

Actividad 6

Controlador de motor por ancho de pulso (PWM)

1.- Introducción y objetivos

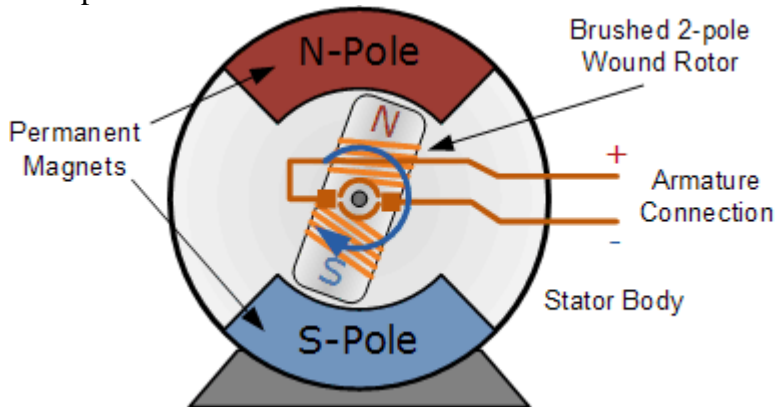
En esta práctica se va a implementar modulo capaz de generar una señal PWM (Pulse Width Modulation) para poder controlar el driver de un motor. Además de la descripción, la simulación y la síntesis, se completarán todas las fases del diseño digital en FPGA y se observará el resultado en una placa de desarrollo basys2.

Pulse Width Modulation

Hay muchas maneras de controlar un motor de DC pero la más común de ellas es mediante una señal PWM. Los motores de DC son los dispositivos más usados para producir en movimiento continuo rotacional y que puede ser fácilmente controlado.

Un motor de corriente continua consiste básicamente de dos partes, el cuerpo estacionario del motor llamado el "estator" y la parte interior que gira produciendo el movimiento llamado el "Rotor". Para las máquinas de corriente continua del rotor se denomina comúnmente la "armadura".

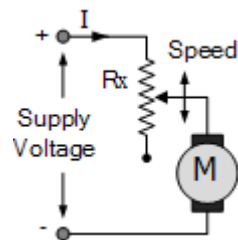
En general, en pequeños motores de corriente continua el estator consta de un par de imanes permanentes fijos que producen un flujo magnético uniforme y estacionario dentro del motor dando a estos tipos de motores de su nombre (PMDC) motores "de imán permanente de corriente continua".



La armadura consiste en bobinas eléctricas individuales conectados entre sí en una configuración circular alrededor de su cuerpo metálico para producir un polo norte y luego un Sur-Pole luego un polo norte etc, tipo de configuración de sistema de campo.

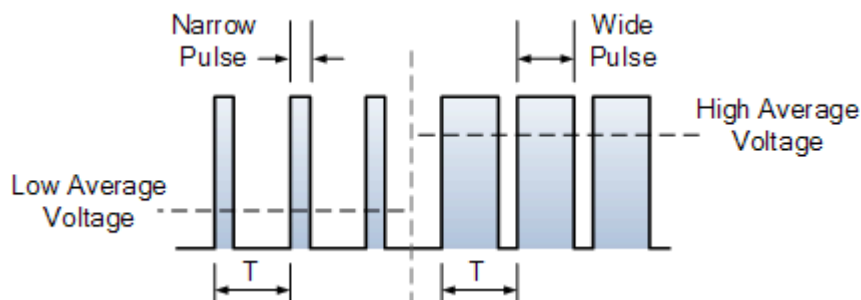
La corriente que fluye dentro de estas bobinas del rotor produce el campo electromagnético necesario para generar el movimiento. El campo magnético circular producido por los armaduras produce tanto polos norte los cuales son se repelidos o

atraído por los imanes permanentes del estator, esto produce un movimiento de rotación alrededor del eje central de motores como se muestra.



Una forma de controlar el flujo de corriente que fluye por motor de DC es mediante el uso de una Resistencia variable (ver figura anterior) en serie con el motor. Esta solución puede producir perdidas por calor en la resistencia. Una manera más simple de controlar la velocidad del motor es regular el voltaje en sus terminales mediante una señal PWM ver figura 3.

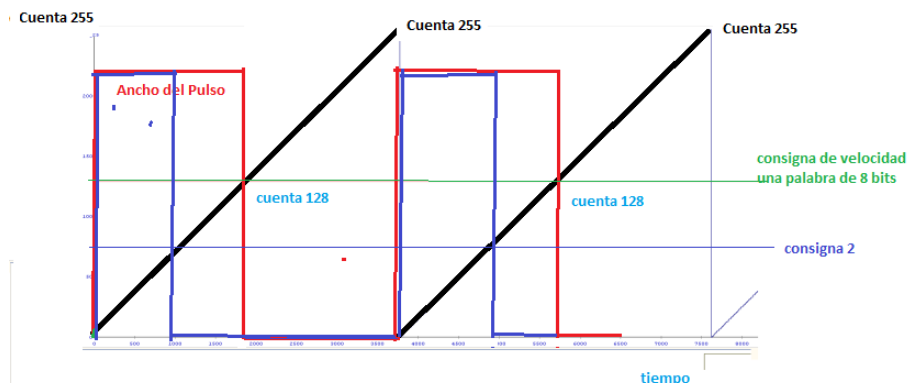
Como su nombre indica, PWM controla la velocidad del motor con una serie de pulsos ON-OFF y variando su duty cycle, fracción de tiempo en ON con respecto a OFF, con frecuencia constante. La potencia aplicada al motor puede ser controlada variando el ancho del pulso y por lo tanto variando el voltaje promedio de DC aplicado al motor. Cambiando el tiempo de estos pulsos en estado ON aumentara la velocidad del motor y viceversa para tiempos más pequeños en estado ON ira más despacio.



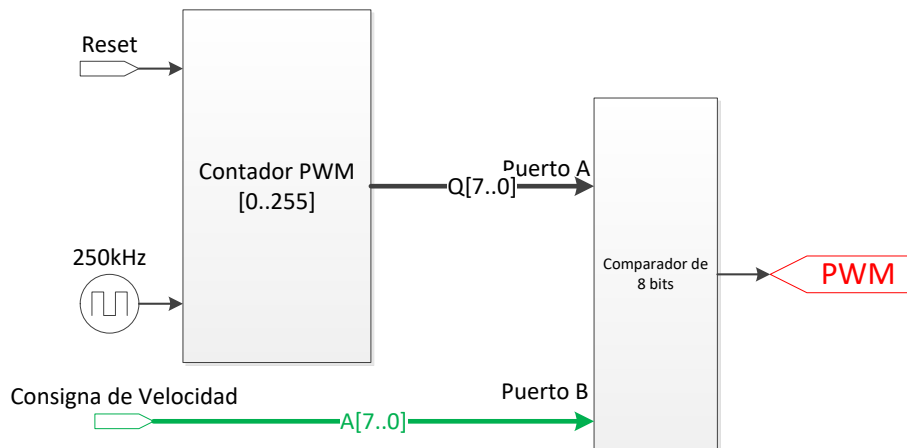
2.- Generación de la señal PWM

El diseño del generador de PWM estaría formado un comparador, un contador PWM y un prescaler.

El comparador de 8 bits tendría como entrada la salida del contador PWM y la consigna de ocho bits que viene determinar la velocidad deseada.

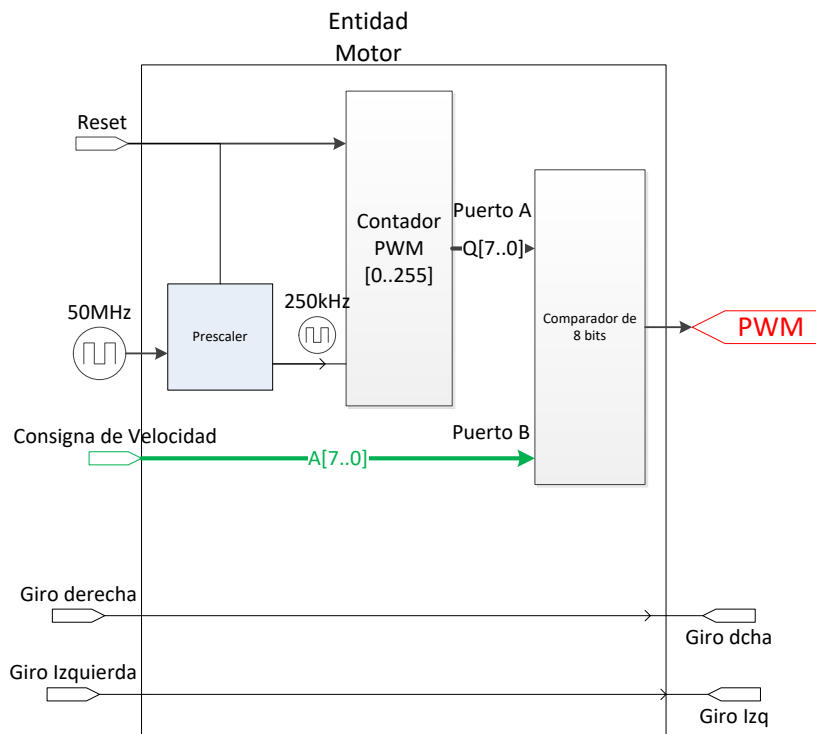


La figura anterior ilustra cómo trabaja el comparador, mientras la cuenta del contador (recta en negro) sea menor o igual que la consigna (ejemplo 128) recta verde este estaría dando una salida a '1', a partir de ahí la señal pwm cae a cero hasta que la cuenta del contador llegue a su máximo valor. Este se resetea y empieza a contar de nuevo con lo cual se vuelve a cumplir que la cuenta es menor que la consigna. Variando la consigna conseguimos que el ancho del pulso cambie de la siguiente forma, a mayor consigna mayor ancho de pulso y viceversa a menor consigna menor ancho. Como comentamos anteriormente con esto conseguimos que el voltaje promedio en los terminales del motor varíe.



2.- Control del motor

Una vez generada la señal PWM es necesario controlar el sentido de giro de del motor. En nuestro caso particular para suministrarle la energía necesaria el motor usaremos un circuito integrado (L). Dicho circuito tiene tres entradas una para la velocidad (Señal PWM), y otras dos para el sentido de giro. Para probar dicha función usaremos dos swich de la pala que conectaremos a dichas entradas a través de nuestro módulo PWM (en la FPGA). Crear un módulo que se llame motor como el que se muestra a continuación.



3.- Diseño del Preescaler,

El Contador Preescaler es un contador que genera una nueva señal de Reloj con una frecuencia de 250Khz, cada vez que exista un flanco positivo del reloj de entrada incrementa la variable cuenta y la compara con un numero 976, que viene hacer el semiperiodo de la señal de reloj de salida. Una vez que se cumpla la igualdad se niega dicha señal y se resetea la variable cuenta. Con esto obtenemos una nueva frecuencia de reloj **clk_250Khz**;

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

entity cont_preescalado is
    port (clk :          in std_logic; -- Reloj de la tarjeta 50MHz
          reset :       in std_logic;
          clk_250Khz :   out std_logic); --Salida: reloj que usará el contador PWM
                                (250KHz)
end cont_preescalado;

architecture Behavioral of cont_preescalado is
    constant preescal:std_logic_vector(11 downto 0):= CONV_STD_LOGIC_VECTOR(976, 12);
    signal clk_out_aux : std_logic:= '0';
    begin
    output: process(clk,reset)
```

```

variable cuenta : std_logic_vector(11 downto 0):="0000000000000";
begin
    if (reset='1') then -- Reset asíncrono
        cuenta:=(others=>'0'); -- Inicializar contador
    elsif (clk'event and clk='1') then -- Flanco de subida en reloj
        cuenta:=(cuenta+1); -- Incrementar contador
        if (cuenta=preescal) then -- usamos la mitad del periodo
            clk_out_aux <= not clk_out_aux;
            cuenta:=(others=>'0'); -- Inicializar contador
        end if;
    end if;
end if;

end process;
clk_250Khz <= clk_out_aux;
end Behavioral;

```

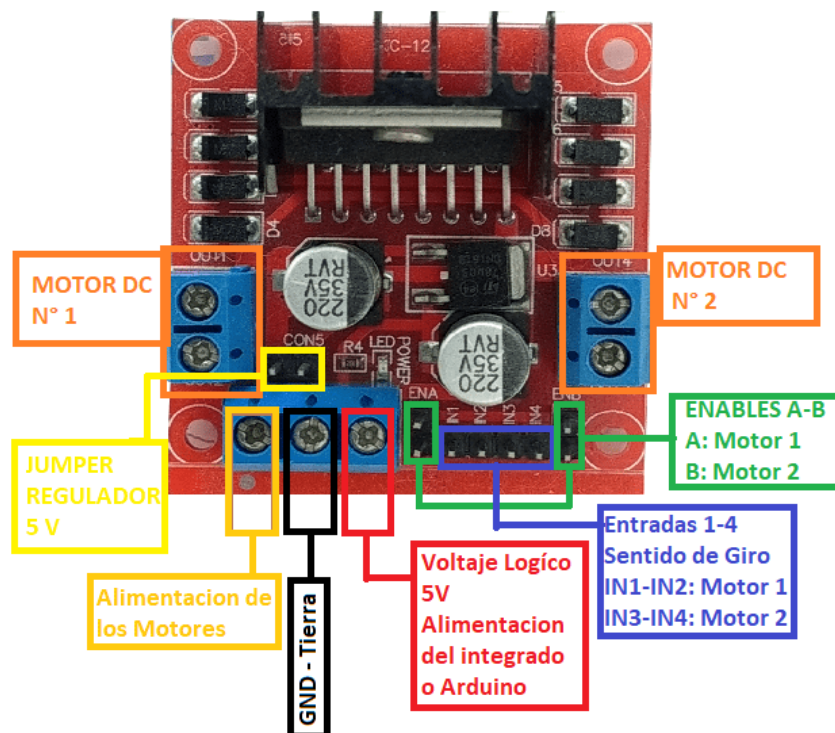
4. Implementacion del controlador en FPGA.

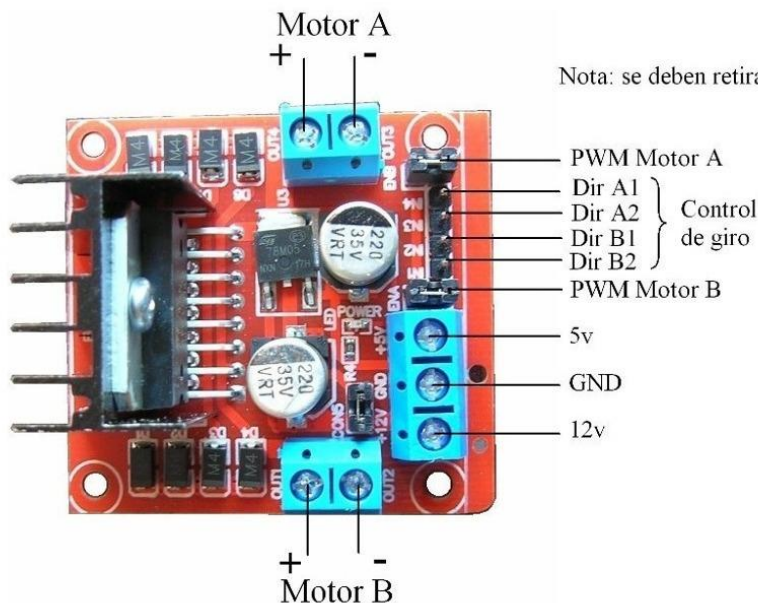
Una vez Simulado el circuito y comprobar que funciona perfectamente. Pasamos a la Fase de Implenetacion en la FPGA. Ya sea en la Nexys A o Basys 2.

Pines Basys 2

Pines Nexy A7

Puente en H L298. Interface de conexión y tabla de funcionamiento.





Driver PWM
Puente H
L298N

Conexión con FPGA placa Basys 2.

Por ejemplo, Usaremos los conectores JA de la Basys 2 para el motor A y el conector JB para el motor B. Cada uno de estos conectores tiene 4 pines conectados a la FPGA. Ver Manual de referencia para pines .

Conexión con FPGA placa Nexys A7.

Por ejemplo, Usaremos los conectores JA de la Nexys A7 para el motor A y el conector JB para el motor B. Cada uno de estos conectores tiene 4 pines conectados a la FPGA. Ver Manual de referencia para pines

Ver figura (Conexión con Basys 2)

