PARCIAL DISPRO 60%

Edward Duarte, Andrés Ramírez, Nicolas Pedraza, David Orozco
Ingeniería Electrónica, Departamento de Ingeniería Pontificia Universidad Javeriana, Bogotá D.C, Colombia
edwardduarte@javeriana.edu.co
camilo_ramirez@javeriana.edu.co
pedraza_n@javeriana.edu.co
davidorozco@javeriana.edu.co

Abstract— The objective to be carried out was to propose 2 functions, with their respective finite state diagram and code, which would allow us to generate and edit a PWM signal from any digital Pin.
The programming of this code was done using Arduino.

I. INTRODUCCIÓN

El objetivo a realizar fue plantear 2 funciones, con su respectivo diagrama de estados finitos y código, las cuales permitieran generar y editar una señal PWM desde cualquier Pin digital.
La programación de este código se realizó mediante Arduino.

II. DIRECTRICES PARA LA PREPARACIÓN DE LOS MANUSCRITOS

La elaboración de este parcial fue asesorada e instruida por el ingeniero y docente de la Pontificia Universidad Javeriana, Juan Carlos Giraldo, ingeniero electrónico, profesor de planta de la Pontificia Universidad Javeriana de Bogotá, del departamento de ingeniería electrónica, facultad de ingeniería

III. OBJETIVO GENERAL

Mediante Arduino realizar la codificación de una señale PWM en la cual se pueda editar su frecuencia y ciclo útil.

IV. OBJETIVOS ESPECÍFICOS

- Realizar el diagrama de estados finitos del Código.
- Comprender que es una señal PWM y como editar su ciclo útil.
- Implementación de las 2 funciones y el tipo de dato.

V. ALGUNOS ERRORES COMUNES

Al usar la función micros se debe tener en cuenta cuantos microsegundos tiene un segundo para así poder generar una conversión correcta y retornar el tiempo adecuado.

VI. CONTEXTO

El lenguaje ARDUINO permite la creación de distintos tipos de microordenadores, cada uno distinto del otro debido a la personalización y libertad de hardware que tiene, ya que cada usuario puede personalizar su propia placa para darle el uso que quiera.
A continuación, se presenta:

- Estado del arte
- Marco Teórico
- Diagrama de flujo(anexo).
- Esquematicos
- Descripción código.
- Funcionamiento notas en el código.
- Conexiones.
- Conclusiones participantes

VII. ESTADO DEL ARTE

El uso de modulación por ancho de banda de una señal o señal PWM es algo que como ingenieros va a ser de utilidad desde cosas pequeñas como encender led de distintos brillos hasta controlar la cantidad de energía que se envía a una carga. Por esta razón es importante informarse de la aplicación y funcionamiento de esto, así como en el artículo "Modulación por ancho de pulso (PWM) y modulación vectorial (SVM). Una introducción a las técnicas de modulación", publicado por Posada Contreras, Johnny, el 25/07/2005 nos habla de las técnicas, funcionamiento y uso de la señal PWM. [1]

VIII. MARCO TEÓRICO

ARDUINO: Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso [2].

Frecuencia: es el número de repeticiones por unidad de tiempo de cualquier evento periódico.[3]

Pin digital (entrada o salida digital): En Arduino las entradas y salidas digitales comparten pin, motivo por el que se denominan y/o digitales. Esto significa que el mismo pin puede ejecutar funciones tanto de entrada como de salida, aunque, lógicamente, no de forma simultánea. Es necesario configurar un pin y/o como entrada o salida en el código. [4].

Señal digital: es una variación de voltaje entre -Vcc a +Vcc sin pasar por los valores intermedios. Por lo tanto, una señal digital dispone solo de dos estados. Al valor inferior de tensión -Vcc le asociamos un valor lógico LOW o '0', mientras que al valor superior +Vcc le asociamos HIGH o '1' lógico. [4].

Microcontrolador: es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM) [5].

Pines PWM: (modulación por ancho o de pulso) es un tipo de señal de voltaje utilizada para enviar información o para modificar la cantidad de energía que se envía a una carga [6].

Pines analógicos: El arduino uno, posee dos salidas analógicas puras mediante dos conversores digital a analógico. Estos pines pueden usarse para crear salidas de audio usando la librería correspondiente. La función para hacer una salida PWM en un pin es: analogWrite() – escribe un valor analógico (onda PWM) al pin especificado.[7]

IX. ESQUEMATICOS

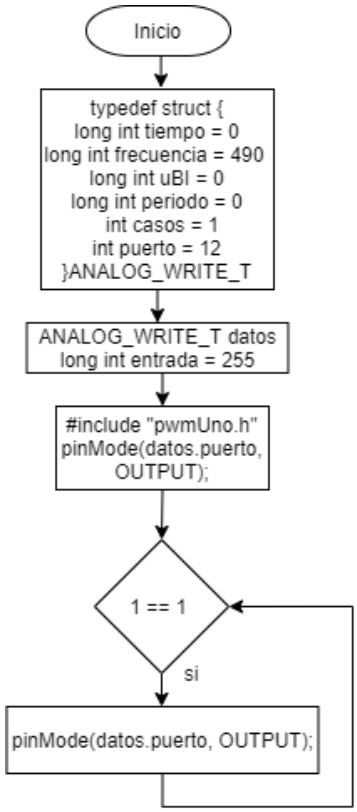


Fig. 1. Diagrama de flujo

El diagrama de flujo del sistema presenta las variables que se utilizaron en el main y los pines de entrada tanto como de salida en el sistema. Se puede observar detalladamente la lógica general del funcionamiento para generar la señal.

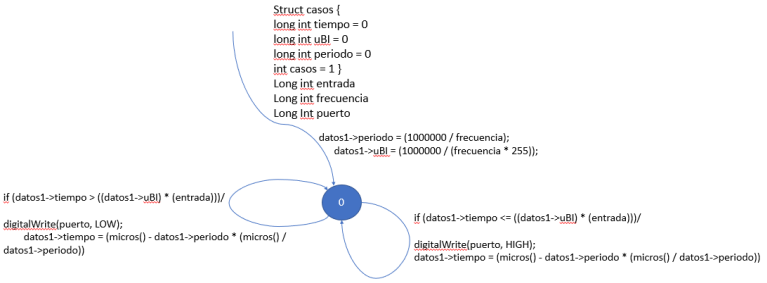
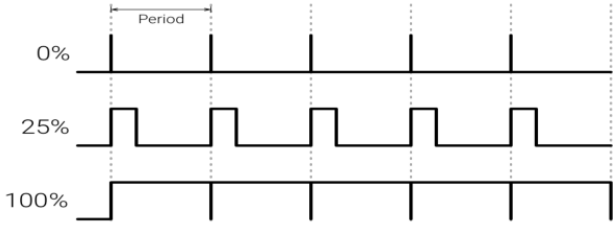


Fig. 2. Diagrama de estados finitos.

El diagrama de estados simplifica la realización de las funciones myanalogicwrite y myanalogicwritesetup en el cual se hace el proceso de cambio de frecuencia y ciclo útil de la señal, también pudiendo observar las variables temporales con su respectiva conversión a micros.

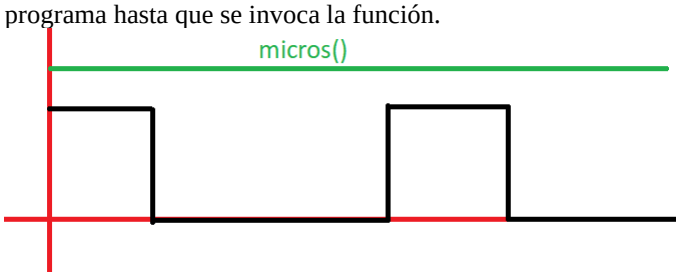
X. Problema y requerimiento

La temática principal del proyecto es generar una señal PWM en un puerto digital que no tiene las características.
Para ello es importante conocer que es una señal PWM, esta como se explicó en el marco teórico.
La señal PWM es una señal cuadrada que varía en su ciclo útil de acuerdo con un parámetro asignado desde cero a 255.



Imag 1. Diferentes Ciclos útiles señal PWM

Para generar la señal es importante considerar 2 factores, la primera es el periodo de la señal y la segunda el voltaje que se quiere seleccionar (con una equivalencia de 0 a 5 voltios con valores de 0 a 255).
Esta señal permite una simulación de una señal analógica a partir de generar pulso de cierta duración en un ciclo útil especifico, con ello es necesario considerar la forma de determinar el tiempo.
Para ello es importante considerar el tiempo en que transcurre el programa, a partir de esto se utiliza la función micros (), debido a que la función milis () a las escalas de frecuencia que se manejan no es capaz de abarcar todos los valores y generara un gran factor de error.
Para el manejo del tiempo es importante considerar que la función micros () toma el tiempo desde que se inició el programa hasta que se invoca la función.



Imag 2. Funcionamiento de micros ()

Este tiempo no es útil debido a que como es una señal periódica es necesaria la información desde donde se comenzó el inicio del periodo hasta donde se quiere observar, para ello se utiliza una variable externa, que almacenara la cantidad de ciclos previos que se han transcurrido desde que se inició el programa. Esto con el fin de realizar el producto con el periodo para determinar el tiempo de inicio de ese ciclo en específico.
Con ello se determina el tiempo dentro del periodo y se puede generar con facilidad la señal PWM.

XI. Descripción código

El código consiste en 2 funciones principales, las 2 tienen un funcionamiento similar por lo tanto solamente será una explicación.

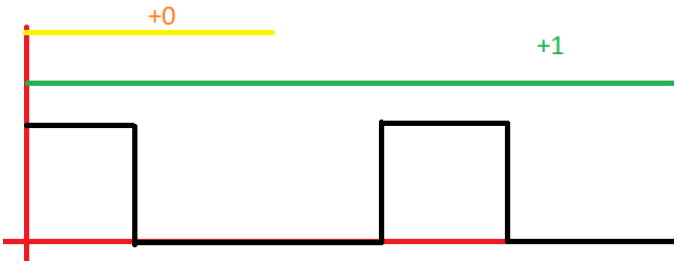
Para generar la señal PWM se tendrá en cuenta 2 variables, la frecuencia del ciclo útil, y la equivalencia del voltaje en un valor de 6 bits, en primera instancia se determinará el periodo y el mínimo valor en el tiempo que puede tomar la señal, para ello se puede determinar como

$$Periodo = \frac{1000000}{frecuencia}$$

Y para la unidad mínima de tiempo se divide el periodo de 255 partes, que son la cantidad de valores que puede tomar la señal PWM.

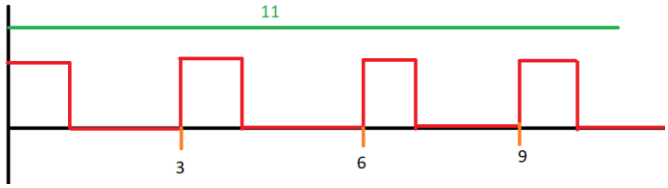
$$U_{minima} = \frac{Periodo}{255} = \frac{1000000}{frecuencia * 255}$$

Con estas variables se manejará la solución, como se mencionó en el problema es necesario tomar el tiempo dentro de un mismo periodo, para ello no se utilizará puramente la función micros () si no que se creara una variable llamada tiempo, que se reiniciará cada vez que se cumpla un periodo.
Para hacer esto se plantearon 2 soluciones, la primera es tener una variable contadora, que añada uno cuando detecte que el tiempo generado por la función micros () sobrepaso el valor del periodo.



Imag 3. Contador de ciclos

Este planteamiento generara una solución teóricamente factible, pero en la ejecución generaron ciertas fallas, la principal fue que, si entre las tomas de los tiempos la señal recorre más de un periodo, el contador solamente tomara que se pasó un único ciclo generando que falle la variable tiempo.
para solucionar este problema, se generó una segunda solución donde la variable contadora no depende de sumar en el valor cuando determina que se cumplió un periodo, sino que se determina mediante las propiedades del cociente y de los números enteros.
Para esta solución toca considerar que el cociente de dos enteros da otro entero.



Imag 4. Señal completa con contador

Cuando se realiza el cociente del periodo con el tiempo total que ha transcurrido, debido a que no tiene valores decimales este dará la cantidad de ciclos que han transcurrido.

$$\frac{11}{3} = 3.66 \approx 3$$

Por lo tanto, el cociente con números enteros es equivalente a la variable contador, además esta tiene ventajas con respecto a

la otra debido a que no toca crear un caso especial para sumarla, esta siempre se puede ejecutar y no modificara el valor.
Con respecto a ello en primera instancia se tiene una condición, si el valor del tiempo es menor al producto de la unidad minina con la equivalencia de voltaje, entonces activara el pin de salida, si no se cumple desactivara el pin, y en cada condición se redefine el tiempo como el sustraendo de la función micros con el producto del periodo por la razón de la función micros con el periodo

$$Tiempo = micros - periodo * \left(\frac{micros}{periodo}\right)$$

Considerando que todas las variables deben estar en un formato de Long int para que el código pueda funcionar.
MyAnalogWrite
Esta función simula el caso de un Arduino uno donde hay una Frecuencia fija de 490Hz en los pines PWM, por ello no es necesario ingresar el valor de frecuencia, únicamente se ingresan los valores del pin que se desea transformar en PWM y la equivalencia de voltaje de 6 bits que se desea ingresar.
MyAnalogWriteSetUp
Esta función simula un pin PWM pero que puede tener una frecuencia indefinida, por ello se ingresan 3 valores, el pin que se desea transformar en PWM, la equivalencia de voltaje de 6 pines y la frecuencia que se desea que se repita el ciclo útil.

XII. Resultados

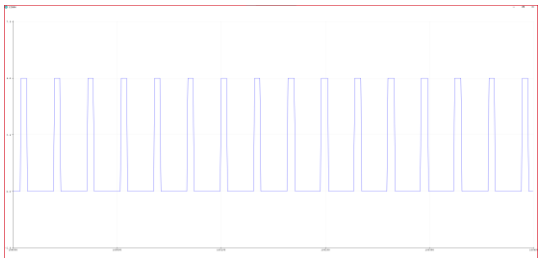
Con el fin de tener resultados palpables en este proyecto se decide realizar una comparativa entre las funciones dadas por Arduino con los puertos PWM, con respecto a la creada. De allí, se realizaron varias pruebas en las que por medio del mismo valor de luminiscencia (de 0 a 255)

Imag 5. Montaje realizando las pruebas de los resultados.

Con esto, se encontró como el ojo humano, no fue capaz de percibir diferencia entre los resultados en el led para cada uno del código. Ello, corrobora de manera directa un correcto funcionamiento en el código y la lógica detrás de este.

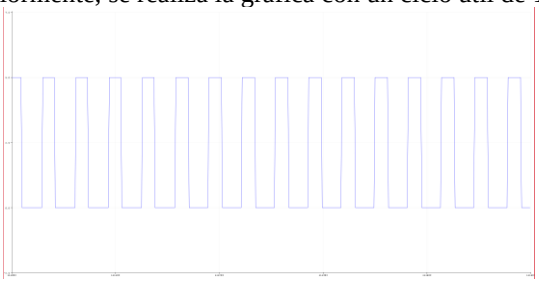
Adicionalmente, se decidió revisar de manera directa si el efecto de menor luminiscencia en el led se causaba por una intermitencia del voltaje como era de esperarse con lógica de PWM.
Allí, se tomó un video inicial a velocidad normal, en donde se observa el led encendido siempre. Sin embargo, al utilizar una frecuencia alta en la toma de muestras se puede apreciar como el led está encendiéndose y apagándose rápidamente para lograr un efecto de menor brillo.

Con esto en cuenta, y sabiendo el buen funcionamiento del código se procedió a realizar las graficas del voltaje del led a lo largo del tiempo para diferentes ciclos útiles (50,100,150,200). A continuación, se muestra la grafica del voltaje con un ciclo útil de 50.



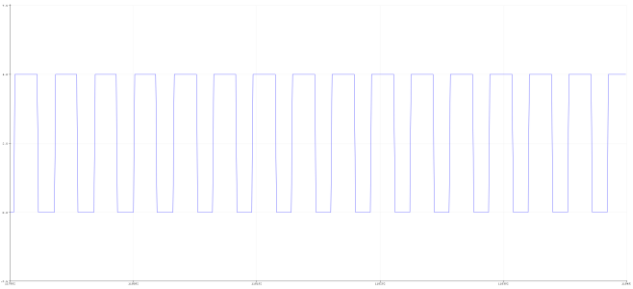
Imag 6. Señal con una frecuencia de 490 a 50 ciclos.

Posteriormente, se realiza la gráfica con un ciclo útil de 100



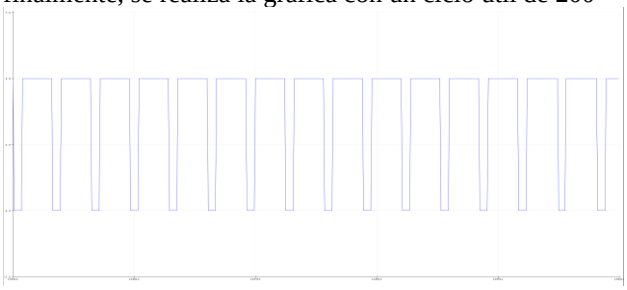
Imag 7. Señal con una frecuencia de 490 a 100 ciclos.

Luego de ello, se realiza la misma grafica con un ciclo útil de 150



Imag 8. Señal con una frecuencia de 490 a 150 ciclos.

Y finalmente, se realiza la gráfica con un ciclo útil de 200



Imag 9. Señal con una frecuencia de 490 a 200 ciclos.

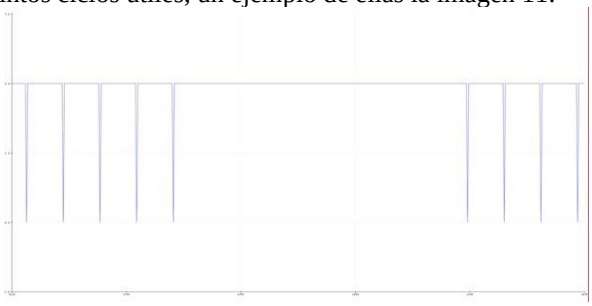
Con todo esto, se obtiene un comportamiento esperado. Dicho comportamiento consiste en un aumento constante en el tiempo de los 5 voltios por periodo. Allí, teóricamente un ciclo útil mayor debe dar como resultado un mayor tiempo en 5 voltios. Lo cual, concuerda con el resultado.

XIII. ANÁLISIS

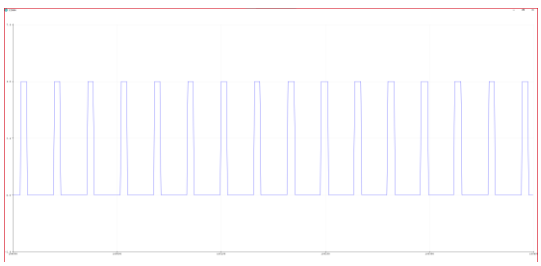
Para este parcial, para realizar el análisis de una señal PWM es importante conocer de antemano que se puede editar su ciclo útil y frecuencia, y de esta manera es pertinente afrontar adecuadamente ambos elementos de la señal.

Al momento de codificar se puede observar que dicha frecuencia mencionada presentaba un límite. De manera, que el código funciona con frecuencias menores a los 4 kHz, esto debido a que, al momento de calcular el tiempo, se está realizando una división por enteros, de manera que el resultado que va a devolver va a ser un entero y las frecuencias superiores a 4khz nos retornan un valor de 0, provocando así un error en el tiempo. Al momento de tomar este valor los datos que la involucran tienden a 0, produciendo así un error en el código y fallando pues reflejan datos que no los reales. También se debe apreciar, que al usar la función micros (), el tiempo que se va a trabajar debe ser llevado a dicho tiempo, es por esto que el uso del factor 1000000 es vital a la hora de obtener los tiempos, ya que sin él no se daría la conversión del tiempo y el dato que se devolvería estaría erróneo en unidades.

Una vez llevado a cabo todo este proceso se llevó a cabo la realización de las gráficas de las señales generadas, observando que la función Serial.Println generaba errores en las señales ya que las modificaba un poco como se observa en la imagen 10, luego de tratar con este error se realizaron varias graficas con distintos ciclos útiles, un ejemplo de ellas la imagen 11.



Imag 10. Señal alterada por serial.println



Imag 11. Señal con una frecuencia de 490 a 50 ciclos

XIV. CONCLUSIÓN

- ✓ En frecuencias superiores a 4khz el programa comienza a fallar debido a la función micros.
- ✓ En el proceso del contador la división que se realiza en vez de la suma, causa que gaste un poco más procesamiento, pero nos permite simplificar más el proceso, ya que no necesitamos crear una nueva variable que almacene este.
- ✓ La razón por la cual en ciertas frecuencias el titileo se aprecia con mayor claridad es debido a que como no utilizamos un contador si no una variable integer, este aproxima el dato generando un espacio en tiempo que hace que se observe aparentemente un titileo.

XV. CÓDIGO DEL PROYECTO

Este proyecto es de código abierto y cualquier persona puede acceder a él. A continuación, se adjunta el link que contiene la totalidad de códigos.

https://livejaverianaedu-my.sharepoint.com/:u:/g/personal/edwardduarte_javeriana_edu_co/Ea_EKi07e7lGgWbOAleQgMQBrehDjbw9QLusDzDzD73kdg?e=ZOiJwR

XVI. REFERENCIAS

[1] I. “Modulación por ancho de pulso (PWM) y modulación vectorial (SVM). Una introducción a las técnicas de modulación””, Posada Contreras, Johnny, 2021. [Online]. Available: https://www.redalyc.org/pdf/478/47802507.pdf [Accessed: 06- Jun- 2021].

[2] "¿Que es la programacion Arduino y para que sirve?, BeJOB, 2021 [Online]. Available: https://www.bejob.com/que-es-la-programacion-con-arduino-y-para-que-sirve/ [Accessed: 06- Jun- 2021].

[3] "¿Que es Frecuencia? ", significados, 2021. [Online]. Available: https://www.significados.com/frecuencia/ [Accessed: 06- Jun- 2021].

[4] "ENTRADAS DIGITALES EN ARDUINO”, ingeniería, informática y diseño, 2021. [Online]. Available: https://www.luisllamas.es/entradas-digitales-en-arduino/ / [Accessed: 06- Jun- 2021].

[5] “Microcontrolador ¿Qué es y para que sirve?”,Hetpro, 2021 [Online].Available:https://hetpro-store.com/TUTORIALES/microcontrolador/ [Accessed: 28- May- 2021].

[6] "Salidas PWM Arduino", *Programo Ergo sum* 2021. [Online]. Available: https://www.programoergosum.com/cursos-online/arduino/255-salidas-analogicas-pwm-con-arduino/salidas-analogicas-pwm#:~:text=Salidas%20PWM%20en%20Arduino,se%20env%C3%ADa%20a%20una%20carga. [Accessed: 06- Jun- 2021].

[7] "Estructuras Arduino", Aprendiendo Arduino ,2021. [Online]. Available: https://aprendiendoarduino.wordpress.com/tag/entradas-analogicas/ [Accessed: 06- Jun- 2021].