

## Práctica 3.

- **Encender un LED y cambiarle su intensidad usando PWM.**
- **Controlar un servomotor a través de un potenciómetro.**

### 3.1 Objetivo General:

1. Encender un LED y cambiarle su intensidad usando PWM, señales de modulación de ancho de pulsos (pulse-width modulation).
2. Controlar un servomotor a través de un potenciómetro.
3. Desarrollar el soft para controlar tanto el servo motor como la intensidad del led usando la salida PWM del timer 1 del microcontrolador Atmega328p

---

### 3.2 Materiales y Métodos:

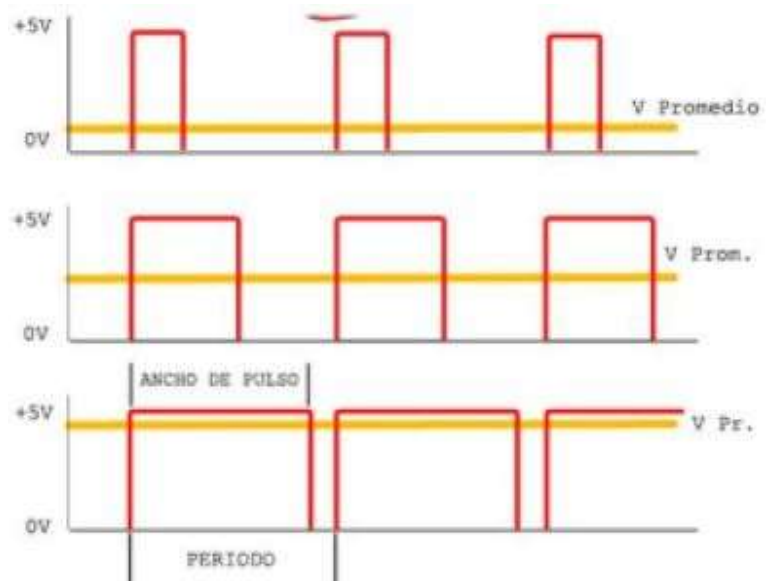
1. Un LED.
2. Un microcontrolador Atmega 328p ó Una tarjeta Arduino Uno-R3
3. Un cable USB impresora.
4. Un computador.
5. Cables para el montaje del circuito.
6. Tarjeta Protoboard.
7. Una Resistencia .
8. 1x Fuente de alimentación externa
9. Un Potenciómetro rotatorio.
10. Un servomotor.

### 3.2.1 Señales PWM

A menudo se necesita algo más que una señal de 0 o 1. Por ejemplo: variar la velocidad de giro de un motor, variar la intensidad de emisión un LED, o transmitir los grados de giro de un servo, entre otros ejemplos. Para todo esto nos servirá las PWM, que emula una señal analógica a partir de una señal digital.

Las siglas PWM vienen de *Pulse Width Modulation*, o *Modulación de Ancho de Pulso*.

Lo que hace este tipo de señal es emitir, en lugar de una señal continua de salida, es. (generar una serie de pulsos que se pueden variar en su duración pero a frecuencia constante. Así, la tensión promedio resultante es directamente proporcional a la duración de los pulsos dentro del rango del periodo indicado. Esto es, cuanto más juntos estén esos pulsos de +5v, mayor será la tensión promedio de nuestra salida, y cuanto más distantes sean estos, menor será la tensión. Lo anterior se puede observar en la Figura 1.



### 3.2.2 Modos de operación del timer 1 del atmega 328p

El Timer/Counter1 del AVR, tiene cuatro modos de funcionamiento: Normal, CTC, Fast PWM, Phase Correct PWM y Phase and Frequency Correct PWM. Estos modos, definen cómo se comporta el contador, y los módulos que hemos visto arriba por su parte dicen que se va a hacer con el valor del contador: poner una salida a alto, o cambiar el valor de un pin.

**En el modo normal**, el contador simplemente va incrementándose con cada ciclo de reloj. Si nuestro reloj va a 16Mhz, el contador incrementará en uno su valor cada 0.0625 microsegundos. Como, nuestro contador es de 16 bits, el valor máximo será de 0xFFFF que es lo mismo que 65535, y por tanto una vez que llegue a 4095.9375 us, se reiniciará.

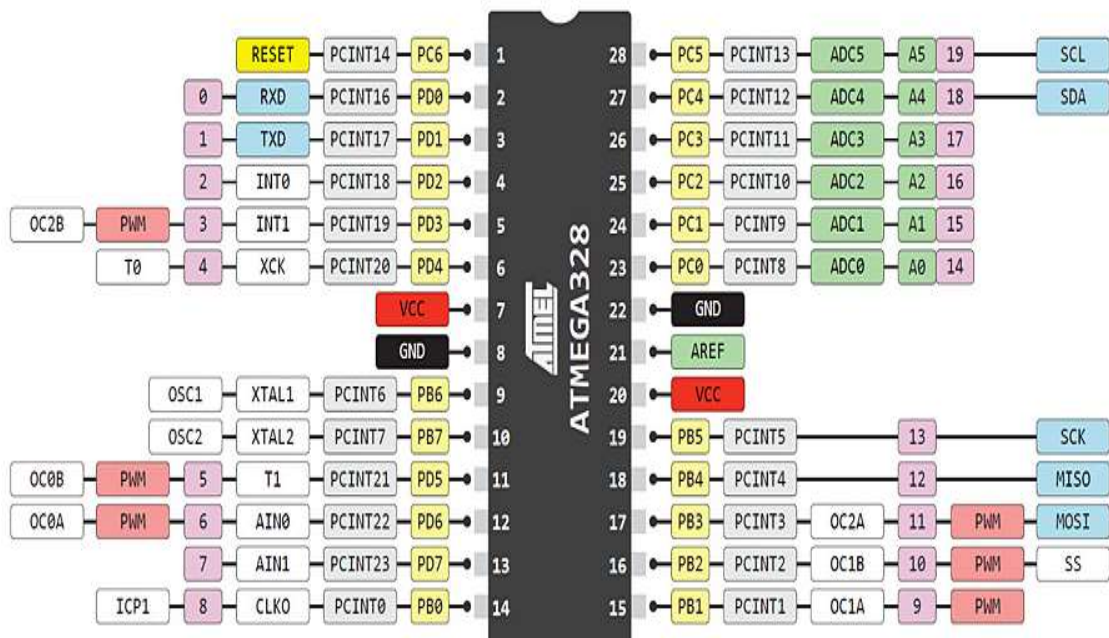
**El modo CTC** (Clear Timer on Compare Match), es muy parecido al modo normal, solo que en este caso, podemos decirle al contador que en vez de llegar a contar hasta el valor máximo (0xFFFF), cuando llegue a cierto valor, se reinicie y

empiece a contar de nuevo. De esta manera, tenemos un temporizador como el normal, solo que podemos configurar cada cuanto queremos que nos avise con una interrupción.

**Los modos de PWM**, son todas diferentes formas de generar señales PWM. Cada uno tiene sus peculiaridades, y es mejor para una aplicación u otra. Como esto es una introducción a los temporizadores, vamos a ir al más sencillo de entender, el Fast PWM.

El modo Fast PWM, consiste en que el contador va incrementando hasta llegar al valor máximo, pero mientras está subiendo, cuando el valor del contador es igual a uno que le hemos definido nosotros, cambia la salida (OCnx) a valor alto. Hasta que al llegar al valor máximo, lo vuelve a poner a nivel bajo. De esta manera, cambiando el valor del Compare Match en el cuál se va cambiar la salida, podemos variar la ancho del pulso.

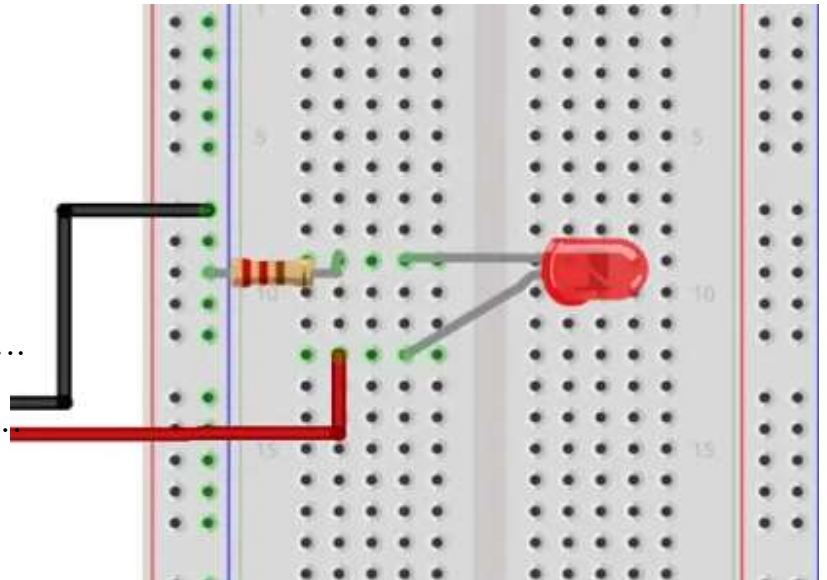
### 3.3 Montaje de la práctica:



## Montaje del circuito para Encender un LED y cambiarle su intensidad usando PWM.

Va a tierra.....

Al pin PWM.....

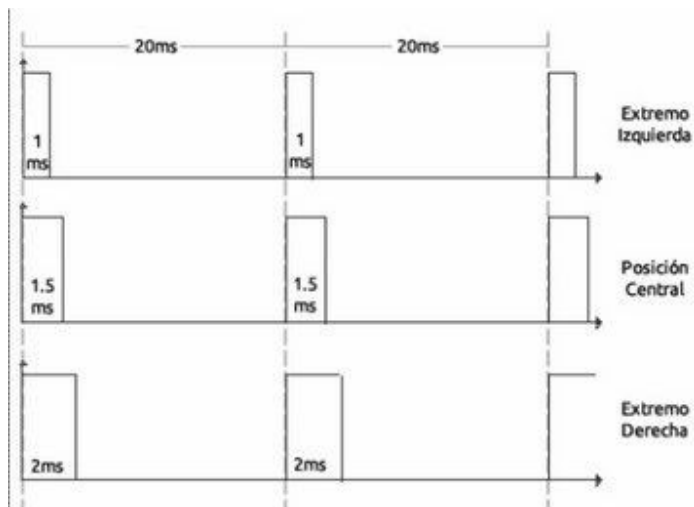


### 3.4 Servomotor

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua pero que tiene la capacidad de ubicarse en una posición específica dentro de su rango de operación. Además, puede mantenerse estable en dicha posición ya que dispone de un circuito electrónico de control.

Dependiendo del tipo de servomotor, este puede ser controlado tanto en velocidad como en posición.

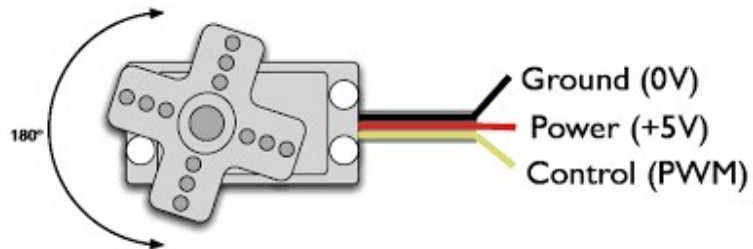
El PWM enviado al motor determina la posición del eje, y se basa en la duración del pulso enviado a través del cable de control; el rotor girará a la posición deseada.



El servomotor espera ver un pulso cada 20 milisegundos (ms) y la longitud del pulso determinará hasta dónde gira el motor. Por ejemplo, un pulso de 1.5ms hará que el motor gire a la posición de 90°.

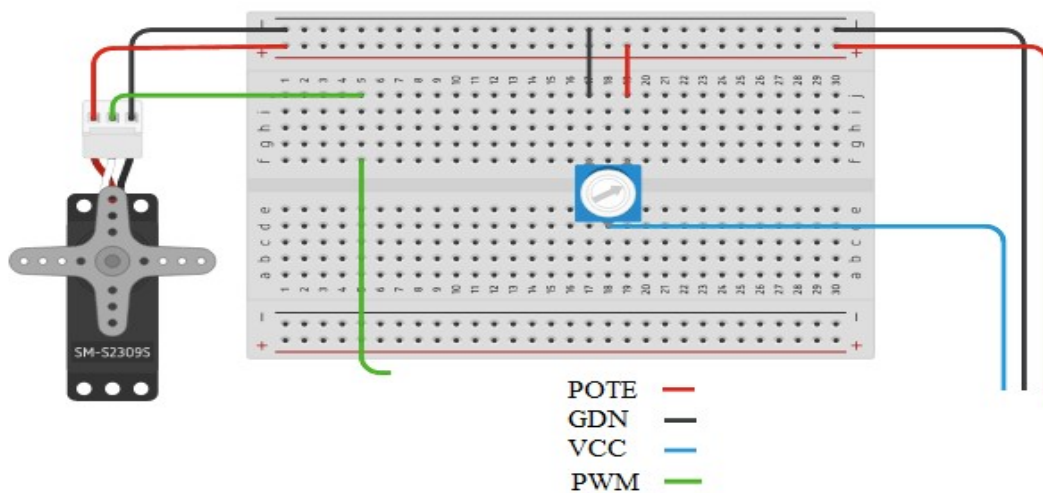
Si el tiempo es inferior a 1,5 ms, se mueve en sentido

contrario a las agujas del reloj hacia la posición de 0°, y si el tiempo es superior a 1,5 ms, el servo girará en sentido de las agujas del reloj hacia la posición de 180°.



Esquema de conexionado

### Montaje del circuito para un servo motor usando PWM.



## 4. Bibliografia

VR Microcontroller and Embedded Systems: Using Assembly and C  
Mazidi, Muhammad Ali,

CHAPTER 9:

AVR TIMER PROGRAMMING IN ASSEMBLY AND C  
SECTION 9.1:  
PROGRAMMING TIMERS 0, 1, AND 2

Chapter 15:

SECTION 15.2: WAVE GENERATION USING TIMER1

Chapter 16:

PWM AND DC MOTOR CONTROL SECTION

16.1: DC MOTOR INTERFACING AND SECTION

16.2: PWM MODES IN 8-BIT TIMERS87SECTION

16.3: PWM MODES IN TIMER1

## **5. Videos ilustrativos**

[https://youtu.be/4NzS7dDQ\\_t8](https://youtu.be/4NzS7dDQ_t8)

[https://youtu.be/2\\_c5i6bT1bc](https://youtu.be/2_c5i6bT1bc)