

**ESCUELA DE INGENIERÍA INFORMÁTICA**

**Grado en Ingeniería Informática**



**PERIFÉRICOS E INTERFACES**

**PRÁCTICAS EN LABORATORIO**

**Práctica 2**

**Timers en el  $\mu$ C ATmega 2560**

Última actualización: 05 octubre 2023

## Contenido

1	Competencias y objetivos de la práctica .....	3
2	Documentación previa .....	3
3	Descripción del Timer 3.....	4
3.1	Introducción.....	4
3.2	Diagrama de bloques .....	6
3.3	Funcionamiento simplificado .....	7
3.4	Interrupciones del TIMER3.....	7
3.5	Modos de funcionamiento .....	8
3.5.1	Modo normal.....	8
3.5.2	Modo Clear Timer on Compare Match (CTC) .....	10
3.5.3	Modo Fast PWM.....	12
3.5.4	Modo Phase Correct PWM .....	15
3.5.5	Modo Phase and Frequency Correct PWM.....	16
3.6	Rangos de frecuencia y modos de funcionamiento .....	18
4	Realización práctica .....	19
4.1	Tareas previas (no entregables).....	19
4.1.1	Tarea 1. Generación de señales en modo NORMAL.....	19
4.1.2	Tarea 2. Generación de señales en Modo CTC.....	19
4.1.3	Tarea 3. Generación de señales en modo FAST PWM.....	20
4.1.4	Tarea 4. Generación de señales en modo PWM, Phase Correct .....	20
4.2	Aplicación a desarrollar (entregable para evaluar).....	21
4.2.1	Sustitución de la interrupción INT3 por otra generada por el Timer 3.....	21
4.2.2	Diseño e implementación de un frecuencímetro .....	23
4.2.3	Mejoras de la aplicación (opcional) .....	24
5	Entrega del informe de práctica .....	25
6	APÉNDICE: Registros del Timer 3 .....	26

## 1 Competencias y objetivos de la práctica

La práctica 2 de la asignatura se centra en el uso de los timers (o temporizadores) del microcontrolador ATmega 2560 considerados como recursos muy útiles en cualquier microcontrolador. Concretamente, el microcontrolador Atmega 2560, dispone de 2 timers de 8 bits y 4 timers de 16 bits, con diferentes modalidades de funcionamiento. Estos dispositivos nos van a permitir generar diferentes tipos de ondas, establecer temporizaciones con el uso combinado de interrupciones e incluso medir tiempos en señales o entre eventos. Con el estudio de los timers se pretende seguir profundizando en los múltiples recursos del microcontrolador y alcanzar las siguientes competencias:

1. Capacidad para entender los diversos aspectos software y hardware involucrados en la gestión de los timers del microcontrolador ATmega 2560.
2. Capacidad para programar y hacer uso de los timers para diferentes finalidades
3. Capacidad para aprender y aplicar nuevos conceptos de forma autónoma e interdisciplinar.
4. Capacidad para emplear la creatividad en la resolución de los problemas.
5. Capacidad para trabajar en equipo y colaborar eficazmente con otras personas.

Para alcanzar estas competencias se plantea la consecución de los siguientes objetivos:

1. Conocer y entender los aspectos básicos de funcionamiento y programación de los timers del microcontrolador ATmega 2560.
2. Realizar programas sencillos que impliquen la generación de ondas rectangulares de diferentes frecuencias incluida la gestión de interrupciones cuando se alcancen las temporizaciones previstas.
3. Realizar programas sencillos para la generación de señales PWM (Pulse Width Modulation) y su utilización en el control de la intensidad lumínica de un led.
4. Medida de tiempos en señales de entrada para determinar frecuencia de la señal o tiempos entre eventos (flancos).

## 2 Documentación previa

La documentación básica a utilizar para la realización de esta práctica (además de los propios relacionados con la teoría del módulo 1 y 2) está disponible en el Campus Virtual de la ULPGC en el curso que corresponde a Periféricos e Interfaces. La documentación mínima a manejar será la siguiente:

- Documento 1: Enunciado de la práctica (este documento)
- Documento 2: Presentación de la práctica
- Transparencias del módulo 1 y 2

- Datasheet Atmel ATmega 2560 (435 páginas)
  - o [http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\\_datasheet.pdf](http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf)
- Software: Entorno de programación de Arduino y simulador Proteus
  - <https://www.arduino.cc/en/Main/Software>
  - <https://www.hubor-proteus.com/>
- Internet: Búsqueda de cualquier otra información que se considere necesaria

## 3 Descripción del Timer 3

### 3.1 Introducción

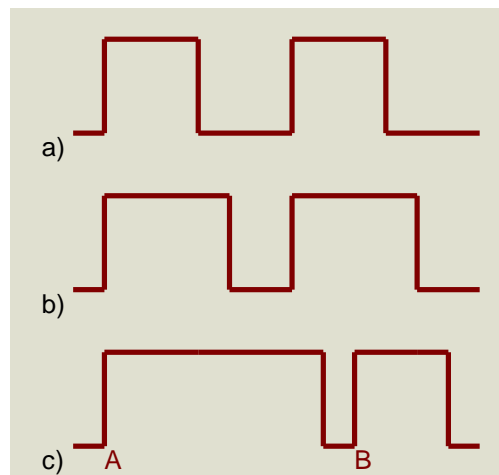
En el desarrollo de la práctica haremos uso del Timer 3 del microcontrolador ATmega 2560 (igualmente aplicable a los timers 1, 4 y 5), que es un timer de 16 bits y no usado por las funciones básicas de la plataforma Arduino. Recuerde que, en dicha plataforma, los timers 0, 1 y 2 tienen un uso específico y son usados por las funciones propias del entorno Arduino. Si los usamos para nuestros propósitos, entonces algunas funciones de Arduino dejarían de funcionar. Por ejemplo, el timer 0 es utilizado por las funciones `delay()` y `delayMicroseconds()`. Si reprogramamos este timer 0 para ser usado por nuestras aplicaciones, entonces las mencionadas funciones dejarían de funcionar correctamente.

En general, el funcionamiento básico de un timer está basado en un contador de “n” bits que cuenta hacia arriba hasta alcanzar un valor máximo definido como el TOP o hacia abajo hasta llegar a 0 definido como el BOTTOM. Dependiendo del modo de funcionamiento, se producen desbordamientos, o avisos de overflow, cuando el contador alcanza el valor máximo 0xFFFF (MAX), el TOP o el BOTTOM. Si se habilitan las interrupciones por overflow (TOIE3=1), entonces éstas se producirán y deberán ser atendidas por la correspondiente rutina de servicio ISR(). En paralelo a este funcionamiento básico, los registros de comparación OCR3A, OCR3B y OCR3C se están comparando con el valor que en cada instante hay en contador del timer (TCNT3). Si se produce una igualdad (match) en alguno de ellos, entonces se activa el flag correspondiente (OCF3A/B/C) y si las interrupciones están habilitadas (OCIE3A/B/C =1), también se producirían interrupciones con vectores `TIMER3_COMPA/B/C_vect`.

Este funcionamiento básico combinado con circuitería adicional permite las siguientes funciones:

- Generar señales rectangulares o cuadradas (solo dos niveles de señal: alto y bajo) de diferentes frecuencias.
- Generar señales PWM para el control de potencia.
- Medidas de tiempos en señales de entrada.
- Generar interrupciones para fines diversos

En la figura 1 se muestra un ejemplo de cada una de esas acciones o funciones.



**Figura 1. a) Señal cuadrada; b) señal PWM y c) medida de tiempo entre dos flancos: A y B**

En los siguientes apartados, realizaremos una breve descripción de la estructura interna del Timer 3 para finalmente plantear las tareas a realizar.

### 3.2 Diagrama de bloques

La figura 2 muestra un diagrama de bloques común para todos los timers de 16 bits (timers 1,3,4,5) del microcontrolador ATmega 2560.

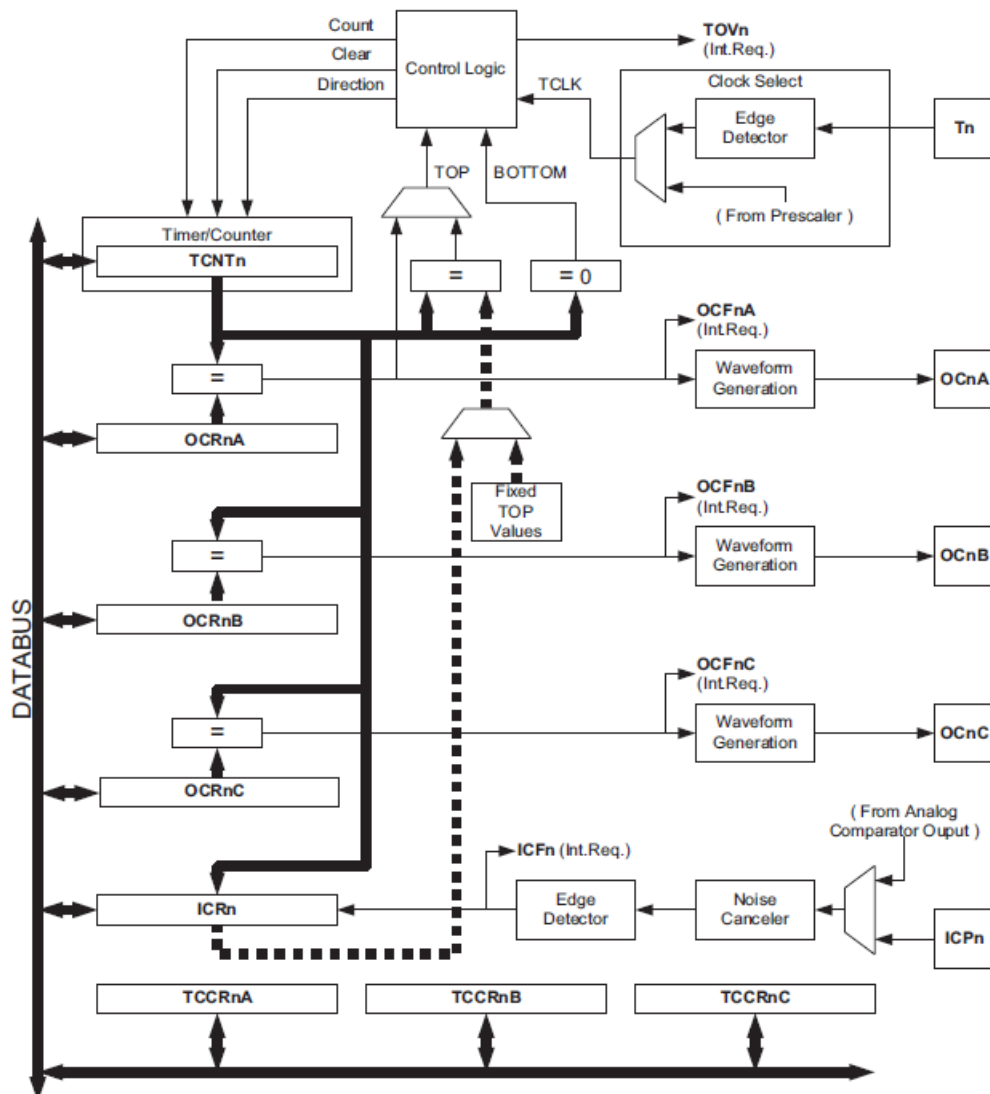


Figura 2. Diagrama de bloques de los timers de 16 bits (1,3,5 y 7)

Antes de comentar los componentes básicos de un timer de 16 bits, conviene fijar algunos conceptos básicos relativos a la terminología a usar:

**BOTTOM:** Mínimo valor que puede alcanzar el contador: 0x0000 (0000)

**MAX:** Máximo valor que puede alcanzar el contador: 0xFFFF (65535). Depende del número de bits del contador (TCNTn).

**TOP:** Valor al que llega el contador para luego retornar a cero (directamente) o iniciar una cuenta hacia abajo, dependiendo del modo de funcionamiento. El TOP puede ser programado con un valor fijo o inmediato (0x00FF, 0x01FF, 0x03FF) o a través del contenido de los registros OCR3A o ICR3, dependiendo del modo de funcionamiento del timer.

### 3.3 Funcionamiento simplificado

Haciendo referencia al diagrama de bloques de la figura 2 y considerando el Timer 3 ( $n=3$ ), destacamos el registro contador de 16 bits (TCNT3) que cuenta pulsos de la señal de reloj (CLKt3) incrementándose o decrementándose (según modo de funcionamiento programado). La señal de reloj, puede ser una señal de reloj externa (T3) o la suministrado por un divisor de frecuencia programable (o prescaler).

Como funcionamiento inicial tenemos que el contador TCNT3 cuenta desde 0x0000 hasta el valor que fije el TOP. Al mismo tiempo, los registros de comparación OCR3A/OCR3B/OCR3C se están comparando con la cuenta del registro contador TCNT3. Cuando el valor de alguno de estos registros coincide con el del contador, entonces se activará los flags que correspondan OCF3A, OCF3B o OCF3C, según en qué registro se haya producido la igualdad (match). La activación de los flags también puede ir acompañada por la generación de interrupciones, si así se programan. A continuación, el generador de onda actuará para generar por los pines de salida del timer (OC3A, OC3B o OC3C) las señales que en cada caso el usuario haya programado.

### 3.4 Interrupciones del TIMER3

Un timer de 16 bits puede generar hasta 5 tipos de interrupciones. En la siguiente tabla se muestran las fuentes de interrupción para el timer 3.

Nombre	Flag de habilitación (TIMSK3)	Número de vector	Nombre del vector para ISR()
Capture	ICIE3	32	TIMER3_CAPT_vect
Compare match A	OCIE3A	33	TIMER3_COMPA_vect
Compare match B	OCIE3B	34	TIMER3_COMPB_vect
Compare match C	OCIE3C	35	TIMER3_COMPC_vect
Overflow	TOIE3	36	TIMER3_OVF_vect

**Tabla 1. Interrupciones del Timer 3**

La primera columna indica el nombre del evento o fuente que causa la interrupción; la segunda columna, muestra los bits de habilitación de interrupción (interrupt enable) de cada uno de los eventos ubicados en el registro de máscara TIMSK3; la tercera columna, corresponde al número de vector asignado a cada interrupción que es utilizado para entrar en la tabla de vectores de interrupción; en la cuarta columna, se muestra el nombre del vector a especificar como parámetro en la función ISR() de la plataforma Arduino.

Los eventos (fuentes) que pueden generar una interrupción son:

- Flanco de subida o bajada en el pin Input Capture (ICP3)
- Igualdad (o match) entre el registro OCR3A y el registro del timer TCNT3
- Igualdad (o match) entre el registro OCR3B y el registro del timer TCNT3
- Igualdad (o match) entre el registro OCR3C y el registro del timer TCNT3
- Overflow. Según el modo de funcionamiento, se produce en el: MAX, TOP o BOTTOM.

### 3.5 Modos de funcionamiento

Los timers de 16 bits tienen un total de 15 modos de funcionamiento que quedan resumidos en la siguiente tabla:

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	—	—	—
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Tabla 2. Modos de funcionamiento del Timer 3

#### 3.5.1 Modo normal

El contador TCNT3 cuenta de 0 hasta 0xFFFF (cuenta hacia arriba). Cada vez que el contador llega a su valor máximo (0xFFFF) y retorna a cero (0x0000), se produce un overflow y se activa el flag TOV3. Si las interrupciones por overflow están habilitadas (TOIE3=1), entonces se generará una interrupción por overflow (vector 36). La figura 3 ilustra el funcionamiento para el timer 3.

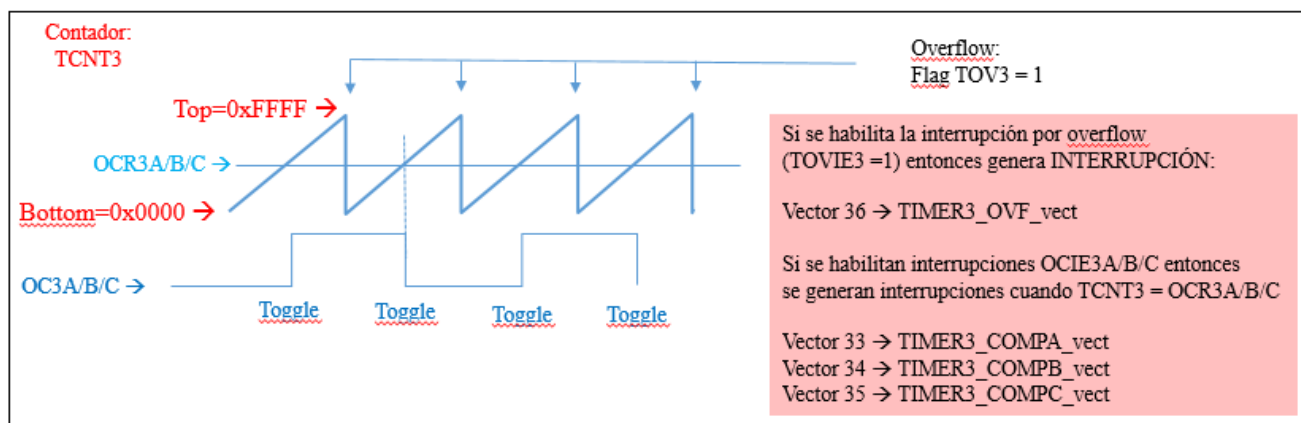


Figura 3. Funcionamiento en modo normal (TOP: 0xFFFF)



Si el generador de onda se programa para que la señal de salida cambie, toggle, (COM3A/B/C1:0 = 01) cada vez que la cuenta del contador iguala (match) a los registros de comparación, entonces la frecuencia de las señales de salida del timer, OC3A/B/C, tendrá el valor:

$$f_{OC3A/B/C} = 16 \text{ MHz} / (2 \times N \times (1 + TOP))$$

TOP= 0xFFFF (fijo)

N: Valor del prescaler (1, 8, 64, 256 o 1024)

Dependiendo del valor de los registros de comparación OCR3A/B/C las señales de salida podrán presentar cierto desfase entre ellas, pero la frecuencia será la misma.

### Ejemplo:

Fclk= 16 MHz; N=1; OCR3A=1; OCR3B=0x7FFF; OCR3C= 0xFFFF

$f_{OC3A/B/C} = 122 \text{ Hz}$ ;  $T = 8.19 \text{ ms}$  (frecuencia de las señales de salida OC3A/B/C)

Señal OC3A: señal de referencia en amarillo

Señal OC3B: desfasada  $\frac{1}{4}$  periodo ( $90^\circ$ ) en azul

Señal OC3C: desfasada  $\frac{1}{2}$  periodo ( $180^\circ$ ) en rojo

Osciloscopio: escala horizontal 1 ms/división

Tick de reloj (16 MHz)  $\rightarrow$  62.5 ns

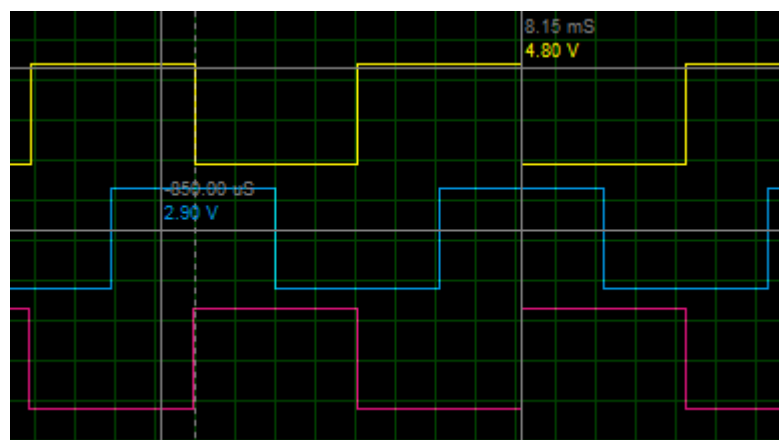


Figura 4. Modo normal ( $f = 122 \text{ Hz}$ ;  $T = 8.19 \text{ ms}$ )

Programa:

```

52 // modo normal
53 // f = 16 MHz / (2 * N * 65536)
54 // N= 1; tclk = 62.5 ns; f = 122Hz; T = 8.19 ms
55
56 TCCR3A = 0;
57 TCCR3B = 0; // no clock
58 OCR3A = 0x0001;
59 OCR3B = 0x7FFF; // desfase de 32768 * 62.5 ns = 2 ms
60 OCR3C = 0xFFFF; // desfase de 65536 * 62.5 ns = 4 ms
61 TCNT3 = 0;
62 TCCR3A = B01010100;
63 TCCR3B = B00000001;

```

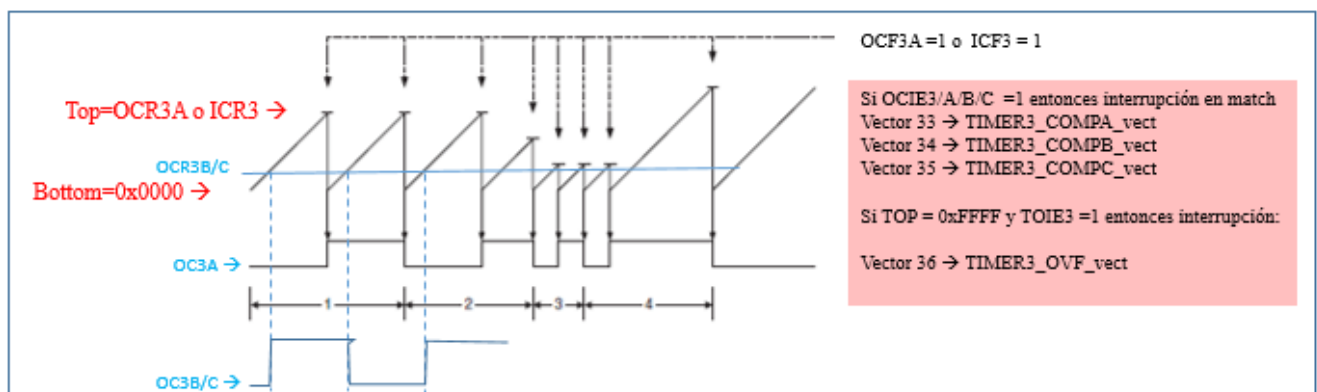
### 3.5.2 Modo Clear Timer on Compare Match (CTC)

Este modo de funcionamiento es similar al modo normal excepto en que el TOP está definido por el contenido de los registros OCR3A o ICR3 (modos de funcionamiento WGM33:0 = 4 o 12). Cuando el contador alcanza el TOP entonces, en el siguiente ciclo de reloj, el contador se resetea a 0x0000, se activa el flag OCF3A y se producen las siguientes acciones:

- Si OCIE3A = 1, entonces se produce la interrupción TIMER3\_COMPA
- Si el TOP se define a 0xFFFF y TOIE3=1, entonces también se produce la interrupción por overflow TIMER3\_OVF
- Acción sobre la salida OC0A según la programación del generador de onda. El cambio de estado (toggle, COM3A1:0 = 01) es la opción más frecuente.

Respecto a las salidas OC3B y OC3C, éstas se activan según la programación del generador de onda cuando el contenido de los registros OCR3B y OCR3C (los contenidos deben ser menores o iguales al TOP) se igualan (match) al contador del timer, instante en el que se activan los flags OCF3B/C, produciéndose las siguientes acciones:

- Si OCIE3B/C = 1, entonces se produce la interrupción TIMER3\_COMPB/C
- Acción sobre la salida OC0B/C según la programación del generador de onda. El cambio de estado (toggle, COM3B1:0 = 01 y COM3C1:0 = 01) es la opción más frecuente.



**Figura 5. Modo CTC (TOP: OCR3A, ICR3)**

La figura 5 ilustra el funcionamiento para el timer 3 en modo CTC.

La frecuencia de la señal en OC3A, si el top lo define el registro OCR3A, viene dada por la siguiente fórmula:

$$f_{OCnA} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

N: prescaler (1,8,64,256 o 1024)

Las señales por los canales B y C, si se habilitan y utilizan, serán de la misma frecuencia, pero desfasadas según los contenidos de los registros de comparación OCR3B y OCR3C.

Ejemplo:

Fclk = 16 MHz; N=1; fclk-timer = 16/1= 16 MHz ; Tclk-timer = 62.5 ns

OCR3A=0x7FFF; OCR3B=0x1FFF; OCR3C= 0x3FFF

$F = 16 \text{ Mhz} / (2 \times N \times (1 + \text{OCR3A}))$

**f<sub>OC3A/B/C</sub> = 244 Hz; T = 4 ms** (frecuencia de las señales de salida OC3A/B/C)

Señal OC3A: señal de referencia, en amarillo

Señal OC3B: desfasada  $8192 \times 62.5 = 0.51 \text{ ms}$ , en azul

Señal OC3C: desfasada  $16384 \times 62.5 \text{ ns} = 1 \text{ ms}$ , en rojo

Osciloscopio: escala horizontal 0.5 ms/división

Tick de reloj (16 MHz) → 62.5 ns

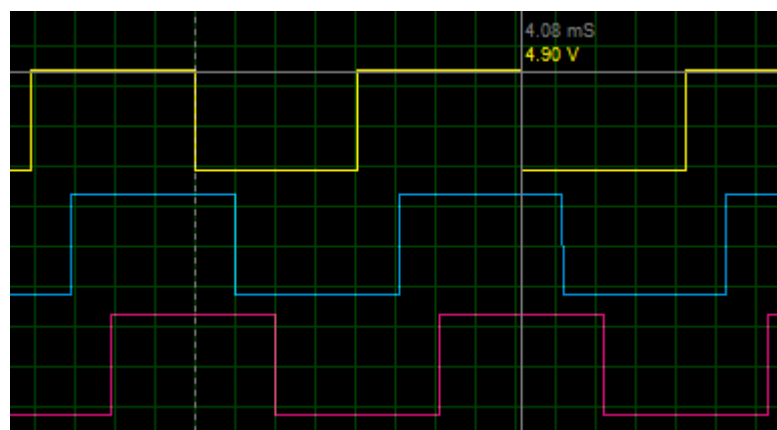


Figura 6. Modo CTC (f = 244 Hz; T= 4 ms)

Programa:

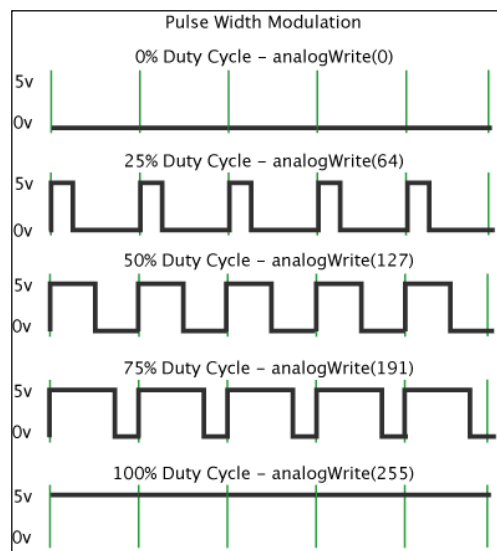
```

69 // modo CTC
70 // f= 16 MHz/(2xNx(1+TOP))
71 // N= 1; tclk = 62.5 ns; f = 244 Hz; T = 4 ms
72 TCCR3A = 0;
73 TCCR3B = 0;
74 TCNT3 = 0;
75 OCR3A = 0x7FFF; // f = 244 Hz; 4 ms
76 OCR3B = 0x1FFF; // 8191 (8K -1)
77 OCR3C = 0x3FFF; // 16383 (16K -1)
78 TCNT3 = 0;
79 TCCR3A = B01010100;
80 TCCR3B = B00001001;

```

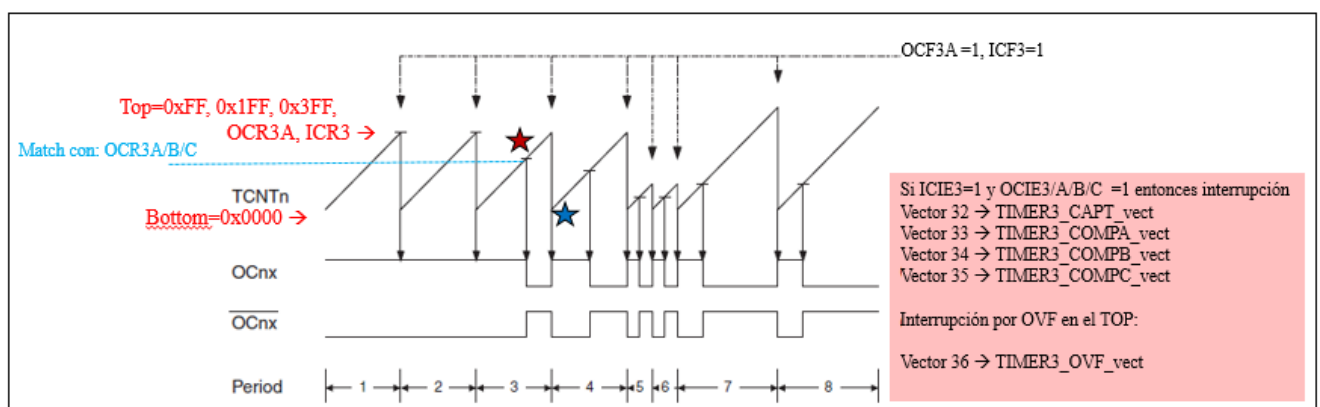
### 3.5.3 Modo Fast PWM

En este modo el timer es capaz de generar señales PWM (Pulse Width Modulation), es decir, señales con forma de onda rectangulares y de ciclo de trabajo (duty cycle) variable. De esta forma, se consiguen señales cuyo voltaje medio puede oscilar entre 0 y 5 voltios, muy útiles para el control de potencia en múltiples dispositivos (control de velocidad en motores, control de intensidad lumínica y otros).



**Figura 7. Señales PWM**

La plataforma Arduino ofrece la función *analogWrite(pin, nivel)* para generar, por un pin con capacidades PWM, una señal con un nivel de potencia o ciclo de trabajo entre 0 y 255 (voltaje medio de 0 a 5 voltios).



**Figura 8. Modo Fast PWM, rampa simple (TOP: 0xFF, 0x1FF, 0x3FF, OCR3A, ICR3)**

La modalidad de fast PWM es una modalidad de simple rampa en la que el contador cuenta desde 0x0000 hasta un TOP que puede ser definido por un valor fijo (0x00FF, 0x01FF, 0x3FF) o por el contenido de los registros OCR3A o ICR3. La frecuencia de la señal PWM vendrá definida por la siguiente fórmula:

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot (1 + TOP)}$$

N: prescaler (1,8,64,256 o 1024)

El funcionamiento básico es el siguiente:

- El contador comienza la cuenta en 0x0000, en sentido ascendente, hasta alcanzar el TOP. Cuando llega al TOP se resetea, activa el flag de TOV3 y se produce una interrupción por overflow si el bit TOIE3 = 1.
- Las salidas OC3A/B/C cuando el TOP es fijo o definido por el registro **ICR3**:
  - Cuando el valor del contador del timer iguala a los contenidos de los registros OCR3A/B/C, entonces se produce una acción en las salidas OC3A/B/C de acuerdo a los valores de programación del generador de onda. Es frecuente usar la opción COM3A/B/C1:0= 10 que pone a cero (clear) las señales OC3A/B/C cuando se produce la igualdad (match) entre los registros OCR3A/B/C y el contador del timer, y pone a 1(set) las señales OC3A/B/C cuando el contador está en 0x0000.
  - Si se habilitan interrupciones (OCIE3A/B/C =1) entonces se generan interrupciones cuando los flags OCF3A/B/C =1 o ICF3=1 (vectores TIMER3\_COMPA/B/C\_vect, TIMER3\_CAPT\_vect).
- Las salidas OC3A/B/C cuando el TOP está definido por **OCR3A**:
  - La salida OC3A no se puede utilizar para generar señal PWM.
  - Las salidas OC3B/C igual que el caso anterior
  - Interrupciones, igual que en el caso anterior.

Ejemplo:

TOP (fijo, 10 bits): 0x3FF → podemos definir hasta 1024 niveles de potencia PWM (0-1023)

Fclk = 16 MHz; N=8; fclk-timer = 16/8 = 2 MHz ; Tclk-timer = 500 ns = 0,5 useg;

$f = 16 \text{ MHz} / (N \times (1 + \text{TOP})) = 1953.125 \text{ Hz}$

$f = 1953.125 \text{ Hz}$ ;  $T = 0.512 \text{ ms} = 512 \text{ useg}$  → periodo de la señal PWM

A partir de estos datos generar tres señales PWM periódicas, con  $T = 0.512 \text{ ms}$ , y con ciclos de trabajo de: 25%, 50% y 75%.

Valores para inicializar los registros OCR3A/B/C:

25% →  $0.25 \times 1024 = 256 = 0x0100$

50% →  $0.50 \times 1024 = 512 = 0x0200$

75% →  $0.75 \times 1024 = 768 = 0x0300$

Señal OC3A: señal PWM 25% duty cycle, en amarillo ( $0.25 \times 512 = 128 \text{ useg}$ )

Señal OC3B: señal PWM 50% duty cycle, en azul ( $0.50 \times 512 = 256 \text{ useg}$ )

Señal OC3C: señal PWM 75% duty cycle, en rojo ( $0.75 \times 512 = 384 \text{ useg}$ )

Osciloscopio: escala horizontal 51.2 useg /división

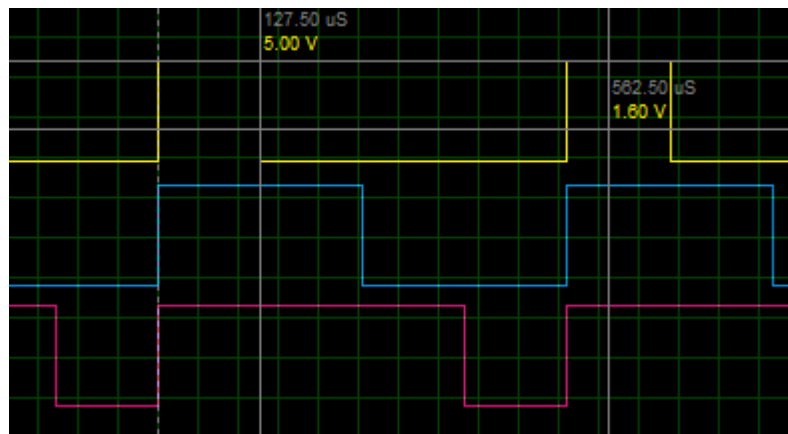


Figura 9. Modo fast PWM ( $f = 1953.125 \text{ Hz}$ ;  $T = 512 \text{ useg}$ )

Programa:

```

83 // modo fast pwm, top_fijo=0x3ff --> hasta 1024 niveles de potencia (0-1023)
84 // f = 16 MHz/(Nx(1+TOP))
85 // N=8; tclk= 500 ns; f = 1953.125 Hz; T = 0.512 ms = 512 useg
86
87 TCCR3A = 0;
88 TCCR3B = 0;
89 TCNT3 = 0;
90 OCR3A=0x0100; // 25% duty cycle (1024/4 = 256)
91 OCR3B=0x0200; // 50% duty cycle (1024/2 = 512)
92 OCR3C=0x0300; // 75% duty cycle /1024x3/4= 768)
93 TCCR3A = B10101011;
94 TCCR3B = B00001010;

```

### 3.5.4 Modo Phase Correct PWM

Es una modalidad de funcionamiento similar a la anterior para generar señales PWM pero, de doble rampa, proporcionando una señal con características simétricas, siendo una modalidad preferida en aplicaciones de control de motores.

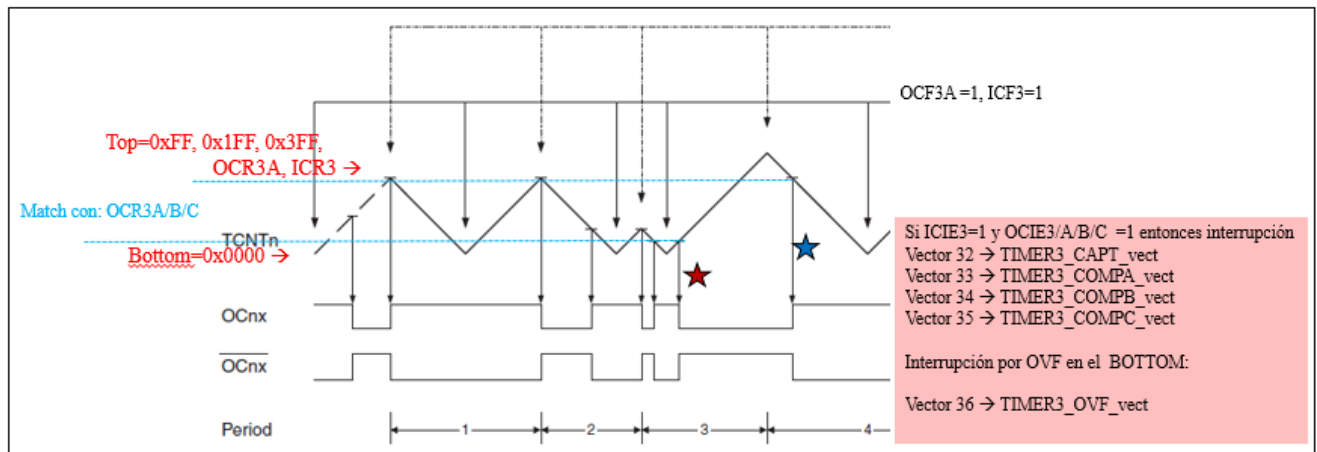


Figura 10. Modo phase correct PWM, rampa doble (TOP: 0xFF, 0x1FF, 0x3FF, OCR3A, ICR3)

La modalidad phase correct PWM es una modalidad de doble rampa en la que el contador cuenta hacia arriba desde el BOTTOM (0x0000) hasta el TOP, y luego hacia abajo, desde el TOP hasta el BOTTOM. El TOP puede ser definido por un valor fijo (0x00FF, 0x01FF, 0x3FF) o por el contenido de los registros OCR3A o ICR3. La frecuencia de la señal PWM viene definida por la siguiente fórmula:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP}$$

N: prescaler (1,8,64,256 o 1024)

El funcionamiento básico es el siguiente:

- El contador comienza la cuenta ascendente hasta el TOP y luego la descendente hasta el BOTTOM, instante en el que se activa el flag TOV3 que puede producir una interrupción si el bit TOIE3=1.
- Las salidas OC3A/B/C cuando el TOP es fijo o definido por el registro **ICR3**:
  - En el modo “no-inverting” del generador de onda, correspondiente a COM3A/B/C1:0 = 10, que es el más frecuente, cuando los contenidos de los registros OCR3A/B/C se igualan a la cuenta del contador del timer en sentido ascendente, las señales OC3A/B/C se resetean (clear). Cuando la igualdad se produce en la cuenta descendente del contador del timer, entonces las señales OC3A/B/C se ponen a 1 (set).

- Si se habilitan interrupciones (OCIE3A/B/C =1) entonces se generan interrupciones cuando los flags OCF3A/B/C =1 o ICF3=1 (vectores TIMER3\_COMPA/B/C\_vect, TIMER3\_CAPT\_vect).
- Las salidas OC3B/OC3C cuando el TOP está definido por **OCR3A**:
  - La salida OC3A no se puede utilizar para generar señal PWM.
  - Las salidas OC3B/C igual que el caso anterior
  - Interrupciones igual que en el caso anterior

### 3.5.5 Modo Phase and Frequency Correct PWM

La modalidad phase and frequency correct PWM es una modalidad de doble rampa en la que el contador cuenta hacia arriba desde el BOTTOM (0x0000) hasta el TOP y, luego, cuenta hacia abajo desde el TOP hasta el BOTTOM. De buenas características simétricas y preferida en las aplicaciones de control de motores. Similar al modo anterior salvo que, la actualización o escritura en los registros de comparación OCR3x, se realiza en el estado de BOTTOM y no en el de TOP como en el caso anterior.

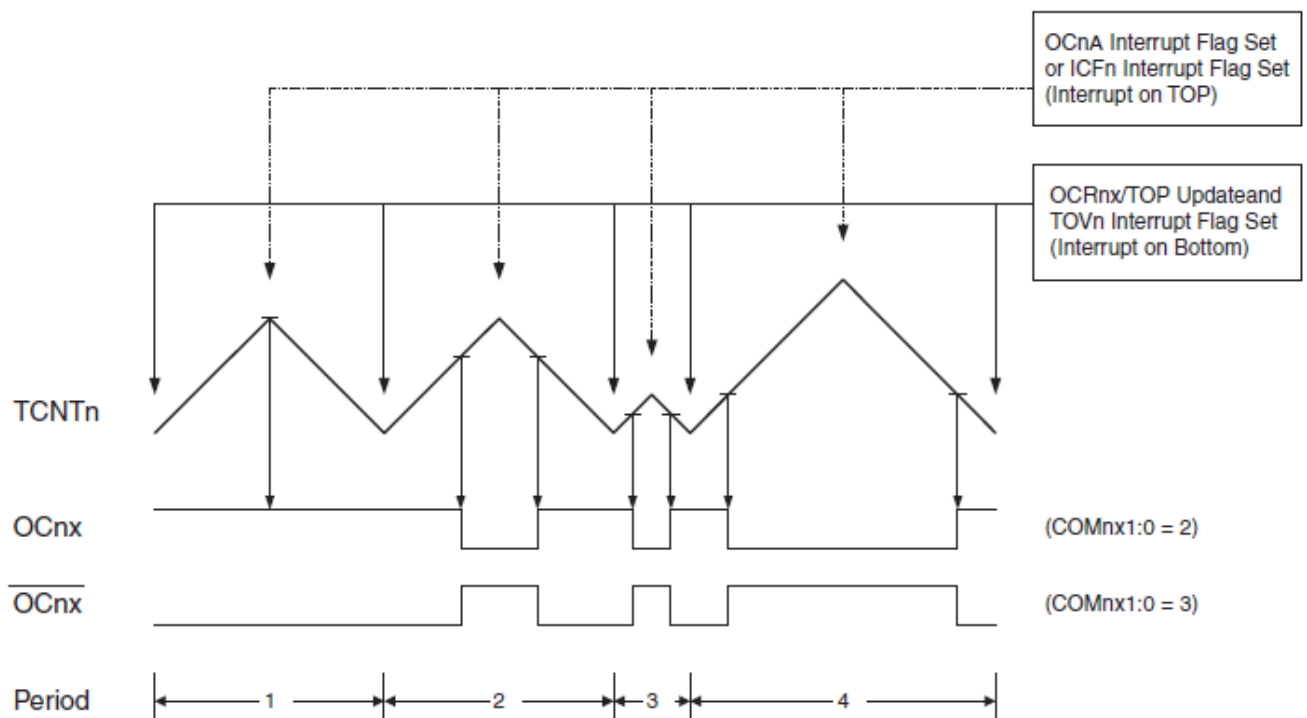


Figura 11. Modo phase and frequency correct PWM (doble rampa)

El TOP puede ser definido por el contenido de los registros OCR3A o ICR3. La frecuencia de la señal PWM vendrá definida por la siguiente fórmula:

$$f_{\text{OCnxPCPWM}} = \frac{f_{\text{clk\_I/O}}}{2 \cdot N \cdot \text{TOP}}$$



N: prescaler (1,8,64,256 o 1024)

El funcionamiento básico es el siguiente:

- El contador comienza la cuenta ascendente hasta el TOP y luego la descendente hasta el BOTTOM, instante en el que se activa el flag TOV3 que puede producir una interrupción si el bit TOIE3=1.
- Las salidas OC3A/B/C cuando el TOP está definido por ICR3
  - En el modo “no-inverting” del generador de onda, correspondiente a COM3A/B/C1:0 = 10, que es el más frecuente, cuando los contenidos de los registros OCR3A/B/C se igualan a la cuenta del contador del timer en sentido ascendente, las señales OC3A/B/C se resetean (clear). Cuando la igualdad se produce en la cuenta descendente del contador del timer, entonces las señales OC3A/B/C se ponen a 1 (set).
  - Si se habilitan interrupciones (OCIE3A/B/C =1) entonces se generan interrupciones cuando los flags OCF3A/B/C =1 o ICF3=1 (vectores TIMER3\_COMPA/B/C\_vect, TIMER3\_CAPT\_vect).
  -
- Las salidas OC3B/OC3C cuando el TOP está definido por OCR3A
  - La salida OC3A no se puede utilizar para generar señal PWM.
  - Las salidas OC3B/C igual que el caso anterior
  - Interrupciones igual que en el caso anterior

### 3.6 Rangos de frecuencia y modos de funcionamiento

Modo	Frecuencias en salidas OC3A, OC3B y OC3C
Normal (toggle en salida)	$F = 16 \text{ MHz} / (2 \times N \times (1+TOP))$ TOP = MAX (0xFFFF)
CTC (toggle en salida)	$F = 16 \text{ MHz} / (2 \times N \times (1+TOP))$ TOP = OCR3A o ICR3 (0x0000-0xFFFF)
Fast PWM, rampa simple	$F = 16 \text{ MHz} / (N \times (1+TOP))$ TOP= 0x00FF, 0x01FF, 0x3FF, OCR3A o ICR3
Phase correct PWM , rampa doble	$F = 16 \text{ MHz} / (2 \times N \times TOP)$ TOP= 0x00FF, 0x01FF, 0x3FF, OCR3A o ICR3
Phase and frequency correct PWM, rampa doble	$F = 16 \text{ MHz} / (2 \times N \times TOP)$ TOP= OCR3A o ICR3
N: prescaler (1,8,64,256 o 1024)	

Tabla 3. Modos de funcionamiento y frecuencias

Normal mode: $F = 16 \text{ MHz} / (2 \times N \times (1+0xFFFF))$						
N	clkT5 (Hz)	tickT5 (useg)	F (Hz)	T (ms)		
1	16000000	0,0625	122,070313	8,192		
8	2000000	0,5	15,2587891	65,536		
64	250000	4	1,90734863	524,288		
256	62500	16	0,47683716	2097,152		
1024	15625	64	0,11920929	8388,608		

CTC mode: $F = 16 \text{ MHz} / (2 \times N \times (1+OCR3A))$						
N	clkT5 (Hz)	tickT5 (useg)	FMIN (Hz)	FMAX (Hz)	TMAX (ms)	TMIN (useg)
1	16000000	0,0625	122,070313	8000000	8,192	0,125
8	2000000	0,5	15,2587891	1000000	65,536	1
64	250000	4	1,90734863	125000	524,288	8
256	62500	16	0,47683716	31250	2097,152	32
1024	15625	64	0,11920929	7812,5	8388,608	128

Fast PWM: $F = 16 \text{ MHz} / (N \times (1+TOP))$						
N	clkT5 (Hz)	tickT5 (useg)	FMIN (Hz)	FMAX (Hz)	TMAX (ms)	TMIN (useg)
1	16000000	0,0625	244,140625	16000000	4,096	0,0625
8	2000000	0,5	30,5175781	2000000	32,768	0,5
64	250000	4	3,81469727	250000	262,144	4
256	62500	16	0,95367432	62500	1048,576	16
1024	15625	64	0,23841858	15625	4194,304	64

Phase correct PWM: $F = 16 \text{ MHz} / (2 \times N \times TOP)$						
N	clkT5 (Hz)	tickT5 (useg)	FMIN (Hz)	FMAX (Hz)	TMAX (ms)	TMIN (useg)
1	16000000	0,0625	122,070313	8000000	8,192	0,125
8	2000000	0,5	15,2587891	1000000	65,536	1
64	250000	4	1,90734863	125000	524,288	8
256	62500	16	0,47683716	31250	2097,152	32
1024	15625	64	0,11920929	7812,5	8388,608	128

Tabla 4. Rangos de frecuencias de las salidas del Timer 3

Nota: Frecuencias en Phase and frequency correct PWM iguales a phase correct pwm

## 4 Realización práctica

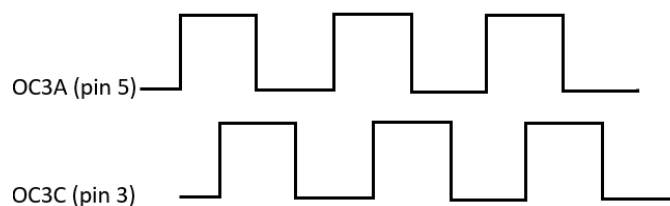
Siguiendo la misma metodología que en la práctica 1, la realización práctica consistirá en un conjunto de tareas previas (no entregables) y una aplicación que será la tarea entregable para su correspondiente valoración.

### 4.1 Tareas previas (no entregables)

Para familiarizarse con el funcionamiento de los timers, y más concretamente del Timer 3 de 16 bits, se propone la realización de un conjunto de actividades en las que se han de tener presente las fórmulas de las frecuencias de las salidas del timer, según los diferentes modos de funcionamiento.

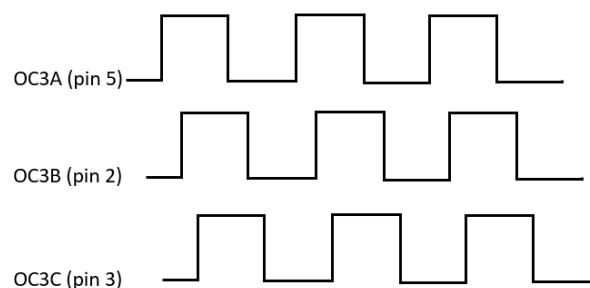
#### 4.1.1 Tarea 1. Generación de señales en modo NORMAL

Generar dos señales de onda cuadrada desfasadas 90° eléctricos con el timer 3 utilizando las salidas **OC3A** (pin 5) y **OC3C** (pin 3) de periodo  $T = 65,536 \text{ ms}$  ( $f = 1/T = 15,2587 \text{ Hz}$ ). Comprobar que las señales generadas por el timer son las correctas haciendo uso del osciloscopio del simulador Proteus.



#### 4.1.2 Tarea 2. Generación de señales en Modo CTC

Generar tres señales desfasadas 45° eléctricos con el timer 3 utilizando las salidas **OC3A** (pin 5), **OC3B** (pin 2) y **OC3C** (pin 3) de periodo  $T = 20 \text{ ms}$ . Hacer uso del registro ICR3 para definir el TOP y escoger  $N=8$  en el divisor de frecuencia del reloj (prescaler). Comprobar que las señales generadas por el timer son las correctas haciendo uso del osciloscopio del simulador Proteus.

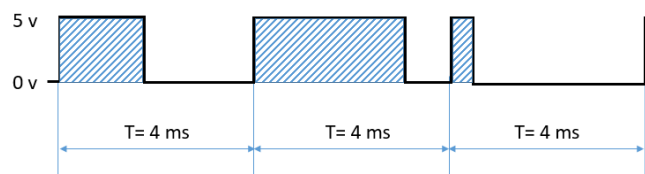


### 4.1.3 Tarea 3. Generación de señales en modo FAST PWM

Generar una señal FAST PWM por la salida **OC3B** (pin 2) cuyo ciclo de trabajo (voltaje medio de la señal) sea controlado por un potenciómetro conectado a la entrada analógica A0 con el que se aplicará un voltaje entre 0 y 5 voltios a la entrada analógica. La lectura de este canal analógico permitirá luego controlar el ciclo de trabajo de la señal PWM entre el 0% (voltaje medio mínimo = 0v) y el 100% (voltaje medio máximo = 5v).

El periodo de la señal PWM será de  $T = 4 \text{ ms}$ ,  $N=8$  (prescaler) y el TOP será definido por el registro OCR3A.

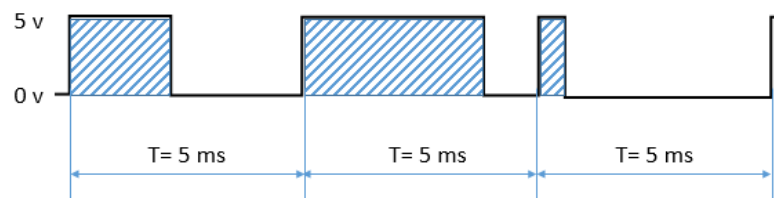
Conecte la señal generada a un LED para ver que el control de potencia se realiza correctamente (el LED se enciende con mayor o menor intensidad según el potenciómetro). Comprobar que la señal generada por el timer es correcta haciendo uso del osciloscopio del simulador Proteus.



### 4.1.4 Tarea 4. Generación de señales en modo PWM, Phase Correct

Generar una señal PWM Phase Correct por la salida **OC3A** (pin 5) cuyo ciclo de trabajo (voltaje medio de la señal) sea controlado por un potenciómetro conectado a la entrada analógica A0 con el que se aplicará un voltaje entre 0 y 5 voltios a la entrada analógica. La lectura de este canal analógico permitirá luego controlar el ciclo de trabajo de la señal PWM entre el 0% (voltaje medio mínimo = 0v) y el 100% (voltaje medio máximo = 5v).

El periodo de la señal PWM Phase Correct será de  $T = 5 \text{ ms}$ ,  $N=8$  (prescaler) y el TOP será definido por el registro ICR3.



Conecte la señal generada (OC3A) a un motor de corriente continua (componente MOTOR-DC) motor para comprobar que el control de potencia se realiza correctamente (el motor gira a mayor o menor velocidad según el potenciómetro). Comprobar que la señal generada por el timer es correcta haciendo uso del osciloscopio del simulador Proteus. En el esquema del proyecto base puede apreciar el uso de un driver (etapa de potencia) para gobernar al motor ya que la salida OC3A del microcontrolador es muy débil para alimentar directamente al motor.

Las propiedades del motor ya están ajustadas a los valores que figuran en la siguiente captura:

The screenshot shows the 'Edit Component' dialog box for a DC motor. The 'Part Reference' is 'MOTOR-DC'. The 'Part Value' is empty. The 'Element' is a dropdown menu with a 'New' button. The 'LISA Model File' is 'DCMOTOR'. The 'Nominal Voltage' is '12v'. The 'Coil Resistance' is '12'. The 'Coil Inductance' is '100mH'. The 'Zero Load RPM' is '1000'. The 'Load/Max Torque %' is '6'. The 'Effective Mass' is '0.0001'. There are 'Hidden' checkboxes for the 'Part Reference' and 'Part Value'. The 'Other Properties' section is empty. At the bottom, there are checkboxes for 'Exclude from Simulation', 'Exclude from PCB Layout' (checked), 'Exclude from Current Variant', 'Attach hierarchy module', 'Hide common pins', and 'Edit all properties as text'. The 'OK', 'Help', and 'Cancel' buttons are on the right.

Property	Value	Hidden
Part Reference	MOTOR-DC	<input type="checkbox"/>
Part Value		<input type="checkbox"/>
Element		
LISA Model File	DCMOTOR	Hide All
Nominal Voltage	12v	Hide All
Coil Resistance	12	Hide All
Coil Inductance	100mH	Hide All
Zero Load RPM	1000	Hide All
Load/Max Torque %	6	Hide All
Effective Mass	0.0001	Hide All

Other Properties:

☐ Exclude from Simulation  
☒ Exclude from PCB Layout  
☐ Exclude from Current Variant  
☐ Attach hierarchy module  
☐ Hide common pins  
☐ Edit all properties as text

## 4.2 Aplicación a desarrollar (entregable para evaluar)

La aplicación que se propone en este apartado consta de dos partes: 1) Sustitución de la interrupción externa INT3 por otra del timer 3 y 2) Diseño e implementación de un frecuencímetro.

### 4.2.1 Sustitución de la interrupción INT3 por otra generada por el Timer 3

Partiendo de la aplicación desarrollada en la práctica 1, sustituir la interrupción externa INT3 por otra interrupción, esta vez generada por un timer, cada 5 ms. Para generar la interrupción se utilizará el Timer 3, programado en el modo 4 (CTC, TOP=OCR3A), con los parámetros de configuración adecuados, para producir una interrupción cada 5 ms y cuyo vector de interrupción es el TIMER3\_COMPA\_vect (vector 33).

La estructura del programa será igual a la de la aplicación de la práctica 1 salvo el apartado de la interrupción externa INT3 que habrá de sustituirse por la programación adecuada del Timer 3 para que genere la interrupción especificada anteriormente.

```
// Las variables que se modifiquen en ISR() y se utilicen
// fuera de ISR() han de declararse "volatile"
// ejemplo:
volatile boolean estado;

void setup() {
  cli();
  // programación del Timer 3, modo 4, interrupción cada 5 ms,
  // vector de interrupción TIMER3_COMPA_vect.
  // Habilitar la interrupción (OCIE3A =1 en TIMSK3)
  sei();
}

ISR (TIMER3_COMPA_vect) {
  // Cuerpo de la rutina de servicio a ejecutar cuando interrumpa
  // el TIMER3 a través del vector TIMER3_COMPA_vect
  // Visualización entrelazada en el display (= aplicación práctica 1)
}

void loop() {
  ...
}
```

### 4.2.2 Diseño e implementación de un frecuencímetro

Añadir a la aplicación del apartado anterior una nueva opción al menú (opción 4) para seleccionar otra funcionalidad para el sistema correspondiente a un frecuencímetro (instrumento que mide la frecuencia de una señal) suponiendo que el Timer 3 está programado según el apartado anterior. Esa programación nos condiciona el rango de frecuencias a medir (frecuencia mínima y frecuencia máxima). Las opciones del menú quedan ahora como:

... opciones anteriores del menú de la aplicación de la práctica 1

4.- Modo frecuencímetro

Entrar opción:

Para ello, utilizaremos las capacidades que tiene el Timer 3 para medir el tiempo entre eventos (por ejemplo, medir el tiempo entre dos flancos de bajada o de subida) haciendo uso de la sección de captura (registro ICR3 y circuitería asociada). Con ello ya se tendría el periodo de la señal cuya inversa sería la frecuencia y que, luego, se mostraría en el display de 7 segmentos. La siguiente figura muestra la sección de captura del timer:

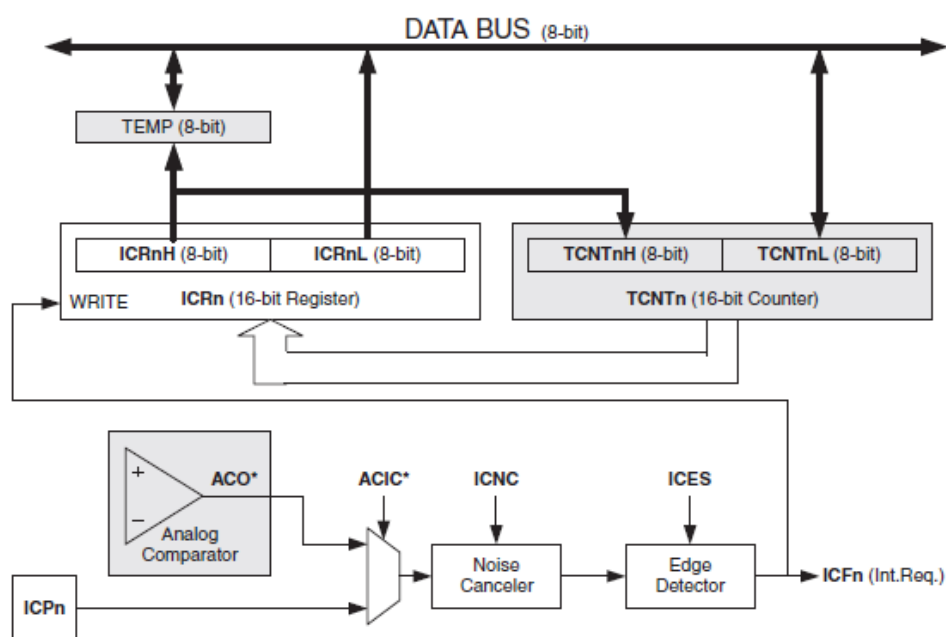
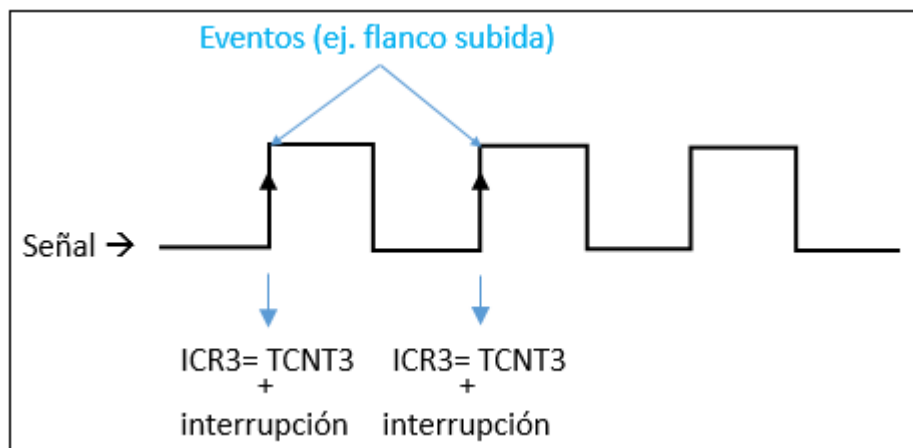


Figura 12. Circuito de captura

La señal de la que queremos medir la frecuencia se conecta al pin ICP3 (pata 9 del microcontrolador, no disponible en los conectores del Arduino mega).



**Figura 13. Medida de tiempos entre eventos (modo captura, ICR3)**

La figura 13 muestra la señal de entrada y cada vez que se produce un evento (flanco de subida o bajada, programable con el bit ICES3 del registro TCCR3B) se guarda el valor del contador TCNT3 en el registro ICR3, se activa el flag ICF3 y se produce una interrupción si está habilitada (bit ICIE3=1 en el registro TIMSK3). La rutina de servicio de la interrupción ha de guardar los últimos dos valores de los ICR3 cuya diferencia nos dará el periodo de la señal ( $T$ ) y cuya inversa será la frecuencia ( $f = 1/T$ )

Con objeto de simplificar la visualización de las frecuencias se le permite las siguientes opciones de visualización, según las frecuencias:

- a) Rango: desde la **frecuencia mínima** detectable hasta 9999 Hz
  - a. Visualización en el display de 7-segmentos
- b) Rango: desde 10000 Hz hasta la **frecuencia máxima** que pueda detectar el frecuencímetro
  - a. Visualización en pantalla o virtual terminal

Explorar o calcular cuáles serían las **frecuencias mínima y máxima** (teóricas y reales, razonar la respuesta) que puede detectar o medir el frecuencímetro en base a la programación del Timer 3 que se ha realizado en el apartado 1 de la aplicación (visualización entrelazada sincronizada por interrupción, vector TIMER3\_COMPA\_vect).

#### 4.2.3 Mejoras de la aplicación (opcional)

Una vez calculadas las frecuencias mínimas y máximas del apartado anterior, trate de visualizar todos los rangos de frecuencias en el display de 7 segmentos de acuerdo a las siguientes especificaciones:

- Visualizar todos los rangos de frecuencias en el display de 7 segmentos de modo que cuando la frecuencia sea superior a 9999 Hz sea dividida por 1000 y se presente de forma escalada en el display de 7 segmentos. Ejemplos de visualización:
  - i. **7834** (7834 Hz)
  - ii. **11.34** (11348 Hz). Activar el punto del dígito de las centenas y no visualice las unidades.



## 5 Entrega del informe de práctica

Una vez desarrollada la aplicación y comprobado su correcto funcionamiento es, absolutamente necesario, subir el informe de la práctica al Campus Virtual de la asignatura, en tiempo y forma, a través del enlace disponible en la sección de prácticas. El informe consistirá en el proyecto Proteus (\*.pdsprj) de la aplicación con los programas adecuadamente comentados. En principio, solo será necesario subir un fichero cuyo nombre seguirá la siguiente nomenclatura:

23-24\_plab2\_iniciales nombre y apellidos.pdsprj

Ejemplo:

**23-24\_plab2\_amf.pdsprj**

Fichero correspondiente al proyecto Proteus de la práctica 2 en laboratorio entregado por Anabel Medina Falcón.

## 6 APÉNDICE: Registros del Timer 3

A continuación, se muestra un resumen de los principales registros del Timer 3. Para un mayor detalle, consultar el manual de referencia del microcontrolador ATmega 2560.

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

### TCCR3A – Timer/Counter 3 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x90)	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### TCCR3B – Timer/Counter 3 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x91)	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**TIMSK3 – Timer/Counter 3 Interrupt Mask Register**

Bit	7	6	5	4	3	2	1	0	
(0x71)	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**TIFR3 – Timer/Counter3 Interrupt Flag Register**

Bit	7	6	5	4	3	2	1	0	
0x18 (0x38)	–	–	ICF3	–	OCF3C	OCF3B	OCF3A	TOV3	TIFR3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Table 17-6.** Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{IO}/1$ (No prescaling)
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

If external pin modes are used for the Timer/Counter, transitions on the Tn pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

**Table 17-3.** Compare Output Mode, non-PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	Toggle OCnA/OCnB/OCnC on compare match
1	0	Clear OCnA/OCnB/OCnC on compare match (set output to low level)
1	1	Set OCnA/OCnB/OCnC on compare match (set output to high level)

Table 17-4 shows the COMnx1:0 bit functionality when the WGMn3:0 bits are set to the fast PWM mode.

**Table 17-4.** Compare Output Mode, Fast PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match, set OCnA/OCnB/OCnC at BOTTOM (non-inverting mode)
1	1	Set OCnA/OCnB/OCnC on compare match, clear OCnA/OCnB/OCnC at BOTTOM (inverting mode)

Note: A special case occurs when OCRnA/OCRnB/OCRnC equals TOP and COMnA1/COMnB1/COMnC1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 146. for more details.

Table 17-5 shows the COMnx1:0 bit functionality when the WGMn3:0 bits are set to the phase correct and frequency correct PWM mode.

**Table 17-5.** Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 = 9 or 11: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match when up-counting Set OCnA/OCnB/OCnC on compare match when downcounting
1	1	Set OCnA/OCnB/OCnC on compare match when up-counting Clear OCnA/OCnB/OCnC on compare match when downcounting

**TCNT3H and TCNT3L – Timer/Counter 3**

Bit	7	6	5	4	3	2	1	0	
(0x95)	TCNT3[15:8]								TCNT3H
(0x94)	TCNT3[7:0]								TCNT3L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**OCR3AH and OCR3AL – Output Compare Register 3 A**

Bit	7	6	5	4	3	2	1	0	
(0x99)	OCR3A[15:8]								OCR3AH
(0x98)	OCR3A[7:0]								OCR3AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**OCR3BH and OCR3BL – Output Compare Register 3 B**

Bit	7	6	5	4	3	2	1	0	
(0x9B)	OCR3B[15:8]								OCR3BH
(0x9A)	OCR3B[7:0]								OCR3BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**OCR3CH and OCR3CL – Output Compare Register 3 C**

Bit	7	6	5	4	3	2	1	0	
(0x9D)	OCR3C[15:8]								OCR3CH
(0x9C)	OCR3C[7:0]								OCR3CL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**ICR3H and ICR3L – Input Capture Register 3**

Bit	7	6	5	4	3	2	1	0	
(0x97)	ICR3[15:8]								ICR3H
(0x96)	ICR3[7:0]								ICR3L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	