# Timer Setup for PWM

*STM32L476VG Discovery board*

## Overview:

The Discovery board has a variety of timers ranging from very substantial supporting functions through very basic timers. This document discusses configuring a basic Pulse Width Modulation (PWM) system. This should apply to all timers.

## Documents:

You need to have the Reference Manual (RM) RM0351 STM32L4x6 advanced ARM-based 32-bit MCUs manual: http://www.se.rit.edu/~swen-563/resources/STM32L476/STM32L476VGT6%20Reference%20manual.pdf.

You also need to have the STM32L476xx datasheet (DS) http://www.se.rit.edu/~swen-563/resources/STM32L476/STM32L476VGT6%20Datasheets.pdf

The following sections will reference these documents as RM and DS respectively.

## Timer Selection:

First determine what features are needed. This board has two advanced timers (TIM1/TIM8), four general purpose timers (TIM2 through TIM5) and three additional reduced feature set general purpose timers (TIM15 through 17). Timers TIM2 and TIM5 are 32 bit timers which provide the obvious range advantage over the 16 bit timers. Using these 32 bit timers will allow you to use a finer resolution for more gradations of position when controlling the servo motor positions.

Refer to the "main features" overviews in the RM Chapter 26 (page 745), Chapter 27 (page 850), and Chapter 28 (page 923) to select an appropriate timer.

## PWM Timing Analysis:

Decide on how fast to run the clock in the timer. For the servo motors you need to have a period of 20 milliseconds with a pulse width ranging from around 0.4 to 2 milliseconds. Setting the prescaler to 100 microseconds per count allows you to use a value of 200 for the period and from 4 to 20 for the pulse width. This provides more than enough gradation to have 6 evenly spaced positions.

## Timer Configuration:

First be sure to enable the clock for the selected timer in the RCC->APB1ENR1 register. Then load your prescaler value into the TIMx->PSC register. Be sure to force the load of the new prescaler value by creating an update event using the TIMx->EGR register.

## Timer PWM Configuration:

For your selected timer set one of its channels as an output to a GPIO pin by connecting via the alternate function on an acceptable input pin on P1 or P2. Refer to the Alternate Functions section of the "GPIO and Alternate Function Setup" guide for how to make this determination.

Refer to the selected timer's "PWM mode" in the RM. For timers TIM2 through TIM5 go to page 874.

Set up the CCMRx (capture/compare mode register) register for output compare mode. For timers 2 through 5 refer to RM page 906 for these bits. Then set the Output Compare Preload Enable bit for this same register.

Next, enable the Auto-Reload of the preload (ARPE bit) in the timer's CRx (Control Register).

Next, enable the corresponding channel output bit in the timer's CCER (capture/compare enable register). For timers 2 through 5 see RM page 911.

Lastly, set the update bit in the EGR register to force an update (this loads all of these settings into the timer).

## PWM Period:

Note that the number loaded here is the number of counts of the pre-scaled counter – not the number of 80 MHz clock cycles.

The ARR (auto-reload register) controls the period of the PWM. See RM page 914 for timers 2 through 5. You may want to set the update bit in the EGR to force it to take effect immediately. This is required if the timer is not currently running a PWM signal. If it is running the ARR value will take effect that the completion of the current period.

## PWM Pulse:

Note that the number loaded here is the number of counts of the pre-scaled counter – not the number of 80 MHz clock cycles.

The CCRx (capture/compare register) controls the pulse width. For timers 2 through 5 see RM page 914. As described for the Period you may want to set the update bit in the EGR register.

## PWM Start/Stop:

To run the PWM set the TIM_CRx_CEN bit in the TIMx->CRx register. Clear this bit to stop input capture.

To confirm it is running you can check that the timer's CNT (counter) register is changing and then resetting back after the period ends. You can initially use a very long period to simplify development.