT < TIMx_CCRx$, в противном случае он становится низким. Если значение сравнения в TIMx_CCRx больше, чем значение автоматической перезагрузки (в TIMx_ARR), тогда tim_oscref сохраняется равным «1». Если значение сравнения равно «0», то tim_oscref сохраняется равным «0». На рис. 388 показаны некоторые выровненные по фронту сигналы ШИМ в примере, где $TIMx_ARR=8$.

Figure 388. Edge-aligned PWM waveforms (ARR=8)



MSv62327V1

Конфигурация обратного отсчета

Понижающий счет активен, когда бит DIR в регистре TIMx_CR1 имеет высокий уровень. См. раздел Режим обратного отсчета на стр. 1234.

В режиме ШИМ 1 опорный сигнал tim_ocrref имеет низкий уровень, пока TIMx_CNT > TIMx_CCRx, в противном случае он становится высоким. Если значение сравнения в TIMx_CCRx больше, чем значение автоматической перезагрузки в TIMx_ARR, то tim_ocrref сохраняется на уровне 100%. ШИМ в этом режиме невозможен.

Режим ШИМ с выравниванием по центру

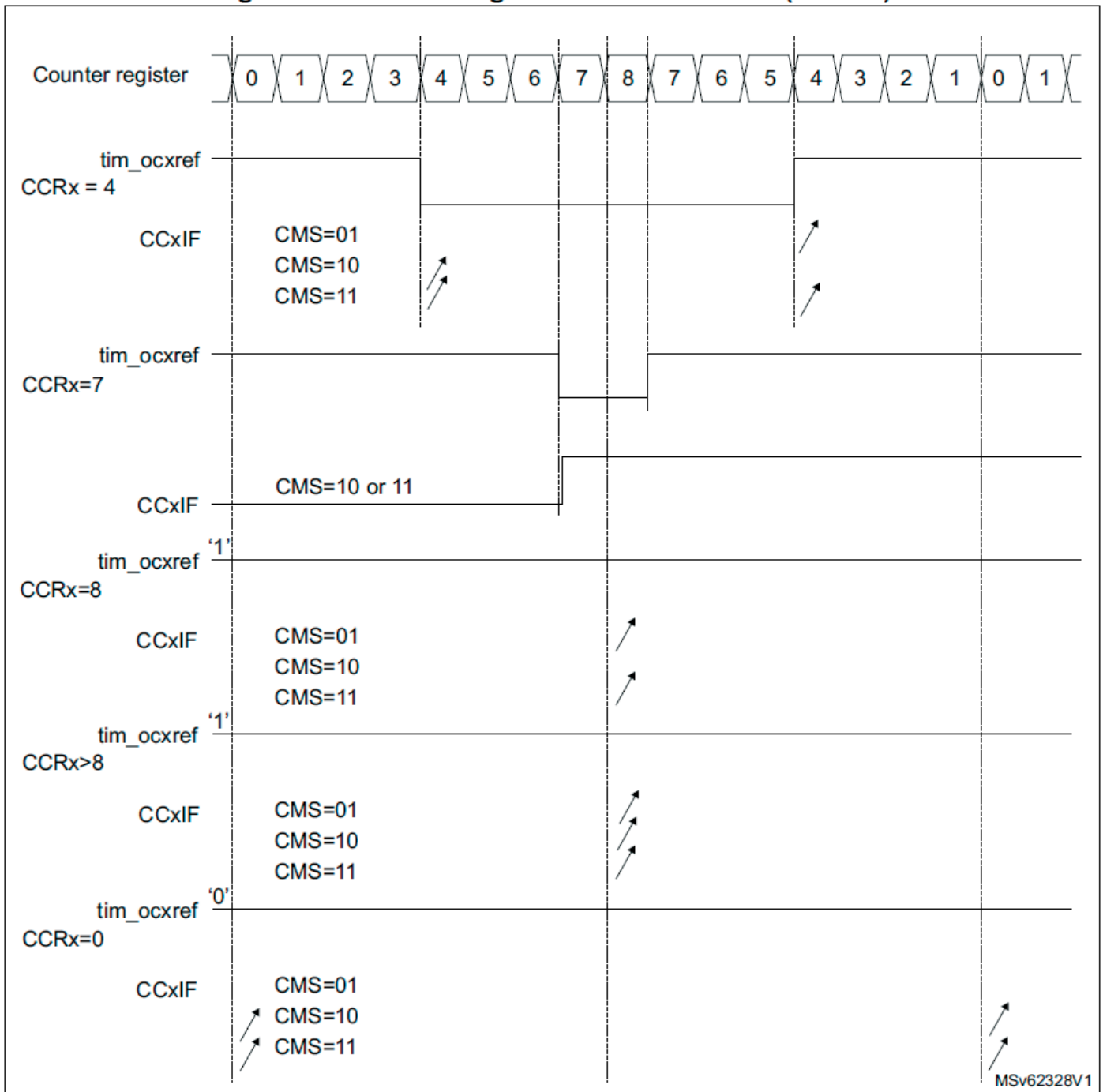
Режим с выравниванием по центру активен, когда биты CMS в регистре TIMx_CR1 отличны от '00 (все остальные конфигурации имеют одинаковый эффект на сигналы tim_ocrref/tim_ocrx). Флаг сравнения устанавливается, когда счетчик считает вверх, когда он считает вниз, или и то, и другое, когда он считает вверх и вниз, в зависимости от конфигурации битов CMS. Бит направления (DIR) в регистре TIMx_CR1 обновляется аппаратно и не должен изменяться программно. См. раздел Режим с выравниванием по центру (счет вверх/вниз) на стр. 1237.

На рис. 389 показаны некоторые выровненные по центру сигналы ШИМ в примере, где:

- TIMx_ARR=8,
- Режим ШИМ — это режим ШИМ 1,

- Флаг устанавливается, когда счетчик ведет обратный отсчет в соответствии с режимом выравнивания по центру 1, выбранным для CMS=01 в регистре TIMx_CR1.

Figure 389. Center-aligned PWM waveforms (ARR=8)



Советы по использованию режима с выравниванием по центру:

- При запуске в режиме с выравниванием по центру используется текущая конфигурация вверх-вниз. Это означает, что счетчик считает вверх или вниз в зависимости от значения, записанного в бит DIR в регистре TIMx_CR1. Кроме того, программное обеспечение не должно одновременно изменять биты DIR и CMS.

- Запись в счетчик при работе в режиме выравнивания по центру не рекомендуется, так как это может привести к неожиданным результатам. В частности:

- Направление не обновляется, если в счетчике записано значение, превышающее значение автоперезагрузки (TIMx_CNT > TIMx_ARR). Например, если счетчик считал вверх, он продолжает считать вверх.

– Направление обновляется, если в счетчик записано 0 или значение TIMx_ARR, но не генерируется событие обновления UEV.

- Самый безопасный способ использования режима с выравниванием по центру — генерировать обновление с помощью программного обеспечения (устанавливая бит UG в регистре TIMx_EGR) непосредственно перед запуском счетчика и не записывать счетчик во время его работы.

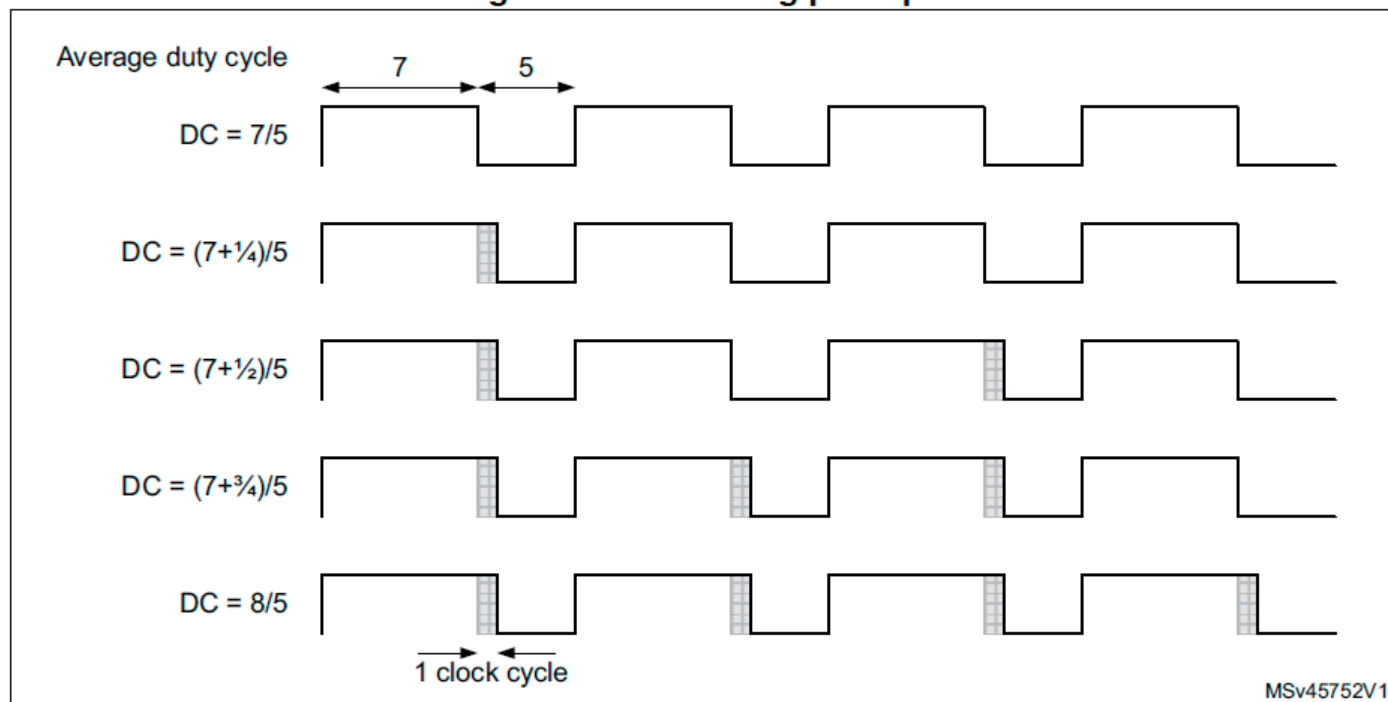
Режим дизеринга

Эффективное разрешение режима ШИМ можно увеличить, включив режим дизеринга с помощью бита DITHEREN в регистре TIMx_CR1. Это относится как к CCR (для увеличения разрешения рабочего цикла), так и к ARR (для увеличения разрешения частоты PWM).

Принцип работы заключается в незначительном изменении фактического значения CCR (или ARR) (добавляя или не добавляя один период тактового сигнала таймера) в течение 16 последовательных периодов ШИМ с predetermined шаблонами.

Это позволяет увеличить разрешение в 16 раз, учитывая средний коэффициент заполнения или период ШИМ. На рисунке 390 ниже представлен принцип дизеринга, примененный к 4 последовательным циклам ШИМ.

Figure 390. Dithering principle



Когда включен режим дизеринга, кодировка регистра изменяется следующим образом (см., например, Рисунок 391):

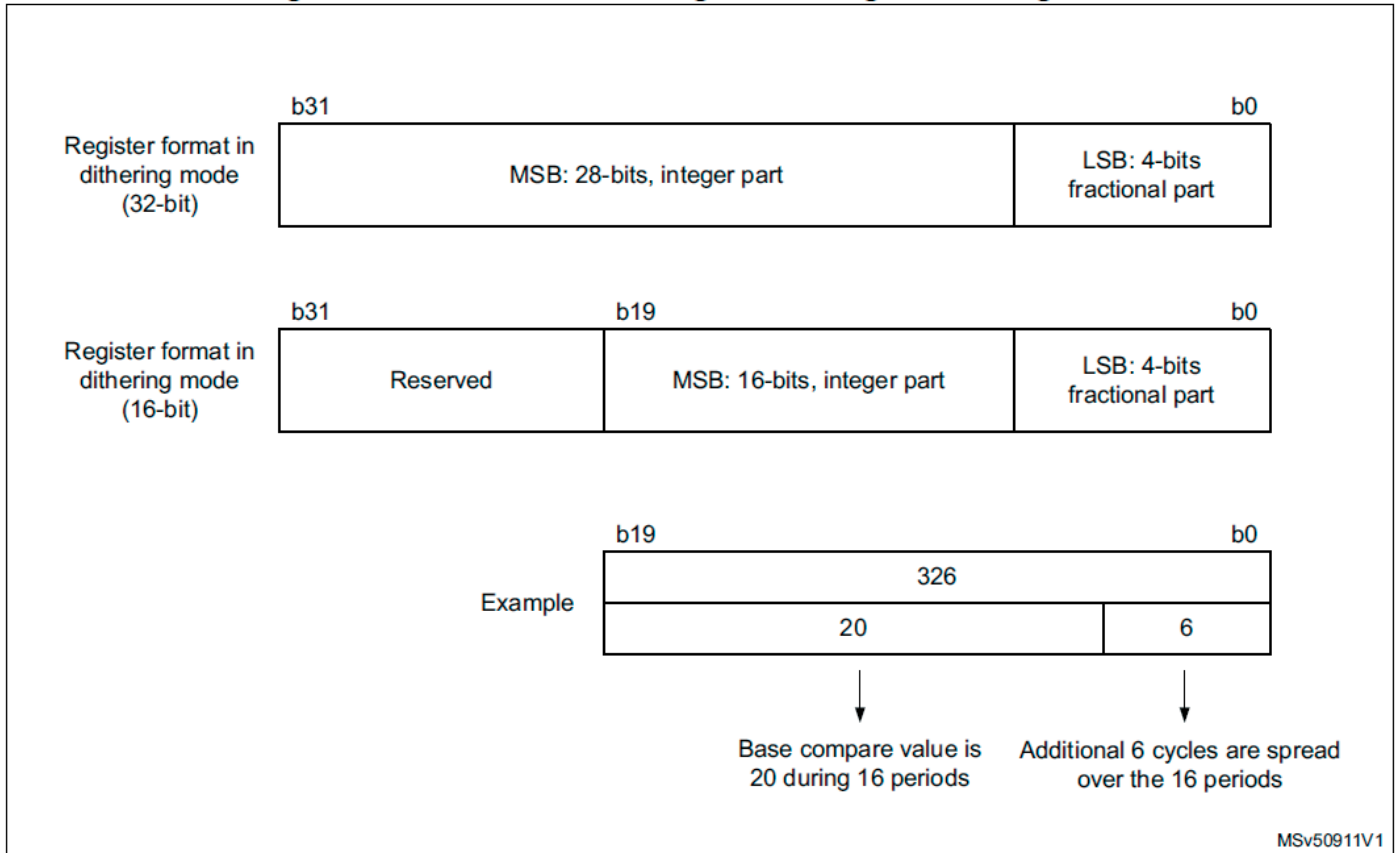
- 4 LSB кодируют часть с повышенным разрешением (дробную часть).
- Старшие биты сдвинуты влево на 4 позиции и кодируют базовое значение. В 16-битном режиме сохраняется 16-битный формат.

Примечание. Значения ARR и CCR будут обновляться автоматически, если бит DITHEREN установлен/сброшен (например, если ARR=0x05 с DITHEREN=0, оно будет обновлено до ARR=0x50 с DITHEREN=1).

При сбросе бита DITHEREN необходимо соблюдать следующую последовательность:

1. Биты CEN и ARPE должны быть сброшены
2. Биты ARR[3:0] должны быть сброшены.
3. Бит DITHEN должен быть сброшен.
4. Флаги CCIF должны быть очищены
5. Можно установить бит CEN (возможно, при ARPE = 1)

Figure 391. Data format and register coding in dithering mode



Минимальная частота определяется по следующей формуле:

$$\text{Resolution} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

$$\text{Dithering mode disabled: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

$$\text{Dithering mode (16-bit timer): } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

$$\text{Dithering mode (32-bit timer): } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{268435454 + \frac{15}{16}}$$

Примечание. Для 16-битных таймеров максимальные значения TIMx_ARR и TIMx_CCRy ограничены 0xFFFFF в режиме сглаживания (соответствует 65 534 для целочисленной части и 15 для сглаженной части).

Для 32-битных таймеров максимальные значения $TIMx_ARR$ и $TIMx_CCRy$ ограничены $0xFFFFFFFF$ в режиме сглаживания (соответствует 264435454 для целочисленной части и 15 для сглаженной части).

Как показано на рисунке 392 и рисунке 393 ниже, режим дизеринга позволяет увеличить разрешение ШИМ.

Figure 392. PWM resolution vs frequency (16-bit mode)

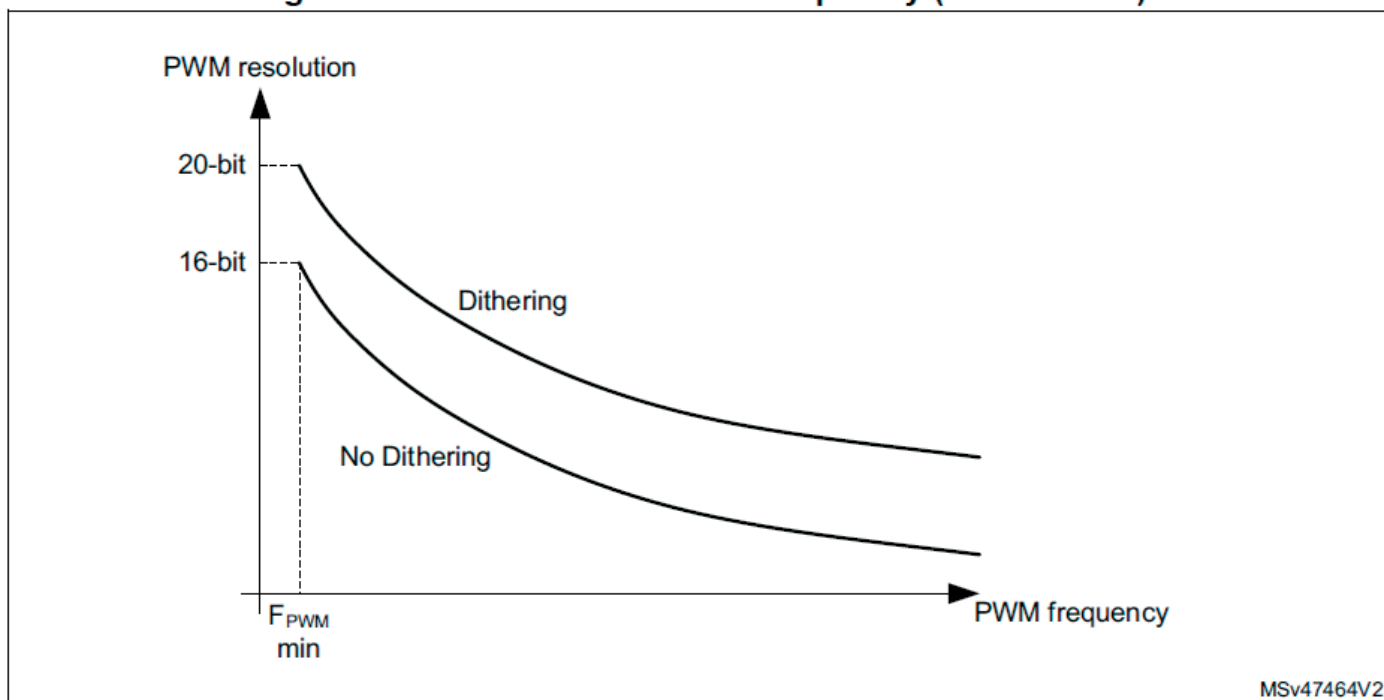
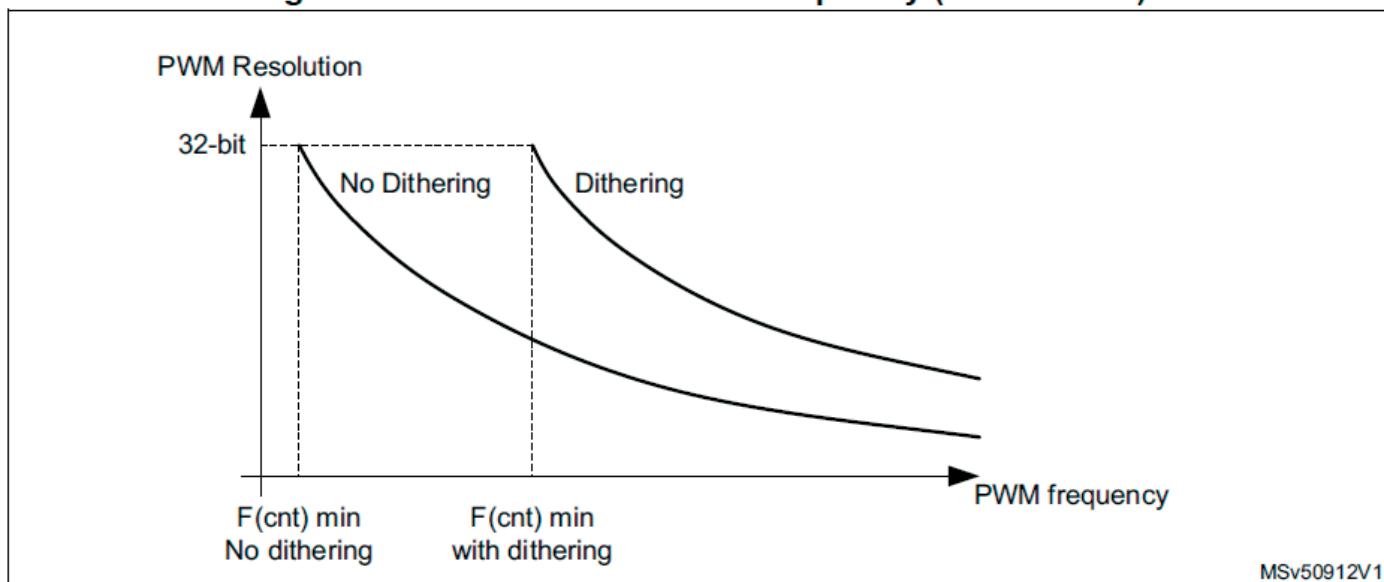


Figure 393. PWM resolution vs frequency (32-bit mode)



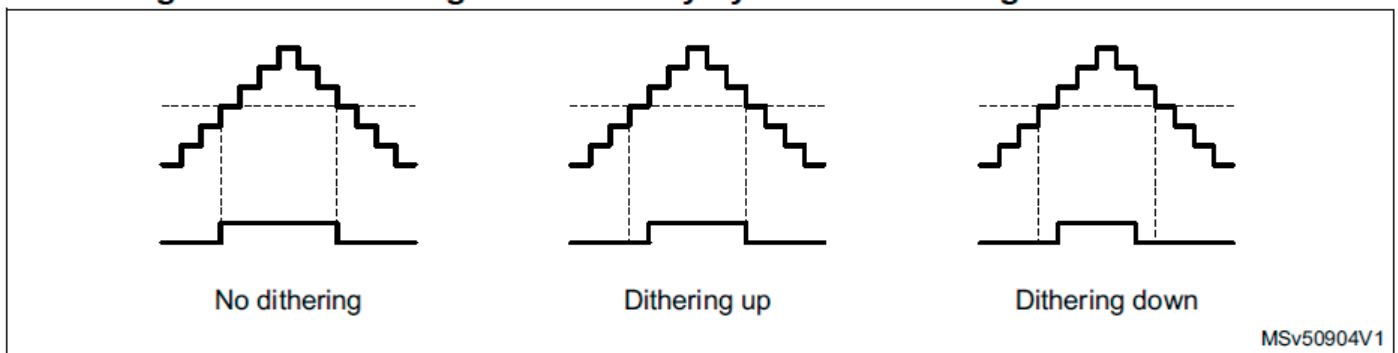
Изменения рабочего цикла и/или периода распределены по 16 последовательным периодам, как показано на рисунке 394 ниже.

Приращения значений автоматической перезагрузки и сравнения распределяются по определенным шаблонам, описанным в Таблице 279 ниже. Последовательность дизеринга выполняется для максимально равномерного распределения приращений и минимизации общей пульсации.

Режим дизеринга также доступен в режиме ШИМ с выравниванием по центру (биты CMS в регистре $TIMx_CR1$ не равны '00'). В этом случае шаблон дизеринга

применяется в течение 8 последовательных периодов ШИМ с учетом фаз прямого и обратного счета, как показано на рисунке 395 ниже.

Figure 395. Dithering effect on duty cycle in center-aligned PWM mode



В таблице 280 ниже показано, как шаблон дизеринга добавляется в режиме ШИМ с выравниванием по центру.

Table 280. CCR register change dithering pattern in center-aligned PWM mode

LSB value	PWM period															
	1		2		3		4		5		6		7		8	
	Up	Dn	Up	Dn	Up	Dn	Up	Dn	Up	Dn	Up	Dn	Up	Dn	Up	Dn
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

29.4.12 Асимметричный режим ШИМ

Асимметричный режим позволяет генерировать два выровненных по центру ШИМ-сигнала с программируемым фазовым сдвигом. В то время как частота определяется значением регистра TIMx_ARR, рабочий цикл и фазовый сдвиг определяются парой регистров TIMx_CCRx. Один регистр управляет ШИМ во время обратного счета, второй — во время прямого счета, так что ШИМ корректируется каждые полпериода ШИМ:

- `tim_oc1refc` (или `tim_oc2refc`) управляется `TIMx_CCR1` и `TIMx_CCR2`.
- `tim_oc3refc` (или `tim_oc4refc`) управляется `TIMx_CCR3` и `TIMx_CCR4`.

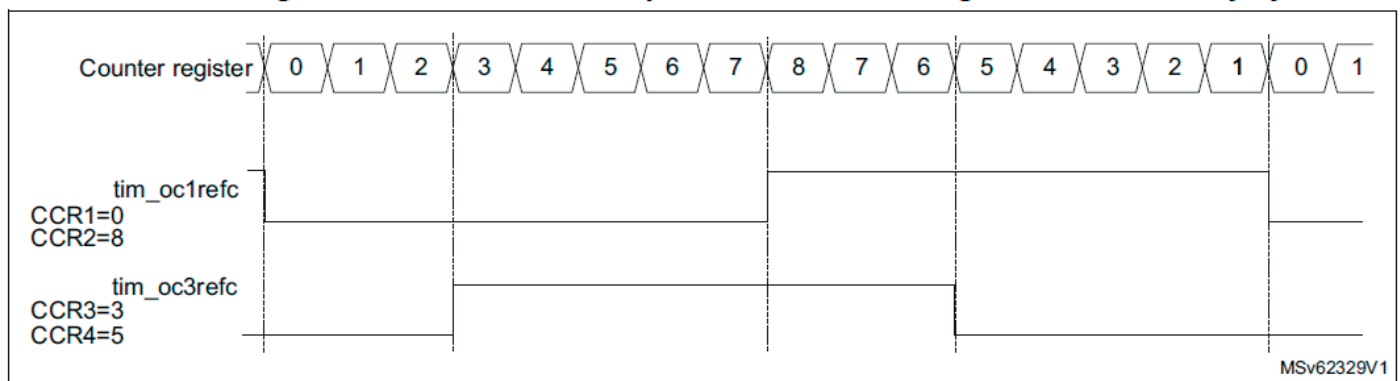
Режим асимметричной ШИМ можно выбрать независимо для двух каналов (один выход `tim_oscx` на пару регистров `CCR`) путем записи «1110» (режим асимметричной ШИМ 1) или «1111» (режим асимметричной ШИМ 2) в битах `OSxM` в регистре `TIMx_CCMRx`.

Примечание. Битовое поле `OSxM[3:0]` разделено на две части по соображениям совместимости, старший бит не граничит с тремя младшими битами.

Когда данный канал используется в качестве асимметричного канала ШИМ, его вторичный канал также может использоваться. Например, если сигнал `tim_oc1refc` генерируется на канале 1 (асимметричный режим ШИМ 1), можно вывести либо сигнал `tim_oc2ref` на канал 2, либо сигнал `tim_oc2refc`, полученный в результате асимметричного режима ШИМ 2.

На рис. 396 показан пример сигналов, которые можно генерировать в режиме асимметричной ШИМ (каналы с 1 по 4 настроены на режим 2 асимметричной ШИМ).

Figure 396. Generation of 2 phase-shifted PWM signals with 50% duty cycle



29.4.13 Комбинированный режим ШИМ

Комбинированный режим ШИМ позволяет генерировать два сигнала ШИМ, выровненных по краю или по центру, с программируемой задержкой и фазовым сдвигом между соответствующими импульсами. В то время как частота определяется значением регистра `TIMx_ARR`, рабочий цикл и задержка определяются двумя регистрами `TIMx_CCRx`. Результирующие сигналы `tim_oscxrefc` состоят из логической комбинации ИЛИ или И двух эталонных ШИМ:

- `tim_oc1refc` (или `tim_oc2refc`) управляется `TIMx_CCR1` и `TIMx_CCR2`.
- `tim_oc3refc` (или `tim_oc4refc`) управляется `TIMx_CCR3` и `TIMx_CCR4`.

Комбинированный режим ШИМ можно выбрать независимо для двух каналов (один выход `tim_oscx` на пару регистров `CCR`) путем записи «1100» (комбинированный режим ШИМ 1) или «1101» (комбинированный режим ШИМ 2) в битах `OSxM` в регистре `TIMx_CCMRx`.

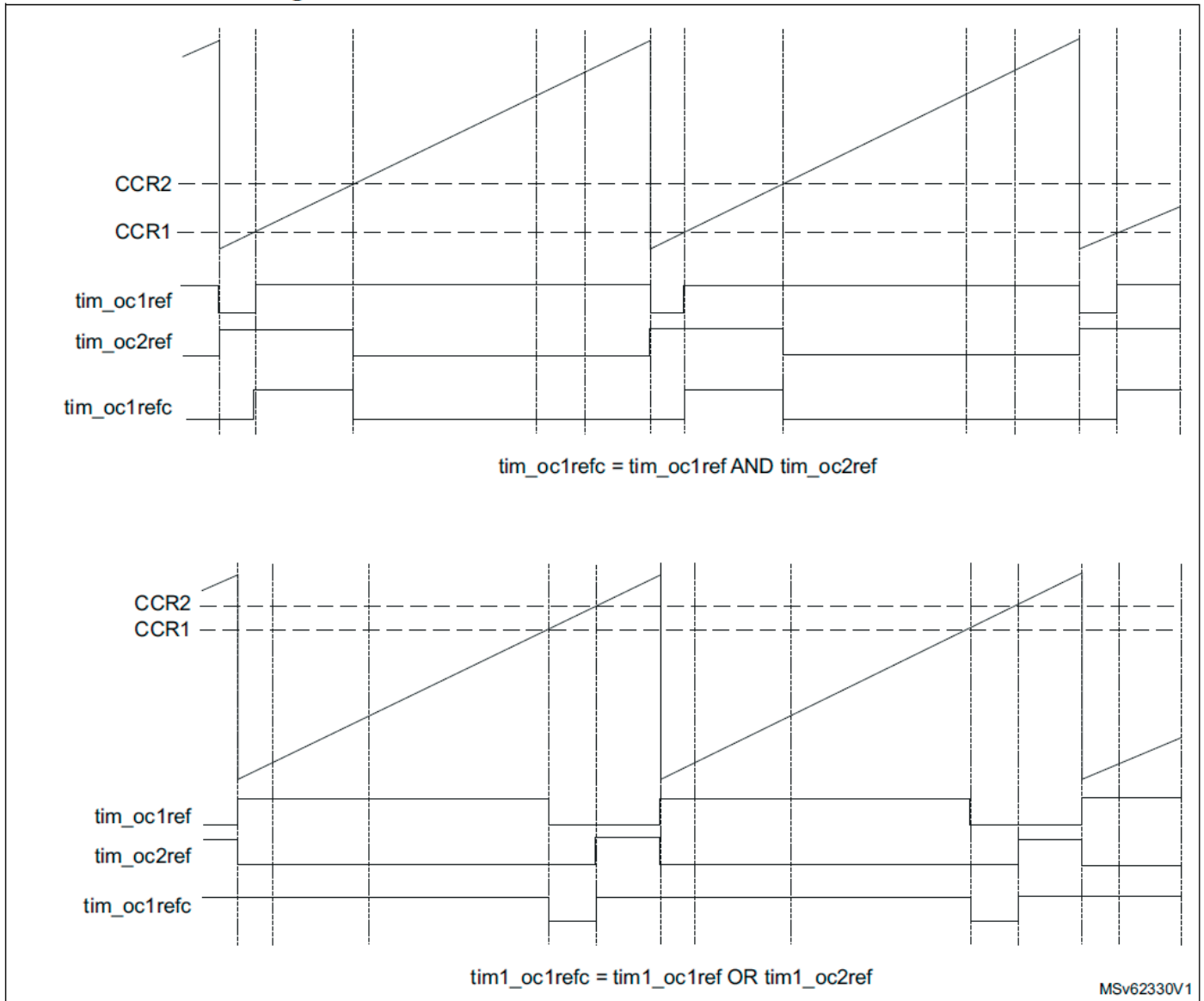
Когда данный канал используется как комбинированный канал ШИМ, его вторичный канал должен быть сконфигурирован в противоположном режиме ШИМ (например, один в комбинированном режиме ШИМ 1, а другой в комбинированном режиме ШИМ 2).

Примечание. Битовое поле `OSxM[3:0]` разделено на две части по соображениям совместимости, старший бит не граничит с тремя младшими битами.

На рис. 397 показан пример сигналов, которые можно генерировать с использованием комбинированного режима ШИМ, полученных при следующей конфигурации:

- Канал 1 настроен на комбинированный режим ШИМ 2,
- Канал 2 настроен на режим ШИМ 1,
- Канал 3 настроен на комбинированный режим ШИМ 2,
- Канал 4 настроен на ШИМ-режим 1.

Figure 397. Combined PWM mode on channels 1 and 3



MSv62330V1

29.4.14 Очистка сигнала `tim_osxref` по внешнему событию

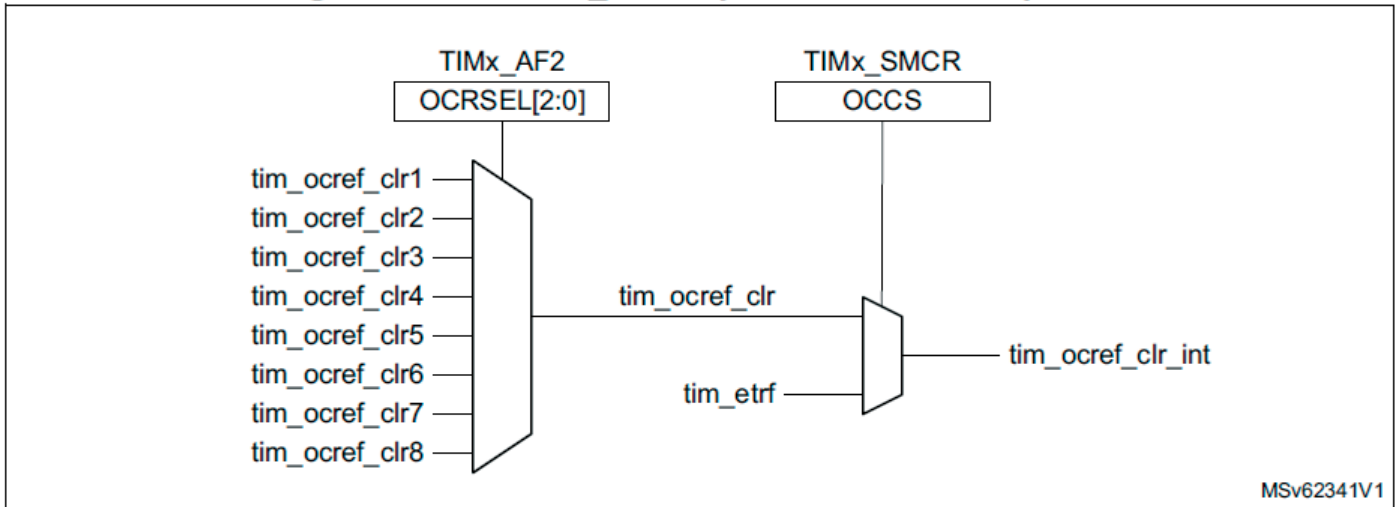
Сигнал `tim_osxref` данного канала может быть очищен при подаче высокого уровня на вход `tim_ocref_clr_int` (бит разрешения OCxSE в соответствующем регистре `TIMx_CCMRx` установлен в 1). `tim_osxref` остается низким до тех пор, пока не произойдет следующее событие обновления (UEV). Эту функцию можно использовать только в режимах сравнения выхода и ШИМ. В принудительном режиме не работает.

Источник `tim_ocref_clr_int` зависит от реализации функции четкого выбора `OCREF`, см. Раздел 29.3: Реализация TIM2/TIM3/TIM4/TIM5.

Если функция выбора очистки `OCREF` реализована, `tim_ocref_clr_int` может быть выбран между входом `tim_ocref_clr` и входом `tim_etr` (`tim_etr_in` после

фильтра) путем настройки бита OCCS в регистре TIMx_SMCR. Вход tim_ocref_clr можно выбрать из нескольких входов tim_ocref_clr[7:0] с помощью битового поля OCRSEL[2:0] в регистре TIMx_AF2, как показано на рис. 398 ниже.

Figure 398. OCREF_CLR input selection multiplexer

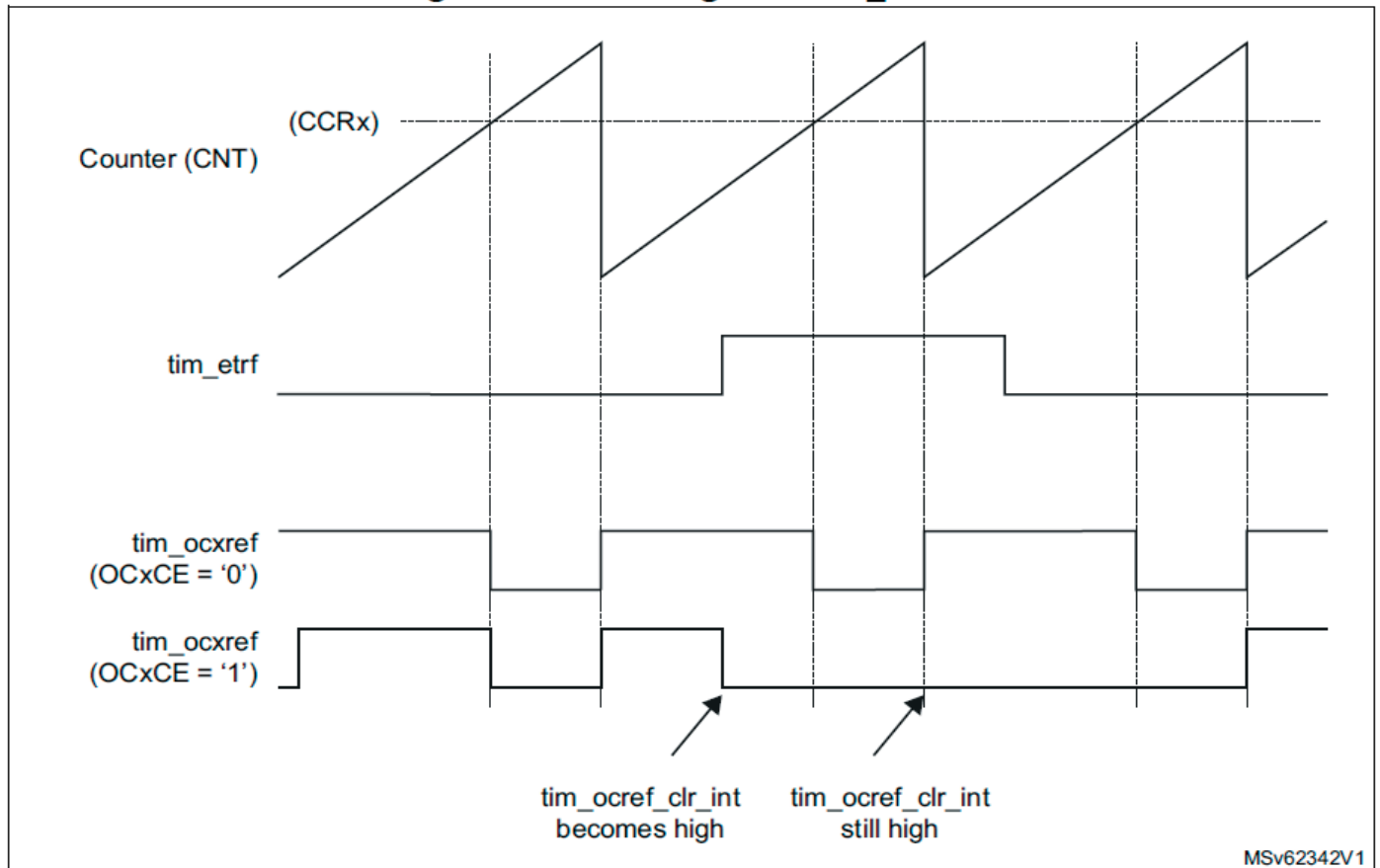


Если функция выбора очистки OCREF не реализована, вход tim_ocref_clr_int напрямую подключается к входу tim_etrif.

Например, сигнал tim_ocref_clr_int может быть подключен к выходу компаратора, который будет использоваться для текущей обработки. В этом случае tim_etrif_in необходимо настроить следующим образом:

1. Предварительный делитель внешнего триггера должен быть отключен: биты ETPS[1:0] в регистре TIMx_SMCR очищаются до 00.
2. Режим внешних часов 2 должен быть отключен: бит ECE в регистре TIM1_SMCR очищается до 0.

Figure 399. Clearing TIMx tim_ocxref



3. Полярность внешнего запуска (ETP) и фильтр внешнего запуска (ETF) можно настроить в соответствии с потребностями приложения.

Рисунок 399 показывает поведение сигнала `tim_ocr1ref`, когда вход `tim_etrif` становится высоким, для обоих значений бита разрешения OCxSE. В этом примере таймер TIMx запрограммирован в режиме PWM.

Примечание. В случае ШИМ со 100% рабочим циклом (если $CCR_x > ARR$) `tim_ocr1ref` снова включается при следующем переполнении счетчика.

29.4.15 Одноимпульсный режим

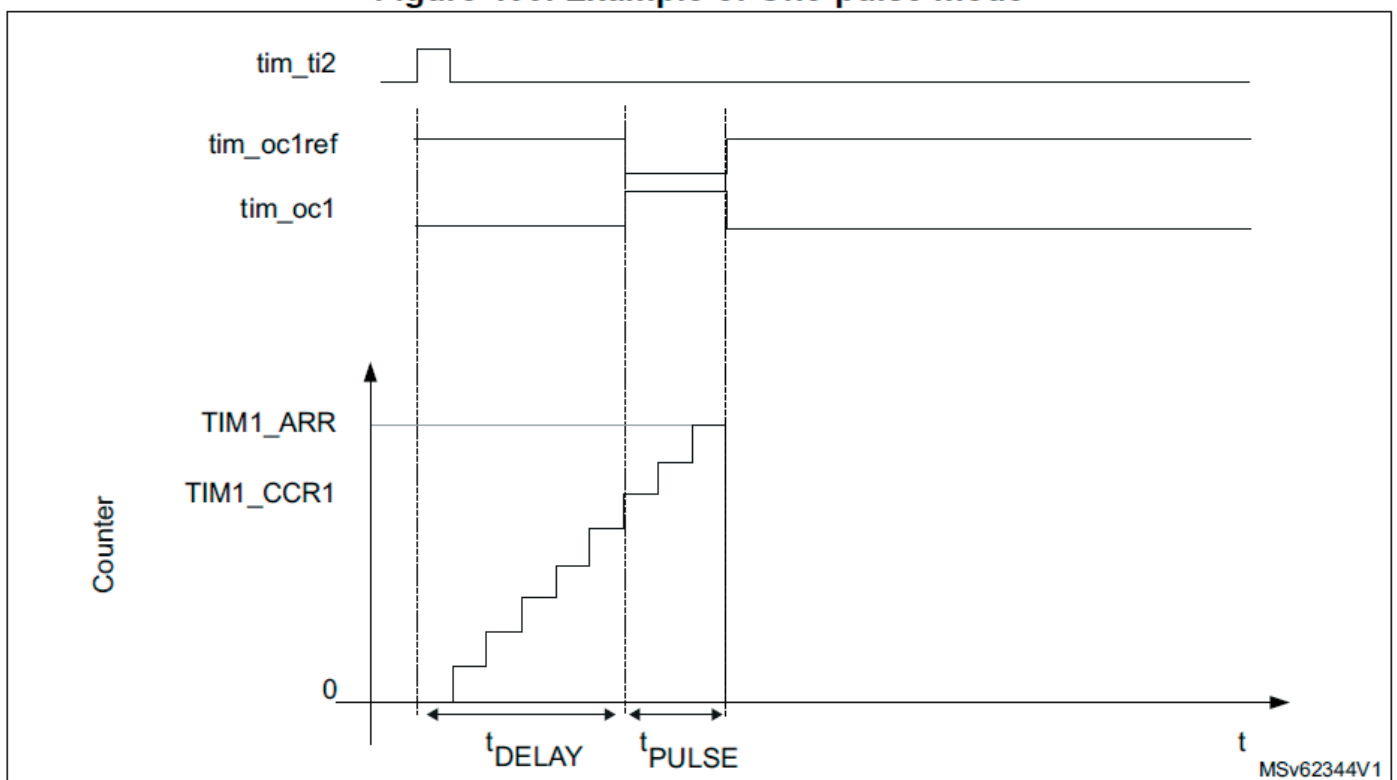
Одноимпульсный режим (ОПМ) является частным случаем предыдущих режимов. Он позволяет запускать счетчик в ответ на стимул и генерировать импульс программируемой длины после программируемой задержки.

Запуском счетчика можно управлять через контроллер ведомого режима. Генерация сигнала может выполняться в режиме сравнения выходных сигналов или в режиме ШИМ. Одноимпульсный режим выбирается установкой бита OPM в регистре TIMx_CR1. Это автоматически останавливает счетчик при следующем событии обновления UEV.

Импульс может быть сгенерирован правильно, только если значение сравнения отличается от начального значения счетчика. Перед запуском (когда таймер ожидает срабатывания) конфигурация должна быть:

- $CNT < CCR_x \leq ARR$ (в частности, $0 < CCR_x$),

Figure 400. Example of One-pulse mode



Например, может потребоваться сгенерировать положительный импульс на `tim_oc1` с длиной t_{PULSE} и после задержки t_{DELAY} , как только на входе `tim_ti2` будет обнаружен положительный фронт.

Давайте используем `tim_ti2fp2` в качестве триггера 1:

1. Выберите правильный источник `tim_ti2_in[15:0]` (внутренний или внешний) с помощью битов `TI2SEL[3:0]` в регистре TIMx_TISEL.

2. Сопоставьте `tim_ti2fp2` с `tim_ti2`, записав `CC2S=01` в регистр `TIMx_CCMR1`.
3. `tim_ti2fp2` должен обнаружить нарастающий фронт, записать `CC2P=0` и `CC2NP='0'` в регистр `TIMx_CCER`.
4. Настройте `tim_ti2fp2` как триггер для контроллера ведомого режима (`tim_trgi`), записав `TS=00110` в регистр `TIMx_SMCR`.
5. `tim_ti2fp2` используется для запуска счетчика путем записи `SMS` на '110 в регистре `TIMx_SMCR` (режим триггера).

Форма волны ОРМ определяется записью регистров сравнения (с учетом тактовой частоты и счетчика прескалера).

- `tDELAY` определяется значением, записанным в регистре `TIMx_CCR1`.
- `tPULSE` определяется разницей между значением автоматической перезагрузки и значением сравнения (`TIMx_ARR - TIMx_CCR1`).
- Допустим, нужно построить сигнал с переходом от «0» к «1», когда происходит совпадение сравнения, и переходом от «1» к «0», когда счетчик достигает значения автоматической перезагрузки. Для этого необходимо включить режим ШИМ 2, записав `OC1M=111` в регистр `TIMx_CCMR1`. Дополнительно можно включить регистры предварительной загрузки, записав `OC1PE=1` в регистр `TIMx_CCMR1` и `ARPE` в регистр `TIMx_CR1`. В этом случае необходимо записать значение сравнения в регистр `TIMx_CCR1`, значение автоматической перезагрузки в регистр `TIMx_ARR`, сгенерировать обновление, установив бит `UG`, и дождаться внешнего триггерного события на `tim_ti2`. В этом примере `CC1P` записывается в «0».

В нашем примере биты `DIR` и `CMS` в регистре `TIMx_CR1` должны быть низкими.

Поскольку требуется только 1 импульс (одиночный режим), в бит `OPM` в регистре `TIMx_CR1` должна быть записана 1, чтобы остановить счетчик при следующем событии обновления (когда счетчик переходит от значения автоматической перезагрузки обратно к 0). Когда бит `OPM` в регистре `TIMx_CR1` установлен на «0», значит, выбран повторяющийся режим.

Частный случай: быстрое включение `tim_oscx`:

В одноимпульсном режиме обнаружение фронта на входе `tim_tix` устанавливает бит `SEN`, который включает счетчик. Затем сравнение между счетчиком и сравниваемым значением приводит к переключению вывода. Но для этих операций требуется несколько тактов, и это ограничивает минимальную задержку `t DELAY min`, которую мы можем получить.

Если нужно вывести сигнал с минимальной задержкой, можно установить бит `OSxFE` в регистре `TIMx_CCMRx`. Затем `tim_oscxref` (и `tim_oscx`) форсируется в ответ на стимул без учета сравнения. Его новый уровень такой же, как если бы произошло совпадение сравнения. `OSxFE` действует, только если канал настроен в режиме `PWM1` или `PWM2`.

29.4.16 Повторно запускаемый одноимпульсный режим

Этот режим позволяет запускать счетчик в ответ на стимул и генерировать импульс программируемой длины, но со следующими отличиями от режима одного импульса без повторного запуска, описанного в разделе 29.4.15:

- Импульс начинается сразу после срабатывания триггера (без программируемой задержки)

- Импульс продлевается, если новый запуск происходит до завершения предыдущего. Таймер должен находиться в режиме Slave с битами $SMS[3:0] = '1000'$ (комбинированный сброс + запуск).

mode) в регистре $TIMx_SMCR$, а биты $OCxM[3:0]$ установлены в «1000» или «1001» для режима Retriggerable OPM 1 или 2.

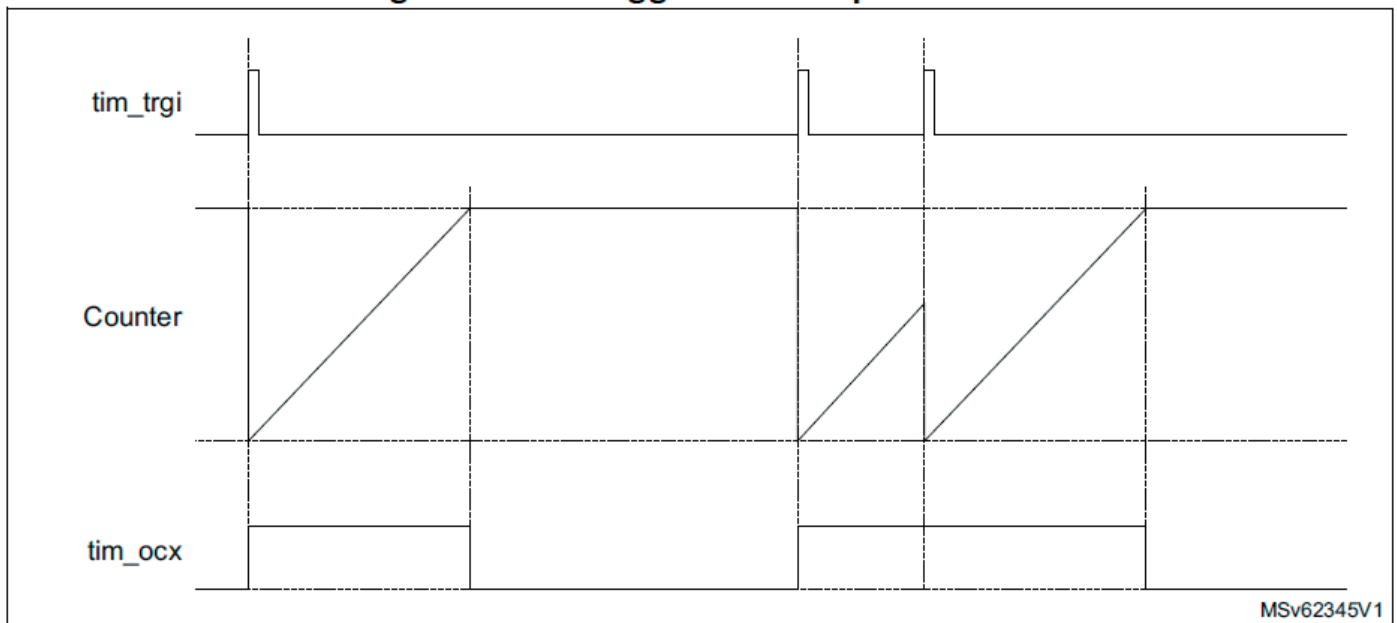
Если таймер сконфигурирован в режиме обратного счета, соответствующий $CCRx$ должен быть установлен в 0 (регистр ARR устанавливает длину импульса). Если таймер настроен на режим обратного счета, $CCRx$ должен быть больше или равен ARR .

Примечание. В одноимпульсном режиме с повторным запуском флаг $CCxIF$ не имеет значения.

Битовые поля $OCxM[3:0]$ и $SMS[3:0]$ разделены на две части по соображениям совместимости, старший бит не граничит с тремя младшими битами.

Этот режим нельзя использовать с режимами ШИМ с выравниванием по центру. Обязательно иметь $CMS[1:0] = 00$ в $TIMx_CR1$.

Figure 401. Retriggerable one-pulse mode



29.4.17 Импульс в режиме сравнения

Импульс может быть сгенерирован при событии сравнения совпадений. Сигнал с программируемой шириной импульса, генерируемый, когда значение счетчика равно заданному значению сравнения, для целей отладки или синхронизации.

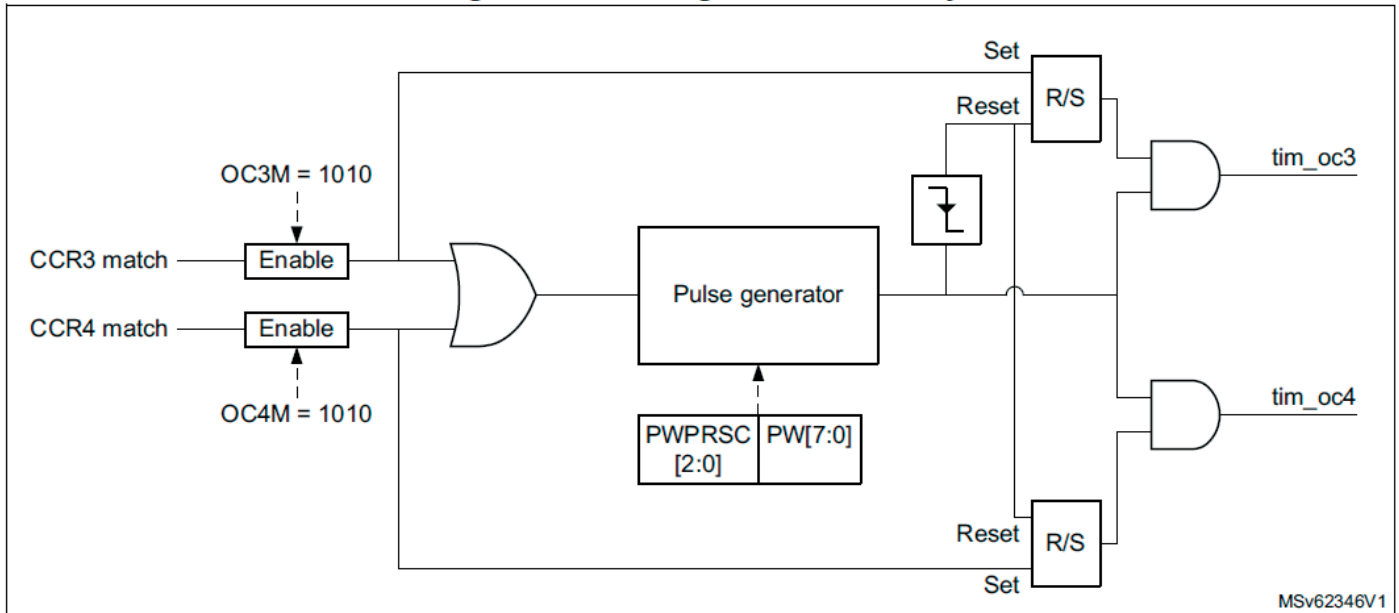
Этот режим доступен для выбора любого подчиненного режима, включая режимы энкодера, в режимах счета с выравниванием по краю и центру. Он доступен только для канала 3 и канала 4. Генератор импульсов уникален и используется совместно двумя каналами, как показано на рисунке 402 ниже.

На рисунке 403 ниже показано, как генерируется импульс для режимов работы с выравниванием по фронту и энкодера.

Этот режим сравнения вывода выбирается с помощью битовых полей $OC3M[3:0]$ и $OC4M[3:0]$ в регистре $TIMx_CCMR2$.

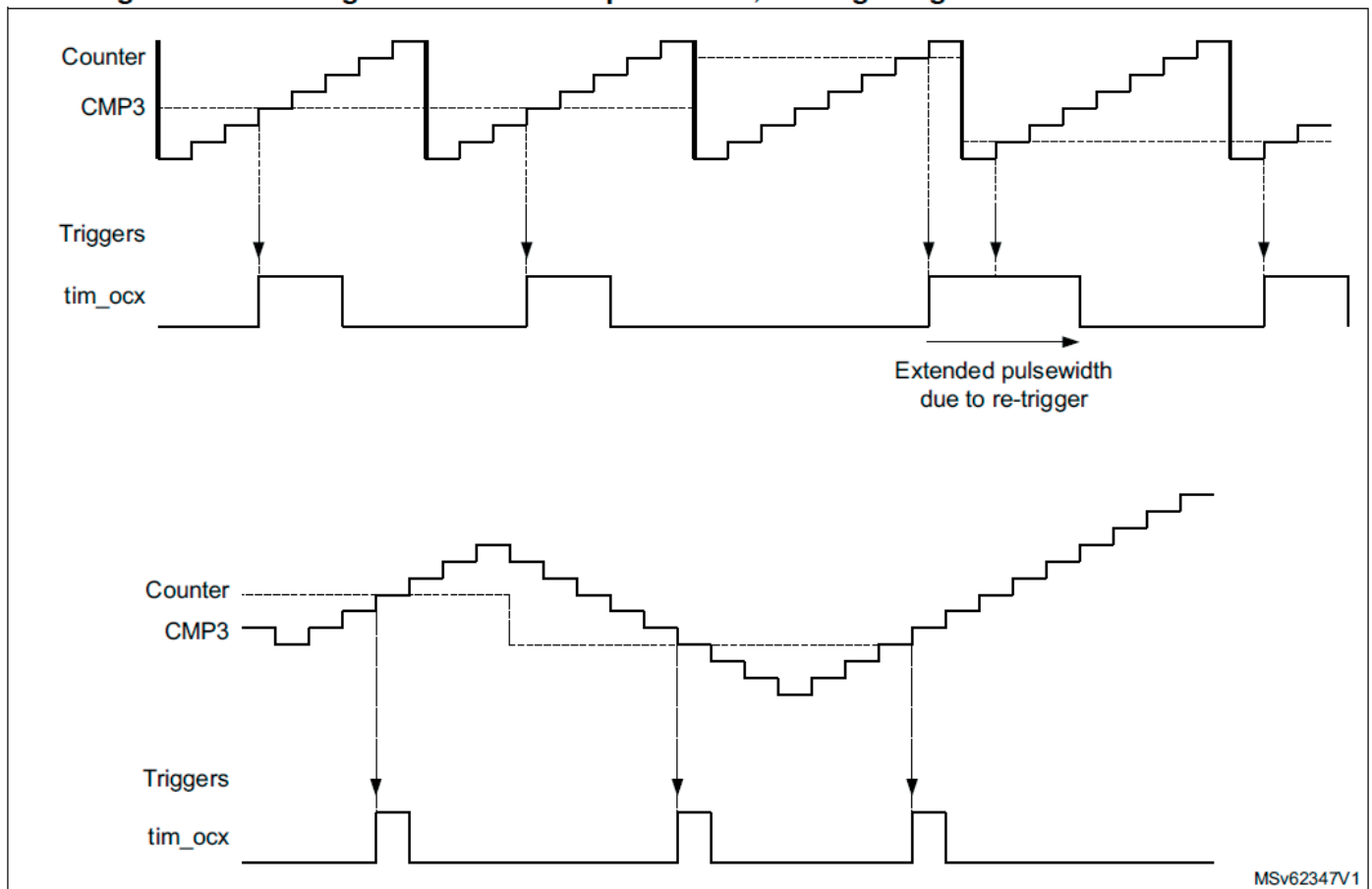
Длительность импульса программируется с использованием битового поля $PW[7:0]$ в регистре, с использованием определенного тактового сигнала, предварительно масштабированного в соответствии с битами $PWPRSC[2:0]$, следующим образом:

Figure 402. Pulse generator circuitry



MSv62346V1

Figure 403. Pulse generation on compare event, for edge-aligned and encoder modes



MSv62347V1

$$t_{PW} = PW[7:0] \times t_{PWG}$$

$$\text{where } t_{PWG} = (2^{PWPRSC[2:0]}) \times t_{tim_ker_ck}$$

Где t_{PWG} дает разрешение и максимальные значения в зависимости от значения предварительного делителя.

Импульс можно запускать повторно: новый триггер во время действия импульса вызывает продление импульса.

Примечание. Если два канала включены одновременно, импульсы выдаются независимо, пока триггер на одном канале не перекрывает импульс, генерируемый на па-

раллельном выходе. Наоборот, если два триггера перекрываются, ширина импульса, относящаяся к 1-му прибывшему триггеру, увеличивается (из-за повторного триггера), в то время как ширина импульса последнего пришедшего триггера является правильной (как показано на рисунке 404 ниже).

Figure 404. Extended pulse width in case of concurrent triggers

