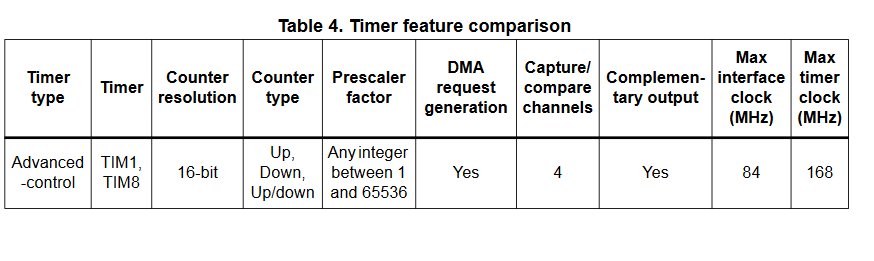
ODHS -Timer interrupt LED toggle

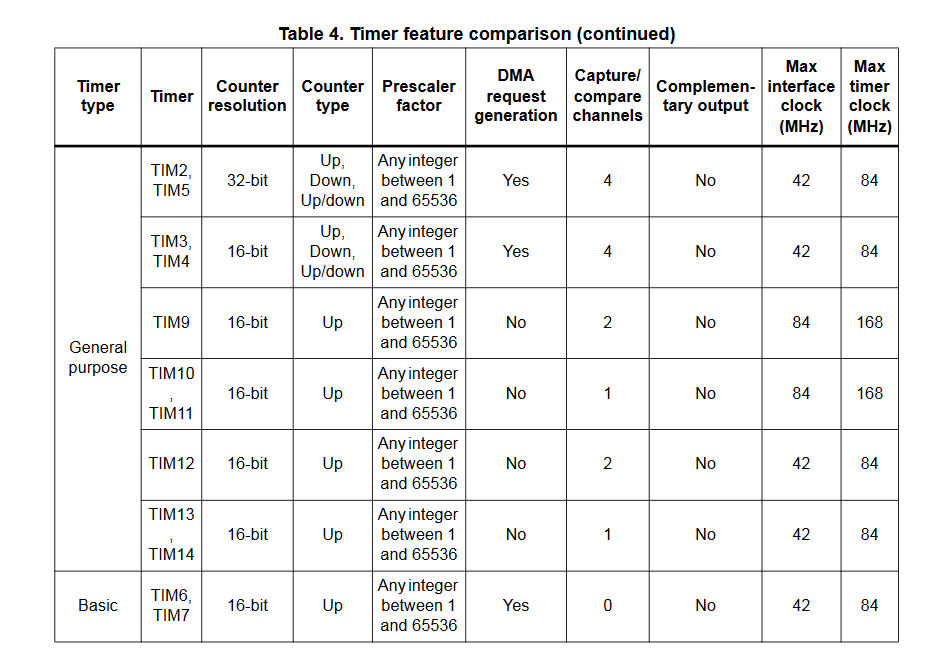
*When a timer overflows, it causes an interrupt which toggles an led.*

Open STM32cubeIDE and select the STM32F407VG as the processor

Connect the RCC clock HSE to crystal/ceramic resonator

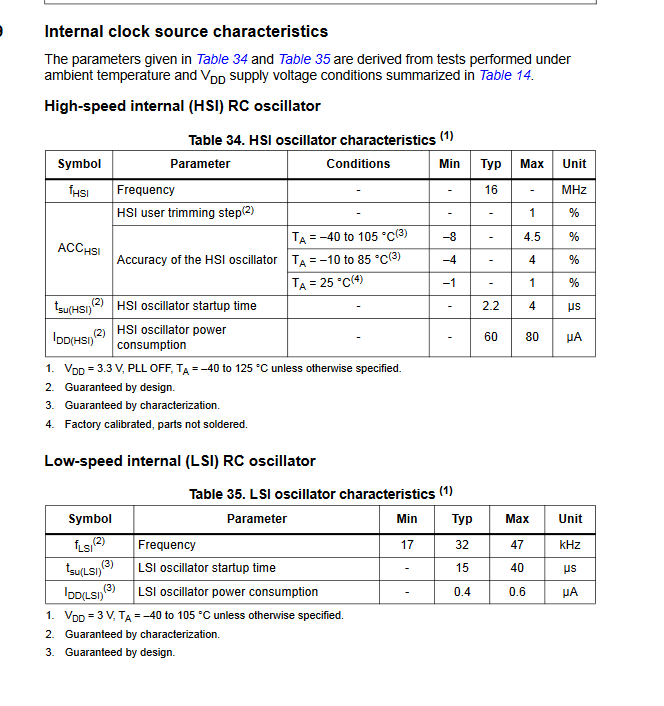
According to the datasheet the timers available are:-



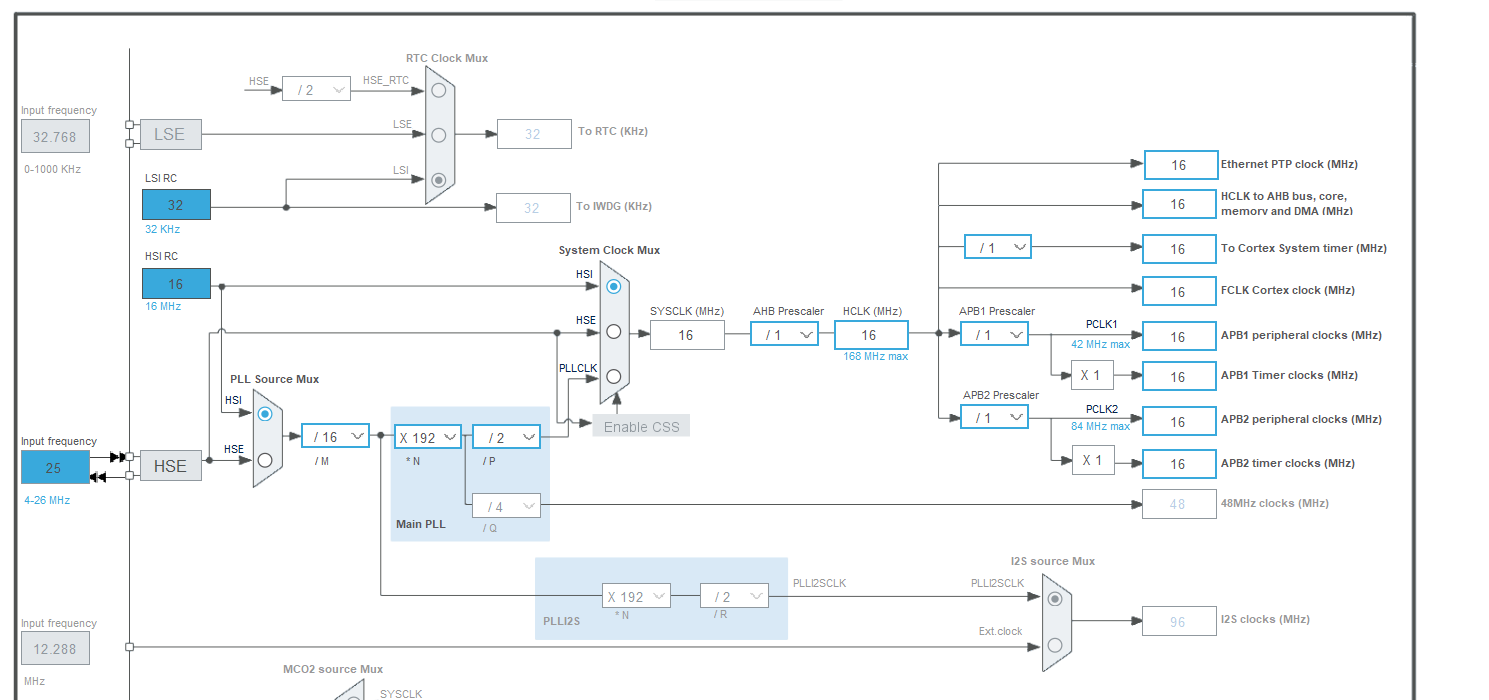


Here we use the general purpose timers say timer TIM3.

Now the clock speed according to the datasheet



As the RCC is set to HSI mode the clock speed is 16MHz

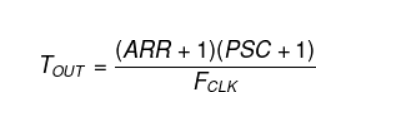


In the pinout of TIM3 , clock source > internal clock

To configure the timer

Tout is the desired output time interval as in for how long should the timer

The timer interval is set by prescale value, AutoReload/counter Period and clock speed



To calculate how long the timer will run for

Clock frequency=16 Mhz

Now when PSC prescaler = (say 2)

This means timer clock speed = clock frequency/Prescaler value = 16MHz/2=8MHz

Thus for each count/ tick of the time will take time = 1/(8MHz) =1.25\*10-7 second

Now ,the maximum value of the timer depends upon the ARR register size (in the case of TIM3 =16 bits)

the maximum ARR register value = 216 -1 = 65536-1= 65535

Hence the time taken for timer to overflow = 65536\*1.25\*10-7  = 8.192 ms

Now if the prescaler =2000

Then timer clock frequency = 16MHz/2000= 8Khz

1 tick time= 1/(8KHz) =0.125 ms

ARR register value = 216 -1 = 65536-1= 65535 (kept default)

time taken for timer to overflow = Tout = 65536\*0.125 = 8192 ms =8.192 second

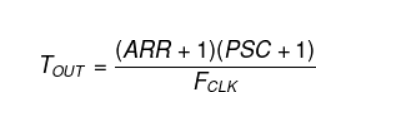
**NOTE:-** putting ARR and PSC values corresponding to desired Tout allows us to not use the delay function.

In the case of STM32F407… the clock speed is FCLK =16Mhz and ARR register width for TIM3 is 16 bits(see table above)

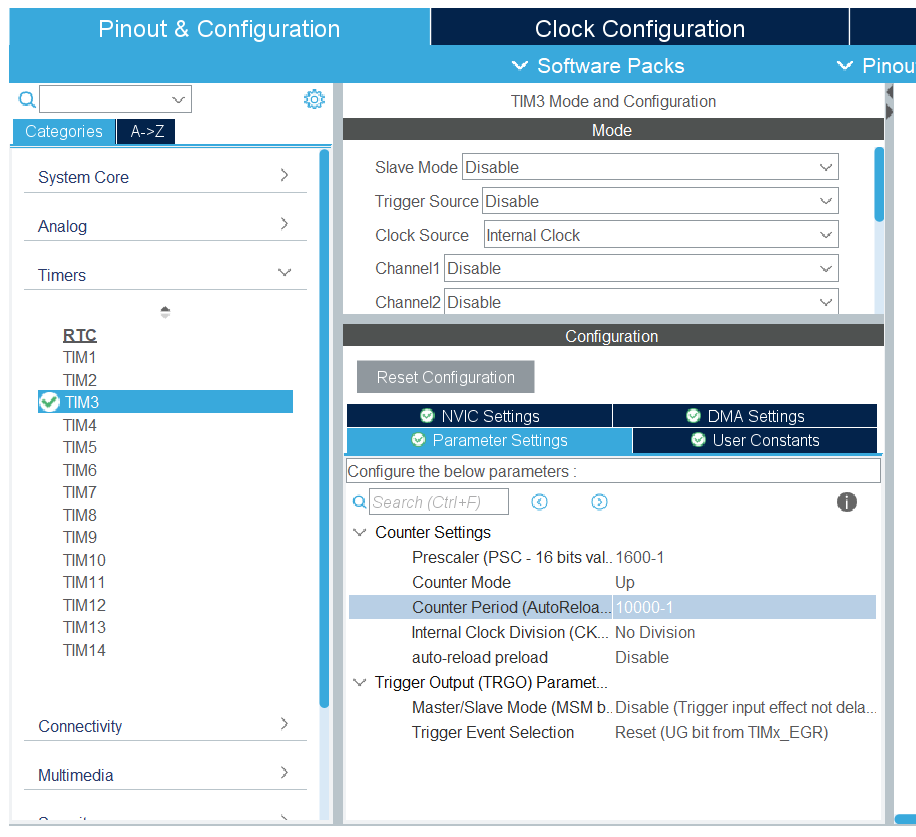
Thus for a timer interval (time of say 1 sec (1000ms) thus the ARR and PSC should be :-

FCLK =16Mhz

Let PSC (Pre Scaler) = 1600 -1 = 1599

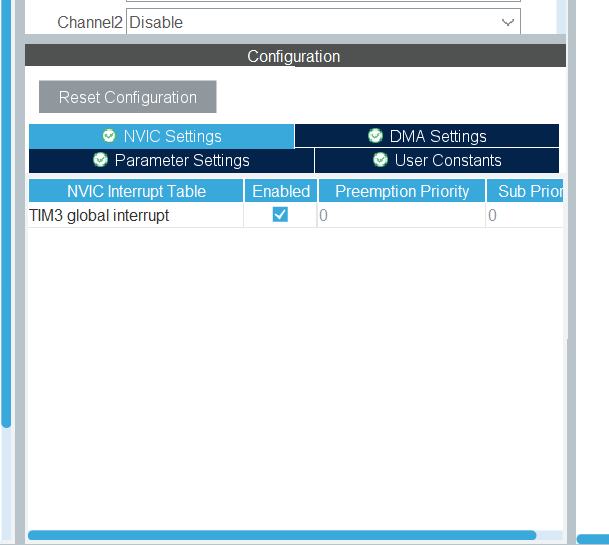


Then from equation above Counter Period (ARR) = 10000-1=9999



In the NVIC (Nested Vectored Interrupt Controller ??) Settings

The TIM3 global interrupt is enabled

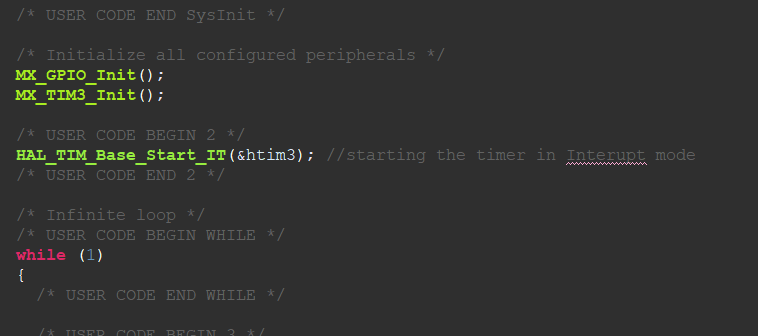


The project is saved and code is generated.

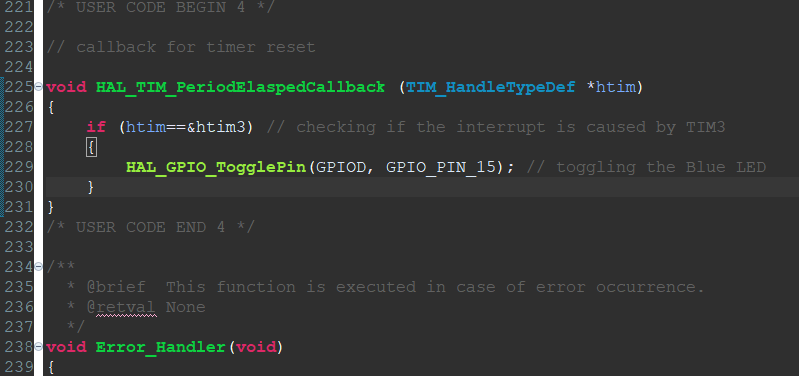
In the main function the timer needs to be started and every time it overflows and causes an interrupt , the LED say blue LED (PD15) should be toggled.

In main add the following to start the timer

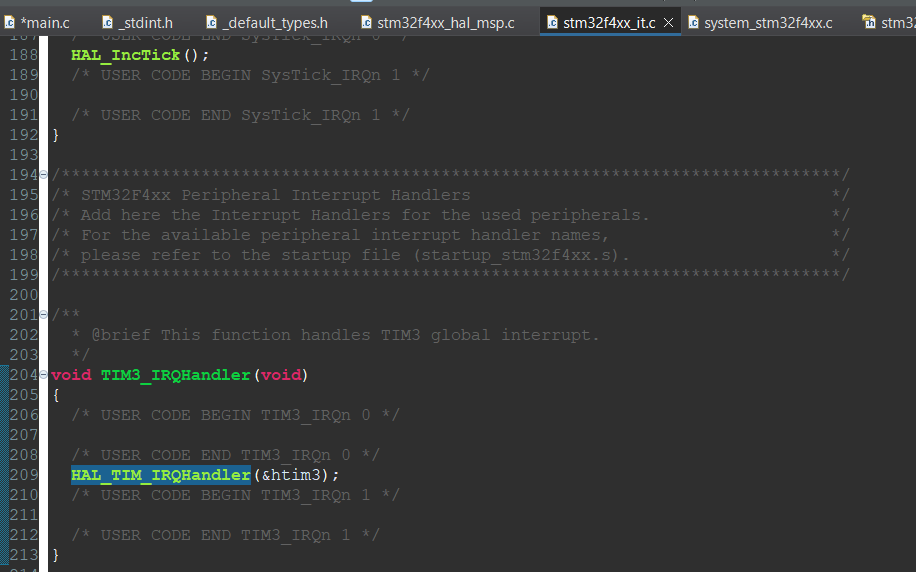
Here htim3 refers to the register corresponding to TIM3

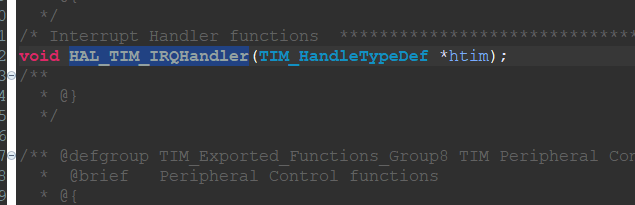


Write a new function in the main file to callback when timer resets



The interrupt here is handled by HAL library

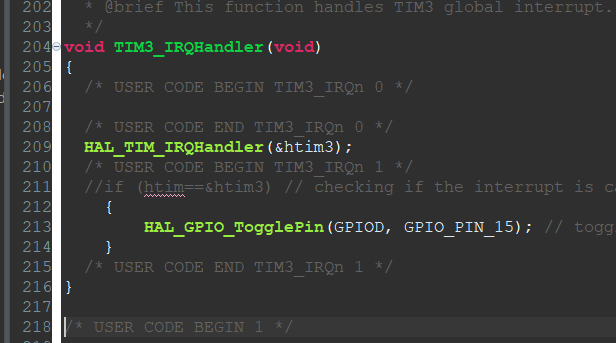




as here any timer can cause an interrupt, thus in the function it is checked whether the timer is TIM3 (htim3)

OR

Rather than having a new function to handle call back, modify the TIM3\_IRQHandler in stm32f4xx\_it.c



And in main just start the timer

NOTE:- as the code is working on an interrupt basis, no need to output the value and any general purpose timer works as the interrupt is a global interrupt which is handled by HAL library.

hopefully this works lol :)

Videos to follow

<https://youtu.be/AjN58ceQaF4?si=WcqRbqqTHd_oTHs2>

<https://www.youtube.com/watch?v=VfbW6nfG4kw&t=151s>

<https://community.st.com/t5/stm32-mcus/how-to-generate-a-one-second-interrupt-using-an-stm32-timer/ta-p/49858>

<https://deepbluembedded.com/stm32-timers-tutorial-hardware-timers-explained/>

<https://www.digikey.in/en/maker/projects/getting-started-with-stm32-timers-and-timer-interrupts/d08e6493cefa486fb1e79c43c0b08cc6>

timer calculations

<https://deepbluembedded.com/stm32-timer-interrupt-hal-example-timer-mode-lab/>