

```

#include "stm32f4xx.h" // STM32F4xx serisi için kütüphane
static volatile unsigned int brightness_step = 16; // Parlaklık adımı değişkeni
/**@brief TIM4 kesme işleyicisi. Bu fonksiyon, Zamanlayıcı 4 (TIM4) tarafından oluşturulan kesme isteklerini işler.
 * @retval None
 */
void TIM4_IRQHandler(void)
{
    TIM4->SR &= ~TIM_SR_CC1IF; // Kesme bayrağını sıfırla
    TIM4->CCR1 += brightness_step; // PWM parlaklığını artır veya azalt
    if (TIM4->CCR1 >= 16000 - 16 || TIM4->CCR1 <= 16) // Eğer parlaklık sınırlara ulaştıysa,
        brightness_step = -brightness_step; // Parlaklık adımını tersine çevir
}
int main(void)
{
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN; // GPIOD portunu etkinleştir
    GPIOD->MODER |= GPIO_MODER_MODE12_1; // GPIOD'nin 12. pinini alternatif fonksiyona ayarla
    GPIOD->AFR[1] |= GPIO_AFRH_AFSEL12_1; // GPIOD'nin 12. pininin alternatif fonksiyonunu seç
    RCC->APB1ENR |= RCC_APB1ENR_TIM4EN; // Zamanlayıcı 4'ü etkinleştir
    TIM4->CCER |= TIM_CCER_CC1E; // Zamanlayıcı 4, kanal 1 çıkışını etkinleştir
    TIM4->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; // Çıkış modunu PWM moduna ayarla
    TIM4->CCMR1 |= TIM_CCMR1_OC1PE; // Çıkışı ön yükleme etkinleştir
    TIM4->DIER |= TIM_DIER_CC1IE; // CC1 kesme etkinleştir

    NVIC_EnableIRQ(TIM4_IRQn); // TIM4 kesmesini etkinleştir
    TIM4->ARR = 16000 - 1; // Sayıcıyı önceden yükleyiciyi ayarla
    TIM4->CCR1 = 16; // Kanal 1 çıkışını ayarla
    TIM4->CR1 = TIM_CR1_CEN; // Zamanlayıcıyı etkinleştir

    while (1); // Sonsuz döngü
}

```

Bu örnek, STM32F4xx serisi mikrodenetleyicilerdeki PWM (Pulse Width Modulation - Darbe Genişlik Modülasyonu) işlevselliğini kullanarak bir LED'in parlaklığını kademeli olarak artırıp azaltır. Kod, Timer 4 (TIM4) kullanılarak gerçekleştirilir.

Timer 4, çıkış kanalı 1 (CC1) üzerinden PWM sinyali üretmek için yapılandırılır. PWM sinyali, LED'in parlaklığını kontrol etmek için kullanılır.

Örneğin başlangıcında, Timer 4 periferikinin saat sinyali etkinleştirilir ve ilgili GPIO pinleri yapılandırılır. Timer 4, PWM modunda çalışacak şekilde yapılandırılır ve gerekli ayarlar yapılır.

TIM4_IRQHandler() fonksiyonu, Timer 4 tarafından üretilen kesme isteklerini ele alır. Bu kesme işleyicisi, PWM sinyalinin darbe genişliğini kademeli olarak artırıp azaltmak için kullanılır.

Ana döngüde, kesmeler etkinleştirilir ve Timer 4 başlatılır. Program sonsuz bir döngüde bekler, bu nedenle LED'in parlaklığı kademeli olarak artar ve azalır.

Bu örnek, Timer 4 periferikini kullanarak kademeli artan-azalan PWM sinyali üretmek için temel bir yapı sağlar.