

## FreeRTOS' DA BULUNAN SOFTWARE TIMERS NE İŞE YARAR?

Belirli bir süre sonra çalışması istenen bir işlevi belirli bir zaman aralığında tekrarlayan veya tek bir kez çalışması istenen bir zamanlamayı sağlar. Bunlar, harici bir donanım zamanlayıcısına ihtiyaç duymadan yazılım tabanlı olarak çalışır ve RTOS işletim sistemi tarafından yönetilir. Örneğin, bir görev sürekli olarak veri toplamaya çalışabilir, ancak belirli bir aralıkla toplanan verilerin işlenmesi gerektiğinde software timer kullanılabilir. Software timer, belirli bir süre sonra verilerin işlenmesi için bir işlev çağırabilir. Bu işlev, verilerin toplanması ve işlenmesi gibi görevleri yapabilir. Software timer'lar, öncelikli işlemlerin aksamasını önlemek ve sistem performansını artırmak için kullanılabilir.

## HARDWARE TIMER İLE SOFTWARE TIMER ARASINDAKİ FARKLAR NEDİR?

**Hardware Timer:** Donanım tarafından desteklenir ve harici bir zamanlayıcısı olan sistemlerde daha hızlı ve doğru çalışır. Hardware timer'lar ayrıca daha fazla zamanlama özelliği sunabilir, ancak daha kompleks ayrıca harici donanım gerektirir (TIM1, TIM2 vb) olabilir. **Software Timer:** RTOS tarafından desteklenir ve yazılım tabanlı olarak çalışır. Software timer'lar, harici bir zamanlayıcıya ihtiyaç duymadan daha kolay kullanılabilir harici bir donanıma ihtiyaç duymaz. Software timer'ların çalışma hızı RTOS'un performansına bağlıdır. Bu farklı özellikler, hangi zamanlayıcıyı kullanmanız gerektiği konusunda sizi yönlendirecektir. Eğer uygulamanızın çok yüksek doğruluk ve hız gerektiren zamanlama gereksinimleri varsa, hardware timer tercih edilmelidir. Eğer uygulamanızın zamanlama gereksinimleri daha az hassas ve daha az maliyetli bir çözüm arıyorsa, software timer tercih edilmelidir.

## FreeRTOS' DA BULUNAN SOFTWARE TIMER İLE LEDİ TOOGLE YAPMA PROJESİ

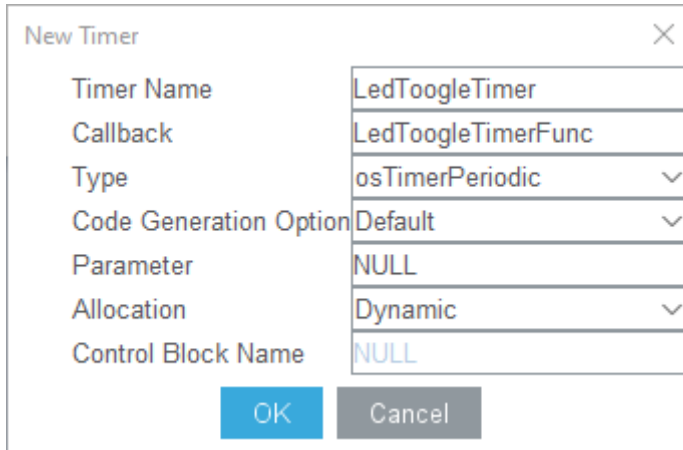
Bu projede Software timer' ı aktifleştirip 100 milisaniyede bir ledleri toogle yapmasını sağlayacağız.

PA5 pinini çıkış ve PC13 pinini giriş olarak ayarlayıp saat ve konfigrasyon ayarlarını yaptıktan sonra FreeRTOS' u aktifleştirip **Software timer definitions** kısmından **USE\_TIMER** parametresini ENABLED olarak düzenliyoruz

The screenshot shows the STM32CubeIDE interface. The 'Pinout & Configuration' tab is active. The 'FREERTOS Mode and Configuration' section is expanded, showing the 'Configuration' tab. The 'Software timer definitions' section is highlighted with a red box, showing the following settings:

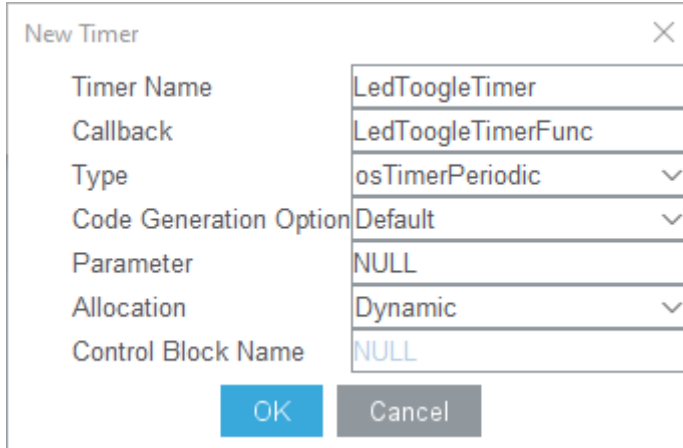
Parameter	Value
USE_TIMERS	Enabled
TIMER_TASK_PRIORITY	2
TIMER_QUEUE_LENGTH	10
TIMER_TASK_STACK_DEPTH	256 Words

Daha sonra Timers and Semaphores kısmından Software Timer' ımızı oluşturuyoruz.



CTRL + S basıp kodumuzu oluşturuyoruz.

Timer' ımızın 1000 milisaniyede 1 tetiklenip TimerLedToogleFunc() adlı fonksiyonu çağırması için gerekli kodları yazıyoruz.



Ardından LedToogleTimerFunc() adlı fonksiyonu aşağıdaki gibi dolduruyoruz.

```
/* LedToogleTimerFunc function */  
void LedToogleTimerFunc(void const * argument)  
{  
    /* USER CODE BEGIN LedToogleTimerFunc */  
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);  
    /* USER CODE END LedToogleTimerFunc */  
}
```

Software Timer' ın zamanlamasını başlatmak için osTimerStart() fonksiyonunu çağırıyoruz.

```
/* Create the timer(s) */  
/* definition and creation of LedToogleTimer */  
osTimerDef(LedToogleTimer, LedToogleTimerFunc);  
LedToogleTimerHandle = osTimerCreate(osTimer(LedToogleTimer), osTimerPeriodic, NULL);  
osTimerStart(LedToogleTimerHandle, 1000);  
//LedToogleTimer adlı timer 1 saniyede 1 tetiklenecek ve LedToogleTimerFunc adlı fonksiyon  
//çağırılıp led' e toggle işlemi yaptırılacak.
```

## NOT:

Bu projeyi xTimerCreate() ve xTimerStart() API' lerini kullanarakta yapabiliriz

```
TimerHandle_t xLedToggleTimerHandler;
xLedToggleTimerHandler = xTimerCreate("LedToggleTimer", pdMS_TO_TICKS(50), pdTRUE, 0, LedToggleTimerFunc);
//5 adet parametre alır bu parametreler
// 1) Timer' ın adı
// 2) Timer' ın kaç milisaniyede bir tetikleneceğini belirtir
// 3) Bu parametre timer' ın 1 keremi yoksa sürekli mi çalışacağını belirtir (pdTRUE: SÜREKLİ)
// 4) Timer' ın ID' si
// 5) Timer callback fonksiyonu

if(xTimerStart(xLedToggleTimerHandler, 0) == pdPASS)
{
    //Timer doğru bir biçimde başlatıldı
}
//Not: xTimerStart API' si eğer doğru bir biçimde çalışırsa pdPASS dönerken çalışmaz ise pdFAIL döndürür
//Not: xTimerStart API' sinin 2. parametresi ilk timer' ın ne kadar süre sonra başlayacağını belirtir.
```

xTimerCreate() fonksiyonu ile timer' ı oluşturup ardından xTimerStart() fonksiyonu ile timer' ı başlatıyoruz.



