

PRÁCTICA INTERACTIVA: Simulación de un Timer de 16 bits de los microcontroladores AVR

Objetivo: Comprender el funcionamiento de un temporizador con funciones de Input Capture, Output Compare y PWM. Aprender sobre los registros a configurar para poner en funcionamiento un temporizador de 16 bits de AVR en sus distintos modos. Analizar estos modos de trabajo y evaluar su conveniencia en diferentes aplicaciones.

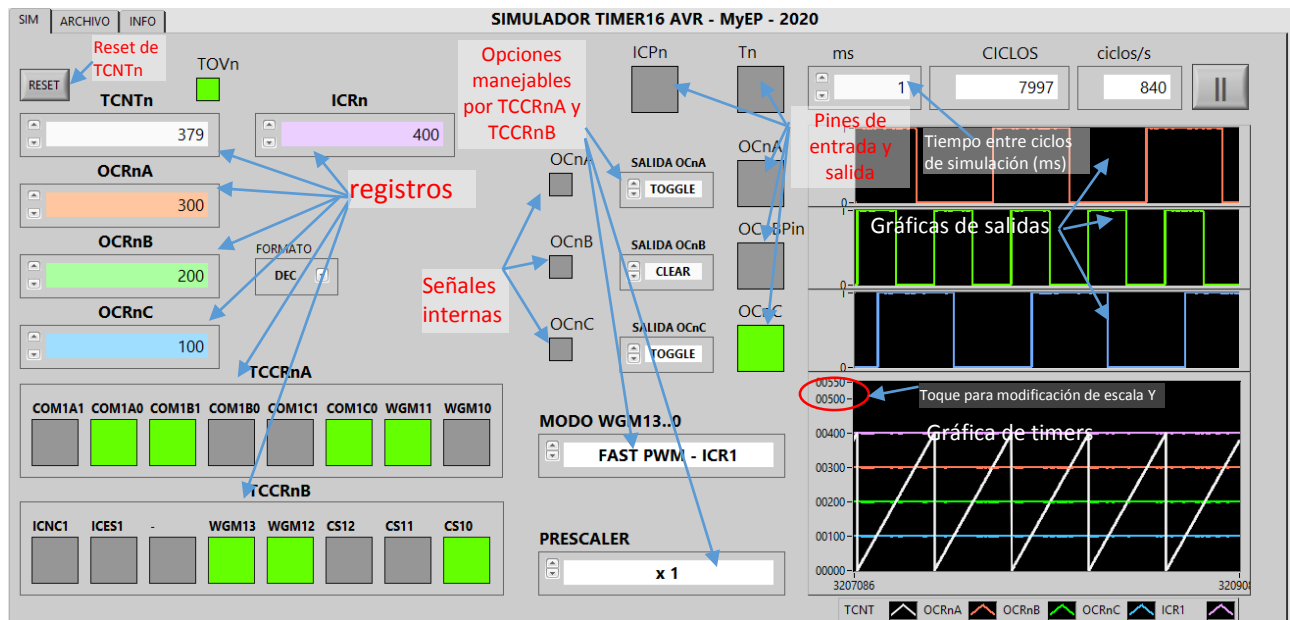
Requisitos previos: Haber leído el apunte “MyEP: Programación de Microcontroladores”, sección 7: temporización (introducción, págs 38 y 39) y sección 7.2: Temporización en AVR. Páginas 43 a 47. Se sugiere también visitar la sección **16** del manual de Atmega48...328P y la sección **18** del manual de Atmega640...2561 indicados en las referencias de aulaabierta, carpeta AVR.

Procedimiento

Bajar software de simulación SimTimer16AVR.exe, de aulaabierta. Descargar e instalar además el Runtime de Labview 8.6, enlace también indicado en la página de aulaabierta.

Ejecutar SimTimer16AVR.exe

La pestaña principal presenta el siguiente panel:



Allí observará siguientes los registros y su gráfica temporal simultánea:

TCNTn (TCNT1 para Timer1, TCNT3 para Timer3 etc)

OCRnA (OCR1A para Timer1, OCR3A para Timer3 etc)

OCRnB

OCRnC (no presente en Atmega328, sí en Atmega2560)

ICRn

Registros de control **TCCRnA** y **TCCRnB**

Los pines:

Salidas **OCnA**, **OCnB** y **OCnC** (este último no presente en el Atmega328) y sus gráficas temporales.

Entradas **ICPn** (input capture) y **Tn** (clock externo).

Las señales internas OCnA, OCnB y OcnC.

Los selectores MODO, PRESCALER y SALIDA OCnX que permiten configurar todas las posibilidades del módulo temporizador. Esta configuración se puede realizar también escribiendo los bits de los registros, como se realizará en el microcontrolador.

El **TCCRnC** no está implementado, los bits FOCnA, FOCnA y FOCnC se puede simular oprimiendo respectivamente los indicadores de **señales internas** OcnA, OCnB y OcnC. (Considerar que esto solamente es válido en los modos NO PWM)

Procedimiento recomendado de ensayo, para apreciar todos los modos de trabajo:

1. Activar el bit **CS10** del registro **TCCRnB**. Observará que **TCNTn** comienza a incrementarse, el **PRESCALER** se pone en **x1**, y en la gráfica de Timers verá la **rampa creciente** de la señal blanca. Para poner el TCNTn en 0 puede oprimirse el botón de Reset
2. Activar también el bit **CS11** del registro TCCRnB. Observará que TCNTn se incrementa más lentamente, el indicador PRESCALER se pone en **x64**, y la pendiente de la rampa resulta más suave.
3. Volver a poner PRESCALER en x1. Modificar el tiempo de simulación de ciclos a **1 ms**.
4. Poner **escala Y** en 600 en gráfica de timers
5. Poner valores **OCRnA = 300**, **OCRnB = 200**, **OCRnC = 100**
6. Poner **WGM13..0** en 0001 (en selector, PWM Phase Correct 8 bits).
7. Reiniciar **TCNTn**. Observar cómo ahora asciende hasta 255 y luego desciende hasta 0, repitiendo el ciclo. 255 es el TOP Fijo.
8. Coloque los selectores SALIDA OCnX (X=A, B,C) en **CLEAR** y observe las salidas correspondientes. ¿Qué sucede con OcnA, OcnB y OCnC? Justifique
9. Poner **WGM13..0** en 0010 (en selector, PWM Phase Correct 9 bits) y en 0011 (en selector, PWM Phase Correct 10 bits). ¿Qué sucede ahora con OCnA, OCnB y OCnC?. ¿Cómo cambia el período de las señales y el Duty Cycle?. ¿Cambia su tiempo en alto (TON)?.
10. Repita 8 y 9 con los selectores SALIDA OCnX (X=A, B,C) en **SET**. Compare con lo obtenido anteriormente.
11. Poner **WGM13..0** en 0100 (CTC-OCR1A). ¿Qué sucede ahora con TCNTn?. Modifique el valor de OCR1A, por ejemplo a 500 u otro valor, y observe.
12. Coloque la salida OCnA en **TOGGLE**. Observe la salida **OCnA**. ¿Cuál es el valor del *duty cycle*?. ¿De qué depende su período?.
13. Experimente los modos **WGM13..0 = 5, 6 y 7** (FAST PWM 8b, 9b y 10b), con las salidas en modo SET y en modo CLEAR, y compare con los de los puntos 8 y 9 (PWM Phase Correct 8b, 9b y 10b). ¿Qué puede comentar respecto a las transiciones de los valores de las salidas?
14. Poner **WGM13..0 = 8** (PWM Phase and Frequency Correct-ICR1). Colocar ICRn en un valor como 400 (puede probar otros). Reinicie **TCNTn**. ¿Cómo trabaja?. Compare con los modos



PWM de los puntos anteriores. ¿En qué se parece y se diferencia respecto a los modos de los puntos 8 y 9 (PWM Phase Correct 8b, 9b y 10b).?

15. Poner **WGM13..0** = 9 (PWM Phase and Frequency Correct-OCR1A). Observe similitudes y diferencias con el punto anterior.
16. Los modos 10 y 11 (PWM Phase Correct-OCR1A y PWM Phase Correct-ICR1) no difieren de 8 y 9 mientras el TOP se mantiene constante. La diferencia se produce cuando se quiere variar el TOP (OCR1A o ICR1) para variar la frecuencia de la onda PWM generada. En los modos 10 y 11 la actualización de OCR1A o ICR1 se realiza cuando TCNT1 alcanza el TOP, mientras que en los modos 8 y 9 se realiza cuando TCNT1 alcanza el BOTTOM. En estos modos (8 y 9), los períodos “instantáneos”, tomados entre dos ceros del TCNT1, resultan exactamente los programados.
17. Probar los modos Fast **PWM-ICR1** (14) y Fast **PWM-OCR1A**.
18. Opcionalmente puede pasar a la pestaña ARCHIVO y conectar la tabla (que representa una señal trifásica, o cargar la tabla con un archivo externo, y conectar cada columna con uno de los 4 registros (OCRnA, OCRnB, OCRnC o ICRn), para ver el comportamiento en una situación posible en una aplicación.
19. Puede ensayar los modos de prescaler Ext Flanco bajada y Ext Flanco subida. En estos modos el Timer se incrementa con los pulsos aplicados al pin Tn.
20. Puede ensayar también la captura de TCNT1 en el registro ICRn, que está disponible en todos los modos excepto en los que ICRn funciona como TOP.