

Práctica 7:

# Modulación por Ancho de Pulso (PWM)

*Prof. Rubén Galicia Mejía*

Equipo:

**Los Pingüinos de Madagascar**

014 Integrantes:

- Ayala Fuentes Sunem Gizeht
- Dávila de Jesús Sandro Armando
- García García Francisco
- Guevara Badillo Areli Alejandra

Fecha de realización: *lunes, 10 de junio de 2024*

Fecha de entrega: *miércoles, 12 de junio de 2024*

## MATERIALES

- 3 servo motores MG996R
- Placa de pruebas (protoboard)
- Cables y conectores
- Placa o base para montar la FPGA

## INTRODUCCIÓN

En el desarrollo de robots, el control preciso de servomotores es crucial para garantizar movimientos exactos. Los servomotores se controlan mediante señales PWM (Modulación por Ancho de Pulso), que modulan el ancho de los pulsos para determinar la posición angular del motor. En esta práctica, programaremos señales PWM utilizando VHDL para controlar los servomotores de un robot.

## MARCO TEÓRICO

La modulación por ancho o de pulso (en inglés pulse width modulation PWM) es un tipo de señal de voltaje utilizada para enviar información o para modificar la cantidad de energía que se envía a una carga. Esta acción tiene en cuenta la modificación del proceso de trabajo de una señal de tipo periódico.

PWM es una técnica que se usa para transmitir señales analógicas cuya señal portadora será digital. En esta técnica se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga. El ciclo de trabajo (duty cycle) de una señal periódica es el ancho de su parte positiva, en relación con el período. Está expresado en porcentaje, por tanto, un duty cycle de 10% indica que está 10 de 100 a nivel alto, como se ilustra en la Ilustración 1.

$$\text{Duty cycle} = \frac{t}{T} * 100\%$$

t = tiempo en parte positiva  
T = Período, tiempo total

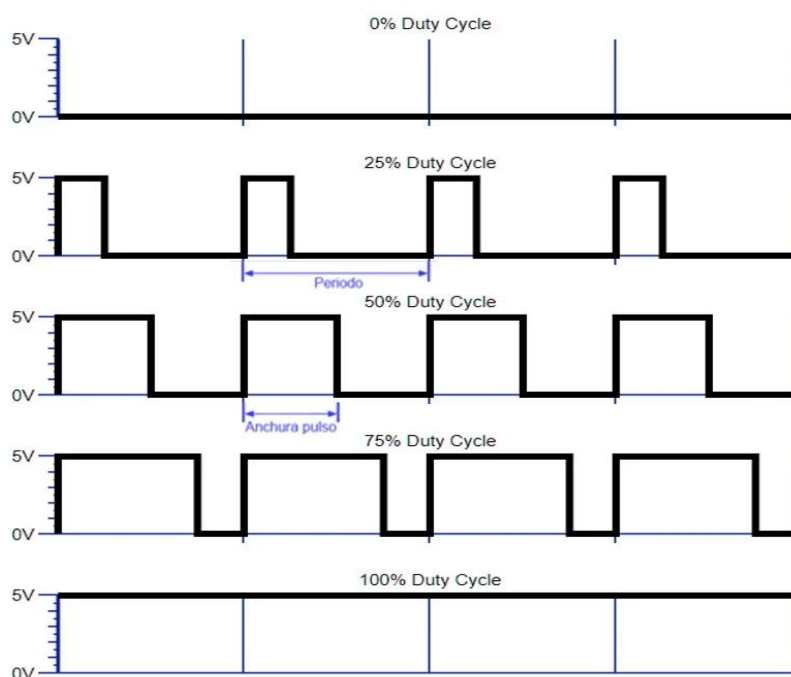


Ilustración 1. Ejemplos de ciclos de trabajo.

Básicamente, consiste en activar una salida digital durante un tiempo y mantenerla apagada durante el resto, generando así pulsos positivos que se repiten de manera constante. Por tanto, la frecuencia es constante (es decir, el tiempo entre disparo de pulsos), mientras que se hace variar la anchura del pulso, el duty cycle. El promedio de esta tensión de salida, a lo largo del tiempo, será igual al valor analógico deseado.

### PWM para el control de Servomotores

Los servomotores se manejan enviando un **pulso eléctrico de ancho variable**, también conocido como **modulación de ancho de pulso (PWM)**, a través del cable de control. Este pulso puede variar en duración, con un pulso mínimo, un pulso máximo y una frecuencia de repetición.

En términos generales, un servomotor sólo puede girar **90°** en cualquier dirección, lo que da un movimiento total de **180°** (Ilustración 2).



Ilustración 2. Colorimetría y alcance de un servomotor MG996R

Piensa en un reloj: la posición neutra del motor es como las 12 en punto, donde el servo tiene la misma cantidad de rotación potencial tanto en el sentido de las agujas del reloj como en el sentido contrario.

El **PWM** que se envía al motor es el coreógrafo de esta danza. Determina la posición del eje, y se basa en la duración del pulso enviado a través del cable de control. Así, el **rotor** girará a la posición deseada, como un bailarín girando en el escenario.

El servomotor, como un bailarín atento, espera ver un pulso cada **20 milisegundos (ms)**. La longitud de este pulso determinará hasta dónde gira el motor. Por ejemplo, un pulso de 1.5ms hará que el motor gire a la posición de 90°, como un bailarín girando un cuarto de vuelta.

Si el tiempo del pulso es inferior a **1,5 ms**, el servo se moverá en sentido contrario a las agujas del reloj hacia la posición de **0°**, como un bailarín girando hacia la izquierda. Y si el tiempo del pulso es superior a **1,5 ms**, el servo girará en sentido de las agujas del reloj hacia la posición de **180°**, como un bailarín girando hacia la derecha.

Cuando se les da la orden de moverse, los servos se moverán a la posición y mantendrán esa posición. Si una fuerza externa empuja contra el servo mientras mantiene una posición, el servo se resistirá a salir de esa posición, como un bailarín manteniendo su equilibrio ante una ráfaga de viento.

## DESARROLLO DE LA PRÁCTICA

En esta práctica, combinaremos el desarrollo de código en VHDL para generar señales PWM con el ensamblado y prueba de un robot controlado por servomotores. La Modulación por Ancho de Pulso (PWM) es esencial para controlar la posición y velocidad de los servomotores, componentes críticos en robótica. Utilizando VHDL, diseñaremos un generador de PWM, ensamblaremos el robot, y verificaremos su funcionamiento, integrando teoría y práctica en un proyecto completo.

### Desarrollo del código en VHDL

En esta sección, desarrollaremos el código en VHDL para generar señales PWM que controlarán los servomotores de nuestro robot. Usaremos una frecuencia de reloj de 50 MHz y ajustaremos los pulsos de ancho para controlar los grados de giro de los servomotores.

### Cálculos para los Grados de Giro

Frecuencia del Reloj: 50 MHz (50,000,000 Hz)

Periodo del Reloj:  $T = \frac{1}{50,000,000} \text{ seg} = 20 \text{ nanosegundos}$

Periodo del PWM: 20 ms (50 Hz), correspondiente a 1,000,000 ciclos de reloj.

Ancho del Pulso:

- 0 grados: 1 ms  $\rightarrow 1 \text{ ms} / 20 \text{ ns} = 50,000$  ciclos de reloj
- 90 grados: 1.5 ms  $\rightarrow 1.5 \text{ ms} / 20 \text{ ns} = 75,000$  ciclos de reloj
- 180 grados: 2 ms  $\rightarrow 2 \text{ ms} / 20 \text{ ns} = 100,000$  ciclos de reloj

Con estos cálculos, procederemos a implementar el código VHDL.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity pwm is
    Port (
        clk : in  STD_LOGIC;
        reset: in  STD_LOGIC;
        pwm_out : out  STD_LOGIC;
        pwm_out_r : out  STD_LOGIC;
        pwm_out_l : out  STD_LOGIC
    );
end pwm;

architecture Behavioral of pwm is
    signal counter : integer := 0;
    signal pwm_signal : STD_LOGIC := '0';
    signal pwm_signal_r : STD_LOGIC := '0';
    signal pwm_signal_l : STD_LOGIC := '0';

    signal pos : integer range 0 to 3 := 0;
    signal pos_counter : integer := 0;
begin
    process(clk, reset)
    begin
        if reset = '0' then
            pwm_signal <= '0';
            pwm_signal_r <= '0';
            pwm_signal_l <= '0';

            counter <= 0;
            pos_counter <= 0;
        elsif rising_edge(clk) then
            counter <= counter + 1;
            if counter = 1000000 then -- 1,000,000 ciclos corresponden a 20 ms
                counter <= 0;
                pos_counter <= pos_counter + 1;

                if pos_counter = 50 then -- Cambiar de posición cada 1 segundo (50 * 20 ms =
1s)
                    pos_counter <= 0;
                    pos <= (pos + 1) mod 4; -- Ciclo entre 0, 1, 2 y 3
                end if;
            end if;

            case pos is
                when 0 =>
                    if counter < 50000 then -- 0 grados (1 ms)
```

```

        pwm_signal <= '1';
    else
        pwm_signal <= '0';
    end if;
when 1 =>
    if counter < 75000 then -- 90 grados (1.5 ms)
        pwm_signal <= '1';
    else
        pwm_signal <= '0';
    end if;
when 2 =>
    if counter < 100000 then -- 180 grados (2 ms)
        pwm_signal <= '1';
    else
        pwm_signal <= '0';
    end if;
when 3 =>
    if counter < 50000 then -- 0 grados (1 ms)
        pwm_signal <= '1';
    else
        pwm_signal <= '0';
    end if;
when others =>
    pwm_signal <= '0';
end case;
end if;
end process;

pwm_out <= pwm_signal;
end Behavioral;

```

## Explicación del Código

1. Declaración de la Entidad y Arquitectura:
  - La entidad pwm define las entradas y salidas del módulo: el reloj (clk), el reset, y las salidas PWM para tres servomotores (pwm\_out, pwm\_out\_r, pwm\_out\_l).
2. Señales Internas:
  - counter: cuenta los ciclos de reloj hasta alcanzar 1,000,000 ciclos (20 ms).
  - pwm\_signal, pwm\_signal\_r, pwm\_signal\_l: señales de PWM generadas.
  - pos: indica la posición actual (0, 90, 180 grados).
  - pos\_counter: cuenta el tiempo para cambiar de posición.
3. Proceso Principal:
  - Reset: Inicializa las señales y contadores.
  - Rising Edge del Reloj: Incrementa el contador y gestiona el cambio de estado el PWM cada 20 ms.
  - Generación de PWM: Según el valor de pos, se generan pulsos de diferente ancho para las posiciones 0, 90, y 180 grados.

## Ensamblado del circuito y Funcionamiento

En nuestro proyecto, ensamblamos un robot denominado "Bonnie", que consta de un torso y una cabeza hechos de cartón y foami. Utilizamos tres servomotores para proporcionar movimiento: uno para la cabeza y uno para cada brazo.

A continuación, se describe el proceso de ensamblaje y las dificultades encontradas durante la construcción.

## Construcción del Muñeco

Primero, diseñamos el torso y la cabeza de Bonnie utilizando cartón y foami. Recortamos y ensamblamos las piezas de cartón para formar la estructura del torso y la cabeza. Luego, cubrimos estas estructuras con foami para darle una apariencia más agradable y suave (Ilustración 3).

## Colocación de los Servomotores

Inicialmente, planeamos utilizar servomotores SG90 debido a su disponibilidad y tamaño compacto. Sin embargo, al montar estos servos en los orificios del torso, descubrimos que no podían soportar el peso de los brazos y la cabeza. Los brazos se caían y la cabeza no mantenía su posición correctamente.

Debido a esta dificultad, reemplazamos los SG90 por servomotores MG996R, que tienen mayor capacidad de carga. Los MG996R proporcionaron el torque necesario para mantener y mover los brazos y la cabeza de Bonnie de manera efectiva (Ilustración 4).



Ilustración 3. Diseño y construcción del muñeco.

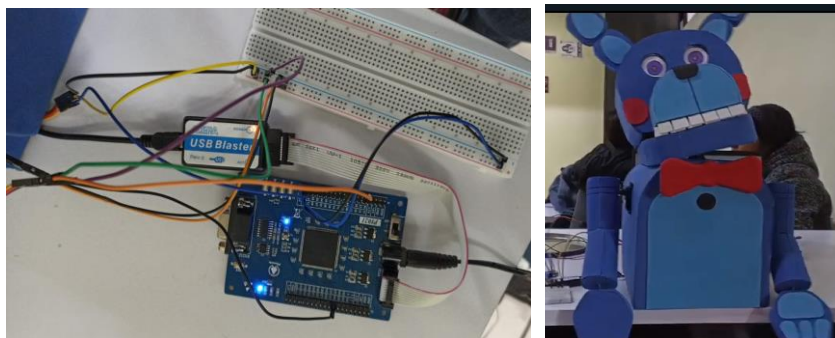


Ilustración 4. Robot ya completamente ensamblado y sus conexiones.

### 1. Cabeza:

- Colocamos el primer servomotor MG996R en el orificio superior del torso, diseñado específicamente para la cabeza.
- Fijamos la cabeza de cartón y foami al servo, asegurando que estuviera bien equilibrada y podía girar sin problemas.

### 2. Brazos:

- Insertamos un servomotor MG996R en cada orificio lateral del torso, destinados a los brazos.
- Montamos los brazos en cada servomotor, asegurándonos de que estuvieran firmemente sujetos y pudieran moverse adecuadamente.

## Prueba del Funcionamiento

Con todos los servomotores montados y los brazos y la cabeza fijados, procedimos a probar el funcionamiento del robot. Programamos los servomotores para realizar movimientos básicos y verificamos que cada uno respondiera correctamente a las señales PWM generadas por nuestro módulo VHDL.

### 1. Prueba de la Cabeza:

- La cabeza giró suavemente entre las posiciones programadas, demostrando que el servomotor MG996R podía soportar su peso y proporcionar un control preciso.

### 2. Prueba de los Brazos:

- Los brazos se movieron correctamente, levantándose y bajándose según las señales PWM. El reemplazo de los SG90 por los MG996R resultó en un movimiento estable y confiable.

Video del funcionamiento del robot:

[https://drive.google.com/file/d/1iDIBET7pYVO\\_oLiDIcQ6ilLbtqKonJF/view?usp=sharing](https://drive.google.com/file/d/1iDIBET7pYVO_oLiDIcQ6ilLbtqKonJF/view?usp=sharing)

## **CONCLUSIONES**

### **CONCLUSIONES 1 (Ayala Fuentes Sunem Gizeth)**

El ensamblaje del robot Bonnie, utilizando cartón y foami para su torso y cabeza, nos permitió experimentar de primera mano la relación crítica entre el diseño mecánico y la implementación de componentes electrónicos. La integración de los servomotores en la estructura del robot destacó cómo una planificación inicial adecuada puede influir significativamente en el éxito del montaje y el funcionamiento general. A pesar de las dificultades con los servomotores iniciales, la experiencia de realizar ajustes y mejoras nos mostró la importancia de una evaluación continua durante todo el proceso de desarrollo. Esta práctica nos ha enseñado a ser más meticulosos en la planificación y a estar siempre preparados para hacer modificaciones en función de los resultados obtenidos.

### **CONCLUSIONES 2 (Dávila de Jesús Sandro Armando)**

La práctica de ensamblaje y programación del robot Bonnie subrayó la importancia crucial de una selección adecuada de componentes. Inicialmente optamos por servomotores SG90, atraídos por su disponibilidad y tamaño compacto. Sin embargo, enfrentamos el desafío de que estos servos no podían soportar el peso de los brazos y la cabeza del robot, lo que resultó en movimientos ineficaces y problemas de estabilidad. La decisión de reemplazarlos por los servomotores MG996R, que tienen una mayor capacidad de carga, fue un giro clave en nuestro proyecto. Este proceso nos enseñó a evaluar mejor las especificaciones técnicas y a planificar en base a los requisitos reales del proyecto. Además, nos mostró la importancia de la flexibilidad y la adaptación rápida a los cambios necesarios, habilidades esenciales para el éxito en proyectos de ingeniería.

### **CONCLUSIONES 3 (García García Francisco)**

Nuestro equipo aprendió que los desafíos prácticos, como el cambio de servomotores, pueden ser una oportunidad para mejorar nuestras habilidades en resolución de problemas y adaptación. La necesidad de reemplazar los servos SG90 por los MG996R debido a problemas de carga nos obligó a reconsiderar nuestras elecciones iniciales y a realizar ajustes técnicos significativos. Esta experiencia reforzó la importancia de no solo basar nuestras decisiones en la disponibilidad de componentes, sino también en su capacidad para cumplir con los requisitos específicos del proyecto. Además, trabajar juntos para superar estos desafíos mejoró nuestra capacidad de colaboración y comunicación, esenciales para cualquier equipo de ingeniería.

### **CONCLUSIONES 4 (Guevara Badillo Areli Alejandra)**

La integración del sistema de control de servomotores a través de la programación en VHDL y la construcción física del robot Bonnie nos proporcionó una comprensión profunda y práctica de todo el proceso de desarrollo robótico. Desde la creación de las estructuras de cartón y foami hasta la programación de las señales PWM para controlar los movimientos, cada etapa del proyecto contribuyó a una visión holística del diseño y la implementación de sistemas robóticos. Esta experiencia no solo consolidó nuestros conocimientos técnicos, sino que también subrayó la importancia de la integración interdisciplinaria. La colaboración efectiva entre las áreas de diseño mecánico, programación y montaje físico fue fundamental para el éxito del proyecto. En resumen, esta práctica nos ha preparado mejor para enfrentar proyectos futuros con una comprensión sólida de la teoría y la práctica, así como con habilidades mejoradas en trabajo en equipo y resolución de problemas.

## BIBLIOGRAFÍAS

- Que es pwm y cómo funciona. (n.d.). Recuperado de <https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/4517-que-es-pwm-y-como-funciona-0689593953254.html>
- ¿Qué es PWM y cómo usarlo? (n.d.). Recuperado de <https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38>
- Kohlhase, K. (2020, September 15). La PWM: ¿Qué es? ¿Cómo puedo utilizarla? Recuperado de <https://www.digikey.com.mx/es/blog/pulse-width-modulation>
- Magazine. (2022, March 26). ¿Sabes lo que es la Función PWM? Recuperado de <https://ibertronica.es/blog/tutoriales/funcion-pwm/>
- Aula. (2023, December 12). Servomotores: Héroes Silenciosos de la Tecnología Moderna. Recuperado de <https://www.cursosaula21.com/que-es-un-servomotor/>