

CURSO DE STM32



Activar Windows
Vé a Configuración para activar Windows.

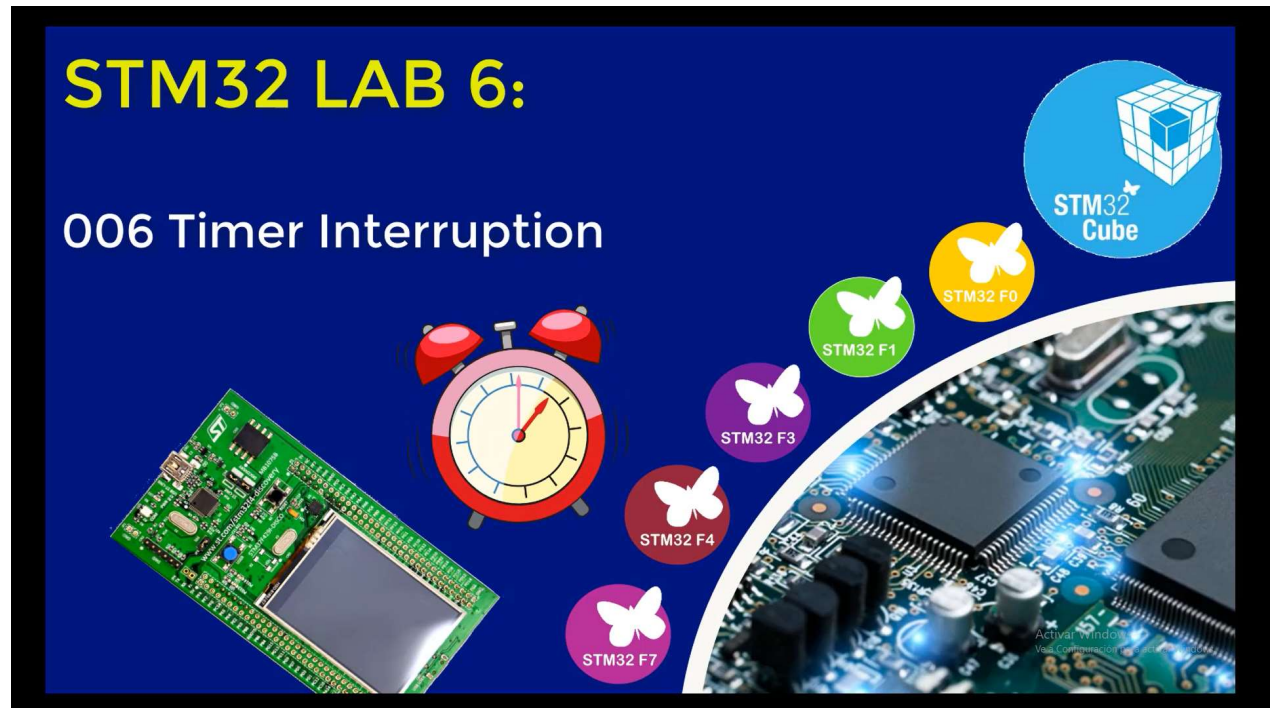
25 ENERO 2021

006_Timer_Interruption

Creado por: Ing. Christian Salazar

SECCION 6

006_Timer_Interruption



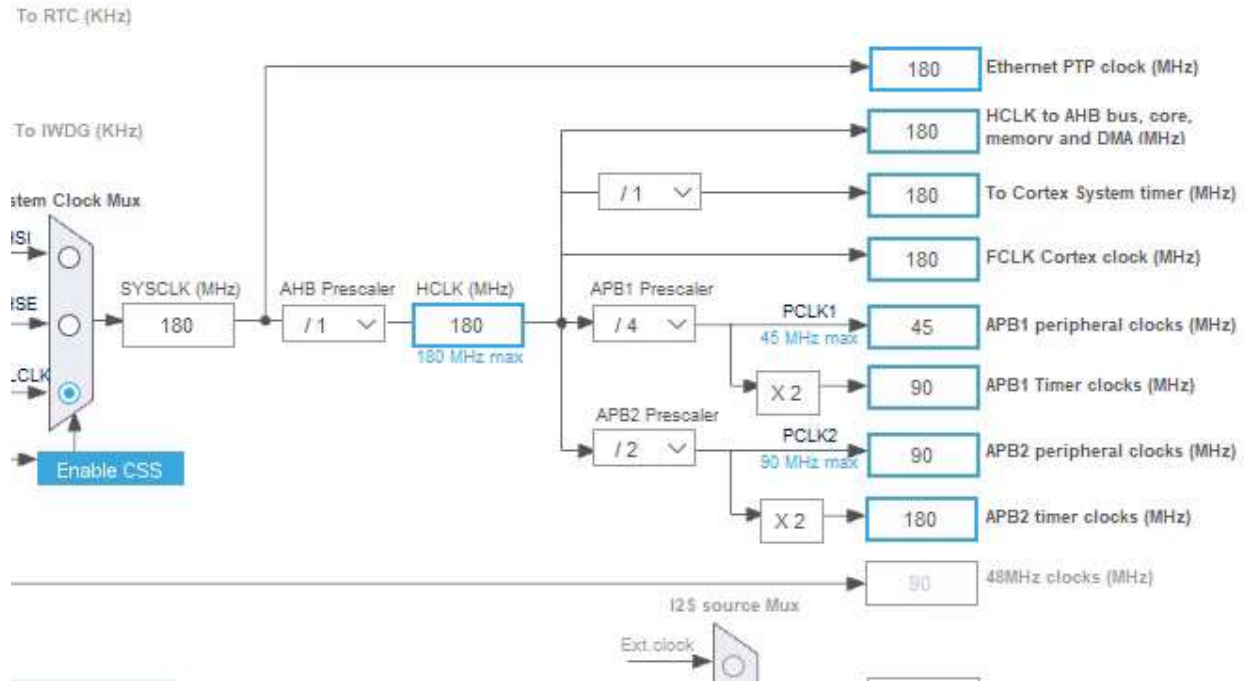
Que aprenderemos?

En este video calcularemos los valores del periodo y prescaler para que un timer nos genere una interrupción, probaremos cuando se habilita la opción de que el timer genera la interrupción periódicamente o solo una vez, usaremos el datasheet del microcontrolador para determinar la frecuencia a la que está oscilando el timer seleccionado.

“Usaremos HAL Drivers, lo cual nos ayudará en gran medida a portar y reciclar rutinas de código de un procesador de una Familia a otro de otra Familia”.

Puntos Clave

FRECUENCIA DEL TIMER



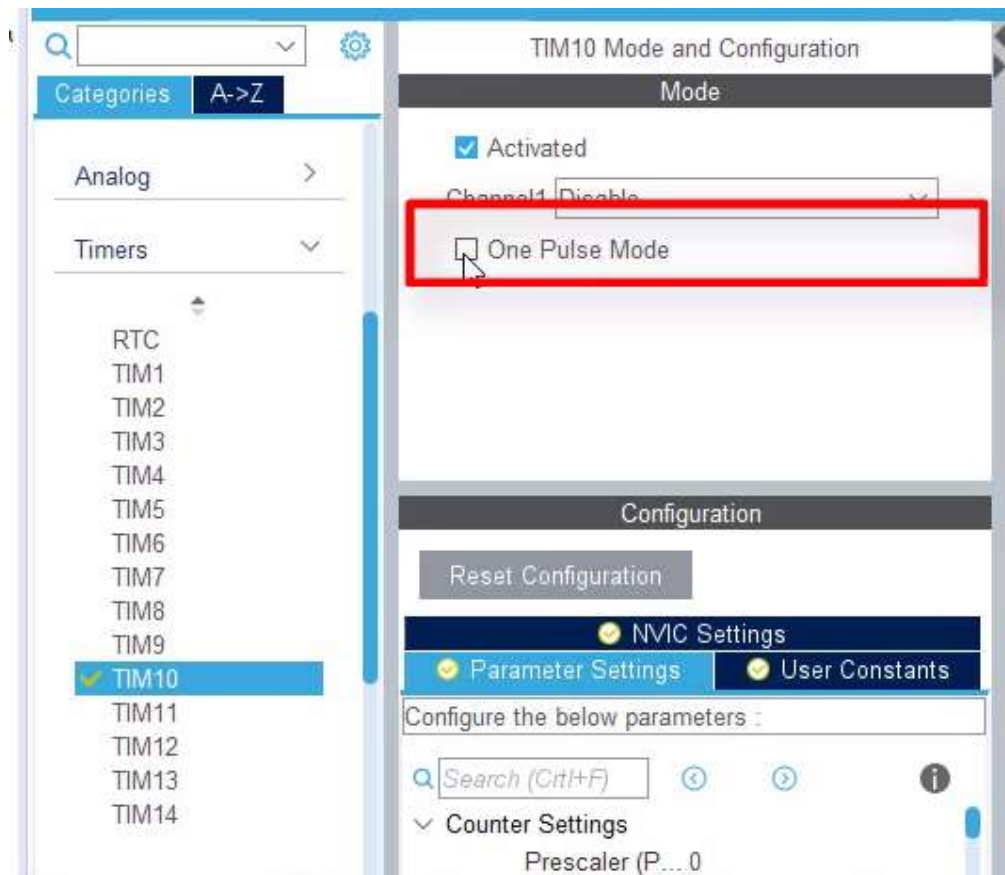
Como podemos observar en este caso tenemos 2 frecuencias a las que trabajan los timers (APB1 Timer Clocks y APB2 timer clocks) a 90 y 180 MHz respectivamente, entonces si por ejemplo como en el video seleccionamos el TIMER10 debemos buscar en el datasheet del microcontrolador a que Bus (APB1 or APB2) está conectado dicho timer.

Table 1. STM32F4xx register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4001 4800 - 0x4001 4BFF	TIM11	APB2	Section 19.5.12: TIM10/11/13/14 register map on page 694
0x4001 4400 - 0x4001 47FF	TIM10		
0x4001 4000 - 0x4001 43FF	TIM9		Section 19.4.13: TIM9/12 register map on page 684
0x4001 3C00 - 0x4001 3FFF	EXTI		Section 12.3.7: EXTI register map on page 387
0x4001 3800 - 0x4001 3BFF	SYSCFG		Section 9.2.8: SYSCFG register maps for STM32F405xx/07xx and STM32F415xx/17xx on page 294 and Section 9.3.8: SYSCFG register maps for STM32F42xxx and STM32F43xxx on page 301
0x4001 3400 - 0x4001 37FF	SPI4	APB2	Section 28.5.10: SPI register map on page 925
0x4001 3000 - 0x4001 33FF	SPI1	APB2	Section 28.5.10: SPI register map on page 925
0x4001 2C00 - 0x4001 2FFF	SDIO		Section 31.9.16: SDIO register map on page 1074
0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3		Section 13.13.18: ADC register map on page 430
0x4001 1400 - 0x4001 17FF	USART6		Section 30.6.8: USART register map on page 1018
0x4001 1000 - 0x4001 13FF	USART1		
0x4001 0400 - 0x4001 07FF	TIM8		Section 17.4.21: TIM1 and TIM8 register map on page 587
0x4001 0000 - 0x4001 03FF	TIM1		
0x4000 7C00 - 0x4000 7FFF	UART8	APB1	Section 30.6.8: USART register map on page 1018
0x4000 7800 - 0x4000 7BFF	UART7		

Podemos ver en el PDF RM0090 del STM32F429xx que el TIM10 está en el dominio de APB2 es decir que esta corriendo a 180 MHz, esta frecuencia es importante para poder calcular cierto tiempo requerido.

CONTEO DE TIMER UNA SOLA VEZ



One Pulse Mode es la opción que si la habilitamos hara que el timer realice su conteo al tiempo calculado UNA SOLA VEZ, lo hará nuevamente si le ordenamos nuevamente y asi sucesivamente contará una sola vez y nos generará la interrupción de tiempo solo cuando le ordenemos iniciar el conte con la función adecuada.

AUTOCARGA DEL TIMER (PERIODICO)

Asimismo con el caso anterior si no marcamos la opción One Pulse Mode estamos diciendo que nuestro timer va a generar una interrupción periódica del tiempo establecido con el cálculo del periodo y prescaler cuando le indiquemos con la función de inicio de conteo del timer, en ambos casos tenemos la opción de poner en

STOP el conteo del timer y manipular el valor actual del conteo a 0 para reiniciar cuando así lo desemos (TIM10->CNT=0).

CALCULO PERIODO Y PRESCALER PARA DETERMINAR UN TIEMPO

Para calcular y generar una interrupción a 1 milisegundo:

1. Calcular la frecuencia del tiempo requerido.

Para 1 mS, sabemos que $F(\text{Frecuencia}) = 1/T(\text{Tiempo})$, lo que nos da una frecuencia de 1 KHz.

2. Conocer la frecuencia del Timer (Datasheet), como vimos anteriormente el TIM10 está a 180 MHz.

3. Eliminamos la frecuencia requerida dividiendo.

$$\text{FreqReq} = 180\,000\,000\text{ Hz} / 1000\text{ Hz (1 KHz es 1000 Hz)}$$

$$\text{FreqReq} = 180\,000.$$

4. Calcular Periodo y Prescaler

Periodo y Prescaler son 2 factores que se multiplican para conocer el valor hasta el cual el timer debe contar.

Estos dos Factores son registros de 16 bits sin signo que quiere decir que pueden tener un valor desde 0 a 65535.

Con la fórmula $\text{FreqReq} = \text{Periodo} * \text{Prescaler}$, con lo antes calculado tenemos:

$$180\,000 = \text{Periodo} * \text{Prescaler}$$

Entonces lo que debemos hacer es asumir uno de los valores de tal manera que el número asumido sea entre 0 y 65535 y que el resultado del despeje de la otra variable en la fórmula anterior con el número asumido sea de igual manera entre 0 y 65535.

Asumiremos para el Periodo = 100

Despejando Tenemos que el Prescaler = 1800

Como vemos ambos cumplen que son de 0 a 65535.

Para configurar el timer y poner estos valores se pone restado en 1 es decir:

Periodo = $100 - 1 = 99$

Prescaler = $1800 - 1 = 1799$

Estos son los valores que realmente debemos asignar en la configuración, se pone estos valores ya que el timer empieza su conteo desde 0 y hay un ciclo en el salto desde 0 a 1.

Es como un Vector de una variable:

`int MyVariable [1800];`

El último espacio del vector poner a 0, tendríamos

`MyVariable [1799] = 0;`

Vemos que la posición 0 tiene su propio valor.