

14.1 Introduzione a TIM1 e TIM8

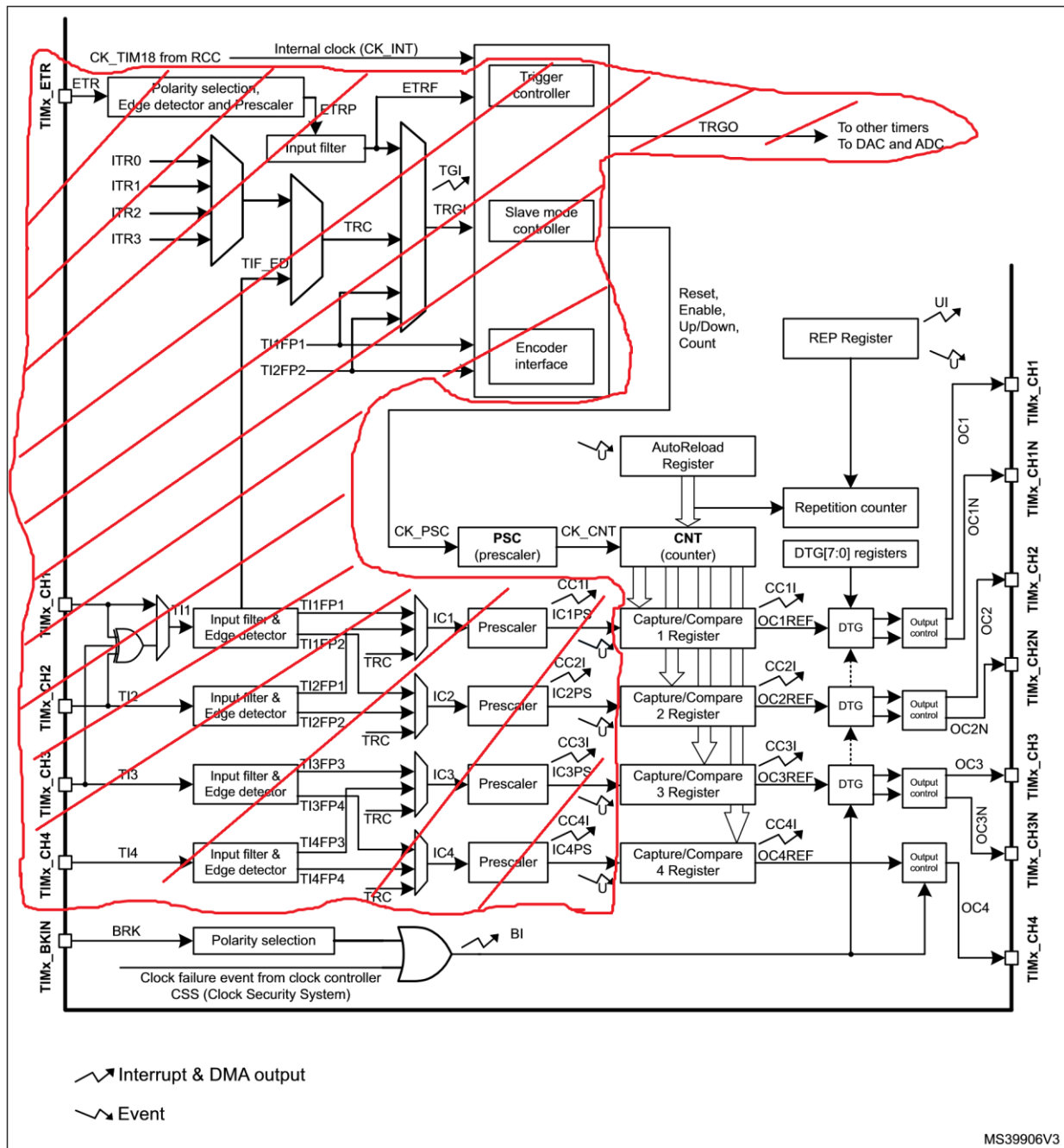
Le lunghezze degli impulsi e i periodi della forma d'onda possono essere modulati da pochi microsecondi a diversi millisecondi utilizzando il prescaler timer e i prescaler del controller di clock RCC.

I timer di controllo avanzato (TIM1 e TIM8) e generici (TIMx) sono completamente indipendenti e non condividono alcuna risorsa. Possono essere sincronizzati insieme come descritto nella [Sezione 14.3.20](#).

14.2 Caratteristiche principali di TIM1 e TIM8

Le caratteristiche del timer TIM1 e TIM8 includono:

- Contatore di ricarica automatica up, down, up/down a 16 bit.
- Prescaler programmabile a 16 bit che consente di dividere (anche "al volo") la frequenza di clock del contatore per qualsiasi fattore compreso tra 1 e 65536.
- Fino a 4 canali indipendenti per:
 - Generazione PWM (modalità allineata al bordo e al centro) + dead time
- Contatore di ripetizioni per aggiornare i registri del timer solo dopo un determinato numero di cicli del contatore.
- Interrompi ingresso per mettere i segnali di uscita del timer in stato di ripristino o in uno stato noto.
- Generazione di interrupt/DMA sui seguenti eventi:
 - Aggiornamento: overflow/underflow del contatore, inizializzazione del contatore (tramite software o trigger interno/esterno)
 - Evento trigger (avvio, arresto, inizializzazione o conteggio del contatore tramite trigger interno/esterno)



Tutta la parte cancellata in rosso è la parte ridotta, quella che ho rimosso dalla documentazione perché non la userò per l'inverter. Questo documento parla solo della parte non in rosso !

14.3 Descrizione funzionale di TIM1 e TIM8

14.3.1 Unità di base dei tempi

Il blocco principale del timer programmabile per il controllo avanzato è un contatore a 16 bit con il relativo registro di ricarica automatica. Il contatore può contare verso l'alto, verso il basso o sia verso l'alto che verso il basso. Il contatore può essere diviso per un prescaler.

Il contatore, il registro di ricarica automatica e il registro prescaler possono essere scritti o letti dal software. Questo vale anche quando il contatore è in esecuzione.

L'unità di base dei tempi comprende:

- Registro contatori (TIMx_CNT)
- Registro prescaler (TIMx_PSC)
- Registro di ricaricamento automatico (TIMx_ARR) [Auto-reload register](#)
- Registro dei contatori di ripetizioni (TIMx_RCR) [Repetition counter register](#)

Il registro di ricarica automatica è precaricato. La scrittura o la lettura dal registro di ricaricamento automatico accede al registro di precaricamento. Il contenuto del registro di precaricamento viene trasferito nel registro shadow in modo permanente **o a ogni evento di aggiornamento (UEV)**, a seconda del bit di abilitazione del precaricamento di ricaricamento automatico (**ARPE** [auto-reload preload enable](#)) nel registro TIMx_CR1. L'evento di aggiornamento viene inviato quando il contatore raggiunge l'overflow (o l'underflow durante il downcount) **e se il bit UDIS è uguale a 0** nel registro TIMx_CR1. Può anche essere generato da un software.

Bit 1 UDIS: Disabilitazione aggiornamento

Questo bit viene impostato e cancellato dal software per abilitare/disabilitare la generazione di eventi UEV.

0: UEV abilitato. L'evento Update (UEV) viene generato da uno dei seguenti eventi:

- [Counter overflow/underflow](#)
- [Setting the UG bit](#)
- Generazione di aggiornamenti tramite il controller in modalità slave I registri bufferizzati vengono quindi caricati con i relativi valori di precaricamento.

1: UEV disabilitato. L'evento Update non viene generato, i registri shadow mantengono il loro valore (ARR, PSC, CCRx). Tuttavia, il contatore e il prescaler vengono reinizializzati se il bit UG è impostato o se viene ricevuto un reset hardware dal controller in modalità slave.

Il contatore conta solo quando è impostato il bit di abilitazione del contatore (CEN) nel registro TIMx_CR1

Si noti che il contatore inizia a contare 1 ciclo di clock dopo aver impostato il bit **CEN** nel registro TIMx_CR1.

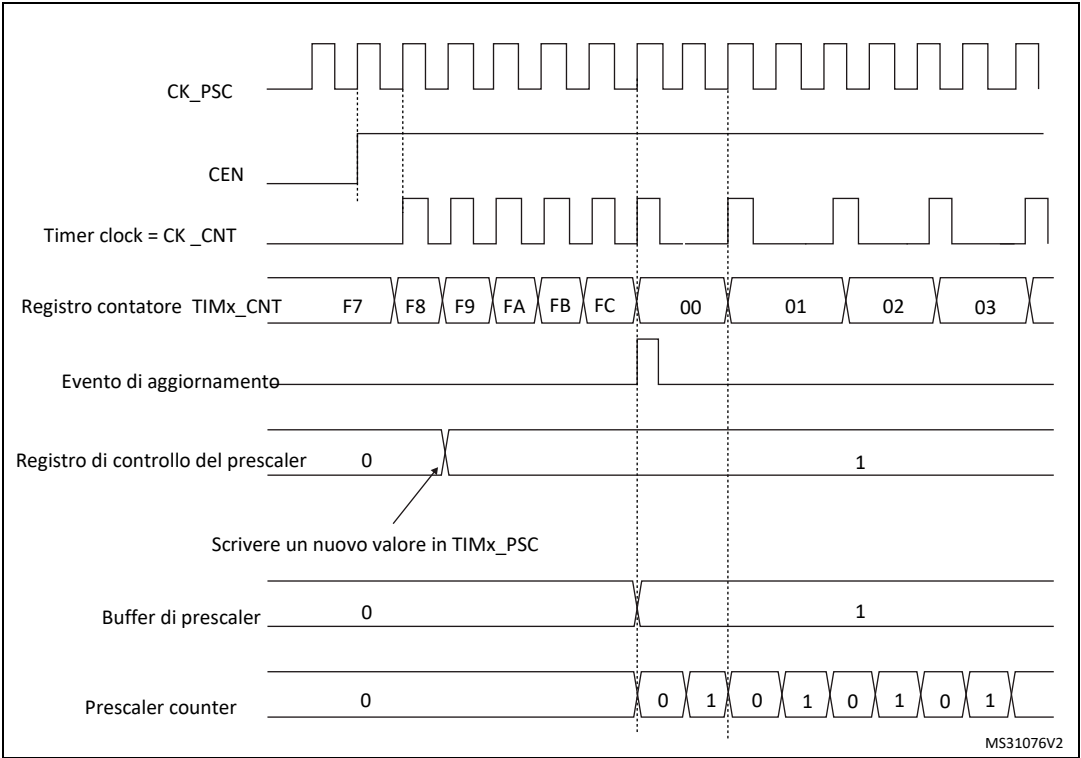
Descrizione del prescaler

Il prescaler può dividere la frequenza di clock del contatore per qualsiasi fattore compreso tra 1 e 65536. Si basa su un contatore a 16 bit controllato tramite un registro a 16 bit (nel registro TIMx_PSC). Può essere modificato al volo poiché questo registro di controllo è bufferizzato. Il nuovo rapporto prescaler viene preso in considerazione all'evento di aggiornamento successivo.

[Nota che dice che è sempre bufferizzato ARPE lavora solo sul TIMx_ARR.](#) (sarei curioso di vedere se è vero ! comunque non penso che varierò il prescaler a runtime)

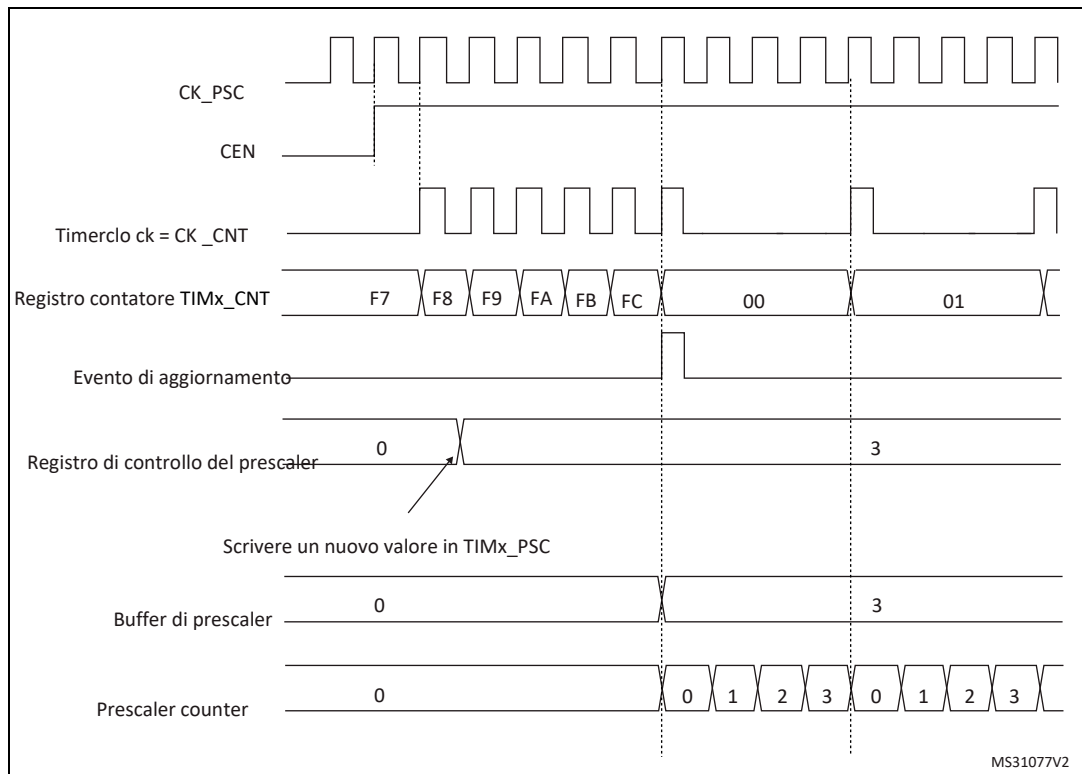
La Figura 53 e *la Figura 54* forniscono alcuni esempi del comportamento del contatore quando il rapporto del prescaler viene modificato al volo:

Figura 53. Diagramma temporizzatore contatore con variazione della divisione prescaler da 1 a 2



Buffer di prescaler e prescaler counter esisteranno ma non sono accessibili non li ho trovati ne li fa vedere. Ok

Figura 54. Diagramma di temporizzazione del contatore con cambio di divisione del prescaler da 1 a 4



14.3.2 Counter modes

Upcounting mode

In modalità upcounting, il contatore conta da 0 al valore di ricarica automatica (contenuto del registro **TIMx_ARR**), quindi riparte da 0 e genera un evento di overflow del contatore.

Se si utilizza il contatore di ripetizioni, l'evento di aggiornamento (UEV) viene generato dopo che il conteggio è stato ripetuto per il numero di volte programmato nel registro del contatore di ripetizioni più uno (**TIMx_RCR**+1). In caso contrario, l'evento update viene generato a ogni overflow del contatore.

Anche l'impostazione del bit UG nel registro TIMx_EGR (tramite software o tramite il controller in modalità slave) genera un evento di aggiornamento.

Bit **UG**: Generazione di aggiornamenti Update generation

Questo bit può essere impostato dal software, viene cancellato automaticamente dall'hardware.

0: Nessuna azione

1: Reinizializza il contatore e genera un aggiornamento dei registri. Si noti che anche il contatore del prescaler viene cancellato (in ogni caso il rapporto del prescaler non viene influenzato). Il contatore viene azzerato se è selezionata la modalità allineata al centro o se DIR=0 (upcounting), altrimenti prende il valore di ricarica automatica (TIMx_ARR) se DIR=1 (downcounting).

L'evento UEV può essere disabilitato dal software impostando il bit UDIS nel registro TIMx_CR1. In questo modo si evita di aggiornare i registri shadow durante la scrittura di nuovi valori nei registri di precaricamento. Quindi non si verifica alcun evento di aggiornamento fino a quando il bit UDIS

non è stato scritto su 0. Tuttavia, il contatore riparte da 0, così come il contatore del prescaler (ma la velocità di prescalabilità non cambia). Inoltre, se il bit URS (selezione della richiesta di aggiornamento) nel registro TIMx_CR1 è impostato, l'impostazione del bit UG genera un evento di aggiornamento UEV ma senza impostare il flag UIF (quindi non viene inviata alcuna richiesta di interrupt o DMA).

Bit **URS**: Aggiorna l'origine della richiesta Update request source

Questo bit viene impostato e cancellato dal software per selezionare le origini degli eventi UEV.

0: uno qualsiasi dei seguenti eventi genera un interrupt di aggiornamento o una richiesta DMA, se abilitata. Questi eventi possono essere:

- Counter overflow/underflow
- Impostazione del bit UG
- Generazione di aggiornamenti tramite il controller in modalità slave

1: Solo l'overflow/underflow del contatore genera un interrupt di aggiornamento o una richiesta DMA, se abilitata.

Quando si verifica un evento di aggiornamento, tutti i registri vengono aggiornati e viene impostato il flag di aggiornamento (bit UIF nel registro TIMx_SR):

- Il contatore di ripetizioni viene ricaricato con il contenuto di TIMx_RCR registro,
- Il registro shadow di ricaricamento automatico viene aggiornato con il valore di precaricamento (TIMx_ARR),
- Il buffer del prescaler viene ricaricato con il valore di precaricamento (contenuto del registro TIMx_PSC).

Le figure seguenti mostrano alcuni esempi del comportamento del contatore per diverse frequenze di clock quando TIMx_ARR=0x36.

Figura 55. Diagramma temporizzatore contatore, orologio interno diviso per 1

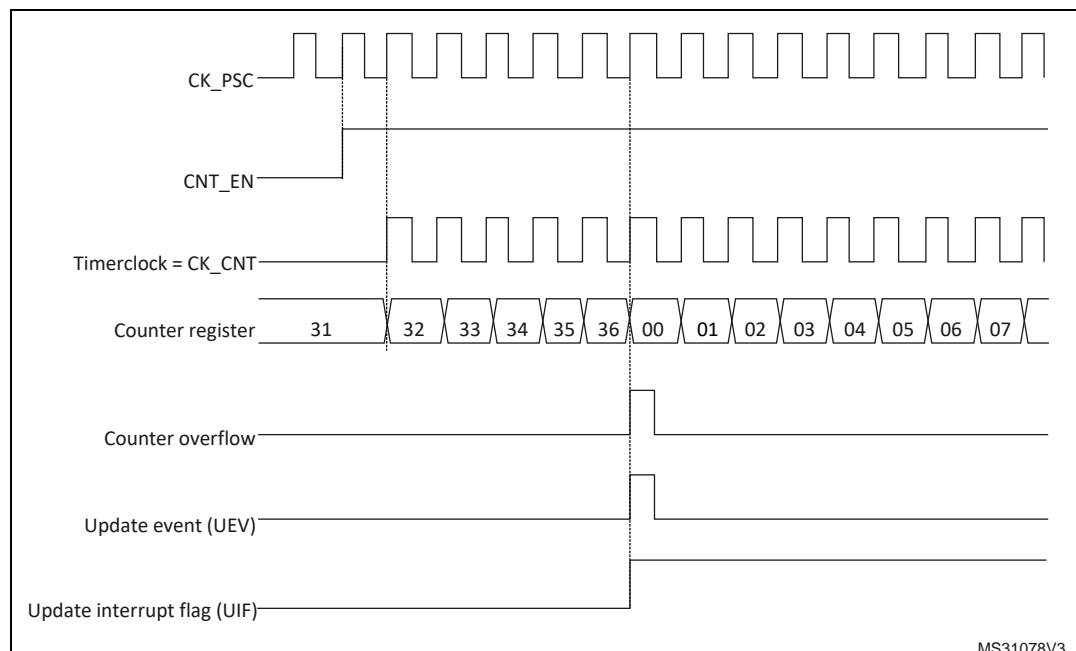


Figura 56. Diagramma temporizzatore contatore, orologio interno diviso per 2

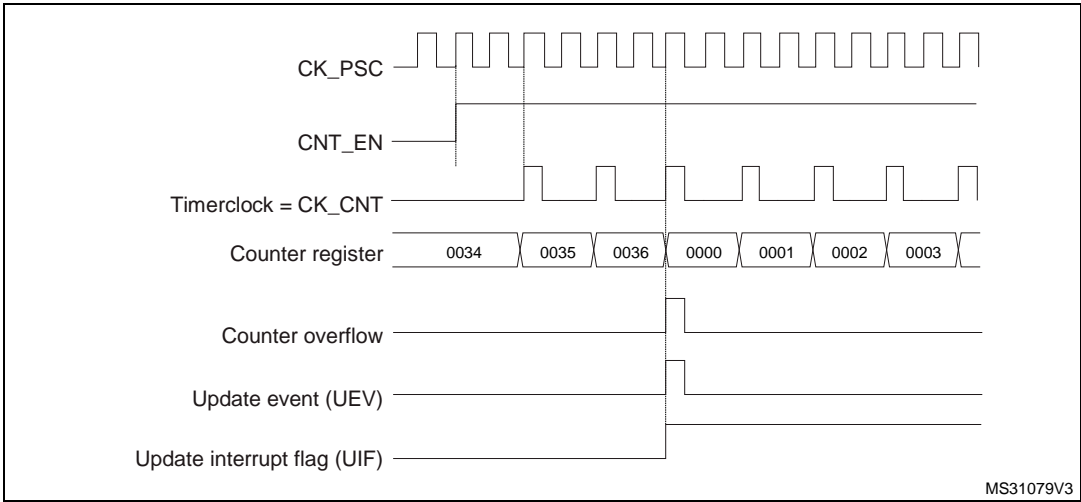


Figure 57. Counter timing diagram, internal clock divided by 4

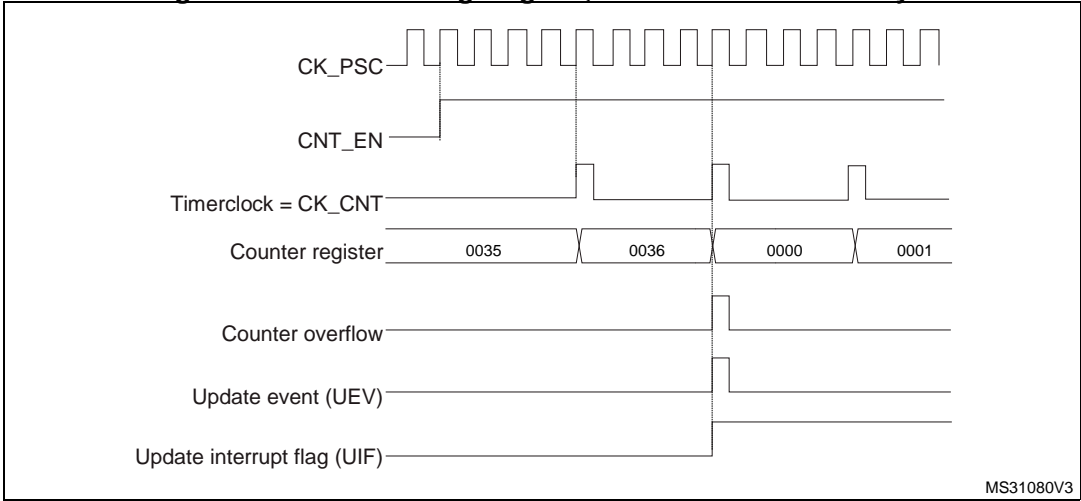
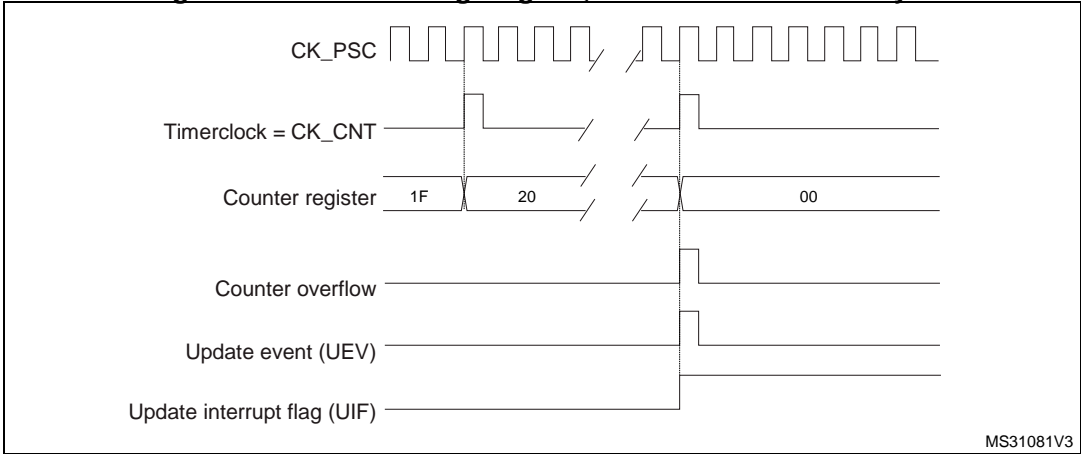


Figure 58. Counter timing diagram, internal clock divided by N



Vabbe da fig 56 a fig 58 abbastanza inutili poteva evitare di ripetersi.

Le due seguenti invece 59 e 60 sono importanti...

Figure 59. Counter timing diagram, update event when ARPE=0 (TIMx_ARR not preloaded)

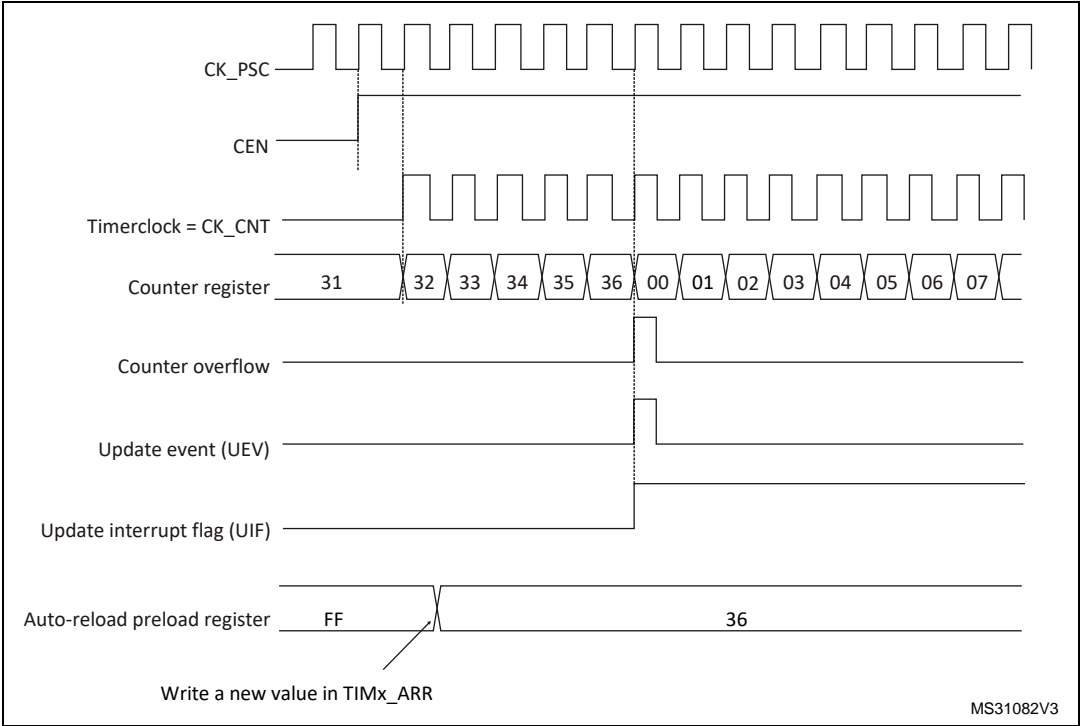
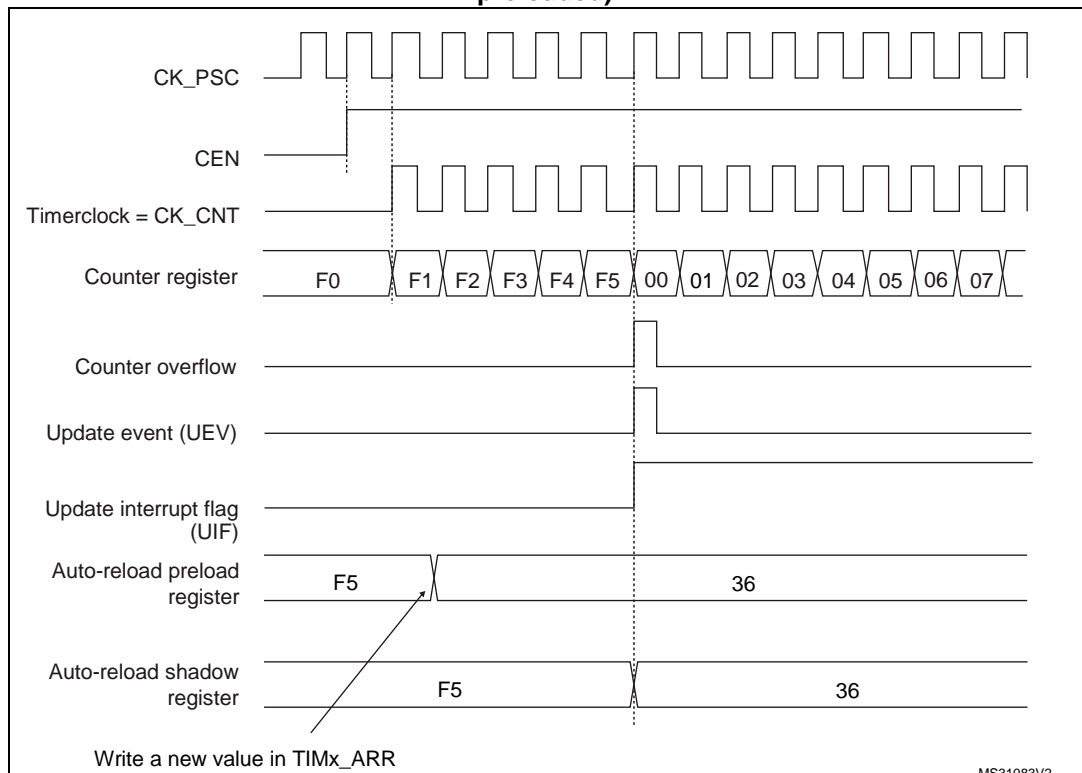


Figure 60. Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)



Modalità di downcounting

In pratica fa vedere circa le stesse cose di sopra ma per il downcounting e nota che non ti dice come fare per impostare il downcounting ! lo scopri forse molto più avanti !

In modalità di downcounting, il contatore conta dal valore di ricarica automatica (contenuto del registro TIMx_ARR) fino a 0, quindi riparte dal valore di ricaricamento automatico e genera un evento di underflow del contatore.

Se si utilizza il contatore di ripetizioni, l'evento di aggiornamento (UEV) viene generato dopo che il downcount è stato ripetuto per il numero di volte programmato nel registro del contatore di ripetizioni più uno (TIMx_RCR+1). In caso contrario, l'evento update viene generato a ogni underflow del contatore.

Anche l'impostazione del bit UG nel registro TIMx_EGR (tramite software o tramite il controller in modalità slave) genera un evento di aggiornamento.

L'evento di aggiornamento UEV può essere disabilitato dal software impostando il bit UDIS nel registro TIMx_CR1. In questo modo si evita di aggiornare i registri shadow durante la scrittura di nuovi valori nei registri di precaricamento. Quindi non si verifica alcun evento di aggiornamento fino a quando il bit UDIS non è stato scritto su 0. Tuttavia, il contatore viene riavviato dal valore di ricaricamento automatico corrente, mentre il contatore del prescaler viene riavviato da 0 (ma la frequenza di prescalabilità non cambia).

Inoltre, se è impostato il bit URS (selezione della richiesta di aggiornamento) nel registro TIMx_CR1, impostando il

Il bit UG genera un evento di aggiornamento UEV ma senza impostare il flag UIF (quindi non viene inviata alcuna richiesta di interrupt o DMA). (In questo modo si evita di generare interrupt di aggiornamento e di acquisizione quando si cancella il contatore sull'evento di acquisizione. È input capture che sopra ho rimosso qui ho lasciato per curiosità)

Quando si verifica un evento di aggiornamento, tutti i registri vengono aggiornati e viene impostato il flag di aggiornamento (bit UIF nel registro TIMx_SR):

- Il contatore di ripetizioni viene ricaricato con il contenuto del registro TIMx_RCR
- Il buffer del prescaler viene ricaricato con il valore di preload (contenuto del registro TIMx_PSC)
- Il registro attivo di ricarica automatica viene aggiornato con il valore di precaricamento (contenuto del registro TIMx_ARR). Si noti che il ricaricamento automatico viene aggiornato prima che il contatore venga ricaricato, in modo che il periodo successivo sia quello previsto

Le figure seguenti mostrano alcuni esempi del comportamento del contatore per diverse frequenze di clock quando TIMx_ARR=0x36.

Figure 61. Counter timing diagram, internal clock divided by 1

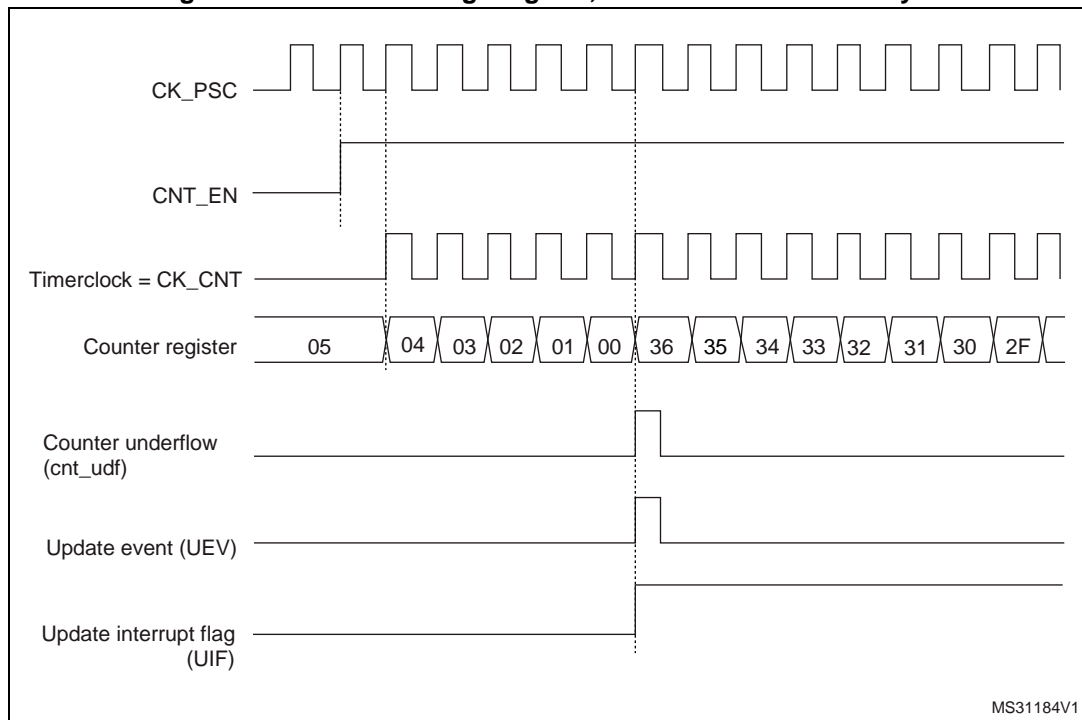


Figure 62. Counter timing diagram, internal clock divided by 2

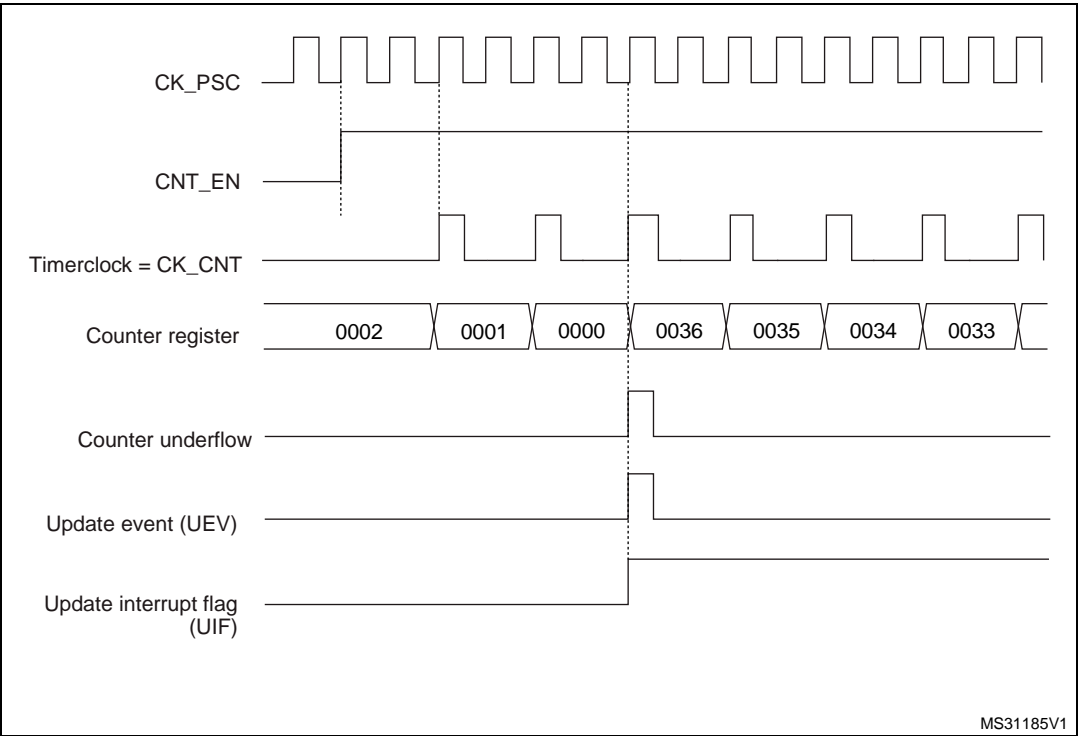


Figure 63. Counter timing diagram, internal clock divided by 4

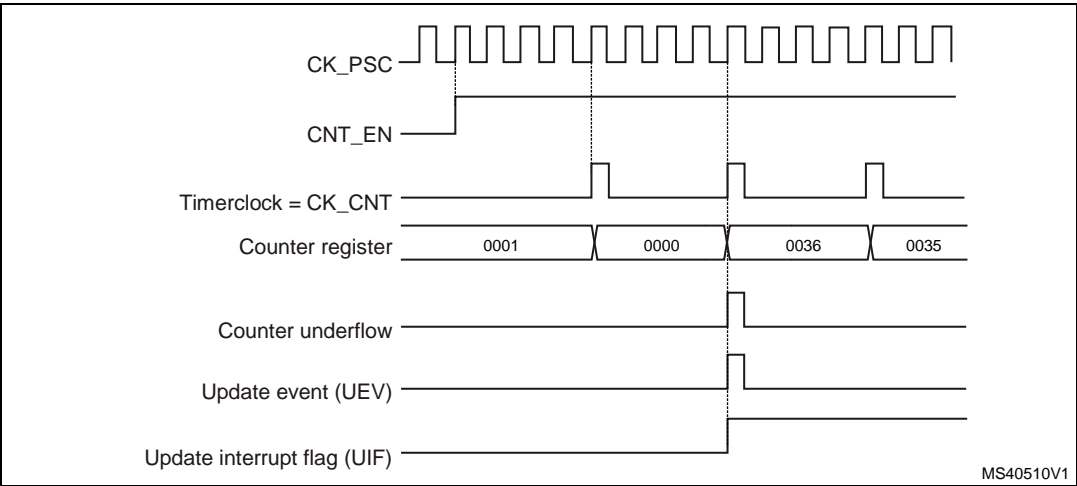


Figure 64. Counter timing diagram, internal clock divided by N

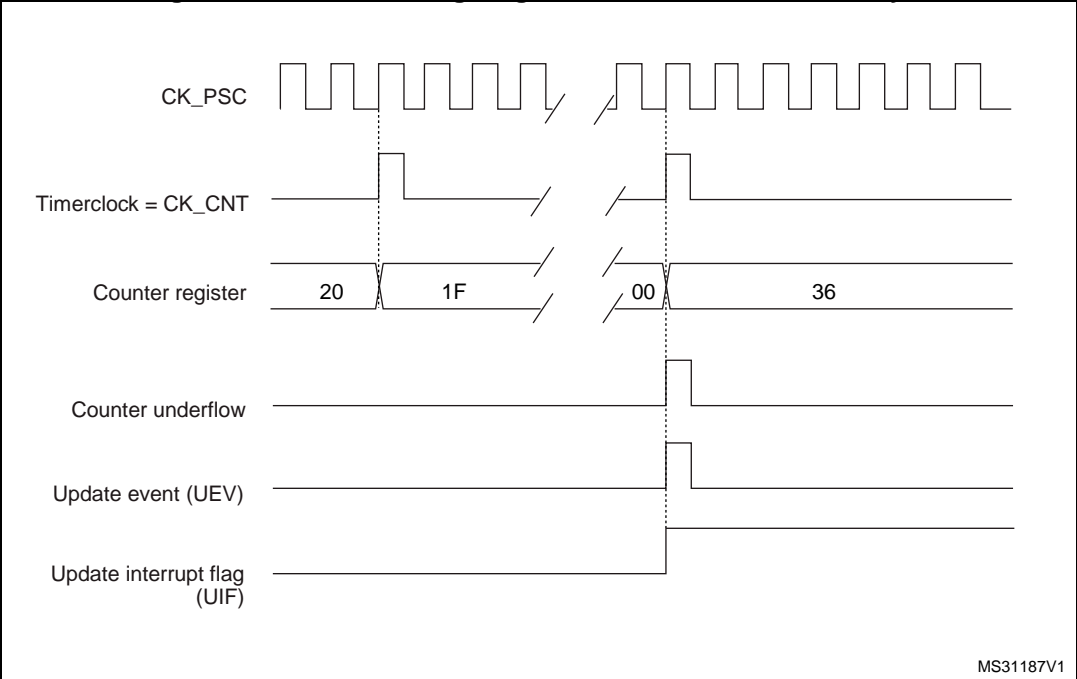
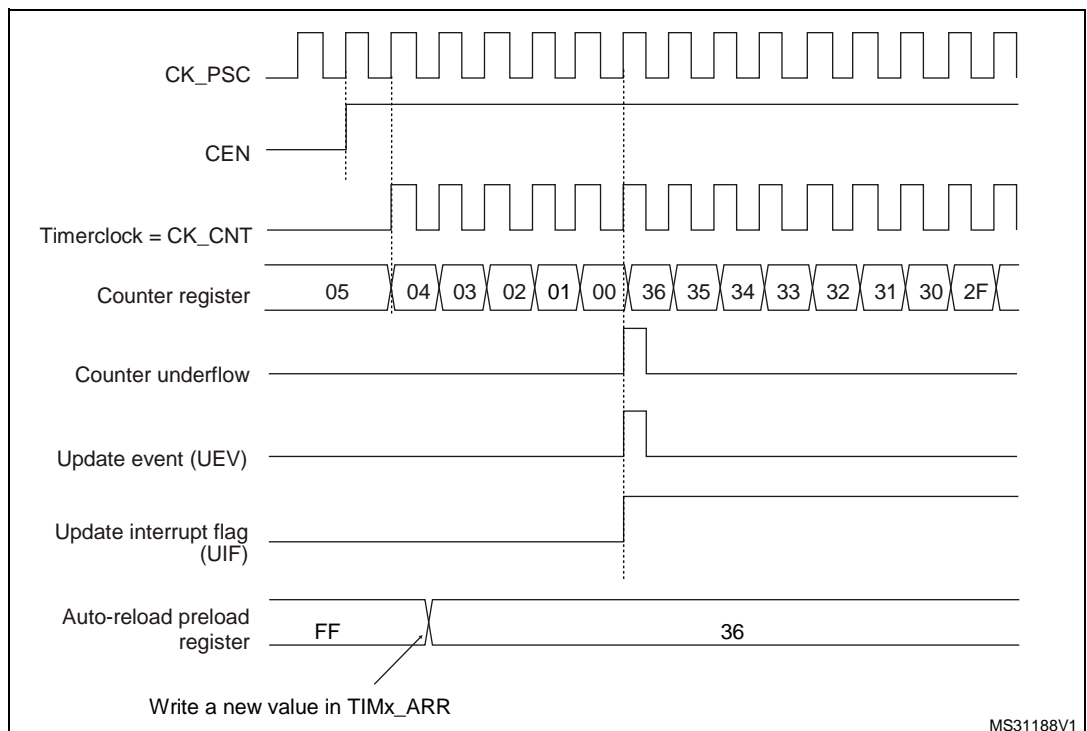


Figure 65. Counter timing diagram, update event when repetition counter is not used



Vabbè questa sopra importante le altre da 61 a 64 poteva evitarle. E nota bene che non c'è la faccenda del ARPE=1 precaricamento bufferizzazione registro shadow... Chiaro arriva a zero e prende il valore presente nel TIMx_ARR senza giri shadow ARPE ecc..

Modalità allineata al centro (conteggio su/giù) Questa è quella che userò per inverter...

In modalità allineata al centro, il contatore conta da 0 al valore di ricaricamento automatico (contenuto del registro TIMx_ARR) - 1, genera un evento di overflow del contatore, quindi conta dal valore di caricamento automatico fino a 1 e genera un evento di underflow del contatore. Quindi riprende a contare da 0. [Nota che rispetto alla modalita conteggio in su descritta sopra quando è al centro l'evento in cima scatta prima -1... ok.](#)

Center-aligned mode is active when the CMS bits in TIMx_CR1 register are not equal to '00'. The Output compare interrupt flag of channels configured in output is set when: the counter counts down (Center aligned mode 1, CMS = "01"), the counter counts up (Center aligned mode 2, CMS = "10") the counter counts up and down (Center aligned mode 3, CMS = "11"). [In pratica è abbastanza difficile capire cosa sta dicendo, volutamente ho lasciato originale inglese ! si capisce di piu vedendo avanti dove documenta CMS](#)

Bit **CMS[1:0]**: Selezione della modalità allineata al centro

00: Modalità allineata ai bordi. Il contatore conta verso l'alto o verso il basso a seconda del bit di direzione (DIR). Cioè quello che ha spiegato sopra conteggio in su o conteggio in giù, cioè non allineato al centro !

01: Center-aligned mode 1. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting down.
10: Center-aligned mode 2. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set only when the counter is counting up.
11: Center-aligned mode 3. The counter counts up and down alternatively. Output compare interrupt flags of channels configured in output (CCxS=00 in TIMx_CCMRx register) are set both when the counter is counting up or down.

E comunque questo CMS sta dicendo quando devono scattare gli interrupt, vedi avanti figura 85 !

Nota: Non è consentito passare dalla modalità allineata ai bordi alla modalità allineata al centro mentre il contatore è abilitato (CEN=1)

In questa modalità non è possibile scrivere il bit di direzione DIR nel registro TIMx_CR1. Viene aggiornato dall'hardware e fornisce la direzione corrente del contatore.

L'evento di aggiornamento può essere generato ad ogni overflow del contatore e ad ogni underflow del contatore oppure impostando il bit UG nel registro TIMx_EGR (tramite software o utilizzando il controller della modalità slave). In questo caso, il contatore ricomincia a contare da 0, così come il contatore del prescaler. *Se si usa UG il contatore riparte da zero, vabbè fa così...*

L'evento di aggiornamento UEV può essere disabilitato dal software impostando il bit UDIS nel registro TIMx_CR1. In questo modo si evita di aggiornare i registri shadow durante la scrittura di nuovi valori nei registri di precaricamento. Quindi non si verifica alcun evento di aggiornamento fino a quando il bit UDIS non è stato scritto su 0. Tuttavia, il contatore continua a contare su e giù, in base al valore di ricarica automatica corrente.

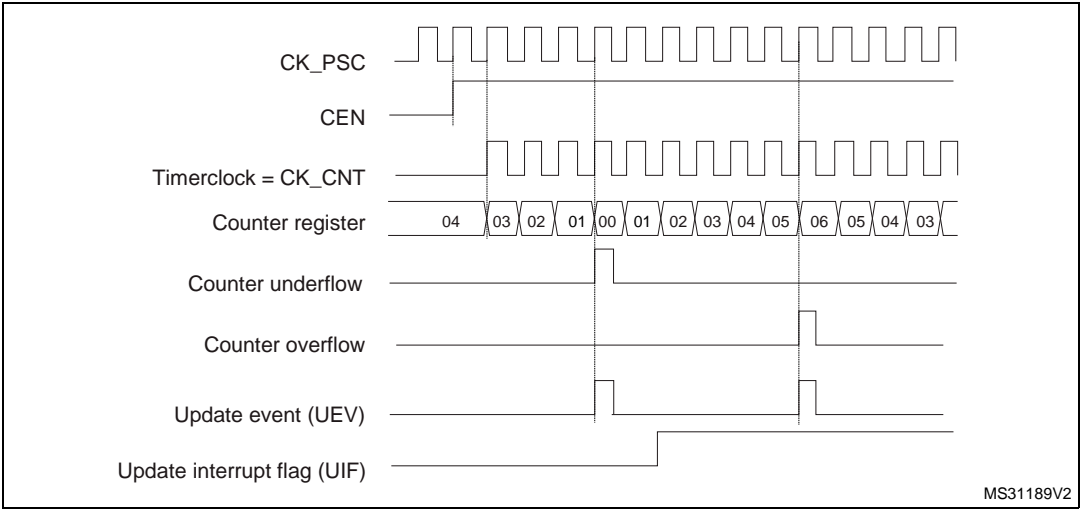
Inoltre, se è impostato il bit URS (selezione della richiesta di aggiornamento) nel registro TIMx_CR1, impostando il bit UG genera un evento di aggiornamento UEV ma senza impostare il flag UIF (quindi non viene inviata alcuna richiesta di interrupt o DMA). In questo modo si evita di generare interrupt di aggiornamento e di acquisizione quando si cancella il contatore sull'evento di acquisizione.

Quando si verifica un evento di aggiornamento, tutti i registri vengono aggiornati e viene impostato il flag di aggiornamento (bit UIF nel registro TIMx_SR):

- Il contatore di ripetizioni viene ricaricato con il contenuto del registro TIMx_RCR
- Il buffer del prescaler viene ricaricato con il valore di preload (contenuto del registro TIMx_PSC)
- Il registro attivo di ricarica automatica viene aggiornato con il valore di precaricamento (contenuto del registro TIMx_ARR). Si noti che se l'origine dell'aggiornamento è un overflow del contatore, il caricamento automatico viene aggiornato prima che il contatore venga ricaricato, in modo che il periodo successivo sia quello previsto (il contatore viene caricato con il nuovo valore).

Le figure seguenti mostrano alcuni esempi del comportamento del contatore per diverse frequenze di clock.

Figure 66. Counter timing diagram, internal clock divided by 1, TIMx_ARR = 0x6



Qui sopra non mi torna UIF penso sia un errore vedi anche foto seguenti dove è giusto penso.

1. Here, center-aligned mode 1 is used (for more details refer to [Section 14.4: TIM1 and TIM8 registers](#)).

Figure 67. Counter timing diagram, internal clock divided by 2

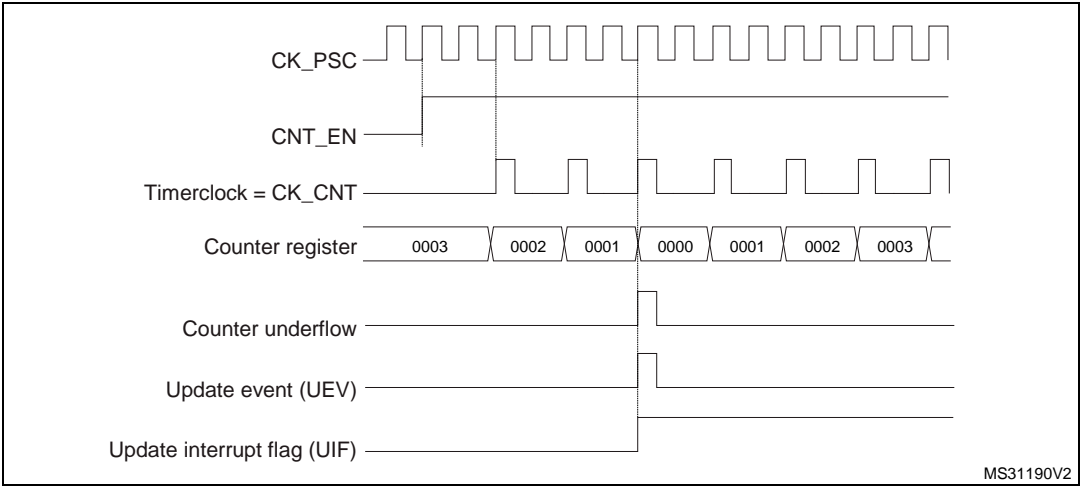
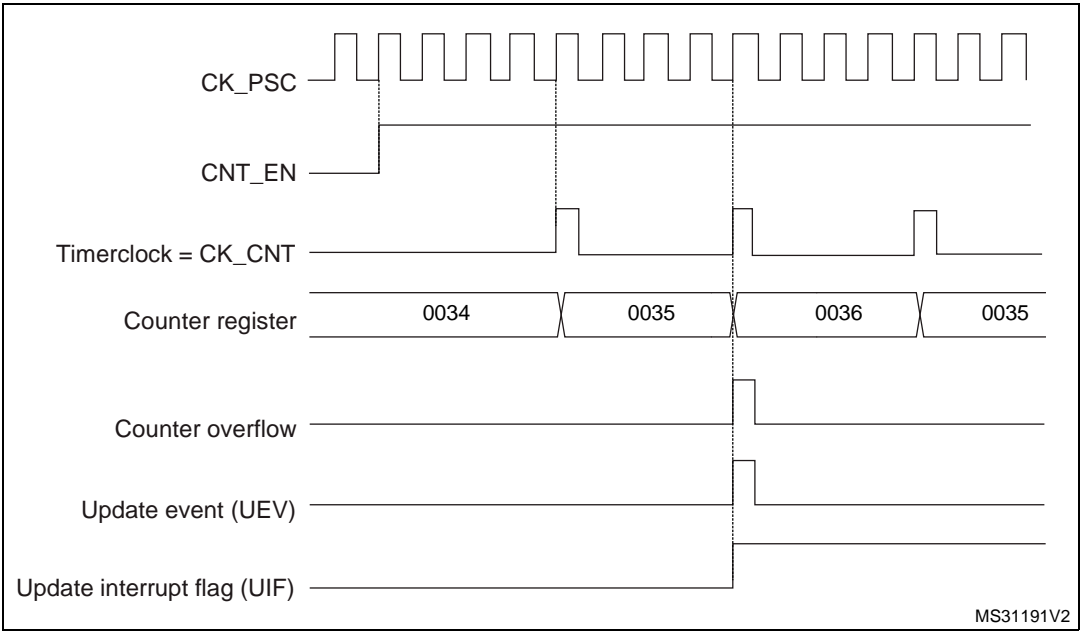


Figure 68. Counter timing diagram, internal clock divided by 4, TIMx_ARR=0x36



1. Center-aligned mode 2 or 3 is used with an UIF on overflow.

Figure 69. Counter timing diagram, internal clock divided by N

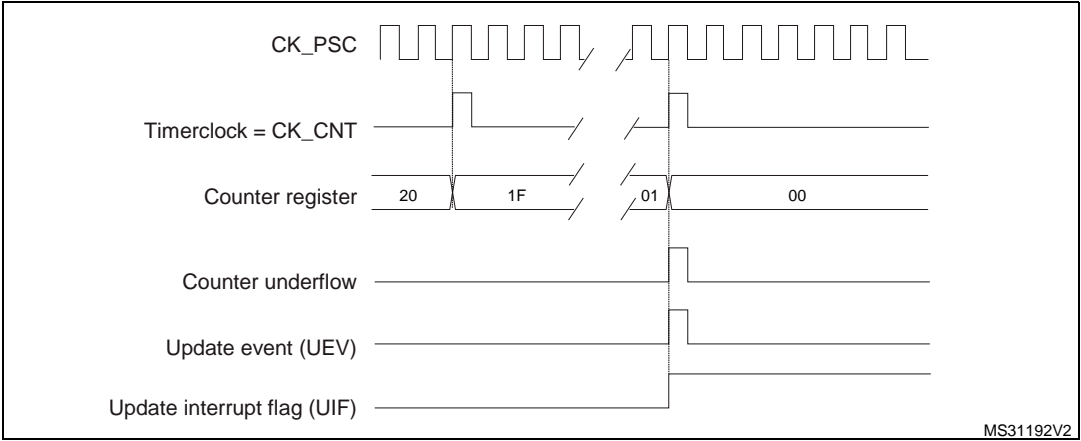


Figure 70. Counter timing diagram, update event with ARPE=1 (counter underflow)

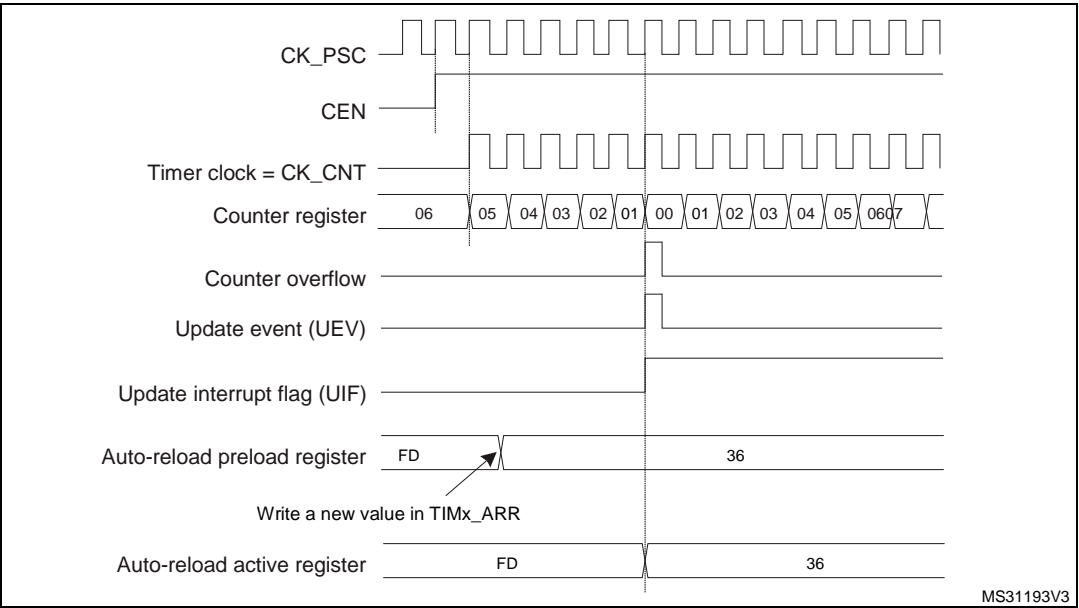
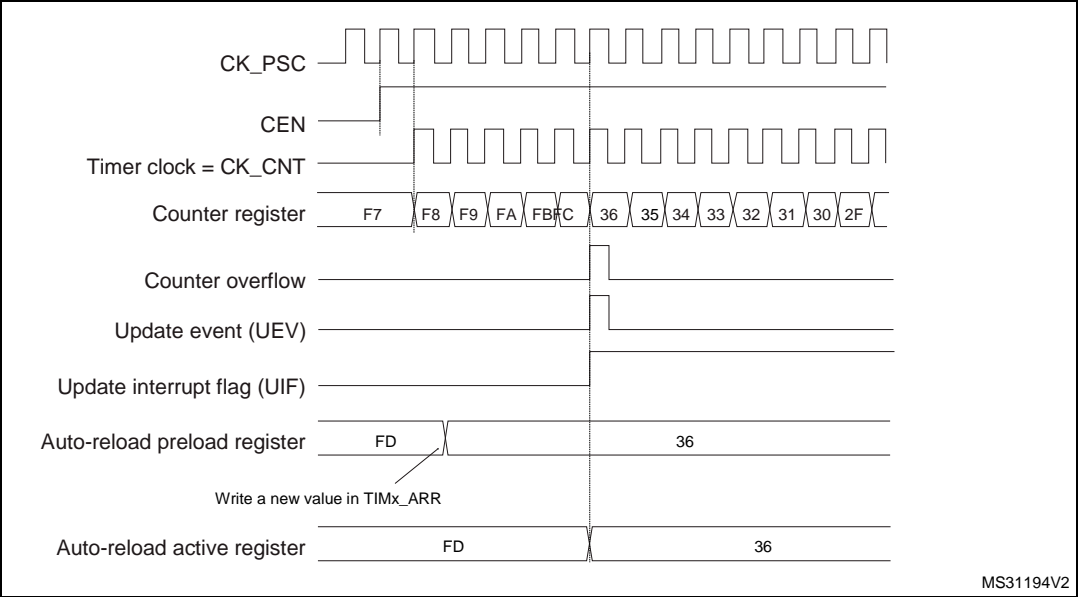


Figure 71. Counter timing diagram, Update event with ARPE=1 (counter overflow)



Le due qua sopra importanti.

14.3.3 Contatore di ripetizioni Repetition counter RCR.

Sezione 14.3.1: L'unità di base temporale descrive il modo in cui viene generato l'evento di aggiornamento (UEV) rispetto agli overflow/underflow del contatore. In realtà viene generato solo quando il contatore di ripetizioni ha raggiunto lo zero. Questo può essere utile quando si generano segnali PWM.

Ciò significa che i dati vengono trasferiti dai registri di precaricamento ai registri shadow (TIMx_ARR registro di ricaricamento automatico, TIMx_PSC registro di prescaler, ma anche TIMx_CCRx registri di acquisizione/confronto in modalità di confronto) every N+1 counter overflows or underflows, where N is the value in the TIMx_RCR repetition counter register.

Il contatore di ripetizioni è decrementato:

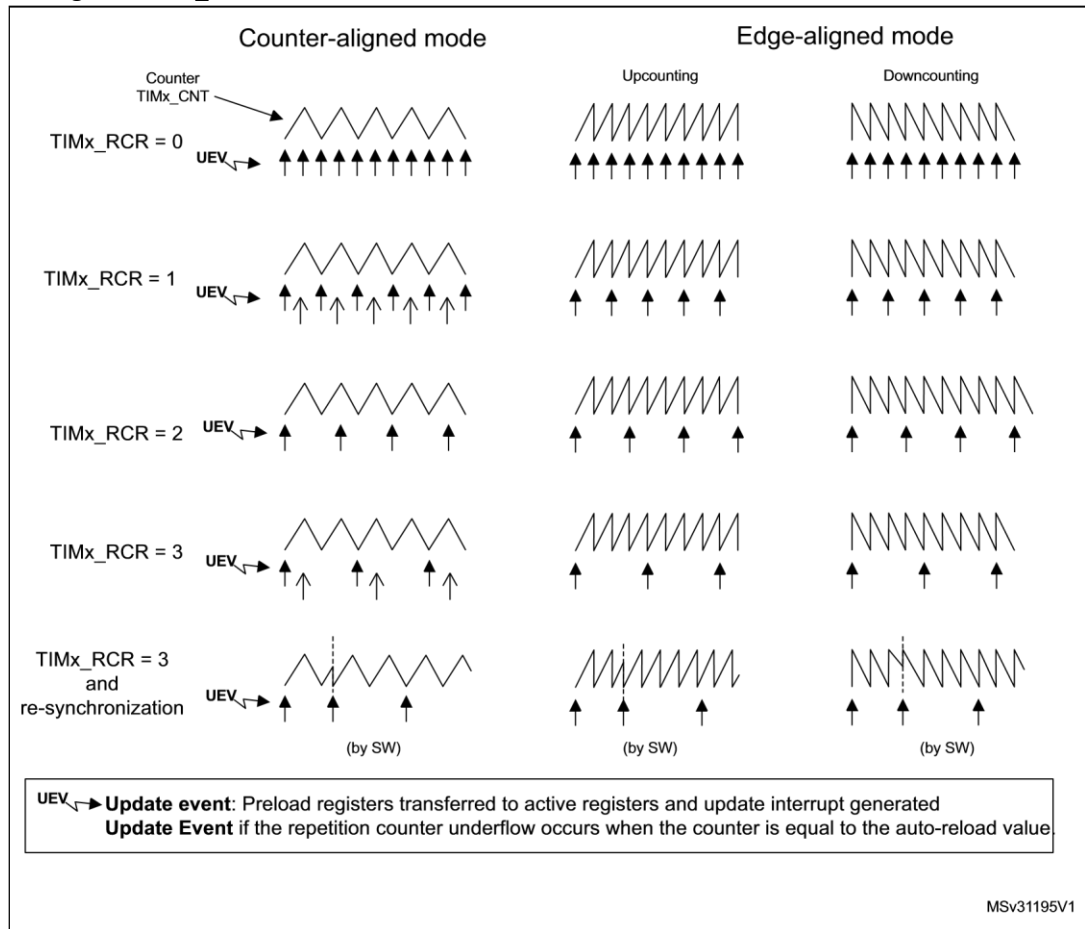
- At each counter overflow in upcounting mode,
- At each counter underflow in downcounting mode,
- At each counter overflow and at each counter underflow in center-aligned mode.
Although this limits the maximum number of repetition to 128 PWM cycles, it makes it possible to update the duty cycle twice per PWM period. When refreshing compare registers only once per PWM period in center-aligned mode, maximum resolution is $2xT_{ck}$, due to the symmetry of the pattern.

Il contatore di ripetizioni è di tipo autoricaricante; la frequenza di ripetizione viene mantenuta come definito dal valore del registro TIMx_RCR (fare riferimento alla [Figura 72](#)). Quando l'evento di aggiornamento viene generato dal software (impostando il bit UG nel registro TIMx_EGR) o dall'hardware tramite il controllore in modalità slave, si verifica immediatamente qualunque sia il valore del contatore di ripetizioni e il contatore di ripetizioni viene ricaricato con il contenuto del registro TIMx_RCR.

In modalità allineata al centro, per i valori dispari di RCR, l'evento di aggiornamento si verifica sull'overflow o sull'underflow a seconda di quando è stato scritto il registro RCR e quando è stato avviato il contatore. Se l'RCR è stato scritto prima di avviare il contatore, l'UEV si verifica sull'overflow. Se l'RCR è stato scritto dopo l'avvio del contatore, l'UEV si verifica sull'underflow. Ad esempio, per RCR = 3, l'UEV viene generato a ogni 4° evento di overflow o underflow a seconda di quando è stato scritto RCR.

Che casino, vedi figura ma pure quella non aiuta molto, comunque penso di tenere TCR a zero ! nel disegno freccia nera scura oppure freccia nera chiara, scattano o le scure o le chiare a secondo di quando è stato scritto RCR. Va interpretato così il disegno !

Figura 72. Esempi di frequenza di aggiornamento a seconda della modalità e delle impostazioni del registro TIMx_RCR



14.3.4 Selezione dell'orologio clock.

Il contatore può essere fornito dalle seguenti sorgenti di clock:

- Orologio interno (CK_INT)

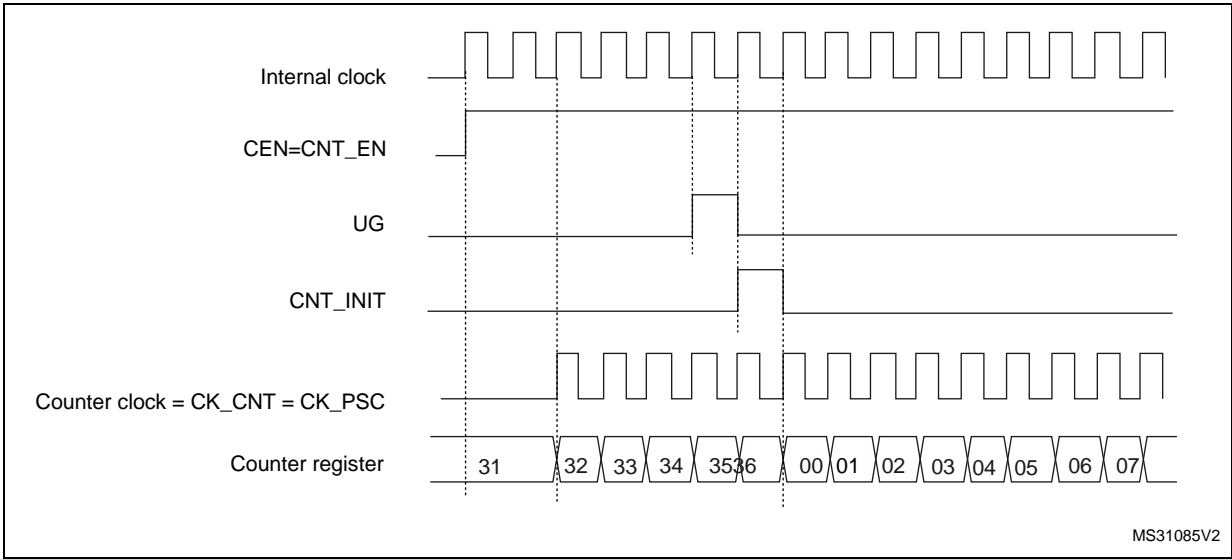
Faccio conto di usare solo CK_INT e non altre fonti esterne ecc ecc. Non tradotto interi pezzi, rimossi...

Sorgente di clock interna (CK_INT)

Se il controllore in modalità slave è disabilitato (SMS=000), i bit CEN, DIR (nel registro TIMx_CR1) e UG (nel registro TIMx_EGR) sono bit di controllo effettivi e possono essere modificati solo dal software (ad eccezione di UG che rimane cancellato automaticamente). Non appena il bit CEN viene scritto su 1, il prescaler viene sincronizzato dal clock interno CK_INT.

La Figura 73 mostra il comportamento del circuito di controllo e dell'upcounter in modalità normale, senza prescaler.

Figure 73. Control circuit in normal mode, internal clock divided by 1



14.3.5 Capture/compare channels

Capture lo salto, rimosso, non tradotto, ma lascio le figure da 79 a 81 perche vi sono parti usate anche nel pwm.

Figure 79. Capture/compare channel 1 main circuit

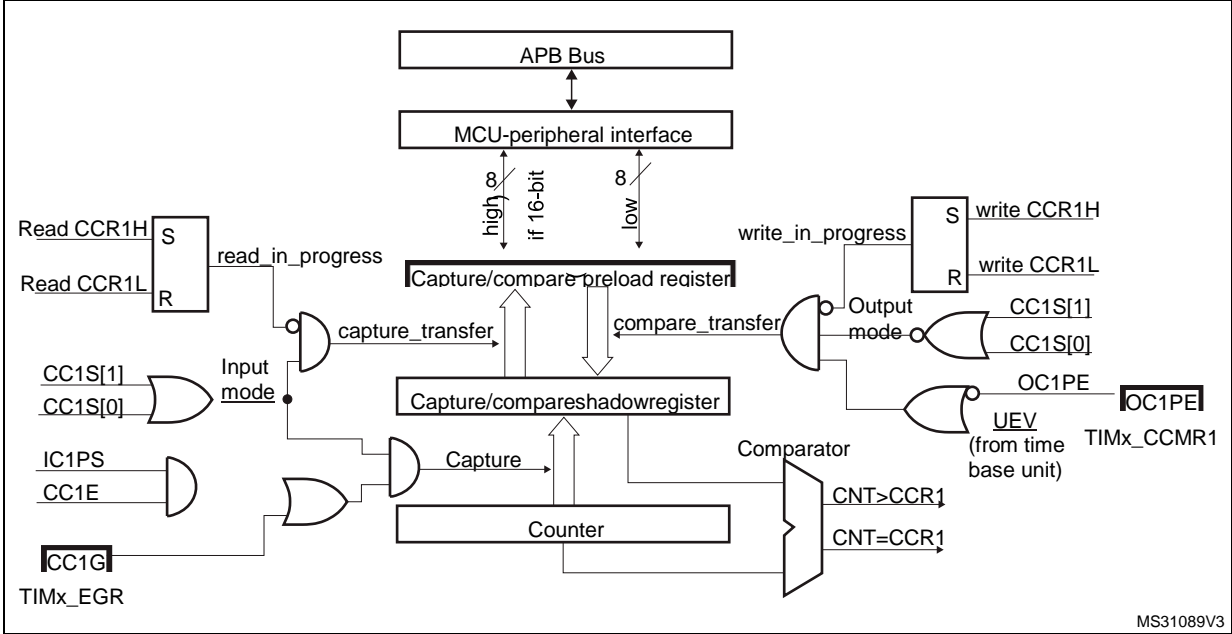


Figure 80. Output stage of capture/compare channel (channel 1 to 3)

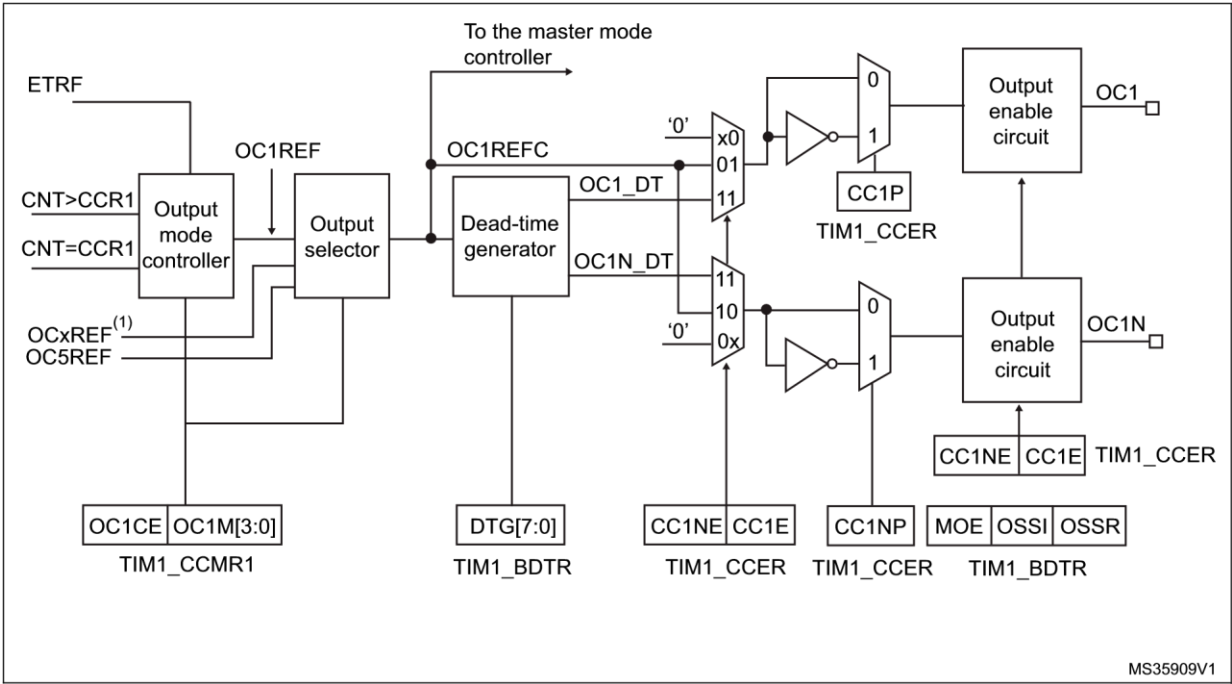
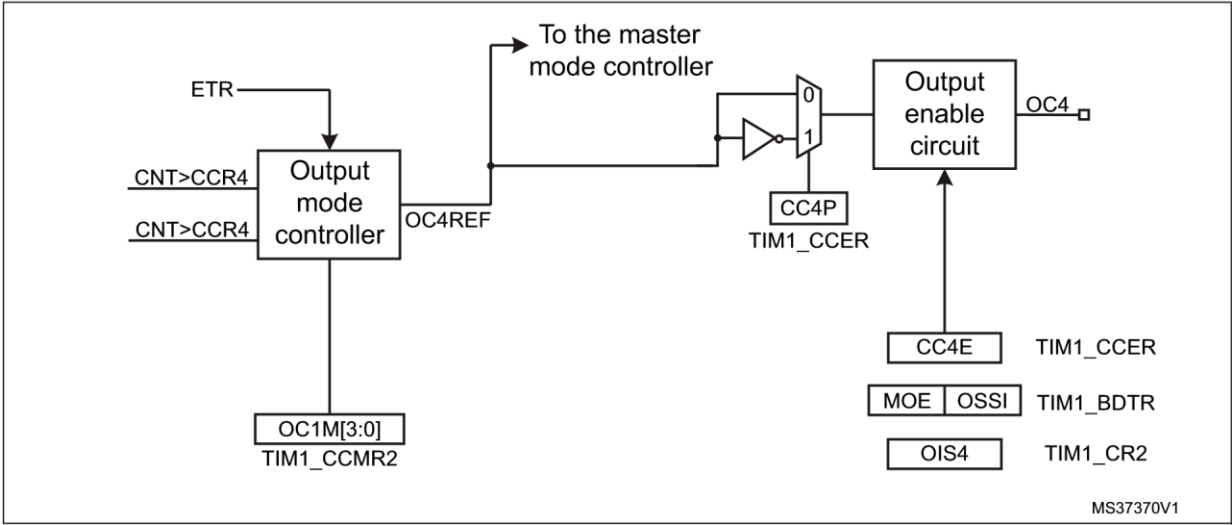


Figure 81. Output stage of capture/compare channel (channel 4)



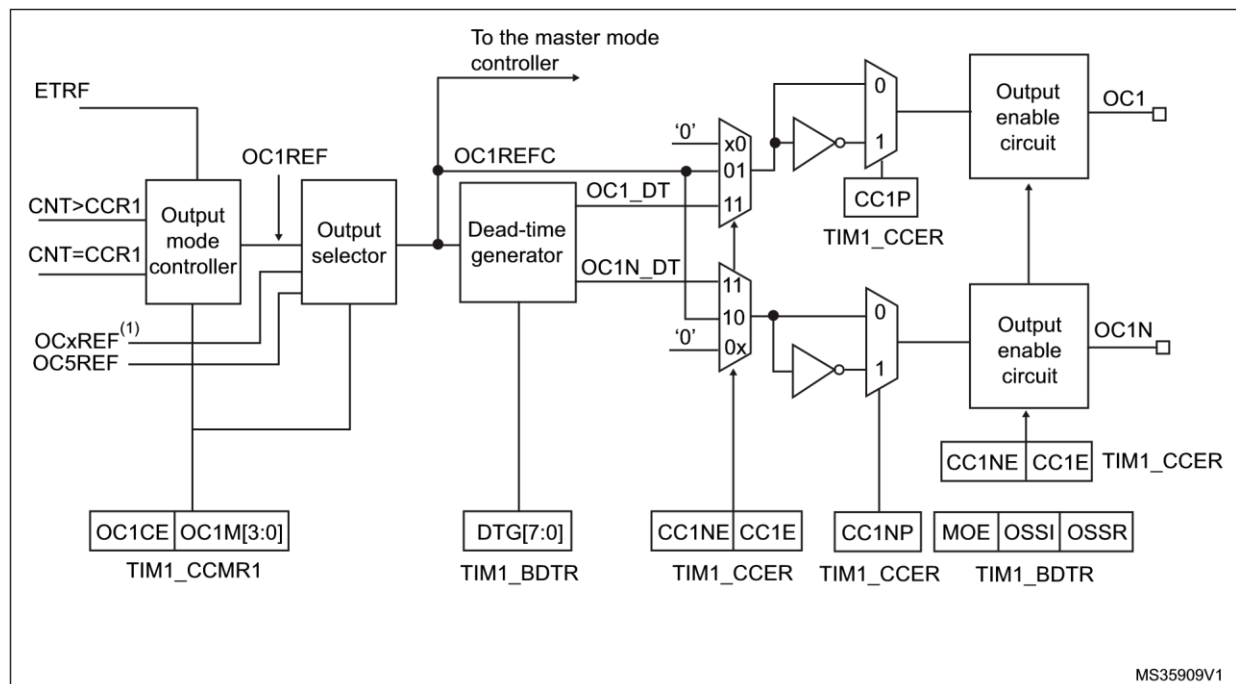
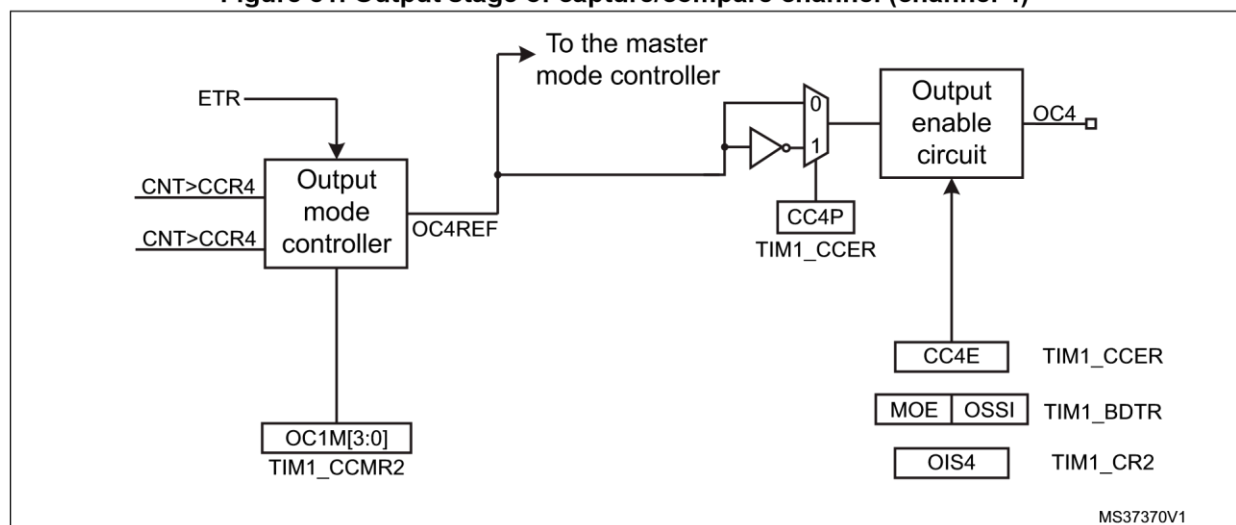


Figure 81. Output stage of capture/compare channel (channel 4)



Il canale 4 è piu semplificato rispetto ai primi 3.

In modalità di confronto, il contenuto del registro di precaricamento viene copiato nel registro shadow che viene confrontato con il contatore.

14.3.6 Input capture mode

rimosso, non tradotto nulla del 14.3.6

14.3.7 PWM input mode

Questa modalità è un caso particolare di modalità di input capture. **rimosso, non tradotto nulla**

14.3.8 Forced output mode

Questo lo documento perchè penso lo usero come modo per forzare i segnali in uscita su un valore prestabilito. (LOW direi...)

In output mode (bit CCxS = 00 nel registro TIMx_CCMRx), ogni output compare signal (OCxREF e quindi OCx/OCxN) può essere forzato a livello attivo o inattivo direttamente dal software.

Per forzare un segnale (OCxREF/OCx) al suo livello attivo, l'utente deve solo scrivere 101 nei bit OCxM nel registro TIMx_CCMRx corrispondente. Quindi OCxREF è forzato alto (OCxREF è sempre attivo alto) e OCx ottiene un valore opposto al bit di polarità CCxP.

Ad esempio: CCxP=0 (OCx attivo alto) => OCx è forzato a un livello alto.

Il segnale OCxREF può essere forzato verso il basso scrivendo i bit OCxM a 100 nel registro TIMx_CCMRx.

In ogni caso, il confronto tra il registro shadow TIMx_CCRx e il contatore viene comunque eseguito e consente di impostare il flag. Le richieste di interrupt e DMA possono essere inviate di conseguenza. Questa operazione è descritta nella sezione relativa alla modalità di confronto dell'output riportata di seguito.

OC1M: Output compare 1 mode

These bits define the behavior of the output reference signal OC1REF from which OC1 and OC1N are derived. OC1REF is active high whereas OC1 and OC1N active level depends on CC1P and CC1NP bits.

000: Frozen - The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on the outputs.(this mode is used to generate a timing base).

001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1).

010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1).

011: Toggle - OC1REF toggles when TIMx_CNT=TIMx_CCR1.

100: Force inactive level - OC1REF is forced low.

101: Force active level - OC1REF is forced high.

110: PWM mode 1 - In upcounting, channel 1 is active as long as TIMx_CNT<TIMx_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF='0') as long as TIMx_CNT>TIMx_CCR1 else active (OC1REF='1').

111: PWM mode 2 - In upcounting, channel 1 is inactive as long as TIMx_CNT<TIMx_CCR1 else active. In downcounting, channel 1 is active as long as TIMx_CNT>TIMx_CCR1 else inactive.

Note: 1: These bits can not be modified as long as LOCK level 3 has been programmed (LOCK bits in TIMx_BDTR register) and CC1S='00' (the channel is configured in output).

2: In PWM mode 1 or 2, the OCREF level changes only when the result of the comparison changes or when the output compare mode switches from "frozen" mode to "PWM" mode.

14.3.9 Output compare mode **documento, traduco:**

Questa funzione viene utilizzata per generare una forma d'onda di uscita o per indicare quando è trascorso un periodo di tempo.

Quando viene trovata una corrispondenza tra il registro di acquisizione/confronto e il contatore, succede questo:

- Assigns the corresponding output pin to a programmable value defined by the output compare mode (OCxM bits in the TIMx_CCMRx register) and the output polarity (CCxP bit in the TIMx_CCER register). The output pin can keep its level (OCxM=000), be set active (OCxM=001), be set inactive (OCxM=010) or can toggle (OCxM=011) on match.
- Imposta un flag nel registro di stato dell'interrupt (bit CCxIF nel registro TIMx_SR).
- Genera un interrupt se è impostata la maschera di interrupt corrispondente (bit CCxIE nel registro TIMx_DIER).
- Invia una richiesta DMA se è impostato il bit di abilitazione corrispondente (bit CCxDE nel registro TIMx_DIER, bit CCDS nel registro TIMx_CR2 per la selezione della richiesta DMA).

I registri TIMx_CCRx possono essere programmati con o senza preload utilizzando il bit **OCxPE** nel registro TIMx_CCMRx.

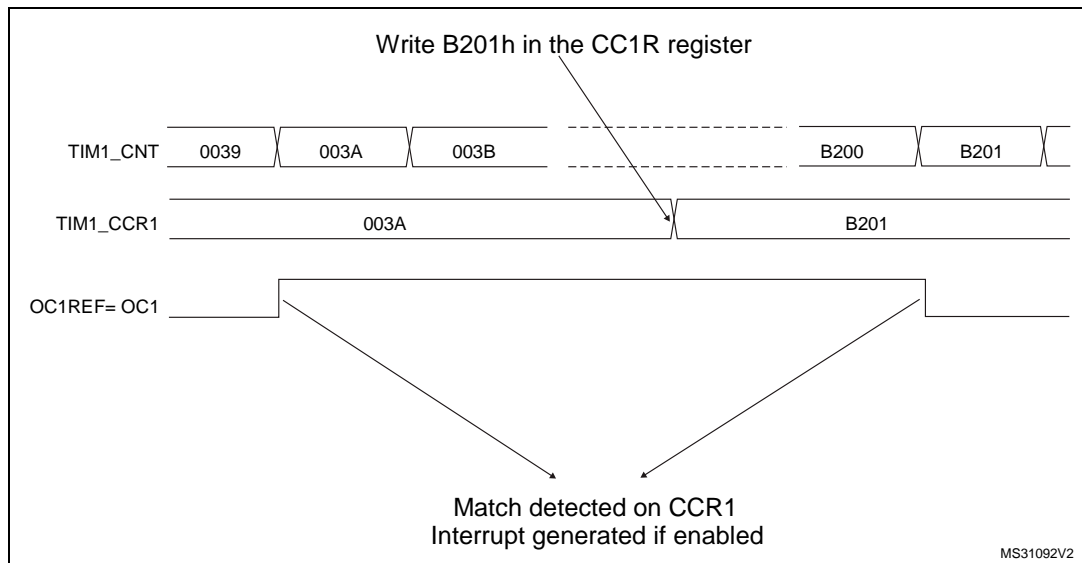
In output compare mode, the update event UEV non ha alcun effetto sull'output OCxREF e OCx. La risoluzione temporale è un conteggio del contatore. La modalità di confronto dell'uscita può essere utilizzata anche per emettere un singolo impulso (in modalità One Pulse).

Procedimento:

1. Selezionare il clock del contatore (interno, esterno, prescaler).
2. Scrivere i dati desiderati nei registri TIMx_ARR e TIMx_CCRx.
3. Impostare il bit CCxIE se deve essere generata una richiesta di interrupt.
4. Selezionare la modalità di uscita. Per esempio:
 - Setta OCxM = 011 per attivare o disattivare il pin di uscita OCx quando CNT corrisponde a CCRx **cioè per fare il toggle del pin.**
 - Setta OCxPE = 0 per disabilitare il preload
 - Setta CCxP = 0 per selezionare l'alta polarità attiva
 - Setta CCxE = 1 per abilitare l'uscita
5. Abilitare il contatore impostando il bit CEN nel registro TIMx_CR1.

Il registro TIMx_CCRx può essere aggiornato in qualsiasi momento dal software per controllare la forma d'onda in uscita, a condizione che il registro di precarico non sia abilitato (OCxPE='0', altrimenti TIMx_CCRx registro shadow viene aggiornato solo al successivo evento di aggiornamento UEV). Un esempio è riportato nella [Figura 83](#).

Figura 83. Modalità di confronto dell'uscita, attiva OC1.



14.3.10 PWM mode

La modalità Pulse Width Modulation consente di generare un segnale con una frequenza determinata dal valore del registro TIMx_ARR e un "duty cycle" determinato dal valore del registro TIMx_CCRx.

La modalità PWM può essere selezionata in modo indipendente su ciascun canale (un PWM per uscita OCx) scrivendo '110' (modalità PWM 1) o '111' (modalità PWM 2) nei bit OCxM nel registro TIMx_CCMRx. Il registro di precarico corrispondente deve essere abilitato impostando il bit OCxPE nel registro TIMx_CCMRx, ed eventualmente auto-reload register (in modalità upcounting o allineato al centro) impostando il bit ARPE nel registro TIMx_CR1.

Poiché i registri di precaricamento vengono trasferiti ai registri shadow solo quando si verifica un evento di aggiornamento, prima di avviare il contatore, l'utente deve inizializzare tutti i registri impostando il bit UG nel registro TIMx_EGR. *Si ma rileggi UG ed anche UDIS ed anche CEN.*

La polarità OCx è programmabile via software utilizzando il bit CCxP nel registro TIMx_CCER. Può essere programmato come attivo alto o attivo basso. L'uscita OCx è abilitata da una combinazione dei bit CCxE, CCxNE, MOE, OSSI e OSSR (registri TIMx_CCER e TIMx_BDTR). Per ulteriori informazioni, fare riferimento alla descrizione del registro TIMx_CCER. *Caspita qui dove ci sarebbe da documentare bene dice vattelo a vedere !*

CCxE e CCxNE sono gli output enable, 1 per enable.

Bit **MOE**: Main output enable

This bit is cleared asynchronously by hardware as soon as the break input is active. It is set by software or automatically depending on the AOE bit. It is acting only on the channels which are configured in output.

0: OC and OCN outputs are disabled or forced to idle state.

1: OC and OCN outputs are enabled if their respective enable bits are set (CCxE, CCxNE in TIMx_CCER register).

See OC/OCN enable description for more details (Section 14.4.9: TIM1 and TIM8 capture/compare enable register (TIMx_CCER)).

Bit **AOE**: Automatic output enable

0: MOE can be set only by software

1: MOE can be set by software or automatically at the next update event (if the break input is not be active)

Note: This bit can not be modified as long as LOCK level 1 has been programmed (LOCK bits in TIMx_BDTR register).

Bit 10 OSSR: Off-state selection for Idle mode

This bit is used when MOE=0 on channels configured as outputs.

See OC/OCN enable description for more details (Section 14.4.9: TIM1 and TIM8 capture/compare enable register (TIMx_CCER)).

0: When inactive, OC/OCN outputs are disabled (OC/OCN enable output signal=0).

1: When inactive, OC/OCN outputs are forced first with their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN enable output signal=1

Note: This bit can not be modified as soon as the LOCK level 2 has been programmed (LOCK bits in TIMx_BDTR register).

POCO CHIARO OSSR...

In modalità PWM (1 o 2), TIMx_CNT e TIMx_CCRx vengono sempre confrontati per determinare se $TIMx_CCRx \leq TIMx_CNT$ o $TIMx_CNT \leq TIMx_CCRx$ (a seconda della direzione del contatore).

Il timer è in grado di generare PWM in modalità allineata ai bordi o in modalità allineata al centro a seconda dei bit CMS nel registro TIMx_CR1.

PWM edge-aligned mode

Upcounting configuration

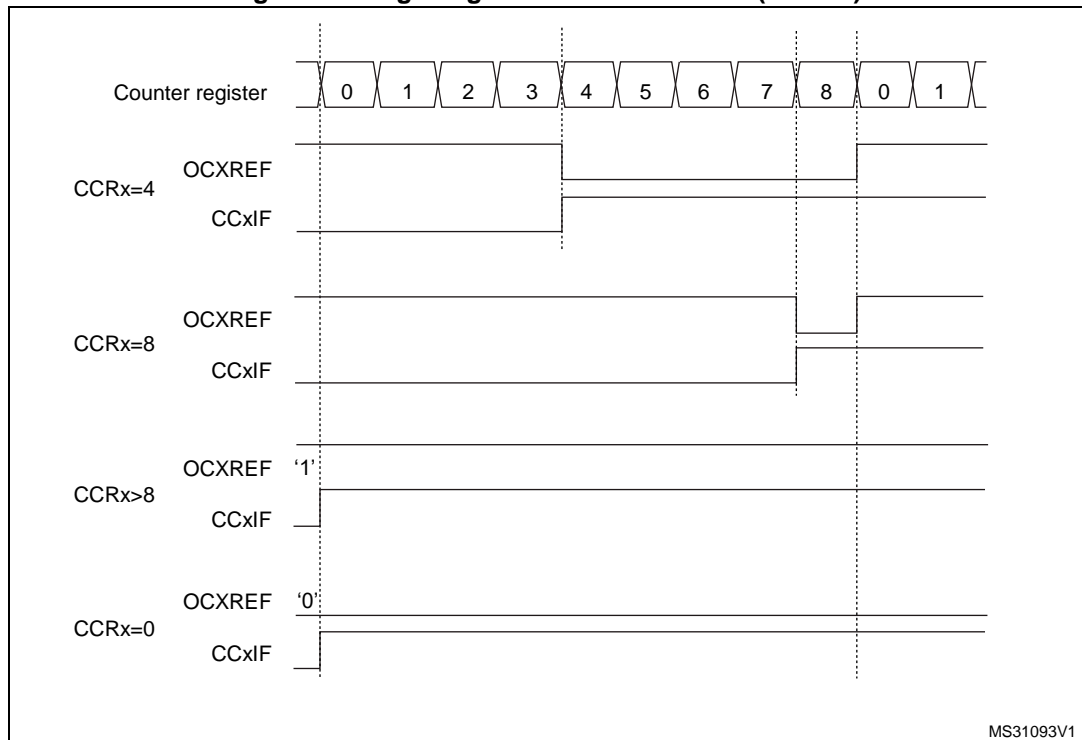
L'upcounting è attivo quando il bit DIR nel registro TIMx_CR1 è basso. Fare riferimento alla [modalità Upcounting](#).

Nell'esempio seguente, consideriamo la modalità PWM 1. Il segnale PWM di riferimento OCxREF è alto fintanto che $TIMx_CNT < TIMx_CCRx$ else diventa basso.

Se il valore di confronto in TIMx_CCRx è maggiore del valore di ricarica automatica (in TIMx_ARR), OCxREF viene mantenuto su '1'.

Se TIMx_CCRx è 0, OCxRef viene mantenuto su '0'. [La Figura 84](#) mostra alcune forme d'onda PWM allineate ai bordi in un esempio in cui TIMx_ARR=8.

Figure 84. Edge-aligned PWM waveforms (ARR=8)



Downcounting configuration

Il downcounting è attivo quando TIMx_CR1 bit DIR nel registro è alto. Fare riferimento alla [modalità di conteggio ridotto](#)

In modalità PWM 1, il segnale di riferimento OCxRef è basso finché $TIMx_CNT > TIMx_CCRx$ else diventa alto.

Se il valore di confronto in TIMx_CCRx è maggiore del valore di ricarica automatica in TIMx_ARR, OCxREF viene mantenuto su '1'.

Lo 0% PWM non è possibile in questa modalità.

PWM center-aligned mode

La modalità allineata al centro è attiva quando i bit CMS nel registro TIMx_CR1 sono diversi da '00' (tutte le configurazioni rimanenti hanno lo stesso effetto sui segnali OCxRef/Ocx)

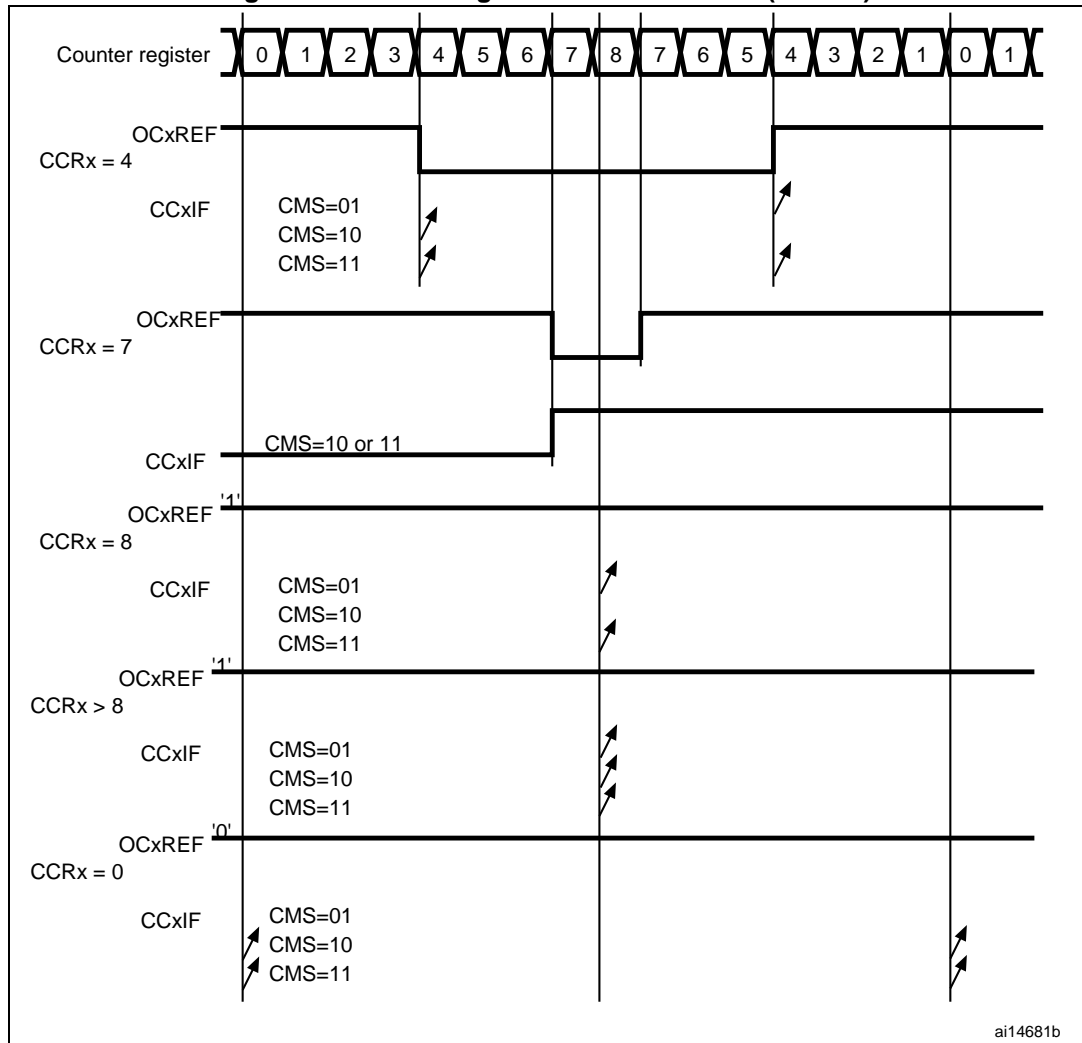
The compare flag is set when the counter counts up, when it counts down or both when it counts up and down depending on the CMS bits configuration.

Il bit di direzione (DIR) nel registro TIMx_CR1 viene aggiornato dall'hardware e non deve essere modificato dal software. Fare riferimento alla [modalità Allineamento al centro \(conteggio su/giù\)](#).

La Figura 85 mostra alcune forme d'onda PWM allineate al centro in un esempio in cui:

- TIMx_ARR=8,
- La modalità PWM è la modalità PWM 1,
- The flag is set when the counter counts down corresponding to the center-aligned mode 1 selected for CMS=01 in TIMx_CR1 register.

Figure 85. Center-aligned PWM waveforms (ARR=8)



Insomma sti eventi di update non sarà banale capirli/usarli, vedi come scattano!

Suggerimenti sull'utilizzo della modalità allineata al centro:

- Quando si avvia in modalità allineata al centro, viene utilizzata la configurazione up-down corrente. Significa che il contatore conta verso l'alto o verso il basso a seconda del valore scritto nel bit DIR nel registro TIMx_CR1. Inoltre, i bit DIR e CMS non devono essere modificati contemporaneamente dal software.
- La scrittura sul contatore durante l'esecuzione in modalità allineata al centro non è consigliata in quanto può portare a risultati imprevisti. Soprattutto:
 - La direzione non viene aggiornata se l'utente scrive un valore nel contatore maggiore del valore di ricarica automatica ($TIMx_CNT > TIMx_ARR$). Ad esempio, se il contatore stava contando verso l'alto, continuerà a contare.
 - La direzione viene aggiornata se l'utente scrive 0 o scrive il valore TIMx_ARR nel contatore, ma non viene generato alcun evento di aggiornamento UEV.
- Il modo più sicuro per utilizzare la modalità allineata al centro consiste nel generare un aggiornamento tramite software (impostando il bit UG nel registro TIMx_EGR) appena prima di avviare il contatore e non scrivere il contatore mentre è in esecuzione.

14.3.11 Complementary outputs and dead-time insertion

I temporizzatori a controllo avanzato (TIM1 e TIM8) possono emettere due segnali complementari e gestire gli istanti di spegnimento e accensione delle uscite.

Questo tempo è generalmente noto come tempo morto e deve essere regolato in base ai dispositivi collegati alle uscite e alle loro caratteristiche (ritardi intrinseci dei traslatori di livello, ritardi dovuti a interruttori di alimentazione...)

L'utente può selezionare la polarità delle uscite (uscita principale OCx o OCxN complementare) indipendentemente per ciascuna uscita. Questa operazione viene eseguita scrivendo nei bit CCxP e CCxNP nel registro TIMx_CCER.

I segnali complementari OCx e OCxN sono attivati da una combinazione di diversi bit di controllo: i bit CCxE e CCxNE nel registro TIMx_CCER e i bit MOE, OISx, OISxN, OSSI e OSSR nei registri TIMx_BDTR e TIMx_CR2. Fare riferimento alla [Tabella 83](#) per maggiori dettagli. In particolare, il tempo morto viene attivato quando si passa allo stato IDLE (MOE che scende a 0).

Dead-time insertion viene abilitato impostando entrambi i bit CCxE e CCxNE e il bit MOE se the break circuit is present. I bit DTG[7:0] del registro TIMx_BDTR vengono utilizzati per controllare la generazione del tempo morto per tutti i canali. Da una forma d'onda di riferimento OCxREF, genera 2 uscite OCx e OCxN. Se OCx e OCxN sono attivi su un valore alto:

- Il segnale di uscita OCx è lo stesso del segnale di riferimento, ad eccezione del fronte di salita, che è ritardato rispetto al fronte di salita di riferimento.
- Il segnale di uscita OCxN è l'opposto del segnale di riferimento, ad eccezione del fronte di salita, che è ritardato rispetto al fronte di discesa di riferimento.

Se il ritardo è maggiore dell'ampiezza dell'uscita attiva (OCx o OCxN), l'impulso corrispondente non viene generato.

Le figure seguenti mostrano le relazioni tra i segnali di uscita del generatore di tempo morto e il segnale di riferimento OCxREF. (supponiamo che CCxP=0, CCxNP=0, MOE=1, CCxE=1 e CCxNE=1 in questi esempi)

Figure 86. Complementary output with dead-time insertion.

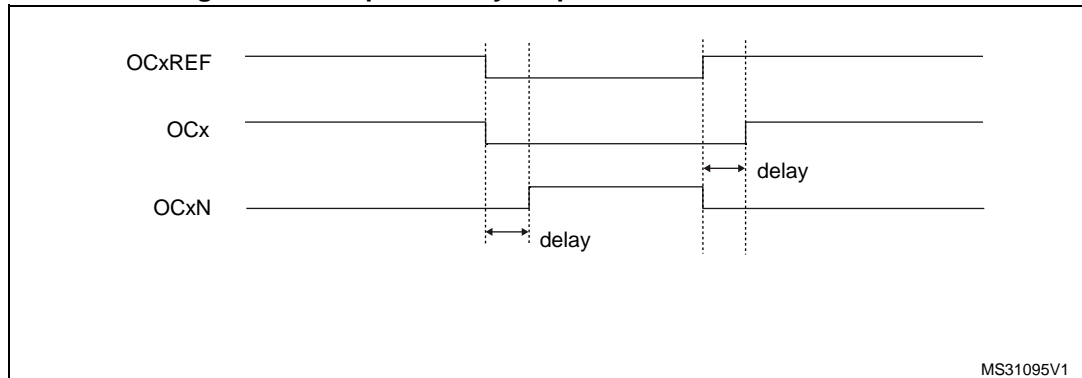


Figure 87. Dead-time waveforms with delay greater than the negative pulse.

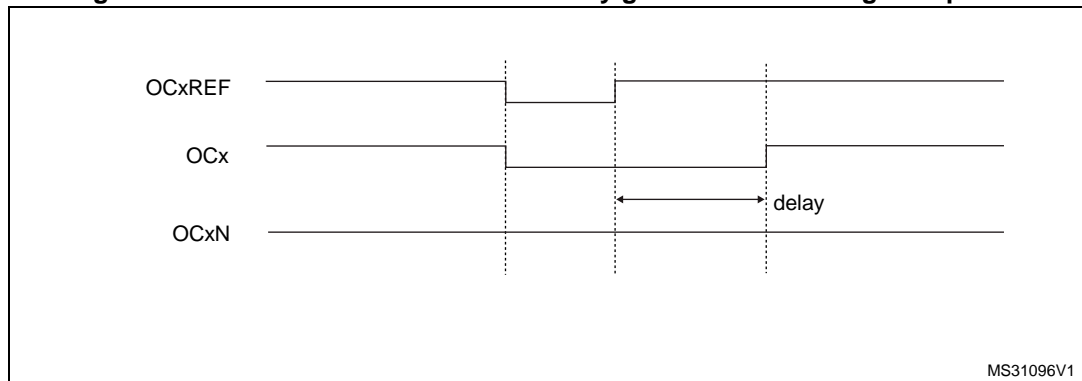
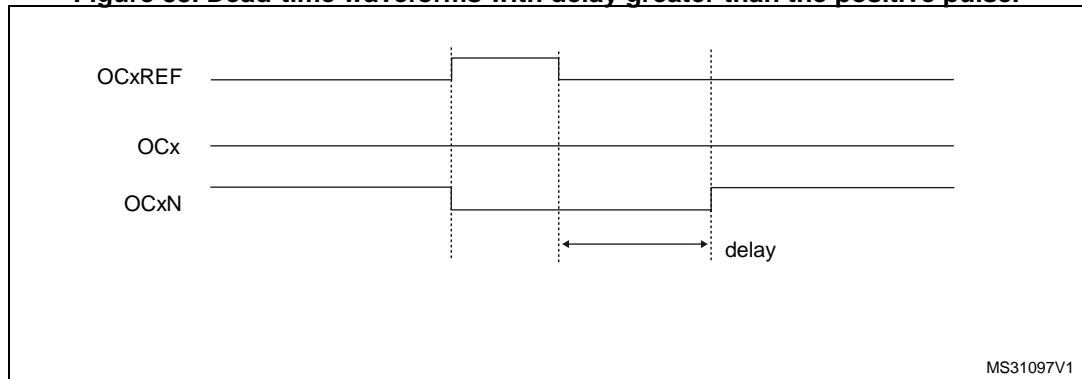


Figure 88. Dead-time waveforms with delay greater than the positive pulse.



Il ritardo del tempo morto è lo stesso per ciascuno dei canali ed è programmabile con i bit DTG nel registro TIMx_BDTR. Fare riferimento alla [Sezione 14.4.18: Registro dei tempi morti e delle interruzioni \(TIMx_BDTR\) TIM1 e TIM8](#) per il calcolo del ritardo. Un'altra bella complicazione ma ok si riesce a capire calcolare quanto farà di deadtime... attenzione che il clock, tempo è quello veloce prima del prescaler...

Reindirizzamento di OCxREF a OCx o OCxN

In output mode (forced, output compare or PWM), OCxREF può essere reindirizzato all'uscita OCx o all'uscita OCxN configurando i bit CCxE e CCxNE nel registro TIMx_CCER.

Ciò consente all'utente di inviare una forma d'onda specifica (come PWM o livello attivo statico) su un'uscita mentre il complementare rimane al suo livello inattivo. Altre possibilità sono quelle di avere entrambe le uscite a livello inattivo o entrambe le uscite attive e complementari con il tempo morto.

Nota: Quando solo OCxN è abilitato ($CCxE=0$, $CCxNE=1$), il pin non è complementare e diventa attivo non appena OCxREF è alto. Ad esempio, se $CCxNP=0$ allora $OCxN=OCxRef$. D'altra parte, quando sia OCx che OCxN sono abilitati ($CCxE=CCxNE=1$), OCx diventa attivo quando OCxREF è alto, mentre OCxN è complementato e diventa attivo quando OCxREF è basso.

14.3.12 Utilizzo della funzione break (da qui in poi non ho controllato bene la traduzione automatica è traduzione grezza da rivedere !)

Quando si utilizza la funzione di interruzione, i segnali di abilitazione dell'uscita e i livelli inattivi vengono modificati in base a bit di controllo aggiuntivi (bit MOE, OSSI e OSSR nel registro TIMx_BDTR, bit OISx e OISxN nel registro TIMx_CR2). In ogni caso, le uscite OCx e OCxN non possono essere impostate entrambe a livello attivo in un dato momento. Fare riferimento alla [Tabella 83](#) per maggiori dettagli.

L'origine dell'interruzione può essere il pin di ingresso dell'interruzione o un evento di errore dell'orologio, generato dal sistema di sicurezza dell'orologio (CSS), dal controller dell'orologio di ripristino. Per ulteriori informazioni sul sistema di sicurezza dell'orologio, fare riferimento alla [Sezione 7.2.7: Sistema di sicurezza dell'orologio \(CSS\)](#).

Quando si esce dal reset, il circuito di interruzione è disabilitato e il bit MOE è basso. L'utente può abilitare la funzione di interruzione impostando il bit BKE nel registro TIMx_BDTR. La polarità di ingresso dell'interruzione può essere selezionata configurando il bit BKP nello stesso registro. BKE e BKP possono essere modificati contemporaneamente. Quando i bit BKE e BKP vengono scritti, viene applicato un ritardo di 1 ciclo di clock APB prima che la scrittura sia effettiva. Di conseguenza, è necessario attendere 1 periodo di clock APB per rileggere correttamente il bit dopo l'operazione di scrittura.

Poiché il fronte di discesa MOE può essere asincrono, è stato inserito un circuito di risincronizzazione tra il segnale effettivo (che agisce sulle uscite) e il bit di controllo sincrono (accessibile nel registro TIMx_BDTR). Provoca alcuni ritardi tra i segnali asincroni e sincroni. In particolare, se MOE viene scritto a 1 mentre era basso, è necessario inserire un ritardo (istruzione fittizia) prima di leggerlo correttamente. Ciò è dovuto al fatto che l'utente scrive un segnale asincrono, ma legge un segnale sincrono.

Quando si verifica un'interruzione (livello selezionato nell'ingresso dell'interruzione):

- Il bit MOE viene cancellato in modo asincrono, mettendo le uscite in stato inattivo, inattivo o in stato di reset (selezionato dal bit OSI). Questa funzione funziona anche se l'oscillatore MCU è disattivato.
- Ogni canale di uscita viene pilotato con il livello programmato nel bit OISx nel registro TIMx_CR2 non appena MOE=0. Se OSSI=0 allora il timer rilascia l'uscita di abilitazione, altrimenti l'uscita di abilitazione rimane alta.
- Quando si utilizzano uscite complementari:

- Le uscite vengono prima messe in stato di reset inattivo (a seconda della polarità). Questa operazione viene eseguita in modo asincrono in modo che funzioni anche se non viene fornito alcun orologio al timer.
- Se il timer è ancora presente, il generatore di tempo morto viene riattivato per pilotare le uscite con il livello programmato nei bit OISx e OISxN dopo un tempo morto. Anche in questo caso, OCx e OCxN non possono essere portati insieme al loro livello attivo. Si noti che, a causa della risincronizzazione su MOE, la durata del tempo morto è un po' più lunga del solito (circa 2 cicli di clock `ck_tim`).
- Se `OSSI=0` allora il timer rilascia le uscite di abilitazione, altrimenti le uscite di abilitazione rimangono o diventano alte non appena uno dei bit `CCxE` o `CCxNE` è alto.
- Viene impostato il flag di stato di interruzione (bit `BIF` nel registro `TIMx_SR`). È possibile generare un interrupt se è impostato il bit `BIE` nel registro `TIMx_DIER`. È possibile inviare una richiesta DMA se è impostato il bit `BDE` nel registro `TIMx_DIER`.
- Se il bit `AOE` nel registro `TIMx_BDTR` è impostato, il bit `MOE` viene impostato automaticamente di nuovo all'evento di aggiornamento successivo `UEV`. Questo può essere utilizzato per eseguire una regolazione, ad esempio. In caso contrario, `MOE` rimane basso fino a quando non viene scritto di nuovo su '1'. In questo caso, può essere utilizzato per la sicurezza e l'ingresso di interruzione può essere collegato a un allarme da driver di potenza, sensori termici o qualsiasi componente di sicurezza.

Nota: Gli ingressi di interruzione agiscono in piano. Pertanto, il MOE non può essere impostato mentre l'ingresso di interruzione è attivo (né automaticamente né via software). Nel frattempo, il flag di stato BIF non può essere cancellato.

L'interruzione può essere generata dall'ingresso `BRK` che ha una polarità programmabile e un bit di abilitazione `BKE` nel registro `TIMx_BDTR`.

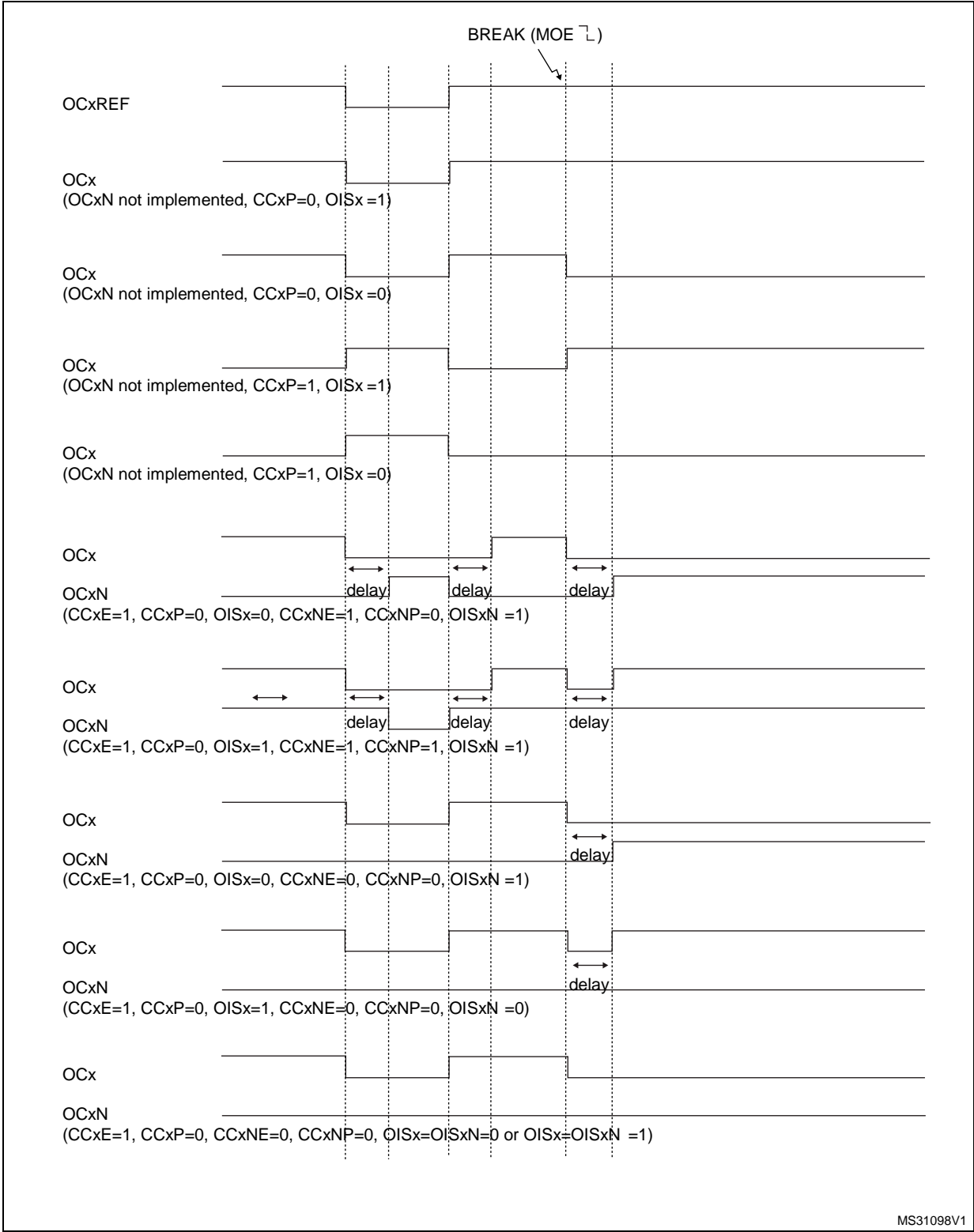
Ci sono due soluzioni per generare una pausa:

- Utilizzando l'ingresso `BRK` che ha una polarità programmabile e un bit di abilitazione `BKE` nel registro `TIMx_BDTR`
- Tramite software tramite il bit `BG` del registro `TIMx_EGR`.

Oltre alla gestione dell'ingresso e dell'uscita di interruzione, all'interno del circuito di interruzione è stata implementata una protezione da scrittura per salvaguardare l'applicazione. Permette di congelare la configurazione di diversi parametri (durata del tempo morto, polarità `OCx/OCxN` e stato quando disabilitato, configurazioni `OCxM`, abilitazione interruzione e polarità). L'utente può scegliere tra tre livelli di protezione selezionati dai bit `LOCK` nel registro `TIMx_BDTR`. Fare riferimento alla [Sezione 14.4.18: Registro dei tempi morti e delle interruzioni \(TIMx_BDTR\) di TIM1 e TIM8](#). I bit `LOCK` possono essere scritti solo una volta dopo un reset MCU.

La Figura 89 mostra un esempio di comportamento delle uscite in risposta a un'interruzione.

Figure 89. Output behavior in response to a break.



14.3.13 Clearing the OCxREF signal on an external event

Il segnale OCxREF per un dato canale può essere pilotato in basso applicando un livello alto all'ingresso ETRF (bit di abilitazione OCxCE del corrispondente registro TIMx_CCMRx impostato su '1'). Il segnale OCxREF rimane basso fino a quando non si verifica l'evento di aggiornamento successivo, UEV.

Questa funzione può essere utilizzata solo in modalità di confronto dell'uscita e PWM e non funziona in modalità forzata.

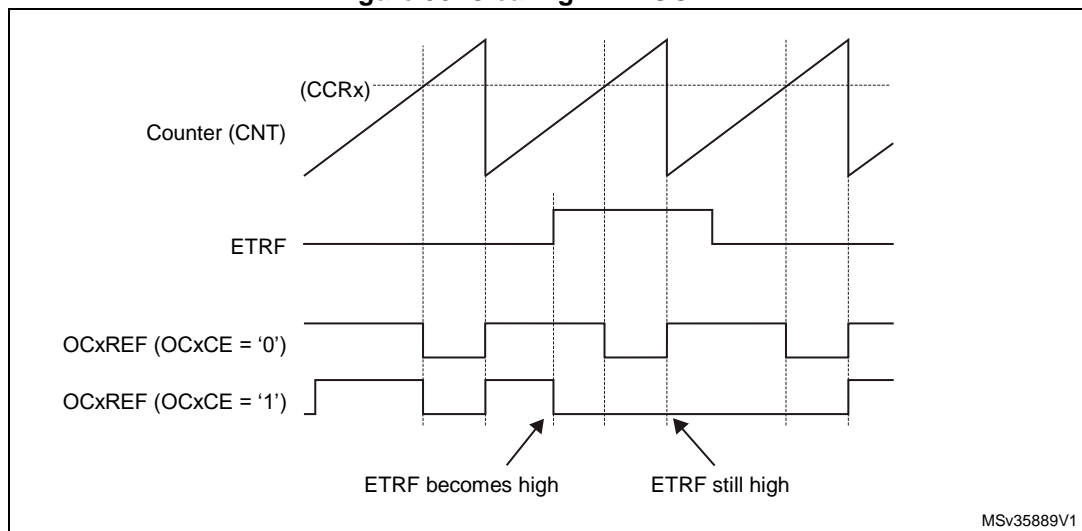
Ad esempio, il segnale ETR può essere collegato all'uscita di un comparatore da utilizzare per la gestione della corrente. In questo caso, l'ETR deve essere configurato come segue:

1. L'External Trigger Prescaler deve essere tenuto disattivato: bits ETPS[1:0] del registro TIMx_SMCR impostato su '00'.
2. La modalità orologio esterno 2 deve essere disabilitata: bit ECE del registro TIMx_SMCR impostato su '0'.
3. La polarità del trigger esterno (ETP) e il filtro del trigger esterno (ETF) possono essere configurati in base alle esigenze dell'utente.

La Figura 90 mostra il comportamento del segnale OCxREF quando l'ingresso ETRF diventa Alto, per entrambi i valori del bit di abilitazione OCxCE. In questo esempio, il temporizzatore TIMx è programmato in

Modalità PWM.

Figure 90. Clearing TIMx OCxREF



14.3.14 6-step PWM generation

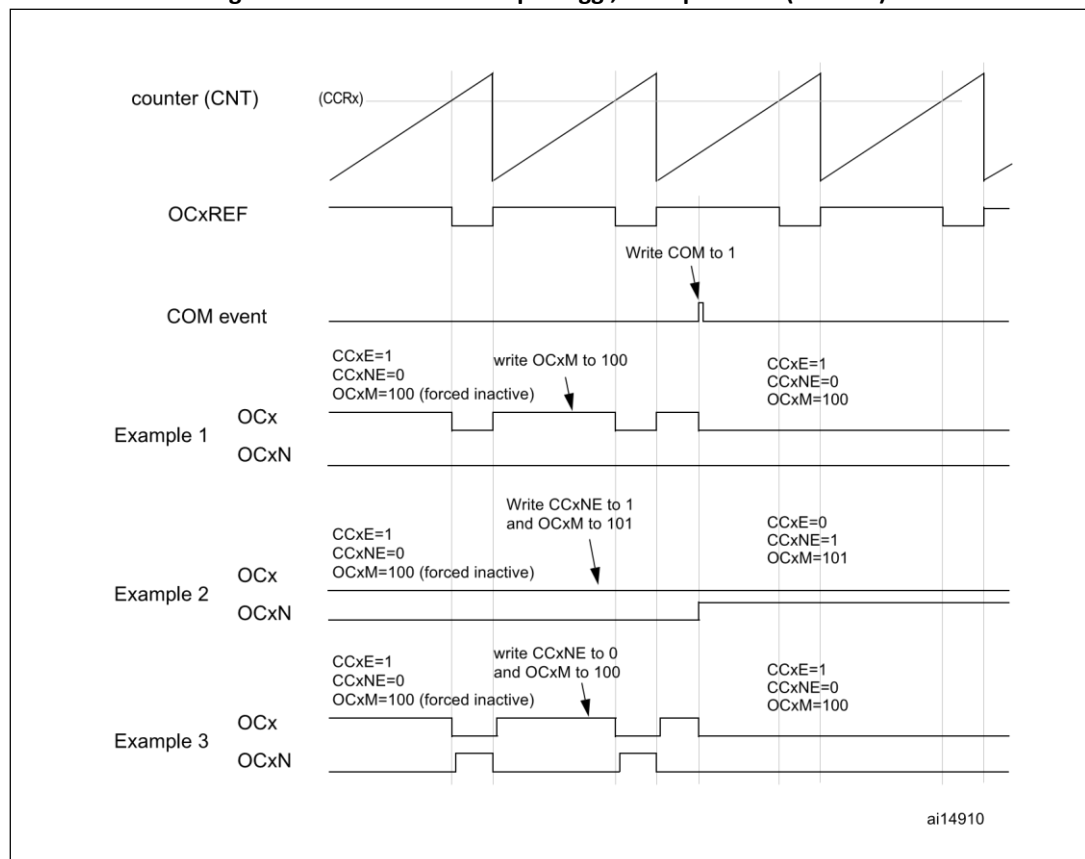
Quando si utilizzano uscite complementari su un canale, i bit di precarico sono disponibili sul

Bit OCxM, CCxE e CCxNE. I bit di precaricamento vengono trasferiti ai bit shadow in corrispondenza dell'evento di commutazione COM. L'utente può così programmare in anticipo la configurazione per la fase successiva e modificare contemporaneamente la configurazione di tutti i canali. COM può essere generato dal software impostando il bit COM nel registro TIMx_EGR o dall'hardware (sul fronte di salita TRGI).

Quando si verifica l'evento COM viene impostato un flag (bit COMIF nel registro TIMx_SR), che può generare un interrupt (se il bit COMIE è impostato nel registro TIMx_DIER) o una richiesta DMA (se il bit COMDE è impostato nel registro TIMx_DIER).

La Figura 91 descrive il comportamento delle uscite OCx e OCxN quando si verifica un evento COM, in 3 diversi esempi di configurazioni programmate.

Figura 91. Generazione in 6 passaggi, esempio COM (OSSR=1)



14.3.15 Modalità a un impulso

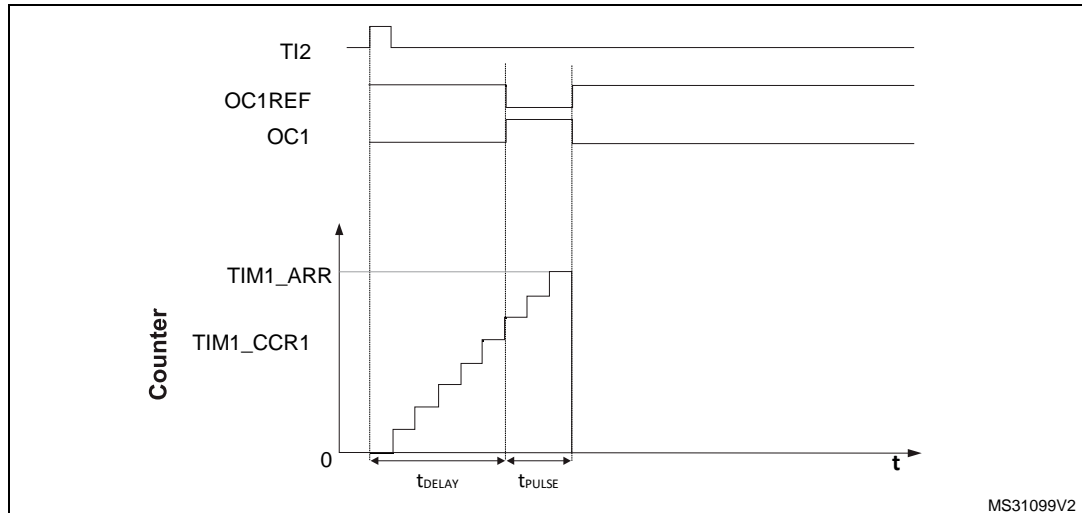
La modalità a un impulso (OPM) è un caso particolare delle modalità precedenti. Permette di avviare il contatore in risposta ad uno stimolo e di generare un impulso con una lunghezza programmabile dopo un ritardo programmabile.

L'avvio del contatore può essere controllato tramite il controller in modalità slave. La generazione della forma d'onda può essere eseguita in modalità di confronto dell'output o in modalità PWM. Selezionare la modalità a un impulso impostando il bit OPM nel registro TIMx_CR1. In questo modo il contatore si arresta automaticamente all'evento di aggiornamento successivo UEV.

Un impulso può essere generato correttamente solo se il valore di confronto è diverso dal valore iniziale del contatore. Prima di iniziare (quando il timer è in attesa dell'innesco), la configurazione deve essere:

- Nell'upcounting: $CNT < CCRx \leq ARR$ (in particolare, $0 < CCRx$)
- In downcounting: $CNT > CCRx$

Figura 92. Esempio di modalità a un impulso.



Ad esempio, l'utente potrebbe voler generare un impulso positivo su OC1 con una lunghezza di t_{PULSE} e dopo un ritardo di t_{DELAY} non appena viene rilevato un fronte positivo sul pin di ingresso TI2.

Usiamo TI2FP2 come trigger 1:

- Mappare TI2FP2 a TI2 scrivendo $CC2S='01'$ nel registro `TIMx_CCMR1`.
- TI2FP2 deve rilevare un fronte di salita, scrivere $CC2P='0'$ nel registro `TIMx_CCER`.
- Configurare TI2FP2 come trigger per il controllore in modalità slave (TRGI) scrivendo $TS='110'$ nel registro `TIMx_SMCR`.
- TI2FP2 viene utilizzato per avviare il contatore scrivendo SMS al $'110'$ nel registro `TIMx_SMCR` (modalità trigger).

La forma d'onda OPM viene definita scrivendo i registri di confronto (tenendo conto della frequenza di clock e del prescaler contatore).

- Il t_{DELAY} è definito dal valore scritto nel registro `TIMx_CCR1`.
- Il t_{PULSE} è definito dalla differenza tra il valore di ricarica automatica e il valore di confronto ($TIMx_ARR - TIMx_CCR1$).
- Supponiamo che l'utente voglia creare una forma d'onda con una transizione da '0' a '1' quando si verifica una corrispondenza di confronto e una transizione da '1' a '0' quando il contatore raggiunge il valore di ricarica automatica. Per fare ciò, abilitare la modalità PWM 2 scrivendo $OC1M=111$ nel registro `TIMx_CCMR1`. L'utente può facoltativamente abilitare i registri di precaricamento scrivendo $OC1PE='1'$ nel registro `TIMx_CCMR1` e $ARPE$ nel registro `TIMx_CR1`. In questo caso il valore di confronto deve essere scritto nel registro `TIMx_CCR1`, il

valore di auto-reload nel registro TIMx_ARR, generare un aggiornamento impostando il bit UG e attendere l'evento trigger esterno su TI2. CC1P è scritto su '0' in questo esempio.

Nel nostro esempio, i bit DIR e CMS nel registro TIMx_CR1 dovrebbero essere bassi.

L'utente desidera un solo impulso (modalità singola), quindi è necessario scrivere '1' nel bit OPM nel registro TIMx_CR1 per arrestare il contatore al successivo evento di aggiornamento (quando il contatore torna dal valore di ricarica automatica a 0). Quando il bit OPM nel registro TIMx_CR1 è impostato su '0', viene selezionata la modalità ripetitiva.

Caso particolare: OCx fast enable:

In modalità a un impulso, il rilevamento del bordo sull'ingresso TIx imposta il bit CEN che abilita il contatore. Quindi il confronto tra il contatore e il valore di confronto fa sì che l'output commuti.

Ma per queste operazioni sono necessari diversi cicli di clock e limita il ritardo minimo $t_{\text{DELAY min}}$ che possiamo ottenere.

Se l'utente desidera emettere una forma d'onda con il ritardo minimo, è necessario impostare il bit OCxFE nel registro TIMx_CCMRx. Quindi OCxRef (e OCx) vengono forzati in risposta allo stimolo, senza tenere conto del confronto. Il nuovo livello è lo stesso che si sarebbe verificato se si fosse verificata una corrispondenza di confronto. OCxFE agisce solo se il canale è configurato in modalità PWM1 o PWM2.

14.3.16 Modalità interfaccia encoder **NON DOCUMENTO !**

14.3.21 Modalità di debug

Quando il microcontrollore entra in modalità debug (core Cortex-M3[®] arrestato), il contatore TIMx continua a funzionare normalmente o si arresta, a seconda DBG_TIMx_STOP bit di configurazione nel modulo DBG. Per maggiori dettagli, fare riferimento alla [Sezione 31.16.2: Supporto del debug per timer, watchdog, bxCAN e I 2C](#).

Per motivi di sicurezza, quando il contatore è fermo (DBG_TIMx_STOP = 1 in DBGMCU_APBx_FZ registro), le uscite sono disabilitate (come se il bit MOE fosse stato resettato). Le uscite possono essere forzate in uno stato inattivo (bit OSS1 = 1) o avere il loro controllo assunto dal controller GPIO (bit OSS1 = 0) per forzarle su Hi-Z.

14.4 Registri TIM1 e TIM8

Dettagli dei registri, ci si potrebbe provare a tradurre ma ha poco senso farlo, vedi originale in inglese.