

SEÑALES DIGITALES Y ANALÓGICAS CON ARDUINO

Alumnos:

José Moisés Martínez Hernández

Sindy Fabiola Perdomo Rapalo

Generando pulsos con digitalWrite()

Tabla 1 Registro de periodo calculado

N o	Frecuencia	Periodo	delay
1	2 Hz	500 ms	250 ms
2	5 Hz	200 ms	100 ms
3	10 Hz	100 ms	50 ms

Tabla 2 Registro de tiempos en HIGH y LOW.

No	Frecuencia	% Duty Cycle	Periodo	delay
1	1 Hz	90%	$t_H = 900 \text{ ms}$ $t_L = 100 \text{ ms}$	$t_H = 900 \text{ ms}$ $t_L = 100 \text{ ms}$
2	1 Hz	60%	$t_H = 600 \text{ ms}$ $t_L = 400 \text{ ms}$	$t_H = 600 \text{ ms}$ $t_L = 400 \text{ ms}$
3	1 Hz	30%	$t_H = 300 \text{ ms}$ $t_L = 700 \text{ ms}$	$t_H = 300 \text{ ms}$ $t_L = 700 \text{ ms}$

Tabla 3 Registro de tiempos en HIGH y LOW.

No	Frecuencia	% Duty Cycle	Periodo	valor
1	490 Hz	90%	$t_H = 1.836\text{ ms}$ $t_L = 0.204\text{ ms}$	$t_H = 230$ $t_L = 26$
2	980 Hz	60%	$t_H = 0.612\text{ ms}$ $t_L = 0.408\text{ ms}$	$t_H = 153$ $t_L = 102$
3	490 Hz	30%	$t_H = 0.612\text{ ms}$ $t_L = 1.428\text{ ms}$	$t_H = 77$ $t_L = 178$

RESULTADOS

Tabla 1

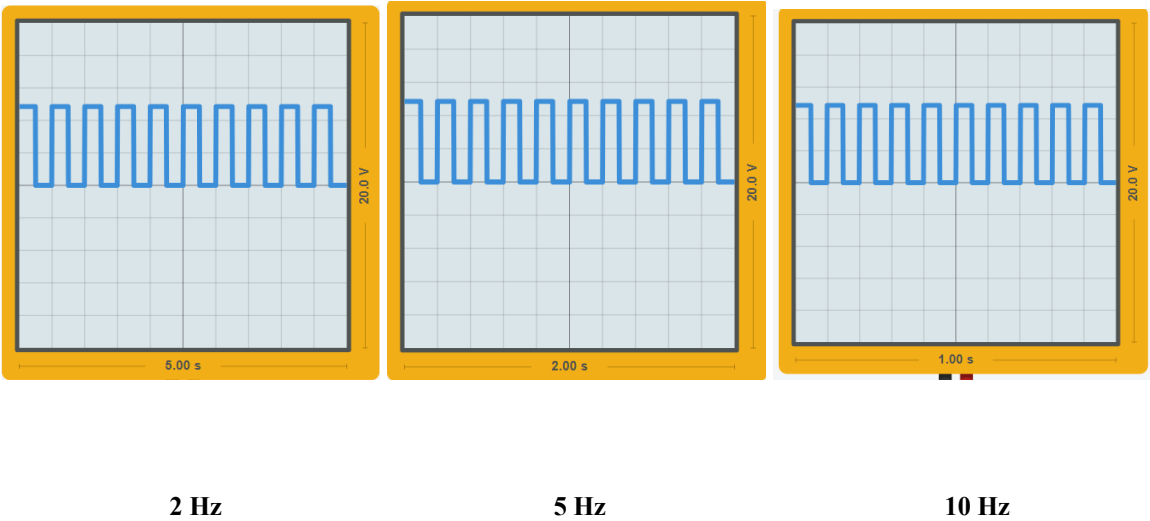
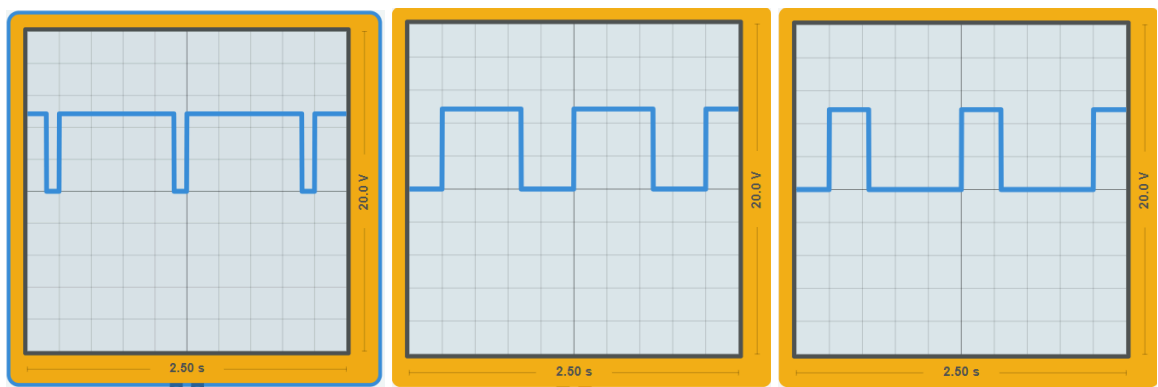


Tabla 2

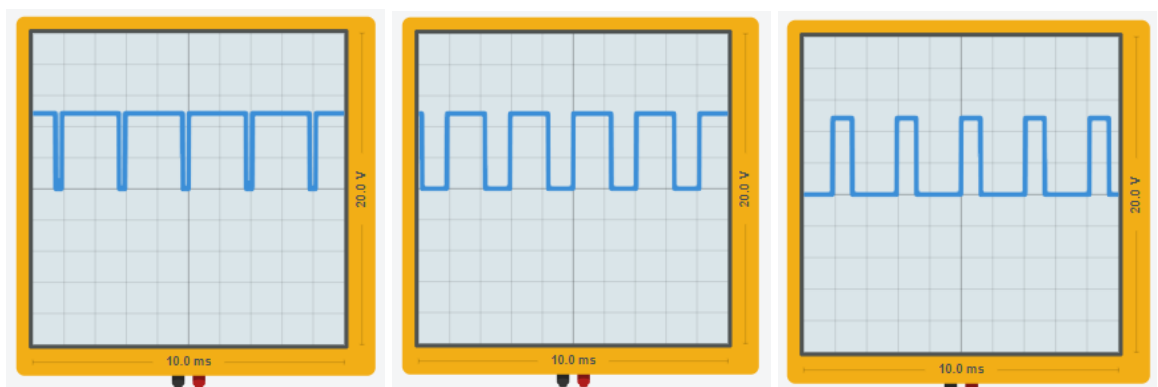


90 %

60 %

30 %

Tabla 3

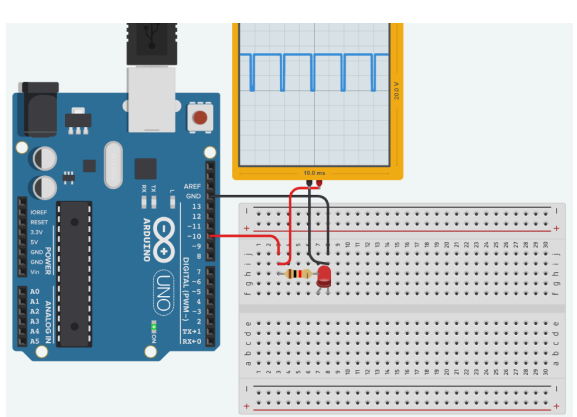


90 %

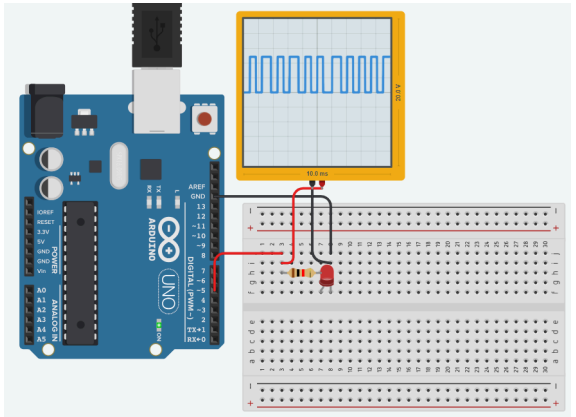
60 %

30 %

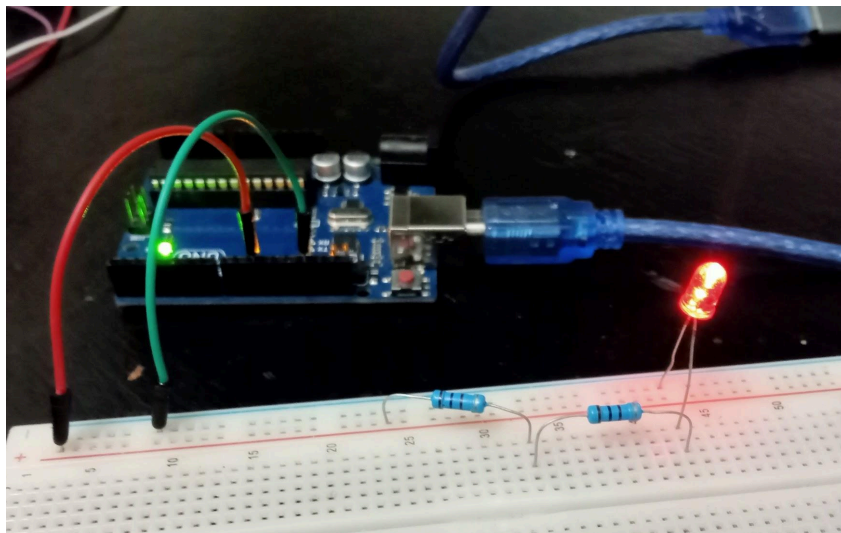
**Configuración
980 Hz**



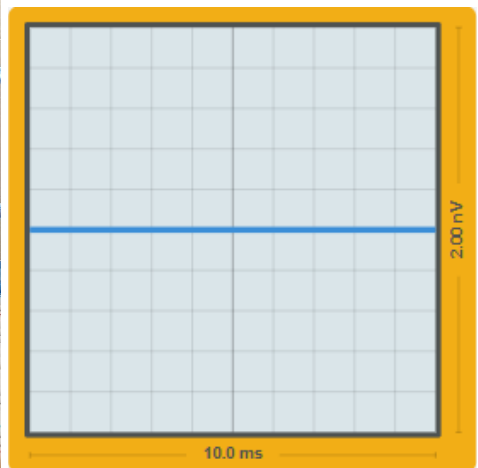
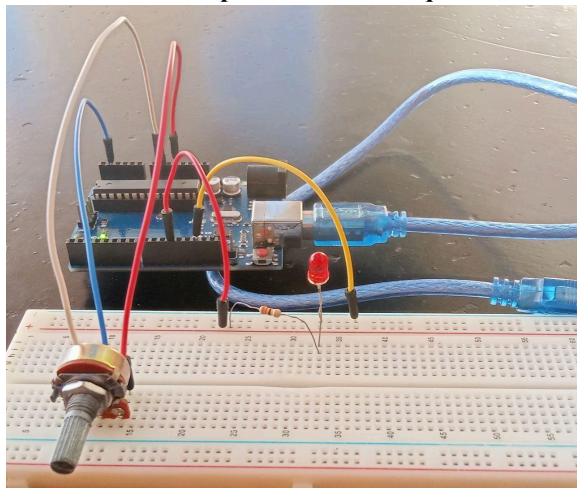
480 Hz



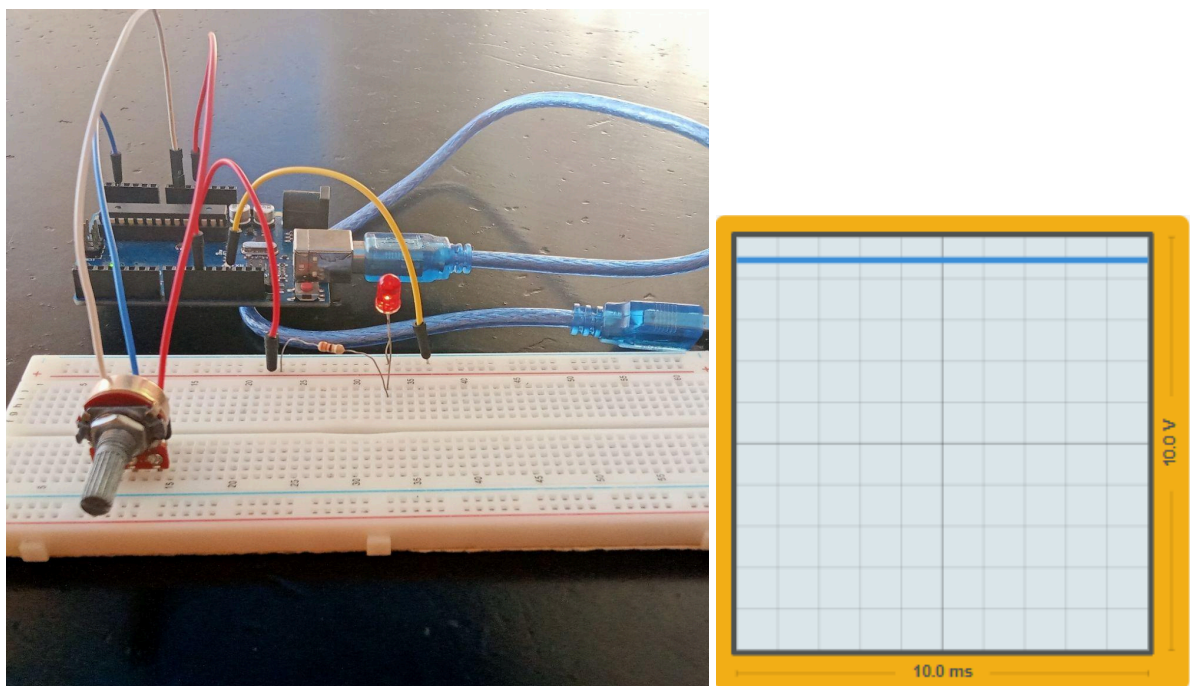
Controlando el brillo de un LED con una señal PWM



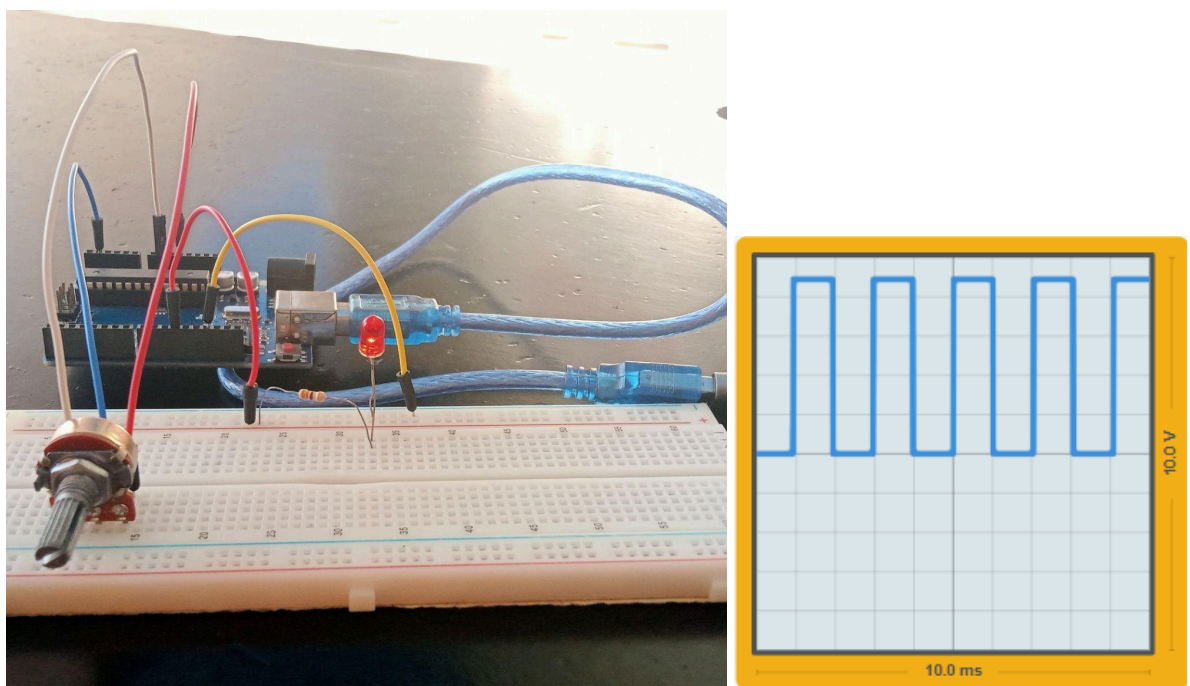
**Controlando el brillo de un LED con analogRead()
Potenciómetro completamente a la izquierda.**



Potenci3metro completamente a la derecha



Potenci3metro a la mitad:



CONCLUSIONES

En esta práctica, exploramos el comportamiento de las señales digitales y su modificación para diferentes aplicaciones. Observamos que estas señales surgen de la conversión de señales continuas, las cuales varían, y que para su procesamiento es esencial representarlas como datos constantes en forma de 0 y 1, o HIGH y LOW.

Durante las actividades realizadas, utilizamos el osciloscopio del simulador [Tinkercad](#) para visualizar la forma de las señales digitales. Comprendimos que estas se generan a partir de pulsos eléctricos que indican si el estado es HIGH o LOW. Se verificó el concepto de Duty Cycle al calcular porcentajes y al modificar el código en Arduino (con los ejemplos de Blink y Fade, en el Delay).

Cabe mencionar que no solo se puede modificar el tiempo de trabajo mediante el código, sino también de manera manual utilizando un potenciómetro, lo cual fue parte de la última actividad.

Con todo esto, se logra una mejor comprensión de la teoría; estas aplicaciones están presentes en nuestra vida cotidiana y podemos implementarlas en proyectos futuros. Al hacerlo, fomentamos el razonamiento y la formación profesional.