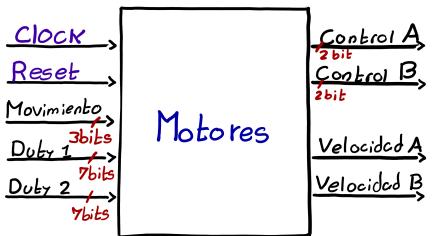
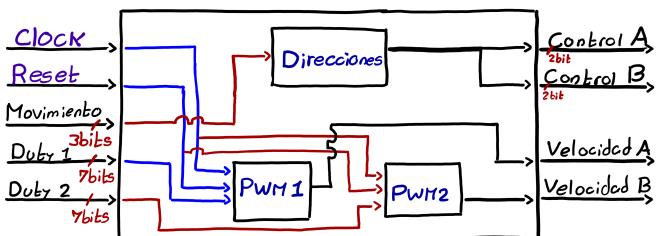


Proyecto Electronica digital → Motores

Modulo Basico (Sin Bus)

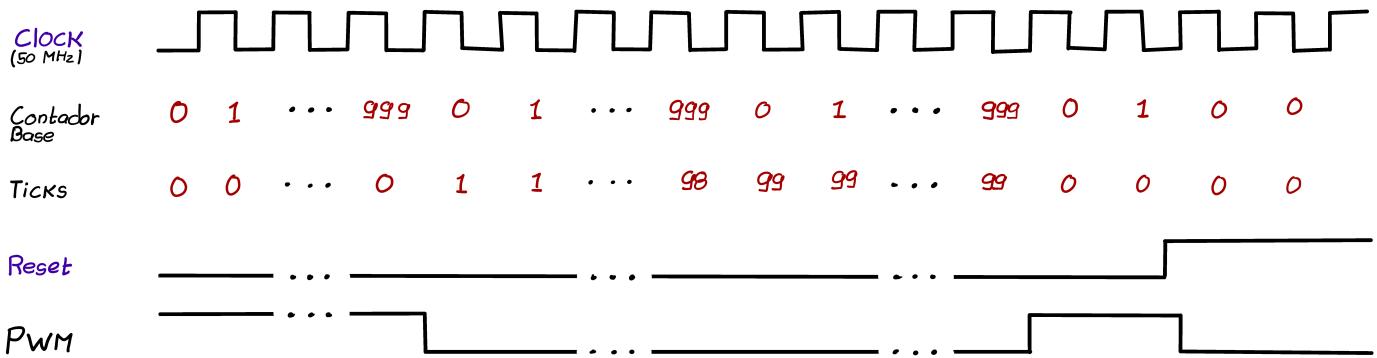


→ Diagrama Estructural



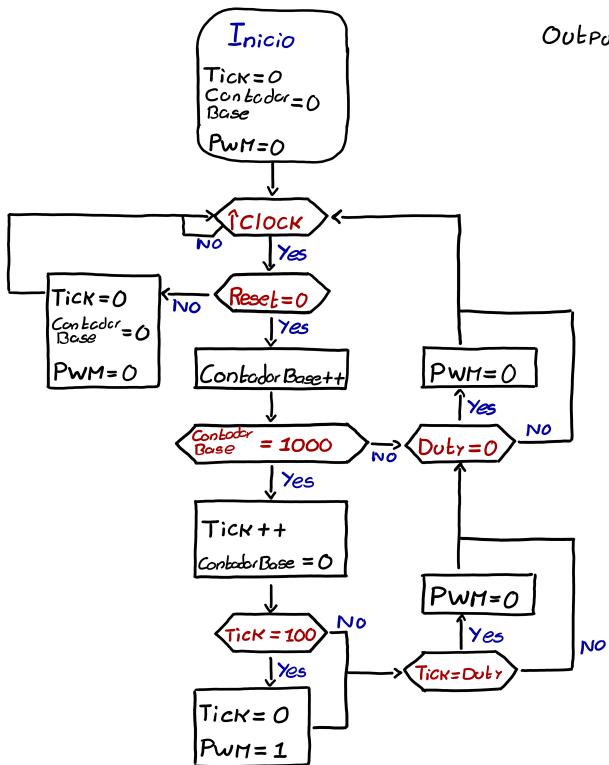
PWM

→ Diagrama de señales



Proyecto Electronica digital → Motores

→ Diagrama Funcional



Output Reg → PWM
 Reg → Contador Base (10 bits)
 Reg → Tick (7 bits)

Input → Clock
 Input → Reset
 Input → Duty (7 bits)

* El PWM se diseño con una frecuencia de 500Hz y un ciclo útil de 100 valores. Por lo que la duración del pulso/resolución es de:

$$\text{Suponiendo } \text{Clock} = 50\text{MHz} \rightarrow \text{Res} = \frac{1}{500\text{Hz} \times 100} = 20\text{ms}$$

$$\text{Ancho del Pulso: } t = 20\text{ms} \times \text{Duty}, \quad \text{Duty} \leq 100$$

Direcciones

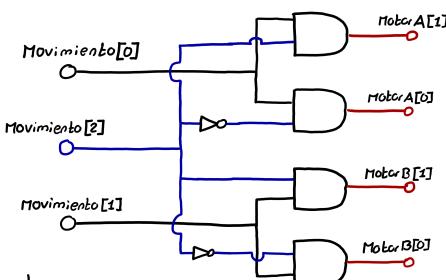
El Controlador L298N tiene una tabla que muestra el sentido de giro del motor según las entradas. Por lo tanto se tiene la siguiente tabla:

In1	In2	Motor
0	0	No hay Giro
1	0	Giro Derecha
0	1	Giro Izquierda

Teniendo esto en cuenta, al combinar 2 motores se pueden generar movimientos. Por lo tanto se quiere que:

Movimiento	Acción	Motor A	Motor B
0 0 0	Estor quieto	0 0 0 0	0 0 0 0
0 0 1	Rotar derecho - Avanzar	0 1 0 0	0 0 0 0
0 1 0	Rotar Izquierda - Avanzar	0 0 0 1	0 0 0 0
0 1 1	Avanzar	0 1 0 1	0 0 0 0
1 0 0	Estor quieto	0 0 0 0	0 0 0 0
1 0 1	Rotar derecho - Retroceso	1 0 0 0	0 0 0 0
1 1 0	Rotar Izquierda - Retroceso	0 0 1 0	0 0 0 0
1 1 1	Retroceder	1 0 1 0	0 0 0 0

Teniendo un comportamiento lógico de entrada y salida, se puede implementar un circuito puramente combinacional como el siguiente:

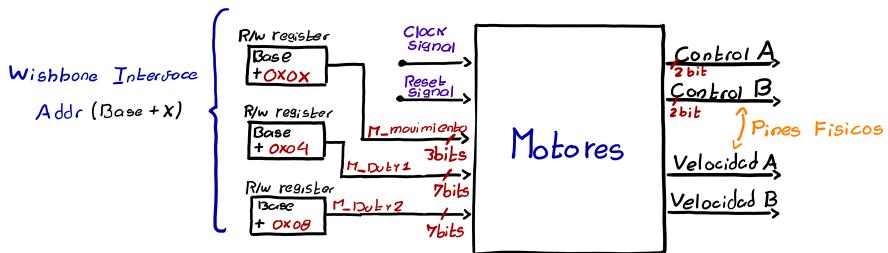


* Ambos motores tienen el mismo giro a pesar de estar invertidos, esto debido a que uno de los motores está cableado al revés.

* En caso de que se avance o retroceda y PWM1 ≠ PWM2 se presentara un giro con eje en el exterior.

Proyecto Electronica digital → Motores

Modulo wishbone



*Clock Signal es el reloj del sistema

*Reset Signal es la señal de reinicio del sistema

Código C (Picavc 32)

Teniendo en cuenta la documentación previa, el código necesario para el correcto de los motores es el siguiente:

```
Motores_Vel(int Pwm1, int Pwm2) {
    Motores_Duty1_Write(Pwm1); —> Escribe el Valor del ciclo útil en el Pwm 1 escribiendo el valor en el registro (Base + 0x04)
    Motores_Duty2_Write(Pwm2); —> Escribe el Valor del ciclo útil 2 en el Pwm2 escribiendo el valor en el registro (base + 0x08)
}
```

```
Motores_Dir(int Direccion) {
    Motores_Movimiento_Write(Direccion); —> Escribe la dirección segun la tabla de la documentacion Actualizando el registro (Base + 0x00)
}
```