

06 TIMER

5 Mayıs 2021 Çarşamba

08:02

06 TIMER

Giriş

- Timer modülünün temel görevi zamanlama yapmaktır. İşlemci frekansasına bağlı olarak çalışırlar. Dışarıdan gelen pulse darbelerini sayarlar. İşlemciye tanıtılan bir süre ile, geçen süreyi karşılaştırma ve belli bir süre sonunda kesme üretme gibi işlemlerde kullanılırlar.
- Sayıcı birimi sabit bir frekans kaynağı ile besleniyorsa Timer olarak çalışır. Zamanlayıcının bir adımı $1/f$ süresine denk gelir. Örneğin 1 kHz ile beslenen bir zamanlayıcının her adımı 1 ms demektir.
- 1kHz ile beslenen zamanlayıcıyı t1 anında okuduğumuzda değeri 100, t2 anında okuduğumuzda değeri 250 ise, t2-t1 arasında geçen süre 150ms demektir. Zamanlayıcılar ile bu şekilde zaman ölçümü ya da periyodik işlemlerin gerçekleştirilmesini sağlarlar.
- Timer, belirli bir süre veya sayım gerçekleştirdikten sonra, sayaç değeri belirli bir sınırı aşarsa veya taşarsa, overflow durumu ortaya çıkar. Bu zamanlayıcı bir belirli sayıya kadar sayıyorsa sayaç bu sayıya ulaştığında, taşma **overflow** gerçekleşir ve sayaç sıfırlanarak yeniden başlar.

Yapılandırma

- STM32 mikrodenetleyicilerinde Timer ayarlarken, **prescaler** ve **period** değerleri zamanlayıcının üreteceği frekansı veya zamanı belirler. Bu değerler genellikle belirli bir zaman aralığı, frekans veya PWM sinyali elde etmek için seçilir.
- Prescaler, zamanlayıcıya giren **saat sinyalini bölmek** için kullanılır. Örneğin, eğer bir zamanlayıcı 72 MHz'lik bir saat sinyaliyle çalışıyorsa ve prescaler değeri 71 olarak ayarlandıysa, zamanlayıcıya gelen sinyalin frekansı 1 MHz olur.
 - Prescaler değeri şu formülle hesaplanır:
$$f_{\text{timer}} = f_{\text{clock}} / (\text{Prescaler} + 1)$$
 - **f_clock**, Timer'in besleme frekansı (genellikle sistem saat frekansı)
 - **Prescaler**, Timer'a uygulanan saat sinyalini bölen değer (0'dan başlar, dolayısıyla +1 yapılır)
- Period değeri, timer'ın kaç clock döngüsünde bir sıfırlanacağını belirler.
 - Period değeri ile timer'ın toplam zaman aralığı şu şekilde hesaplanır:
$$T_{\text{period}} = (\text{Period} + 1) \times 1 / f_{\text{timer}}$$
 - **T_period**: Timer'ın toplam zaman aralığı (örneğin bir PWM sinyali için bir periyod)
 - **Period**: Auto-Reload Register (ARR) değeri (0'dan başlar, bu yüzden +1 yapılır)
 - **f_timer**: Prescaler ile ayarlanmış timer frekansı
- Eğer 1 kHz'lik bir sinyal üretmek istiyorsanız ve sistem saat frekansınız 72 MHz ise:
 - Önce uygun bir prescaler değeri seçin:

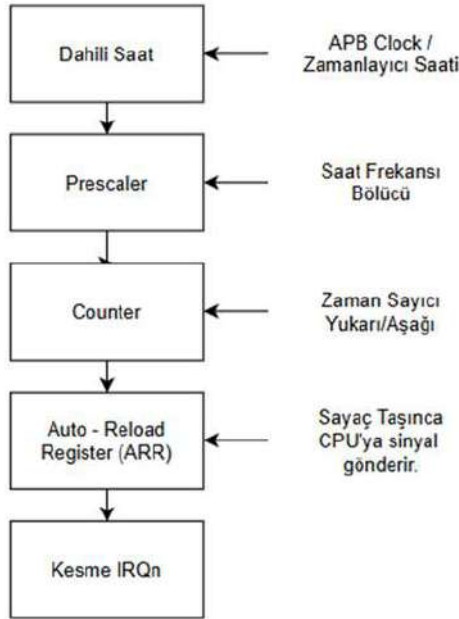
Eğer prescaler = 71999 seçilirse, $f_{\text{timer}} = 72\text{MHz} / 72000 = 1\text{kHz}$ (1 ms per tick)
 - Period değeri olarak ise:

Period = 999 seçilirse, toplam zaman $T_{\text{period}} = 1000 \times 1 / 1\text{kHz} = 1\text{s}$, 1000 tick = 1 saniye
 - 1 ms zaman tabanına sahip bir zamanlayıcı elde ediyorsunuz ve böylece 1000 kere 1 ms aralıklarla sayma yapacak sonucunda 1s sonra taşma olacak.
- Eğer timer ile 100ms'lik bir gecikme yaratmak istiyorsanız:
 - Eğer prescaler = 7199 seçilirse, $f_{\text{timer}} = 72\text{MHz} / 7200 = 10\text{kHz}$
 - Period = 999 olarak seçilirse, $T_{\text{period}} = 1000 \times 1 / 10\text{kHz} = 0,1\text{s} = 100\text{ms}$
 - 0,1 ms zaman tabanına sahip bir zamanlayıcı elde ediyorsunuz ve böylece 1000 kere 0,1 ms aralıklarla sayma yapacak sonucunda 100ms sonra taşma olacak.
- Görseldeki formül, prescaler ve period değerlerini bir araya getirerek timer'ın ne sıklıkla bir güncelleme olayı üreteceğini doğrudan hesaplayan bir yöntem sunar. Daha önce bahsedilen iki aşamalı süreç ise, önce **timer clock frekansını**, ardından **çıkış sinyal frekansını** belirleyen bir yaklaşımı takip eder. Her iki yöntem de timer yapılandırmasında kullanılır; ancak görseldeki formül, daha doğrudan ve kompakt bir yaklaşım sunar.

$$UpdateEvent = \frac{Timer_{clock}}{(Prescaler + 1)(Period + 1)}$$

Çalışma

- Timer sistemi, öncelikle **APB** clock hattından aldığı bir saat sinyaliyle başlar. Bu sinyal, timer modülünün iç saat kaynağı olarak kullanılır. Saat sinyali doğrudan zamanlayıcıya verilmeden önce **Prescaler** adı verilen bir frekans bölücüye girer. Prescaler, gelen saat sinyalini belirli bir oranda yavaşlatarak, zamanlayıcının daha uzun sürelerde sayım yapabilmesini sağlar. Prescaler'den çıkan sinyal **Counter** ünitesine gönderilir. Sayaç, belirlenen yönde **yukarı veya aşağı** sayım yapar ve bu sayım, belirli bir değere ulaşıncaya kadar devam eder.
- Counter, **Auto-Reload Register (ARR)** olarak adlandırılan bir değere ulaştığında **sıfırlanır** ve yeniden başlar. ARR, sayımın ne zaman sıfırlanacağını ve kesmenin hangi periyotla oluşacağını belirleyen ana bileşendir. Counter her **taşma** yaptığında, yani ARR değerine ulaştığında, mikrodenetleyiciye bir kesme sinyali gönderilir. Bu kesme sinyali, IRQ (Interrupt Request) hattı üzerinden CPU'ya iletilerek, belirli bir yazılım fonksiyonunun çalışmasını sağlar. Böylece mikrodenetleyici belirli zaman aralıklarında bir işlemi tekrar edebilir. Bu yapı, zaman tabanlı uygulamalar için son derece esnek ve güçlü bir çözüm sunar.



İşlevler

- **Capture**, zamanlayıcının mevcut değerini özel bir kaydediciye kopyalama işlemidir. Bu, bir dış olayın gerçekleştiği belirli bir zamanı yakalamak için kullanılabilir. Örneğin, dışardan gelen sinyalin belirli bir durumu algılandığında, zamanlayıcı değeri bu anda "yakalanır" ve kaydedilir. Bu, belirli olayların zaman damgalarını elde etmek için sıklıkla kullanılır.
- **Compare**, zamanlayıcı değerini bir belirli değerle karşılaştırma işlemi ifade eder. Zamanlayıcı, belirli bir değere ulaştığında veya onu geçtiğinde, bu bir olayın tetiklenmesine neden olabilir. Örneğin, belirli bir zaman geçtikten sonra bir işlemi başlatmak için compare özelliği kullanılabilir. Bu, periyodik işlemleri kontrol etmek veya belirli bir süreyi takip etmek için yaygın olarak kullanılır.
- **Pulse Width Modulation (PWM)**, genellikle bir dijital sinyalin darbe genişliğini modüle etme tekniğini ifade eder. PWM, bir sinyalin belirli bir süre boyunca HIGH ve belirli bir süre boyunca LOW olduğu bir sinyal üretir. Bu modülasyon tekniği, analog sinyal davranışını taklit etmek veya kontrol etmek için yaygın olarak kullanılır. Çoğu mikrodenetleyicide PWM birimleri de Timer ünitelerine bağlı olarak çalışırlar.

Birimler

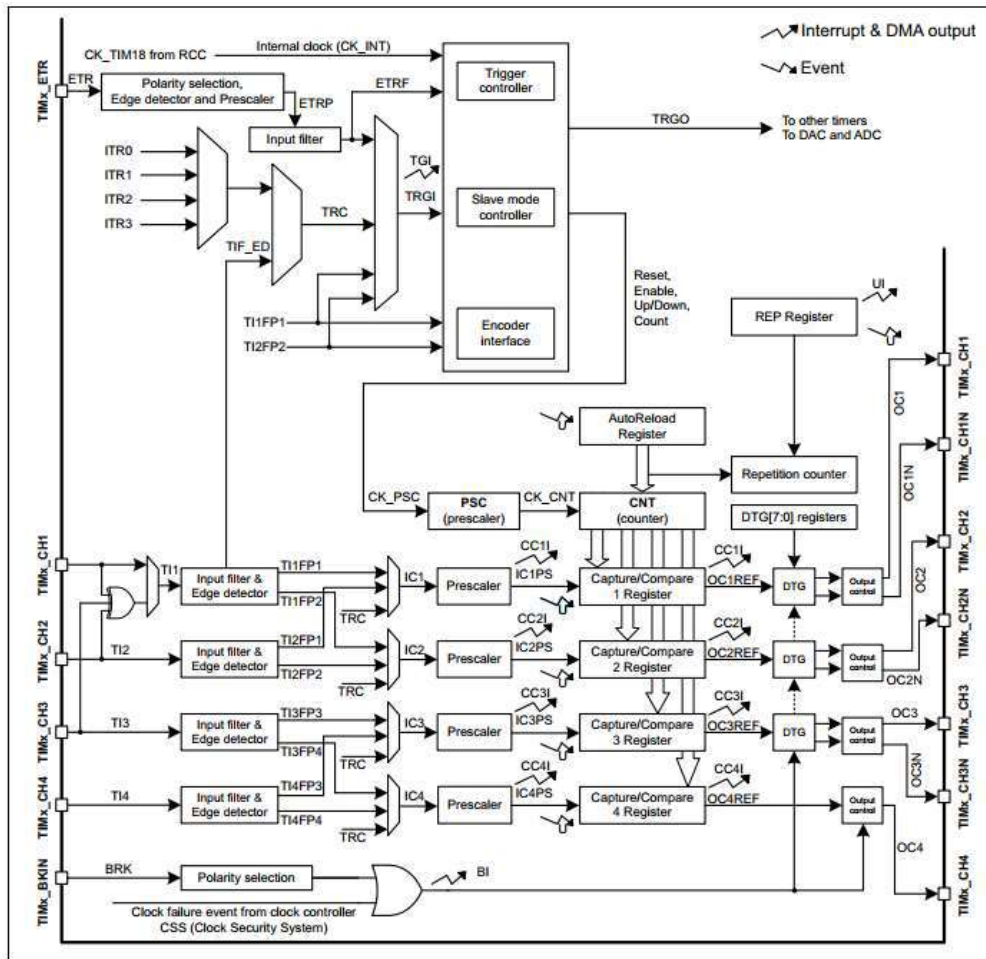
- STM32F407VG işlemcisinde toplam 17 adet timer birimi bulunur. 10 adet **General Purpose**, 2 adet **Advanced Control**, 2 adet **Basic**, 1 adet **Independent Watchdog (IWDG)**, 1 adet **Window Watchdog (WWDG)**, 1 adet **SystemTick (SysTick)** timer var.

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary output	Max interface clock (MHz)	Max timer clock (MHz)
Advanced-control	TIM1, TIM8	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	Yes	84	168
General purpose	TIM2, TIM5	32-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM3, TIM4	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No	84	168
	TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No	84	168
	TIM12	16-bit	Up	Any integer between 1 and 65536	No	2	No	42	84
	TIM13, TIM14	16-bit	Up	Any integer between 1 and 65536	No	1	No	42	84
	TIM15, TIM16	16-bit	Up	Any integer between 1 and 65536	No	1	No	42	84
Basic	TIM6, TIM7	16-bit	Up	Any integer between 1 and 65536	Yes	0	No	42	84

Advanced Control

TIM1, TIM8

- TIM1 ve TIM8, yüksek hızlı APB2 veri yolu (84 MHz) üzerinde bulunurlar. Eğer APB2 prescaler değışkeni 1 değeriinden farklı ise bu timer birimlerinin saat frekansı, APB2'nin frekans değeriinin iki katı olur. Yani, bu timer birimlerinin maksimum çalışma frekansları 168 MHz olabilir.
- TIM1 ve TIM 8 birimleri 16 bitlik sayıcıya sahiptirler.
- Bu sayıcılar; yukarı, aşağı ve merkezlenmiş modlarda sayma yapabilirler.
- Bu sayıcıların otomatik geri yükleme özellikleri bulunmaktadır.
- Bu timer birimlerinde 4x16 adet yüksek çözünürlüklü capture/compare kanalı da bulunur. Bu kanallar giriş çıkış olarak ayarlanabilir, çıkış karşılaştırabilir, PWM sinyali üretebilir, sinyal yakalayabilir ve harici bir PWM sinyaliini algılayabilirler.



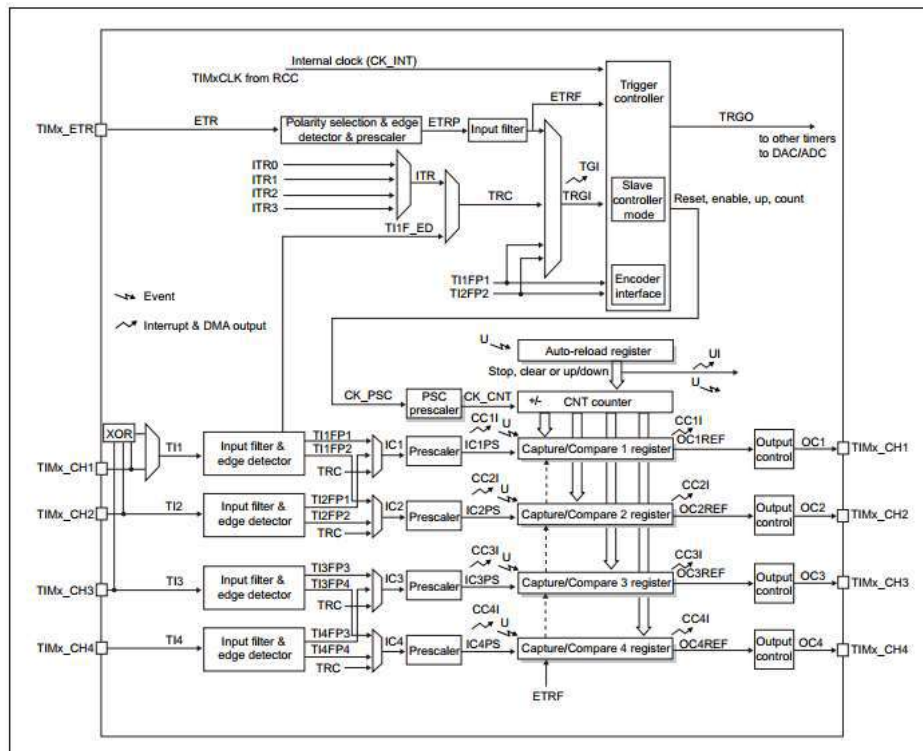
modu seçimi, otomatik yeniden başlatma etkinleştirme gibi ayarları içerir.

- **TIMx_CR2 (Control Register 2)**, Timer'ın özel kontrol ayarlarını içerir. Bu register, master mode seçimi gibi özellikleri kontrol eder.
- **TIMx_SMCR (Slave Mode Control Register)**, Timer'ın slave modunu kontrol eder. Dış bir kaynaktan senkronize olma veya bir başka timer'ı takip etme gibi işlevleri içerir.
- **TIMx_DIER (DMA/Interrupt Enable Register)**, DMA ve kesme interrupt izinlerini kontrol eder. Belirli olayların tetiklenmesi durumunda bir kesme talebi veya DMA transferi başlatma gibi işlevleri etkinleştirir veya devre dışı bırakır.
- **TIMx_SR (Status Register)**, Timer'ın durumuyla ilgili bilgileri içerir. Taşma, karşılaştırma olayları gibi çeşitli olayları takip eder.
- **TIMx_EGR (Event Generation Register)**, Olayların elle tetiklenmesini sağlar. Bu register üzerinden bir olayı (event) hemen tetikleyebilirsiniz.
- **TIMx_CCMR1 ve TIMx_CCMR2 (Capture/Compare Mode Register 1 ve 2)**, Capture/compare modu için ayarları içerir. Timer'ın çeşitli modlarını, giriş ve çıkış ayarlarını belirler.
- **TIMx_CCER (Capture/Compare Enable Register)**, Capture/compare kanallarını etkinleştirme veya devre dışı bırakma işlemlerini kontrol eder.
- **TIMx_CNT (Counter Register)**, Timer'ın ana sayaç değerini içerir. Bu register, zamanlayıcının sayma işlemini temsil eder.
- **TIMx_PSC (Prescaler Register)**, Timer'ın ön bölücü prescaler değerini içerir. Bu değer, timer'ın sayma hızını kontrol eder.
- **TIMx_ARR (Auto-Reload Register)**, Timer'ın otomatik yeniden başlatma değerini içerir. Bu değer, sayacın bir döngü tamamladığında otomatik olarak tekrar başlamasını sağlar.
- **TIMx_RCR (Repetition Counter Register)**, İleri dönüş (overflow) olayının tekrar sayısını kontrol eder.
- **TIMx_CCR1, TIMx_CCR2, TIMx_CCR3, TIMx_CCR4 (Capture/Compare Register 1, 2, 3, ve 4)**, Capture/compare modunda kullanılan karşılaştırma değerlerini içerir. Bu değerler, belirli bir zaman noktasında veya karşılaştırma olayında kullanılır.
- **TIMx_BDTR (Break and Dead-Time Register)**, Timer'ın kesme ve ölü zaman ayarlarını içerir.
- **TIMx_DCR (DMA Control Register)**, DMA transferlerini kontrol eder.
- **TIMx_DMAR (DMA Address Register)**, DMA transferleri için adres bilgisini içerir.

General Purpose

TIM2, TIM3, TIM4, ve TIM5

- TIM2, TIM3, TIM4, ve TIM5 birimleri, düşük hızlı APB1 (42 MHz) veri yolu üzerinde bulunmaktadır. Eğer APB1 prescaler değeri 1'den farklı ise bu timerların clock frekansları beslendikleri frekansların 2 katına çıkar. Yani 84 MHz clock frekansına sahip olur.
- TIM3 ve TIM4 16-bit'lik sayıcıya, TIM2 ve TIM5 32-bit'lik sayıcıya sahiptirler.
- Bu sayıcılar up, down ve auto-reload modlarda sayma yapabilirler.
- Ayrıca bu sayıcıların otomatik yükleme özellikleri de vardır.
- 16-bit genişliğinde kontrol edilebilir prescaler değeri vardır.
- Bu timer biriminde 4x16 adet yüksek çözünürlüklü capture/compare kanalı bulunur. Bu kanallar; Input Capture, Output Compare, PWM, One-Pulse'dır.
- Dahili diğer Timer birimleri ile senkronizasyon
- Interrupt ve DMA üretimi mevcuttur.
- Clock kaynağı seçimi



[illegible]

- **TIMx_CR1 (Control Register 1)**, Timer'ın genel kontrol ayarlarını içerir. Timer'ın etkinleştirme, zamanlama modu seçimi, otomatik yeniden başlatma etkinleştirme gibi ayarları içerir.
- **TIMx_CR2 (Control Register 2)**, Timer'ın özel kontrol ayarlarını içerir. Bu register, master mode seçimi gibi özellikleri kontrol eder.

- **TIMx_SMCR (Slave Mode Control Register)**, Timer'ın slave (köle) modunu kontrol eder. Dış bir kaynaktan senkronize olma veya bir başka Timer'ın takip etme gibi işlevleri içerir.
- **TIMx_DIER (DMA/Interrupt Enable Register)**, DMA (Direct Memory Access) ve kesme (interrupt) izinlerini kontrol eder. Belirli olayların tetiklenmesi durumunda bir kesme talebi veya DMA transferi başlatma gibi işlevleri etkinleştirir veya devre dışı bırakır.
- **TIMx_SR (Status Register)**, Timer'ın durumuyla ilgili bilgileri içerir. Taşma, karşılaştırma olayları gibi çeşitli olayları takip eder.
- **TIMx_EGR (Event Generation Register)**, Olayların elle tetiklenmesini sağlar. Bu register üzerinden bir olayı (event) hemen tetikleyebilirsiniz.
- **TIMx_CCMR1 ve TIMx_CCMR2 (Capture/Compare Mode Register 1 ve 2)**, Capture/compare modu için ayarları içerir. Timer'ın çeşitli modlarını, giriş ve çıkış ayarlarını belirler.
- **TIMx_CCER (Capture/Compare Enable Register)**, Capture/compare kanallarını etkinleştirme veya devre dışı bırakma işlemlerini kontrol eder.
- **TIMx_CNT (Counter Register)**, Timer'ın ana sayaç değerini içerir. Bu register, zamanlayıcının sayma işlemini temsil eder.
- **TIMx_PSC (Prescaler Register)**, Timer'ın ön bölücü (prescaler) değerini içerir. Bu değer, timer'ın sayma hızını kontrol eder.
- **TIMx_ARR (Auto-Reload Register)**, Timer'ın otomatik yeniden başlatma değerini içerir. Bu değer, sayacın bir döngü tamamladığında otomatik olarak tekrar başlamasını sağlar.
- **TIMx_CCR1, TIMx_CCR2, TIMx_CCR3, TIMx_CCR4 (Capture/Compare Register 1, 2, 3, ve 4)**, Capture/compare modunda kullanılan karşılaştırma değerlerini içerir. Bu değerler, belirli bir zaman noktasında veya karşılaştırma olayında kullanılır.
- **TIMx_DCR (DMA Control Register)**, DMA transferlerini kontrol eder.
- **TIMx_DMAR (DMA Address Register)**, DMA transferleri için adres bilgisini içerir.
- **TIMx_OR (Option Register)**, Timer'ın özel seçeneklerini kontrol eder. Bu register, özel özelliklerin etkinleştirilmesi veya devre dışı bırakılması için kullanılır.

TIM9, TIM10, TIM11, TIM12, TIM13, TIM14

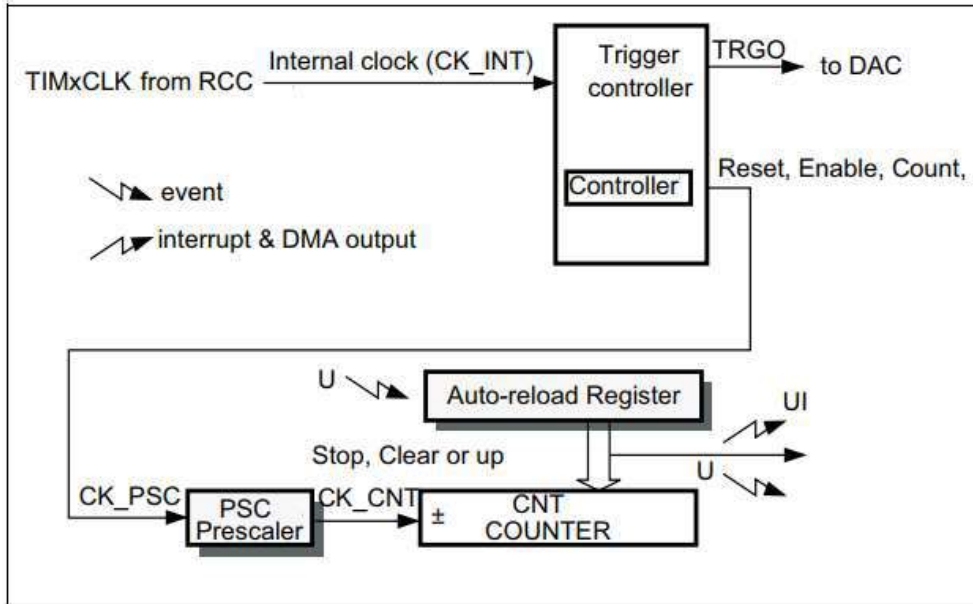
- TIM9 yüksek hızlı APB2 (84 MHz) ve TIM12 düşük hızlı APB1 (42 MHz) üzerinde bulunmaktadır.
- Bu birimlerin frekansları diğerlerinde olduğu gibi veriyolu hızlarının iki katında çalışabilirler.
- TIM9 ve TIM12 birimleri 16 bitlik sayıcıya sahiptirler. Bu sayıcılar sadece yukarı sayma yapabilirler. Ayrıca bu sayıcıların otomatik geri yükleme özellikleri de bulunmaktadır.
- Bu timer birimlerinde 2x16 adet yüksek çözünürlüklü capture/compare kanalı da bulunur. Bu kanallar giriş çıkış olarak ayarlanabilir, çıkış karşılaştırılabilir, PWM sinyali üretebilir, sinyal yakalayabilir ve harici bir PWM sinyalini algılayabilirler.
- TIM10 ve TIM11 yüksek hızlı APB2 (84 MHz) ve TIM13 ve TIM14 düşük hızlı APB1 (42 MHz) üzerinde bulunmaktadır. Bu birimlerin frekansları diğerlerinde olduğu gibi veriyolu hızlarının iki katında çalışabilirler.
- Bu birimler 16 bitlik sayıcıya sahiptirler. Bu sayıcılar sadece yukarı sayma yapabilirler. Ayrıca bu sayıcıların otomatik geri yükleme özellikleri de bulunmaktadır.
- Bu timer birimlerinde 2x16 adet yüksek çözünürlüklü capture/compare kanalı da bulunur. Bu kanallar giriş çıkış olarak ayarlanabilir, çıkış karşılaştırılabilir, PWM sinyali üretebilir, sinyal yakalayabilir ve harici bir PWM sinyalini algılayabilirler.

- **TIMx_CNT (Counter Register)**, Timer'ın ana sayaç değerini içerir. Bu register, zamanlayıcının sayma işlemini temsil eder.
- **TIMx_PSC (Prescaler Register)**, Timer'ın ön bölücü prescaler değerini içerir. Bu değer, timer'ın sayma hızını kontrol eder.
- **TIMx_ARR (Auto-Reload Register)**, Timer'ın otomatik yeniden başlatma değerini içerir. Bu değer, sayacın bir döngü tamamladığında otomatik olarak tekrar başlamasını sağlar.
- **TIMx_CCR1 (Capture/Compare Register 1)**, Capture/compare modunda kullanılan karşılaştırma değerini içerir. Bu değer, belirli bir zaman noktasında veya karşılaştırma olayında kullanılır.
- **TIMx_OR (Option Register)**, Timer'ın özel seçeneklerini kontrol eder. Bu register, özel özelliklerin etkinleştirilmesi veya devre dışı bırakılması için kullanılır.

Basic Timer

TIM6, TIM7

- TIM6 ve TIM7 Basic Timer birimleri genel sayaç olarak kullanılabilecekleri gibi, spesifik olarak DAC biriminin tetikleyicisi olarak da kullanılabilir.
- 16-bit genişliğinde auto-reload upcounter yani otomatik geri yüklenen artan sayaca sahiptir.
- 16-bit genişliğinde kontrol edilebilir prescaler değere sahiptir.
- DAC birimi için tetikleme çıkışlarına sahiptir.
- Interrupt ve DMA üretimi mevcuttur.
- Çalışma prensibi genel amaçlı timer'ların çalışma prensibi ile aynıdır.



- Çalışma akışı
 - RCC'den clock gelir.
 - PSC clock'u böler.
 - CNT sayacı bu clock ile sayar.
 - CNT, ARR'ye ulaştınca sıfırlanır ve Update Event oluşur.
 - Update Event interrupt'a (UI) ya da başka çevrelere (TRGO → DAC) gider.

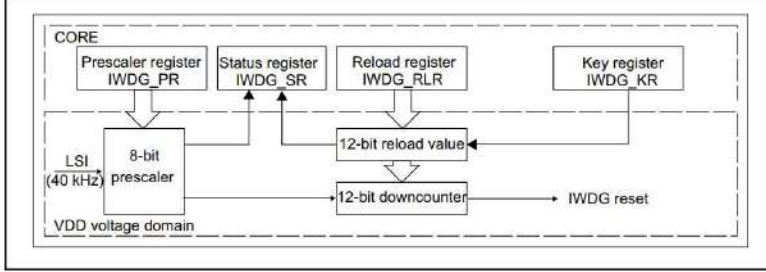
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1	Reserved																								ARPE	Reserved			OPM	URS	UDIS	CEN
	0																									0				0	0	0	
0x04	TIMx_CR2	Reserved																								MMS[2:0]			Reserved				
	0																									0	0						
0x0C	TIMx_DIER	Reserved																								UDE	Reserved						UIE
	0																									0							
0x10	TIMx_SR	Reserved																													UIF		
	0																																
0x14	TIMx_EGR	Reserved																													UG		
	0																																
0x24	TIMx_CNT	Reserved															CNT[15:0]																
	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x28	TIMx_PSC	Reserved															PSC[15:0]																
	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	TIMx_ARR	Reserved															ARR[15:0]																
	1																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

- **TIMx_CR1 (Control Register 1)**, Timer'ın genel kontrol ayarlarını içerir. Örneğin, Timer'ın etkinleştirilmesi, zamanlama modu seçimi, otomatik yeniden başlatma etkinleştirme gibi ayarlar bu register üzerinden yapılmaktadır.
- **TIMx_CR2 (Control Register 2)**, dış tetikleyici konfigürasyonları gibi timer'ın belirli özelliklerini ayarlamanızı sağlar.
- **TIMx_DIER (DMA/Interrupt Enable Register)**, DMA ve interrupt izinlerini kontrol eder. Belirli olayların tetiklenmesi durumunda bir kesme talebi veya DMA transferi başlatma gibi işlevleri etkinleştirir veya devre dışı bırakır.
- **TIMx_SR (Status Register)**, bir taşma durumu overflow olup olmadığını veya bir karşılaştırma olayının gerçekleşip gerçekleşmediğini belirtir.
- **TIMx_EGR (Event Generation Register)**, Olayların elle tetiklenmesini sağlar. Bu register üzerinden bir olayı event hemen tetikleyebilirsiniz.
- **TIMx_CNT (Counter Register)**, Timer'ın ana sayaç değerini içerir. Bu register, zamanlayıcının sayma işlemi temsil eder.
- **TIMx_PSC (Prescaler Register)**, Timer'ın prescaler değerini içerir. Bu değer, timer'ın sayma hızını kontrol eder.
- **TIMx_ARR (Auto-Reload Register)**, Timer'ın otomatik yeniden başlatma değerini içerir. Bu değer, sayacın bir döngü tamamladığında otomatik olarak tekrar başlamasını sağlar.
- **Timer sayaç zamanlayıcısının doğru şekilde yapılandırılması ve çalıştırılması için izlenecek adımlar** açıklanmaktadır.
 - **Clock Yapılandırması**
 - Kullanacağın timer'ın clock'unu aç.
 - **Prescaler (PSC) ve Auto-Reload (ARR) Ayarı**
 - Timer frekansını belirle:
 - **Update Interrupt Enable**
 - DIER registerındaki **UIE (Update Interrupt Enable)** bitini set et.
 - **NVIC Ayarı**
 - NVIC'te ilgili timer kesmesini enable et.
 - **Timer Enable**
 - CR1 registerındaki **CEN** bitini set et.
 - **Kesme Servis Fonksiyonu**
 - Otomatik olarak TIM2_IRQHandler() fonksiyonunu açar.

- İçine **Update Flag (UIF)** kontrol edip temizlemen gerekir.

Independent Watchdog (IWDG)

- IWDG**, işlemci saatinden bağımsız, kendine ait **dahili RC osilatörden** (LSI 32 KHz) beslenen bir watchdog timerdir.
- Mikrodenetleyici içerisindeki amacı da **bekçilik** yapmaktır. Mikrodenetleyici, harici sebeplerden veya kodlardaki bir hata sebebiyle **kilitlenebilir**. Mikrodenetleyici kilitlendiğinde, yürüttüğü işlemler durur. Bu tür durumlarda mikrodenetleyicinin tekrar başlatılması gereklidir. İşte watchdog timerlar burada devreye girerler. Watchdog timerlarda belirlenen bir süre sonunda sıfırlanırlar ve işlemciyi **resetlerler**.

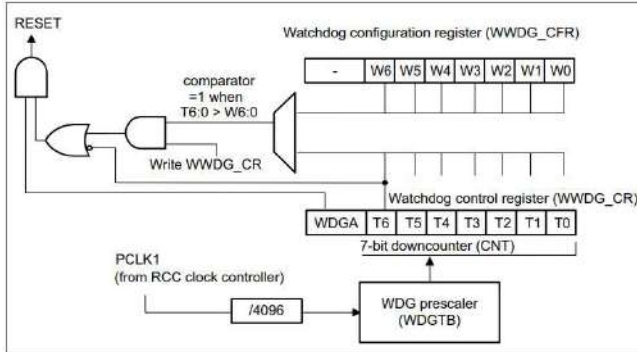


Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IWDG_KR	Reserved																KEY[15:0]															
	Reset value																																
0x04	IWDG_PR	Reserved																PR[2:0]															
	Reset value																																
0x08	IWDG_RLR	Reserved																RL[11:0]															
	Reset value																																
0x0C	IWDG_SR	Reserved																RVU PVU															
	Reset value																																

- IWDG_KR (Key Register)**, IWDG'yi kontrol etmek için kullanılan anahtar değerleri içerir. İlgili anahtar değerleri yazılarak IWDG'nin başlatılması, yeniden başlatılması veya durdurulması gibi işlemler gerçekleştirilir.
- IWDG_PR (Prescaler Register)**, IWDG'nin zamanlayıcı değerini belirlemek için kullanılır. Zamanlayıcı değeri, bu ön bölücü ile çarparak IWDG'nin zamanlamasını elde eder.
- IWDG_RLR (Reload Register)**, IWDG'nin zamanlayıcı değerini reload value içerir. IWDG'nin çalışması sırasında bu değer zaman içinde azalır, eğer bu değer sıfıra ulaşırsa, IWDG bir reset sinyali üretir.
- IWDG_SR (Status Register)**, IWDG'nin durumunu gösteren bilgiler içerir. Örneğin, zaman aşımı durumu gibi bilgiler burada bulunabilir.

Window Watchdog (WWDG)

- WWDG birimi belirli bir pencere içerisinde counter kaydedicisine tekrar değer yüklenebildiği için bu isimle anılmaktadır.
- Ayarlanabilir süre penceresine sahiptir.
- Anormal erken ve anormal geç uygulama davranışını algılayabilir.
- Önceden belirlenen duruma göre işlemciyi resetler.



Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WWDG_CR	Reserved																							WDGA	T[5:0]							
	Reset value																								0	1	1	1	1	1	1	1	
0x04	WWDG_CFR	Reserved																						EWI	WDGTB1	WDGTB0	W[6:0]						
	Reset value																							0	0	0	1	1	1	1	1	1	1
0x08	WWDG_SR	Reserved																															EWIF
	Reset value																																0

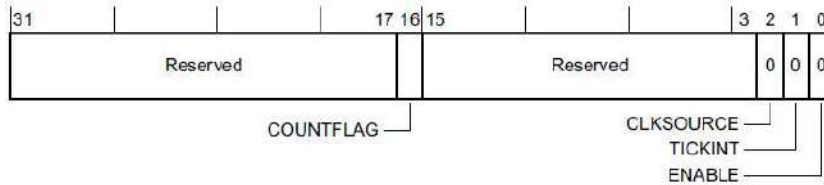
- **WWDG_CR (Control Register)**, WWDG'nin temel kontrol ayarlarını içerir. Özellikle, WWDG'nin etkinleştirilmesi, zamanlayıcı değeri (down-counter) ayarlanması ve bir reset talep biti bulunmaktadır.
- **WWDG_CFR (Configuration Register)**, WWDG'nin daha fazla konfigürasyon ayarlarını içerir. Örneğin, window modunu etkinleştirme, zaman aşımı değeri ve window değeri gibi ayarları içerir.
- **WWDG_SR (Status Register)**, WWDG'nin durumunu gösteren bilgiler içerir. Örneğin, zaman aşımı durumu ve window durumu gibi bilgiler burada bulunabilir.

System Timer (SysTick)

- İşlemcide, **SysTick** adı verilen 24-bitlik bir sistem zamanlayıcısı vardır. Bu zamanlayıcı, **yükleme (reload) değerinden sıfıra kadar geri sayar**, ardından bir sonraki clock darbesinde **SYST_RVR register'ındaki değere tekrar yüklenir (wraps)** ve sonraki clock darbelerinde yeniden geri saymaya devam eder.

Address	Name	Type	Required privilege	Reset value	Description
0xE000E010	SYST_CSR	RW	Privileged	a	<i>SysTick Control and Status Register</i>
0xE000E014	SYST_RVR	RW	Privileged	Unknown	<i>SysTick Reload Value Register</i>
0xE000E018	SYST_CVR	RW	Privileged	Unknown	<i>SysTick Current Value Register</i>
0xE000E01C	SYST_CALIB	RO	Privileged	-a	<i>SysTick Calibration Value Register</i>

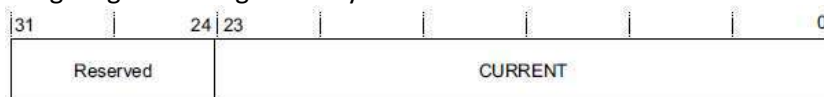
- **SYST_CSR (SysTick Control and Status Register)**, SysTick timer'ı başlatıp durdurmak, clock kaynağını seçmek ve interrupt'u etkinleştirmek için kullanılır.



- **ENABLE**
1: Sayaç çalışır
0: Sayaç durur
- **TICKINT**
1: Sayaç 0'a ulaştığında **interrupt** üretir.
- **CLKSOURCE**
0: Clock kaynağı = External
1: Clock kaynağı = Processor
- **COUNTFLAG**, sayaç 0'a indiğinde **otomatik 1 olur**. Okunduğunda sıfırlanır.
- **SYST_RVR (SysTick Reload Value Register)**, buraya **SysTick'in geri sayacağı başlangıç değeri** yazılır.



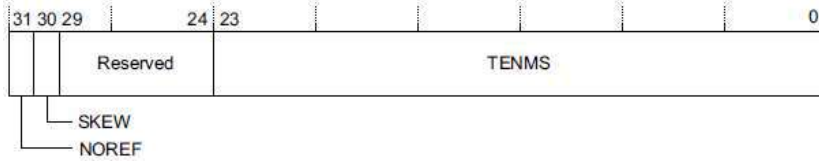
- 24-bit genişliğinde (0x00000001-0x00FFFFFF = 16,777,215)
- Sayaç bu değerden 0'a kadar sayar.
- **SYST_CVR (SysTick Current Value Register)**, anlık sayaç değerini tutar. Yazıldığında sayaç **sıfırlanır**. Her clock darbesinde **1 azalır**. 0'a indiğinde tekrar SYST_RVR değerinden yüklenir. Bu sayede o anda timer'ın hangi değerde olduğunu okuyabilirsin



- **SYST_CALIB (SysTick Calibration Value Register)**, içinde üretici tarafından yazılmış, **kalibrasyon değerleri**

bulunur. Özellikle **10 ms'lik sabit zaman aralığı için reload değeri** içerir. Kullanıcı bu değeri otomatik delay hesaplamaları için kullanabilir. Her mikrodenetleyicide bu register dolu olmayabilir (bazılarında sadece 0 olabilir)

- **TENMS**, 10 ms'lik bir süre için gereken clock sayısı.
- **SKEW**, Kalibrasyon değerinin tam doğru olmadığını gösterir.
- **NOREF**, Harici referans clock desteği yok anlamına gelir.



- **SysTick zamanlayıcısının** doğru şekilde yapılandırılması ve çalıştırılması için izlenecek adımlar açıklanmaktadır.

- **Clock Frekansının Belirlenmesi**
 - SystemCoreClock değeri tespit edilmelidir.
 - Bu değer, **reload** hesaplamasında kullanılacaktır.
- **Reload Değerinin Ayarlanması**
 - Zamanlama periyodu şu formül ile hesaplanır:

$$\text{Periyot} = (\text{Reload} + 1) / \text{Clock_Frekans}$$
 - Örnek: 1 ms kesme için:

$$\text{Reload} = (\text{SystemCoreClock} / 1000) - 1$$
- **Current Value Register'in Sıfırlanması**
 - Sayaç sıfırlanmalıdır.
 - Bu işlem, sayaç değerinin reload'dan doğru şekilde başlamasını sağlar.
- **Control Register Ayarlarının Yapılması**
 - Clock kaynağı seçilir
 - Kesme kullanılacaksa **interrupt enable** biti set edilir.
 - Sayaç başlatılır
- **Kesme Kullanımı Durumunda**
 - SysTick_Handler() fonksiyonu yazılmalıdır.
 - Her kesme oluştuğunda: sayaç artırılabilir, bir flag set edilebilir veya başka bir görev icra edilebilir.
- **Ana Döngü (main while) İşleyişi**
 - **Kesme tabanlı kullanımda**, flag veya sayaç kontrol edilerek süre dolumu tespit edilir.
 - **Polling yönteminde**, COUNTFLAG biti kontrol edilerek sayaç taşması izlenir.

Benzerlikler ve Farklılıklar

- Timer birimlerin sahip oldukları özelliklere göre Basic, General Purpose ve Advanced Control olmak üzere üç grupta incelenir. Aşağıdaki tabloda bu üç timer türünün benzerlikleri ve farklılıkları özetlenmiştir.

Özellik	Basic Timers	General Purpose Timers	Advanced Control Timers
Ana kullanım amacı	Basit zamanlayıcı (delay, periyodik interrupt) veya DAC tetikleme.	Genel amaçlı zamanlama, input capture, output compare, PWM.	Motor kontrolü, güç elektroniği, gelişmiş PWM ve dead-time gerektiren uygulamalar.
Counter boyutu	16-bit (bazı modellerde sadece 16-bit).	16-bit veya 32-bit (ör. TIM2, TIM5 → 32-bit).	16-bit.
Temel özellikler	Yalnızca prescaler + auto-reload (ARR) ile periyodik zamanlayıcı.	Input capture, output compare, PWM üretimi, encoder interface.	General Purpose özelliklerine ek olarak: dead-time insertion, break input, complementary PWM çıkışı.
DMA / Trigger	Update event ile DMA veya DAC tetikleme yapılabilir.	Geniş DMA/trigger desteği.	Daha fazla DMA/trigger imkânı + fault protection.
Kesme desteği	Sadece update interrupt (overflow).	Update + capture/compare interrupt.	Update + capture/compare interrupt + break interrupt.
PWM desteği	Yok.	Var (temel PWM çıkışı).	Var + gelişmiş PWM (komplementer çıkış, dead-time, yüksek çözünürlük).

Tipik kullanım alanı	Delay fonksiyonları, DAC için trigger üretmek.	Input ölçümü (pulse genişliği, frekans), PWM çıkışı, encoder okuma.	BLDC/FOC motor sürme, inverter, SMPS, yüksek güçlü sürücüler.
-----------------------------	--	---	---

Özellik	Basic	General Purpose	Advanced Control
Time-base	✓	✓	✓
Interrupt	✓	✓	✓
PWM	✗	✓	✓
Input Capture	✗	✓	✓
Output Compare	✗	✓	✓
Encoder Interface	✗	✓	✓
Complementary PWM	✗	✗	✓
Dead-time insertion	✗	✗	✓
Break function	✗	✗	✓
Tipik Kullanım	Delay, DAC Trigger	PWM, ölçüm, encoder	Motor kontrol, güç elektroniği

- Window Watchdog (WWDG) ve Independent Watchdog (IWDG), sistem güvenilirliğini artırmak için kullanılan iki farklı gözetim modülüdür. Her ikisi de yazılımın belirlenen sürelerde çalışıp çalışmadığını denetleyerek reset mekanizması sağlar; ancak çalışma prensipleri, bağımlı oldukları clock kaynakları ve kullanım amaçları açısından önemli farklılıklar gösterir.

Özellik	WWDG (Window Watchdog)	IWDG (Independent Watchdog)
Çalışma mantığı	Bir pencere içinde yenilenmeli. Çok erken veya çok geç yenilenirse reset atar.	Sadece zaman aşımı süresi var. Yenilenmezse reset atar.
Zamanlama kaynağı	Ana sistem clock'una (APB1 bus clock) bağlıdır.	Bağımsız LSI (Low-Speed Internal RC – tipik 32 kHz) osilatöründen beslenir.
Bağımsızlık	Sistem clock'una bağımlıdır, clock durursa çalışmaz.	Tamamen bağımsızdır, sistem clock veya PLL dursa bile çalışır.
Kullanım amacı	Daha hassas kontrol: “Ne çok sık, ne çok seyrek” yenileme. Örn. tasklerin belli periyotta çalışmasını garanti etmek.	Genel sistem güvenliği için, “mutlaka belli bir sürede çalışıyor mu?” kontrolü.
Konfigürasyon	Yenileme penceresi ve counter ayarlanır. Yenileme çok erken/çok geç olamaz.	Prescaler ve reload değerleri ile timeout süresi ayarlanır.
Güvenlik seviyesi	Orta seviyeli güvenlik sağlar, CPU clock'a bağımlıdır.	Daha güvenilir (bağımsız clock sayesinde). Genellikle safety-critical sistemlerde tercih edilir.
Reset tipi	Erken refresh veya timeout reset atar.	Sadece timeout reset atar.
Tipik kullanım	Görevlerin doğru periyotlarda çalışmasını izlemek (ör. RTOS task monitoring).	Genel sistem watchdog, CPU donmasını veya clock failure durumunu algılamak.

- Basic Timer** ile **SysTick Timer** ikisi de zamanlama için kullanılır ama farklı amaçlara hizmet eder.

Özellik	SysTick Timer	Basic Timer (TIM6, TIM7)
Temel kullanım amacı	Genellikle sistem zamanlayıcısı olarak kullanılır (RTOS tick, delay, periyodik kesme).	Basit zamanlayıcı veya trigger kaynağı (DAC tetikleme, delay, kesme).
Donanım tipi	Cortex-M çekirdeği içinde çekirdek timer .	STM32'nin genel amaçlı timer birimi (APB bus üzerinde).
Bit genişliği	24-bit down counter.	16-bit up counter (bazı serilerde 32-bit olabilir).
Clock kaynağı	Ana CPU clock (HCLK) veya türetilmiş clock.	APB1/2 timer clock (genellikle HCLK/2 veya HCLK).

Kesme özelliđi	Tek bir kesme hattı vardır (SysTick_IRQn).	Update interrupt ve DMA tetikleme desteđi vardır.
Kullanım kolaylıđı	Donanım basittir, genelde sadece periyodik kesme üretir .	Daha esnektir: prescaler + ARR ile istenen periyotlar ayarlanır, ayrıca trigger/DMAısıyla çevre birimleri besleyebilir.
Bağımlılık	Çekirdeđe gömülü, her Cortex-M’de standarttır.	Mikrodenetleyici serisine bađlıdır (TIM6, TIM7 her STM32’de olmayabilir).
Tipik kullanım alanı	RTOS sistem tick üretmek, delay fonksiyonları, genel zaman tabanı.	DAC tetiklemek, periyodik görevler, basit delay veya kesme.