***STM32 CLOCKS***

* STM32 DE 2 ADET CLK KAYNAGI VARDIR
* DAHİLİ RC OSİLATÖRÜ(HSI)
* KARTIN ÜZERİNDE KRİSTAL OSİLASTÖRDE BULUNUR(HSE)
* KARTIN İÇİNDEKİ OSİLATÖR SICAKLIKTAN ETKİNEBEİLİR
* YÜKSEK HZIDA ÇALIŞAN ÇEVREBİRİMLERİ İÇİN KRİSTAL OSİLATÖR DAHA KULLANIŞLIDIR
* DÜŞÜK HIZLI OSİLATÖR (LSI) , (RTC) GERÇEK ZAMNLI SAAT, BAGIMSIZ GÖZCU(IWDT) ÇEVREBİRİMLERİNİ ÇALIŞTIRMAK İÇİN KULLANILIR.
* İHTİYAÇ DURUMUNA GÖRE FREAKNSIMIZ ARTIRIP AZALTABİLİRZ.

Saat kontrolü;

* HSE kullanılıyorsa, yüksek hızlı osilatör kaynağı seçilir (HSI veya HSE) ve uygun şekilde yapılandırılır.
* SYSCLK'yi yüksek hızlı osilatör tarafından sağlanan frekanstan daha yüksek bir frekansla beslemek istiyorsak, ana PLL'yi (PLLCLK sinyali sağlayan) yapılandırmamız gerekir. Aksi takdirde bu adımı atlayabiliriz.
* Ardından, yüksek hızlı saatin (HCLK - çekirdeği, DMA'ları ve AHB veriyolunu besleyen) istenen frekansa ve gelişmiş frekanslarına ulaşmak için doğru AHB, APB1 ve APB2 (varsa) ön ölçekleyici ayarlarını seçiyoruz. Belirli bir STM32 mikrodenetleyici için PLL'ler ve ön ölçeklerin sadece bazı kombinasyonlar geçerlidir.

Zamanlayıcalar neden kullanılır;

* En temek özelliği zamanlayıcı olmasıdır.
* Harici olaylarda frekans ölçümüne izin verir.
* Bir çıkış dalga formunu kontrol etmek.
* Her kanalda bağımsız olarak pwm sinyallerini ürettirme.

Temel zamanlayıcalar;

* En basit zamanlayıclardır.
* 16bit kullanılır yani 65535 ekadar sayabilir
* Giriş çıkış pinleri yok
* Dacçevre birimlerini beslemek içinde kullanılır
* Temel zaamanlayıcalar diger zamanlayıcalar için master olarak adlandırılır.
* TIM6 VE TIM7

Genel zamanlayıcalar;

* 16 ve 32 bit olabilirler.
* Çıkış karşılaştırması yapabilme
* Giriş yakalama
* Sensör arayüzü
* Dört programanablir giriş/çıkış kanalı sağlar.
* TIM2, TIM3, TIM4, TIM5, TIM12, TIM13, TIM15, TIM16 ve TIM17

Gelişmiş zamanlayıcı;

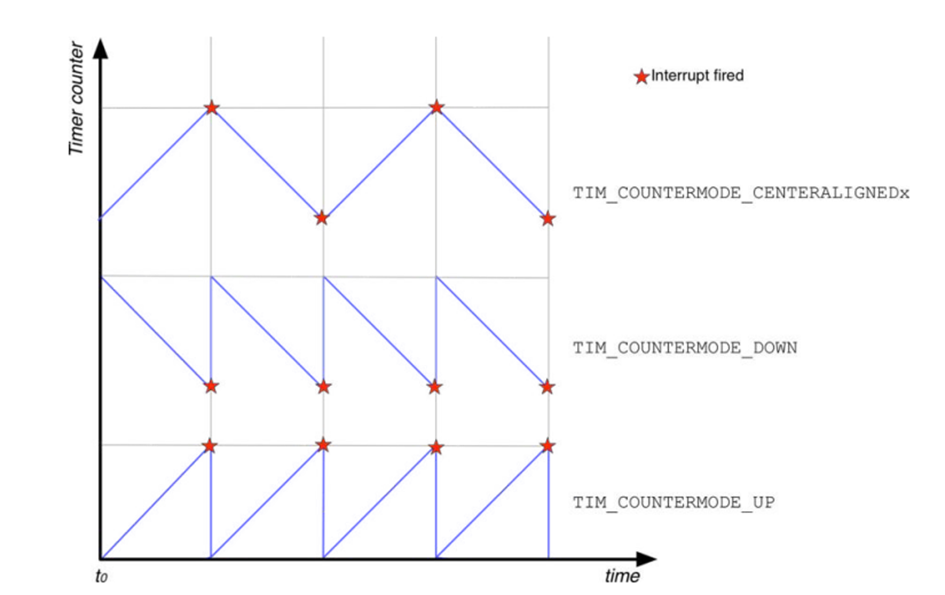
* Stm için eksiksiz zamanlayıcıdır.
* Motor kontrolü ve digital güç dönüştürme uygulamalarında kullanılır.
* TIM1

Terimler;

1.Prescaler; Zamanlayıcı değişen bir faktöre böler.Zamanlyıcı 24Mhz de bağlı ise 24 eş parçaya bölerse 1 Mhz ye getirir.

2.CounterMode; Zamanlayıcının sayım yönünü tanımlar ve aşağıdaki değerlerden birini alabilir. Bazı sayma modları yalnızca genel amaçlı ve gelişmiş zamanlayıcılarda kullanılabilir. Temel zamanlayıcılar için yalnızca TIM\_COUNTERMODE\_UP tanımlanır.

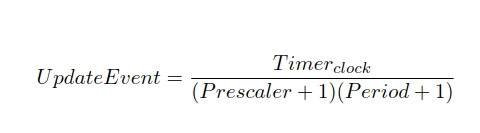
* TIM\_COUNTERMODE\_UP
* TIM\_COUNTERMODE\_DOWN
* TIM\_COUNTERMODE\_CENTERALIGNED

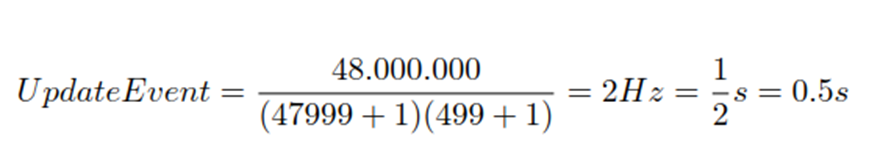


3.clockDıvıson; Bu bit alanı, dahili zamanlayıcı saat frekansı ile ETRx ve TIx pinlerindeki dijital filtreler tarafından kullanılan örnekleme saati arasındaki ayrım oranını gösterir. Aşağıdaki bir değeri alabilir ve yalnızca genel amaçlı ve gelişmiş zamanlayıcılarda kullanılabilir.

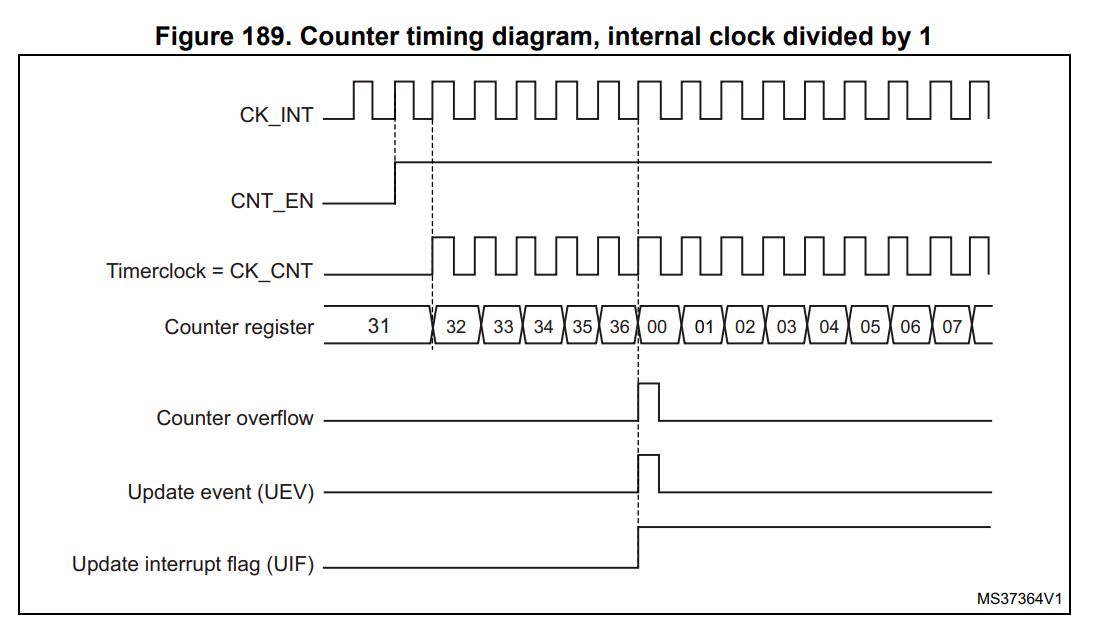
* TIM\_CLOCKDIVISION\_DIV1
* TIM\_CLOCKDIVISION\_DIV2
* TIM\_CLOCKDIVISION\_DIV4

4.RepetitionCounter; Belirlenen aralıkta taşmanın kaç kez gerçekleştiğini söyler.Gelişmiz zamalaıcıda kullanılır.

Timer update even hesaplaması;



COUNTER MODE;Sayaç 0'dan otomatik yeniden yükleme değerine (TIMx\_ARR kaydının içeriği) kadar sayar, ardından 0'dan yeniden başlar ve bir sayaç taşması olayı oluşturur.

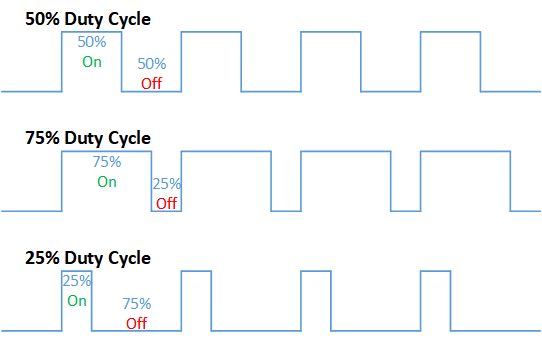


INTERRUPT MODE;

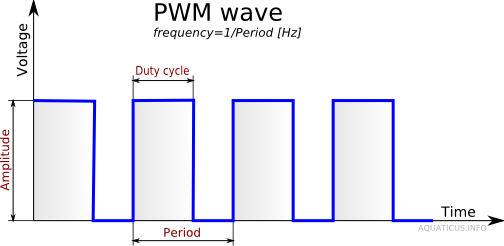
* Timer işlemleriyle birlikte interrupt işlemlerini yapıyoruz.

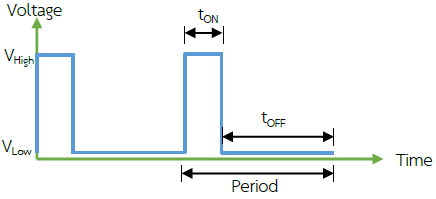
PWM MODE;

**3)PWM Mode(Output Compare):**

**Pulse Width Modulation (Darbe-Pals Genişlik Modülasyonu ):**

PWM sinyalin genişliklerini kontrol ederek, çıkışta üretilmek istenen sinyalin elde edilmesi tekniğidir.



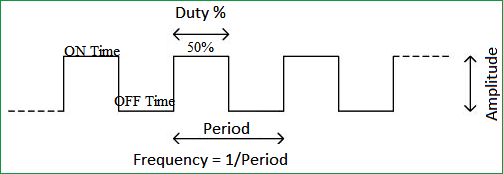


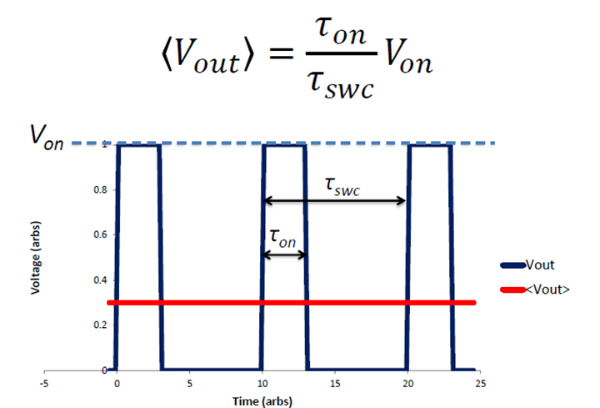
**Duty Cycle(Görev Döngüsü) Formula:**

Görev döngüsü yapılan işlemin periyodunu belirtir.





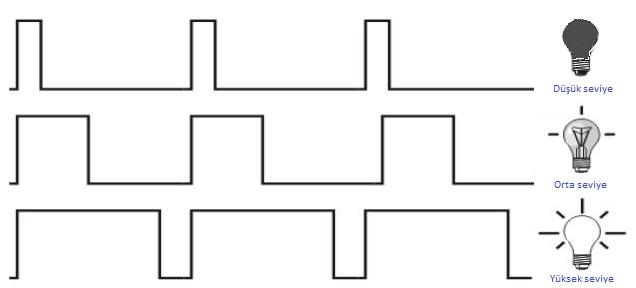


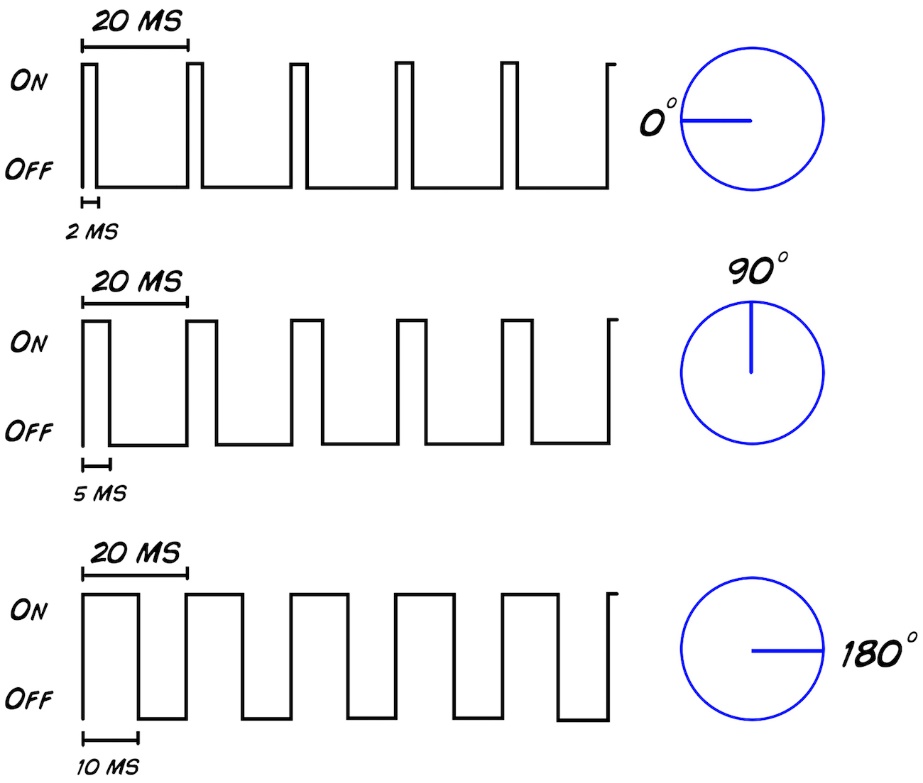
**Output Voltage(Vout) Formula(Çıkış Voltajı):**

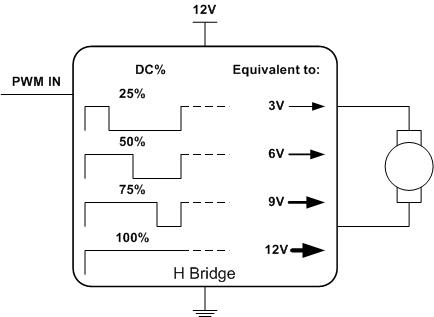




**PWM Uygulama Alanları:**

1)Ledin Parlaklığını Kontrol Edebiliriz.

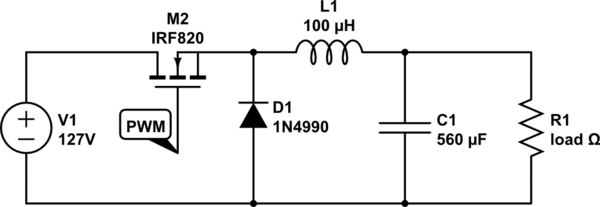
2)Servo Motor Kontrolü Yapabiliriz.

3) Motor Hız Kontrolü Yapabiliriz.

4) Haberleşme Sistemleri:

Telekomünikasyon sistemlerinde sinyallerin bozulmadan taşınmasında PWM tekniği kullanılmaktadır.

5) Voltaj Regülatör Devreleri:(Buck ve Boost Converter)



6) Ses Efektleri ve Yükselticiler:

Osilatörler ile birlikte bazı ses efektlerini elde etmek için PWM tekniği kullanılabilir.

**PWM Advantages:**

1)Güç tüketimi az.

2)Sinyalleri uzun mesafelere bozulmadan taşımak için yüksek verimli.

3)Uygulaması kolay.

**STM32F Kartımız için PWM Frekans ve Duty Cycle Hesaplama:**

**Frekans Hesaplama:**







**Tim Pulse Hesaplama:**

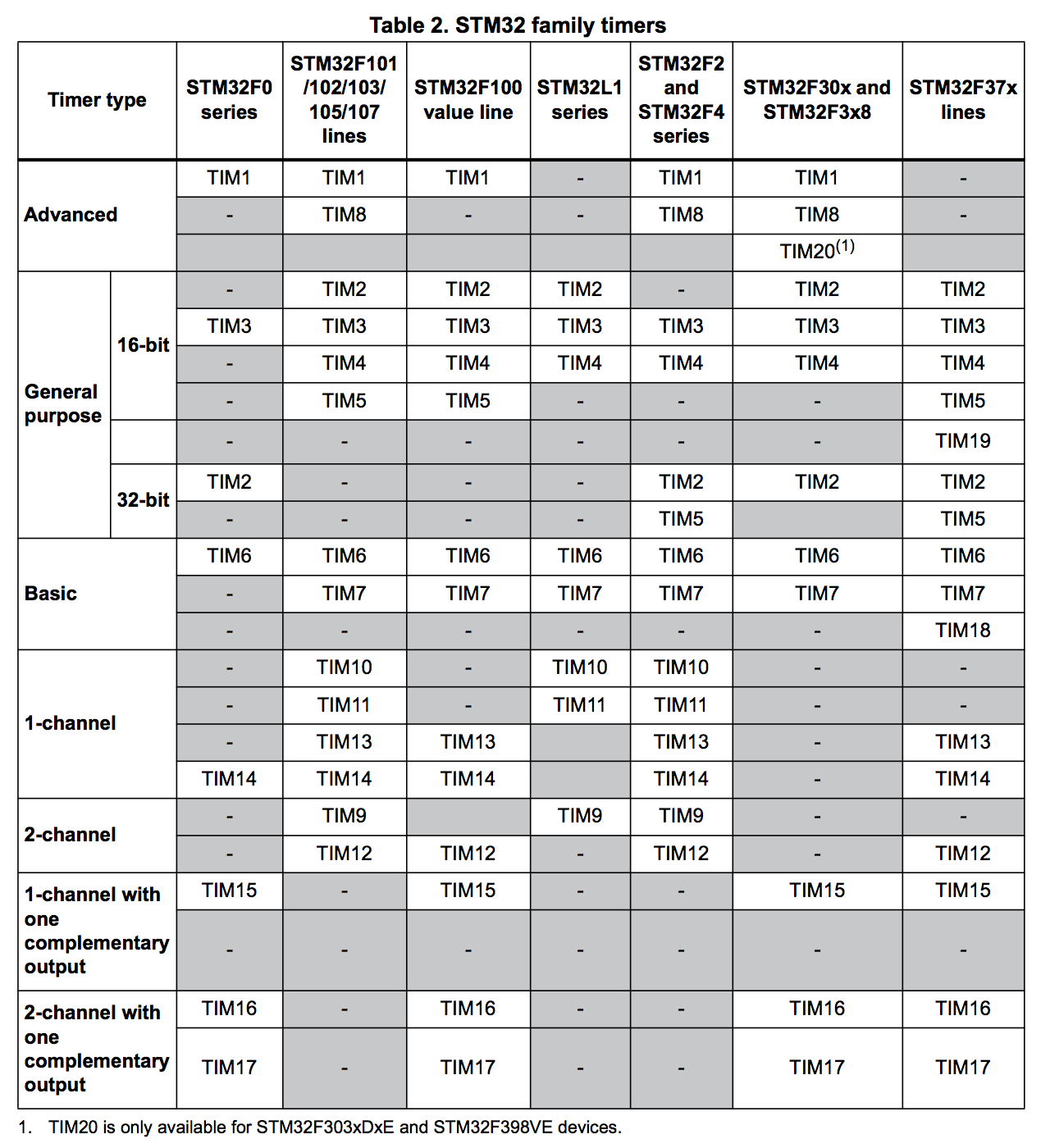


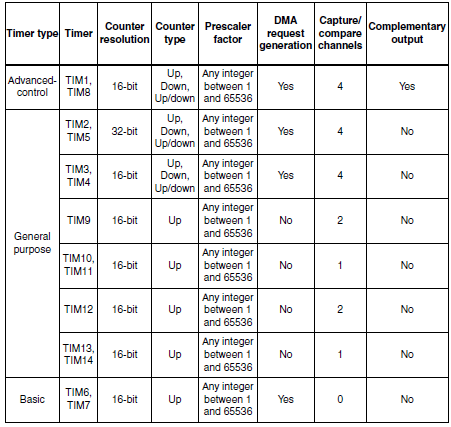


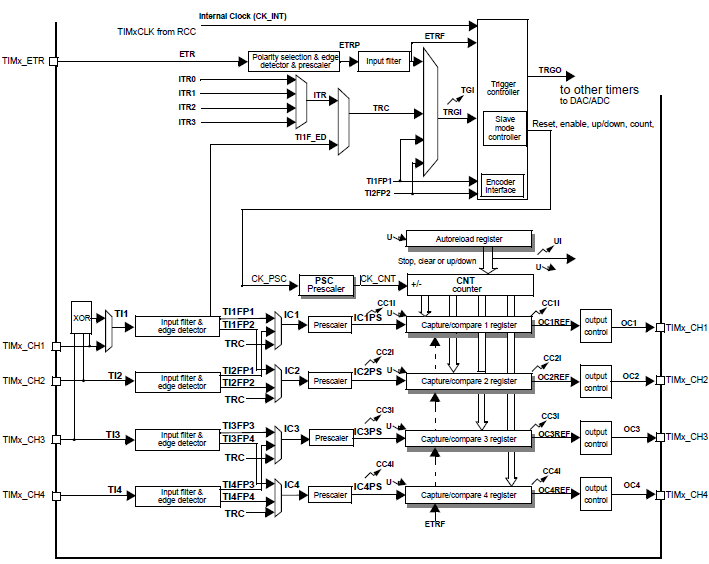






****





• **OCMode:** Çıktı karşılaştırma modunu belirler .

• **Pulse :** Bu alanın içeriği CCRx kaydının içinde saklanır ve çıktı tetiklemek için kullanılır.

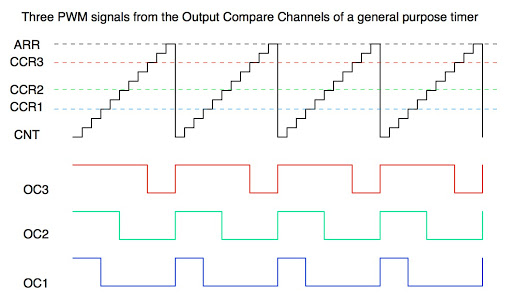
• **OCPolarity:** CCRx kaydı CNT ile eşleştiğinde çıkış kanalı polaritesini tanımlar.

• **OCNPolarity:** Tamamlayıcı çıkış polaritesini tanımlar. Yalnızca TIM1 ve TIM8 gelişmiş zamanlayıcılarında bulunan ve ek özel kanallarda tamamlayıcı sinyaller oluşturmaya izin veren bir moddur. Bu özelliği özellikle motor kontrol uygulamaları için tasarlanmıştır.

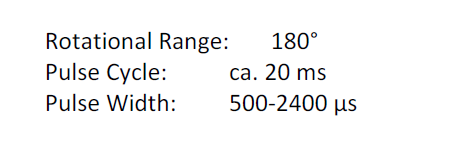
• **OCFastMode:** Hızlı mod durumunu belirtir. Bu parametre yalnızca PWM1 ve PWM2 modunda geçerlidir ve TIM\_OCFAST\_DISABLE ve TIM\_OCFAST\_ENABLE değerlerini alabilir.

• **OCIdleState:** Zamanlayıcı boşta durumu sırasında kanal çıkış karşılaştırması pin durumunu belirtir. TIM\_OCIDLESTATE\_SET ve TIM\_OCIDLESTATE\_RESET değerlerini varsayabilir. Bu parametre yalnızca TIM1 ve TIM8 gelişmiş zamanlayıcılarında kullanılabilir.

**• OCNIdleState:** Zamanlayıcı boşta durumu sırasında tamamlayıcı kanal çıkışının pin durumunu karşılaştırmasını belirtir. TIM\_OCNIDLESTATE\_SET ve TIM\_OCNIDLESTATE\_RESET değerlerini alabilir. Bu parametre yalnızca TIM1 ve TIM8 gelişmiş zamanlayıcılarında kullanılabilir.



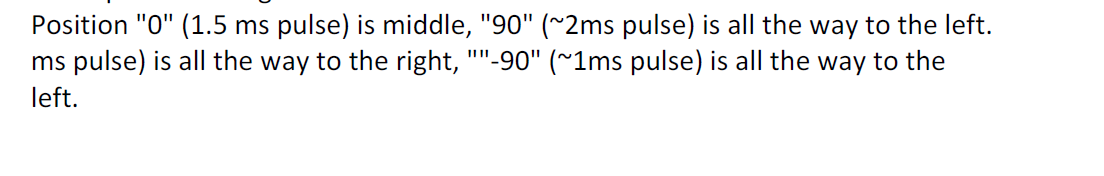
SERVO MOTOR****



**Frekans:**

**Period:**

**Tim Pulse:**



**SysTick Timer:**

SysTick, tüm STM32 mikrodenetleyicileri tarafından sağlanan Cortex-M çekirdeğinin içinde yer alan özel bir zamanlayıcıdır. Diğer zamanlayıcıların input capture / output compare gibi özellikleri olduğu için, bu zamanlayıcıları sıradan zamanlama için kullanmak, işlemci kaynaklarını boşa harcamak anlamına gelmektedir.

Delay foksiyonlarında Systick timer kullanmak işlemcimizi yormaz, güç tüketimini azaltır.

