

# CONFIGURACIÓN DE LOS PWM

Para la maqueta de la bolera, hay que controlar hasta 5 motores. Por tanto, quiero utilizar el mismo timer para hacer el PWM para varios de esos motores.

Antes de nada, supongo que las señales PWM tienen que ser las que vayan a la señal Enable de los motores.

Por ejemplo, pongamos que uso el timer5. Lo primero que hago es ver en el datasheet lo siguiente:

Table 13-33. Port L Pins Alternate Functions

Port Pin	Alternate Function
PL7	–
PL6	–
PL5	OC5C (Output Compare and PWM Output C for Timer/Counter5)
PL4	OC5B (Output Compare and PWM Output B for Timer/Counter5)
PL3	OC5A (Output Compare and PWM Output A for Timer/Counter5)
PL2	T5 (Timer/Counter5 Clock Input)
PL1	ICP5 (Timer/Counter5 Input Capture Trigger)
PL0	ICP4 (Timer/Counter4 Input Capture Trigger)

Aquí veo que en los pines PL3, PL4 y PL5 están los canales A,B y C del timer 5 respectivamente.

En cada uno de los canales, el registro OC5(A,B o C), me sirve para establecer el valor con el que quiero que el PWM cambie de valor. Es decir, en esos registros, establezco el ciclo de trabajo.

Por tanto, de momento ya he visto que tengo que conectar las señales de enable en los pines 3,4,5 del puerto L.

```
#define M1_EN PL3
#define M2_EN PL3
#define M3_EN PL3

void setup(){
    DDRL |= ((1<<M1_EN) | (1<<M2_EN) | (1<<M3_EN));
}
```

Lo siguiente que hago es establecer el timer en modo phase correct; lo busco en el datasheet:

**Table 17-2.** Waveform Generation Mode Bit Description<sup>(1)</sup>

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Para este modo de funcionamiento, necesito los bits WGMn3:0 = “1010”.

#### 17.11.4 TCCR5A – Timer/Counter 5 Control Register A

### 17.11.8 TCCR5B – Timer/Counter 5 Control Register B

Bit (0x121)	7	6	5	4	3	2	1	0	
	ICNC5	ICES5	–	WGM53	WGM52	CS52	CS51	CS50	TCCR5B
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	

Llevándolo al código:

```
TCCR5A |= (1<<WGM51);
TCCR5B |= (1<<WGM53);
```

Ahora ya tengo seleccionado qué timer voy a usar (el 5) y lo tengo en modo phase correct.

Lo siguiente que hago será establecer el prescalado: utilizaré un prescalado de 8: (elijo este por comodidad, ya que la frecuencia del reloj son 8MHz)

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

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (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.

Según la tabla 17-6, tengo que poner los bits CSn2:0 = “010” : Busco en qué registro están:

### 17.11.8 TCCR5B – Timer/Counter 5 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x121)	ICNC5	ICES5	–	WGM53	WGM52	CS52	CS51	CS50	TCCR5B
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	

Añado estos bits a la configuración del TCCR5B que había hecho antes:

```
TCCR5B |= ((1<<WGM53) || (1<<CS51));
```

(sólo tengo que poner a '1' el bit de CS51).

Lo siguiente será determinar hasta qué valor quiero contar, ya que como vi antes, el TOP estará en el registro ICR5:

Con un prescalado de 8, el timer5 contará un pulso cada 1µS. Con un valor TOP de la cuenta de 1000 (ICR5 = 1000), el timer contará 1mS de subida, y 1mS de bajada.

```
ICR5 = 1000;
```

Lo siguiente es hacer que el PWM cambie de valor alto a bajo (o viceversa) al llegar al valor del ciclo de trabajo en la cuenta de subida y de bajada. Quiero trabajar en un modo invertido (Que baje en la igualdad de cuenta ascendente, y suba en la descendente).

Viendo el datasheet:

**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

Note: A special case occurs when OCRnA/OCRnB/OCRnC equals TOP and COMnA1/COMnB1/COMnC1 is set. See ["Phase Correct PWM Mode" on page 148](#) for more details.

Tengo que poner COM5(A,B,C)1 = 1 y COM5(A,B,C)0 = 0. Buscándolos en los registros:

#### 17.11.4 TCCR5A – Timer/Counter 5 Control Register A

Bit (0x120)	7	6	5	4	3	2	1	0	
	COM5A1	COM5A0	COM5B1	COM5B0	COM5C1	COM5C0	WGM51	WGM50	TCCR5A
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	

Y, llevándolo al código, tengo que “unirlos” a la configuración del registro TCCR5A que comencé antes:

```
TCCR5A |= (1<<WGM51) | (1<<COM5A1) | (1<<COM5B1) | (1<<COM5C1);
```

Lo último que me queda es poner el valor con el que quiero que la señal PWM cambie de valor alto a bajo (o viceversa). Es decir, establezco el ciclo de trabajo:

Ya que he utilizado el prescalado de 8, y que cuento hasta 1000 en la subida y en la bajada, para un ciclo de trabajo D, el valor de los registros OC5(A,B,C) será

$\frac{D}{100} \cdot 1000$ . Pongamos que quiero un ciclo del 25% (D = 0,25):

OCR5A = OCR5B = OCR5C = 0,25 \* 1000 = 250.

En el código:

```
#define d_PWM5 250
```

```
OCR5A = d_PWM5;  
OCR5B = d_PWM5;  
OCR5C = d_PWM5;
```

Finalmente, la configuración completa del timer 5 para usarlo en modo PWM phase correct con un ciclo de trabajo del 25% es:

```
#define M1_EN PL3  
#define M2_EN PL4  
#define M3_EN PL5  
  
#define d_PWM5 250  
  
void setup(){  
    DDRL |= ((1<<M1_EN) | (1<<M2_EN) | (1<<M3_EN));  
}  
void setup_timer5(){  
    TCCR5A |= ((1<<WGM51) | (1<<COM5A1) | (1<<COM5B1) | (1<<COM5C1));  
    TCCR5B |= ((1<<WGM53) | (1<<CS51));  
  
    ICR5 = 1000;  
  
    OCR5A = d_PWM5;  
    OCR5B = d_PWM5;  
    OCR5C = d_PWM5;  
}
```