

PWM - Pulse Width Modulation

Pulse Width Modulation o PWM es una técnica común que se utiliza para cambiar el ancho de pulsos digitales. PWM tiene muchas aplicaciones como controlar la velocidad, potencia o giro de motores, la intensidad o intermitencia de LED's, entre otras.

Principio básico

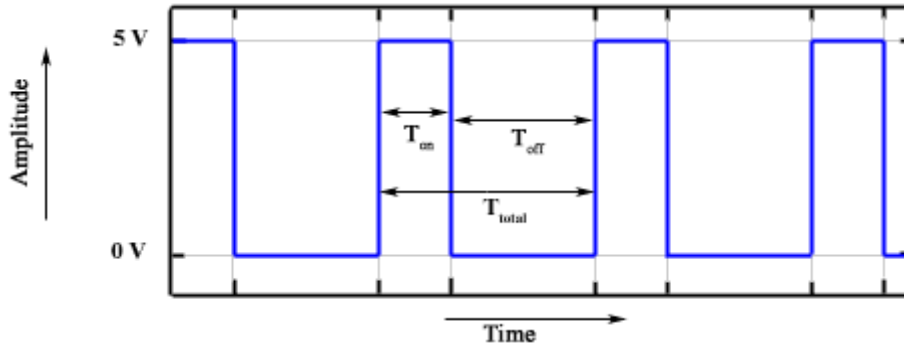


Imagen 1 - Señal PWM

Pulse Width Modulation es básicamente una señal cuadrada con un tiempo de encendido-apagado variable. Hay varios términos asociados con PWM:

- On-Time – tiempo que la señal esta en nivel HIGH.
- Off-Time – tiempo que la señal esta en nivel LOW.
- Period – Se representa por la suma del tiempo de encendido y el tiempo de apagado.
- Duty Cycle – Se representa por el porcentaje del tiempo que la señal dura en nivel HIGH

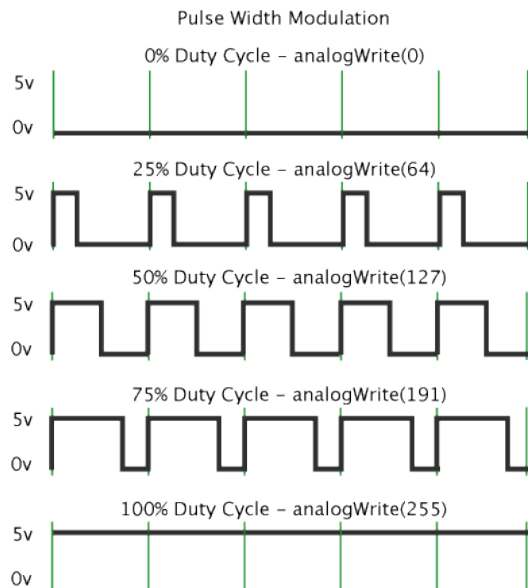


Imagen 2 - Representación del Duty Cycle de una señal PWM

La función `analogWrite()` escribe un valor análogo (onda PWM) a un pin. Después de una llamada a esta función, el pin generará una señal cuadrada estable hasta la siguiente llamada `analogWrite()`, una llamada a `digitalRead()` o `digitalWrite()` al mismo pin.

En las placas Arduino existen pines dedicados exclusivamente para esta función. Esto quiere decir que aunque su funcionamiento es idéntico a los de los otros pines, para aplicaciones con PWM solamente los pines marcados con una tilde (~) podrán ser utilizados para dicha aplicación.



Imagen 3 - Ubicación de los pines PWM en Arduino UNO (3, 5, 6, 9, 10, 11)

Referencias:

Imagen 1 – Señal PWM (2018). [Imagen] Recuperado de:
<https://www.tutorialspoint.com/arduino/images/pwm.jpg>

Imagen 2 - Representación del Duty Cycle de una señal PWM (2018). [Imagen] Recuperada de:
<https://www.arduino.cc/en/uploads/Tutorial/pwm.gif>

Hirzel, T (2018). Arduino – PWM. Recuperado de: <https://www.arduino.cc/en/Tutorial/PWM>

Arduino – Pulse Width Modulation. (2018). Recuperado de:
https://www.tutorialspoint.com/arduino/arduino_pulse_width_modulation.htm