



## EVALUACIÓN POR SIMULACIÓN EN PROTEUS DE SALIDAS PWM DE ARDUINO UNO

Niurka Monteavaro García<sup>1</sup>, Pablo Montejo Valdés<sup>2</sup>

<sup>1</sup>CUJAE, Cuba, calle 114 e/ Rotonda y Ciclovía, Marianao, <sup>2</sup>CUJAE, Cuba

<sup>1</sup>e-mail: niurka@cime.cujae.edu.cu

### RESUMEN

La Modulación por Ancho de Pulso (PWM) es muy utilizada en las telecomunicaciones. En la actualidad gran parte de estos sistemas se basa en el empleo de microprocesadores. Sin embargo, no todos ellos son capaces de dar salidas moduladas por ancho de pulso. El microprocesador Atmega de la plataforma Arduino de código y hardware abiertos, es capaz de entregar salidas moduladas por ancho de pulso, por lo que esta plataforma es ampliamente utilizada. El objetivo de este trabajo es valorar las salidas PWM de Arduino UNO, a través de la simulación, con dos modelos de diferentes bibliotecas para Proteus: ARDUINO REV3 y ARDUINO TEP. La valoración de los resultados de simulación obtenidos con los dos modelos empleados se basa en su comparación con los resultados de la medición experimental. La comparación permite concluir que las bibliotecas de código abierto no siempre son fiables, por lo que se recomienda realizar pruebas de comprobación antes de confiar plenamente en los resultados obtenidos por simulación.

**PALABRAS CLAVES:** Arduino, PWM, Proteus.

## ASSESSMENT BY SIMULATION IN PROTEUS OF PWM OUTPUTS OF ARDUINO UNO

### ABSTRACT

Pulse width modulation (PWM) is widely used in telecommunications. At present, a large part of these systems is based on the use of microprocessors. However, not all of them are capable of giving pulse width modulated outputs. The Atmega microprocessor of the Arduino open source and hardware platform is capable of delivering pulse width modulated outputs, so this platform is widely used. The objective of this work is to evaluate the PWM outputs of Arduino UNO, through simulation, with two models from different libraries for proteus: ARDUINO REV3 and ARDUINO TEP. The assessment of the simulation results obtained with the two models used is based on their comparison with the results of the experimental measurement. The comparison allows to conclude that open source libraries are not always reliable, so it is recommended to carry out verification tests before relying fully on the results obtained by simulation.

**KEY WORDS:** Arduino, PWM, Proteus.

### 1. INTRODUCCIÓN

Según A. Bruce Carlson [1], “En 1838 Morse mandó el primer mensaje de telégrafo a una distancia de 16 km. Este hecho marcó el nacimiento de una nueva era: la de las comunicaciones eléctricas. Ahora, más de un siglo y medio después, la ingeniería de la comunicación ha avanzado hasta el punto de que los televidentes terrestres ven a los astronautas trabajando en el espacio.”

Un sistema de comunicación transmite información desde su fuente hacia un destino distante. Existen muchas aplicaciones diferentes para los sistemas de comunicación y muchos diseños diferentes para lograrlos [1]. Los tipos

## Evaluación por simulación en Proteus de Salidas PWM de Arduino Uno

de modulación empleados y los procesadores encargados de su implementación influyen en los resultados que se obtienen para un diseño concebido.

La modulación por ancho de pulso (PWM por sus siglas en inglés de Pulse Width Modulation) se ha vuelto cada vez más importante en las comunicaciones modernas por su relativa sencillez y robustez.

Actualmente se utiliza en el control de monitores LEDs de alta velocidad [2], en convertidores de bus aislado [3], en fuentes de voltaje inversoras[4] y en moduladores y demoduladores de señal [5]. También se trabaja en obtener formas más eficientes para lograr la modulación por ancho de pulso [6]. Varios procesadores tienen salidas que implementan PWM, descritas la mayoría de las veces como salidas analógicas. Entre estos procesadores están los de la familia ATMEGA, utilizados en las plataformas Arduino.

Varios procesadores tienen salidas que implementan PWM, descritas la mayoría de las veces como salidas analógicas. Entre estos procesadores están los de la familia ATMEGA, utilizados en las plataformas Arduino.

Arduino es una plataforma electrónica de código y hardware abiertos, flexible y fácil de usar. Fue concebido para artistas, diseñadores, como hobby y para cualquier interesado en crear objetos o entornos interactivos [7].

En el sitio oficial de desarrollo de Arduino es posible encontrar un cúmulo de herramientas software, las cuales posibilitan la programación sencilla y rápida de los diferentes modelos que conforman la familia.

Entre las herramientas disponibles en la web se encuentran las bibliotecas para la simulación de aplicaciones desarrolladas con Arduino. La simulación es una herramienta poderosa y necesaria en el diseño de circuitos inteligentes. Las bibliotecas de código abierto contribuyen también a este fin. Uno de los inconvenientes de la utilización de código abierto radica en que la mayoría no se compromete con la ausencia de errores.

En este trabajo se utilizan dos modelos para la simulación de dispositivos Arduino UNO, pertenecientes a diferentes bibliotecas de libre distribución: ARDUINO REV3 y ARDUINO TEP. Con los mismos se realizan diferentes pruebas para obtener y analizar la respuesta de las salidas PWM.

## 2. MODULACIÓN POR ANCHO DE PULSO

La modulación por ancho de pulso de una señal o fuente de energía es una técnica en la cual se modifica el ciclo de trabajo de una señal periódica. Típicamente se emplea sobre señales sinusoidales o cuadradas. Tiene dos usos ampliamente difundidos: para transmitir información a través de un canal de comunicaciones o para controlar la energía que se envía a una determinada carga [8].

Para transmitir información con PWM el valor de la señal analógica se utiliza para determinar el ancho del pulso de la señal digital, variando el ancho del pulso de forma directamente proporcional al valor de la señal analógica, manteniendo una amplitud constante [8]–[10].

La modulación por duración o ancho de pulsos (PWM o PDM) es uno de los esquemas más utilizados en la actualidad en transmisores de amplitud modulada (AM) y permite conseguir eficiencias superiores a las que se tienen con AM tradicional [11]. En cierta forma puede considerarse como una modulación de alto nivel.

En la Figura 1 se muestra el resultado de la modulación PWM por una señal sinusoidal. Se obtiene un tren de pulsos de duración variable. En la imagen se ha superpuesto la señal de información únicamente con el fin de observar que la duración de los pulsos aumenta en los semiciclos positivos de aquella y se reduce a pulsos más estrechos en los semiciclos negativos [11].

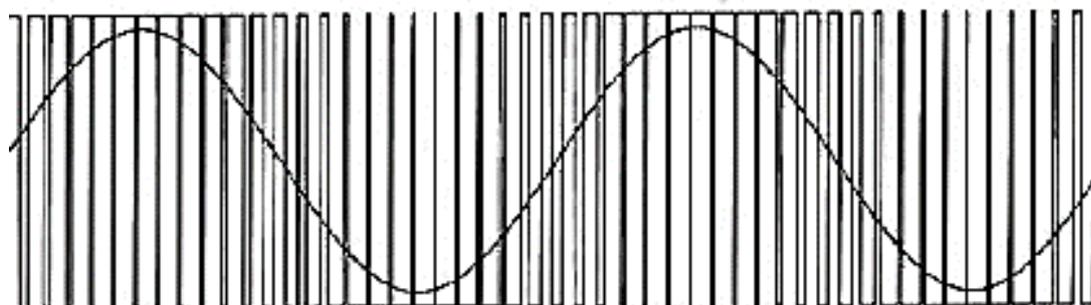


Figura 1: Señal PWM para comparación con la señal de información

Una de las aplicaciones frecuentes de modulación PWM en los sistemas de comunicación es mejorar la eficiencia en la transmisión de la información, reducir el ruido y las interferencias, o lograr la coexistencia de varios canales de comunicaciones sobre un mismo medio de transmisión [1].

Otro empleo de las salidas PWM se vincula con la necesidad de controlar la energía que se suministra a una carga. En estos casos nos encontramos con el inconveniente de que la mayoría de los microcontroladores no son capaces de proporcionar una verdadera salida analógica. Las salidas consisten en una onda cuadrada que varía entre  $-V_{cc}$  y  $V_{cc}$ . (por ejemplo, 0V y 5V). Para salvar esta limitación y simular una salida analógica frecuentemente se utiliza el PWM. Estas aplicaciones están orientadas fundamentalmente al control de fuentes conmutadas, velocidad de motores, posición de un servomotor, elementos termoelectrónicos, interruptores electrónicos (usados para interrumpir una señal bajo el control de otra), convertidores D/A, entre otros.

Con independencia de la aplicación final (sea un sistema de comunicaciones o la implementación de una salida analógica), en una salida PWM se cumple que el promedio de la tensión, a lo largo del tiempo, será igual al valor analógico deseado. Este se calcula como el producto de la tensión máxima y el ciclo útil (DC por sus siglas en inglés de Duty Cycle), según la siguiente expresión:

$$V_{ab} = V_G - V_C \frac{R}{10} \quad (1)$$

El ciclo útil de una señal periódica se define a través del cociente entre el ancho relativo de su parte positiva (W) y el período de la señal (T) expresado en por ciento:

$$DC = \frac{W}{T} 100 \quad (2)$$

### 3. PLATAFORMA ARDUINO UNO

Existen muchos microcontroladores y plataformas de desarrollo disponibles, con las que se pueden obtener salidas moduladas por ancho de pulso. Algunas de las ventajas que ofrece Arduino para profesores, estudiantes y aficionados sobre otros sistemas se mencionan a continuación [7]:

- Bajo Costo
- Multiplataforma

## Evaluación por simulación en Proteus de Salidas PWM de Arduino Uno

- Entorno de programación simple y claro
- Código abierto y software extensible
- Hardware abierto y extensible

El procesador de la placa Arduino UNO R3 es el Atmega328, de 8 bits con 28 pines. Este microcontrolador está provisto de una amplia variedad de prestaciones, que se pueden dividir en sistemas de memoria y de puertos, temporizadores, sistemas de interrupción, conversor análogo – digital y comunicación serie [12].

Arduino UNO cuenta con 6 salidas PWM garantizadas por el hardware de su microcontrolador ATmega328. Estas son muy fáciles de configurar gracias a las herramientas disponibles para el trabajo con Arduino.

Durante el desarrollo de este trabajo se buscaron bibliotecas para la simulación de las salidas PWM en Arduino UNO. En la red fue posible acceder a diferentes bibliotecas con modelos para la simulación de los dispositivos Arduino [13], [14]. Sin embargo, al simular las salidas PWM, con los dos modelos se obtuvieron resultados diferentes.

### 4. IMPLEMENTACIÓN DE LAS SALIDAS PWM DESDE ARDUINO UNO

La programación de los módulos Arduino se realiza desde el software de libre distribución Arduino IDE disponible en el sitio oficial de estos dispositivos [14]. Para ello se emplea un grupo de funciones desarrolladas en C++ que permite de forma muy sencilla acceder a los diferentes recursos de los módulos.

La función *analogWrite()* escribe un valor analógico en un terminal PWM, generando una onda cuadrada de frecuencia constante y ciclo útil especificado. En el Arduino UNO las salidas PWM son los pines 3, 5, 6, 9, 10 y 11. En los pines 5 y 6 la frecuencia es de 980 Hz mientras que en el resto es de 490 Hz [14].

Con la función *analogWrite()* se configura qué valor analógico de tensión se desea establecer en la salida seleccionada, a partir del ciclo útil determinado. Su sintaxis se muestra a continuación:

```
analogWrite(pin, ciclo útil)
```

donde *pin* es el terminal por el que se desea la salida PWM y *ciclo útil* es un valor declarado entre 0 y 255 (8 bits), proporcional al ciclo útil de la salida PWM. Si dicho valor es 0, la señal de salida se mantendrá siempre en cero lógico (0%), de ser 255, siempre se obtendrá un uno lógico (100%).

### 5. SIMULACIÓN Y COMPARACIÓN DE LOS MODELOS

El simulador utilizado es el Proteus versión 7.8 sobre el sistema operativo Windows 7 Professional de 32 bit. Los modelos de los dispositivos Arduino UNO bajo análisis fueron los pertenecientes a la biblioteca ARDUINO REV3 (Figura 1), descargada del sitio [www.microcontrolandos.blogspot.com](http://www.microcontrolandos.blogspot.com) y a la biblioteca ARDUINO TEP (Figura 2), descargada del sitio [www.theengineeringprojects.com](http://www.theengineeringprojects.com).

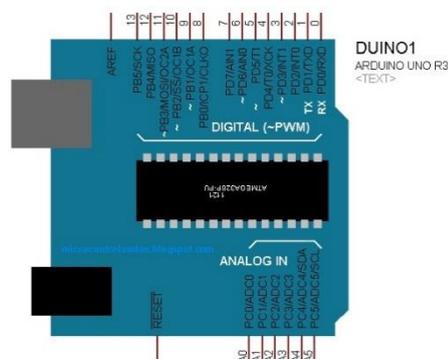


Figura 1: Representación visual del modelo de Arduino UNO de la biblioteca ARDUINO REV3

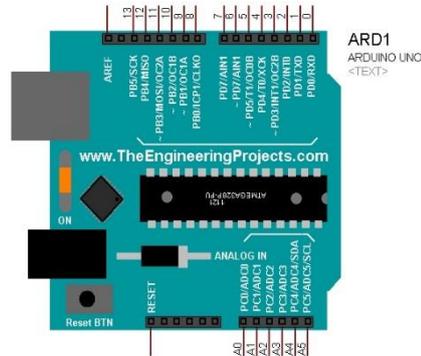


Figura 2: Representación visual del modelo de Arduino UNO de la biblioteca ARDUINO TEP

### Ejemplo de utilización de la función `analogWrite` para generar una señal PWM.

Con el objetivo de comprobar el uso de la función `analogWrite`, se implementó un diseño donde se obtiene un valor analógico de salida por el pin 9. Dado que es posible obtener cualquier valor analógico entre 0 y 5 V, se escogió el valor de 2 V para hacer la prueba.

Para lograr una salida analógica de 2 V a partir de una alimentación de 5 V, se necesita una salida digital con un ciclo útil de 40%, que se obtiene de la ecuación (1) en el epígrafe “Modulación por Ancho de Pulso”. Teniendo en cuenta que el valor del ciclo útil varía entre 0 y 255, al 40% le corresponde el valor 102. Como salida para la prueba se escogió el pin 9, estableciendo los parámetros de la función `analogWrite`, como se muestra en la Figura 3.

```
void setup() {  
}  
  
void loop() {  
  analogWrite(9, 102);  
  delay(10);  
}
```

Figura 3: Código para un DC = 40 %

### Simulación

La simulación, utilizando los modelos de las bibliotecas ARDUINO REV3 y ARDUINO TEP, dio el mismo resultado para ambos (Figura 4). Se obtuvo una onda rectangular con un período de 2.04 ms (frecuencia de 490 Hz) y un tiempo en uno de 817  $\mu$ s, lo que da un ciclo útil de 40% según la ecuación (1). Con este valor se obtiene un voltaje promedio de 2 V (ecuación 2).

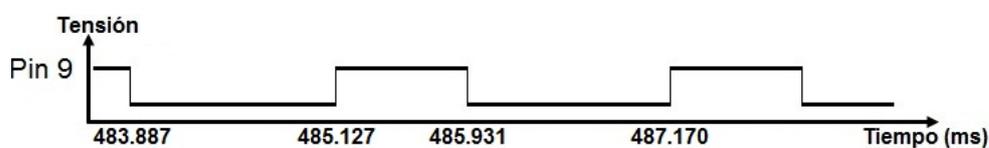


Figura 4: Simulación de la salida 9 con un ciclo útil 40%

**Implementación de PWM en todas las salidas analógicas de forma simultánea.**

Para validar los resultados obtenidos anteriormente, esta vez en todas las salidas PWM, se programó un código capaz de barrer, en intervalos regulares de 3 ms, todos los valores posibles de ciclo útil.

Con el código de la Figura 5 se obtiene una señal con un ciclo útil variable desde 0 (0%) hasta 255 (100%), incrementándose en 1 ( $\approx 0.39\%$ ) cada 3 ms, esto ocurre para cada una de las 6 salidas PWM.

```
void setup() {
}

void loop() {
  for(int dc=0; dc<256; dc=dc+1){
    analogWrite(3, dc);
    analogWrite(5, dc);
    analogWrite(6, dc);
    analogWrite(9, dc);
    analogWrite(10, dc);
    analogWrite(11, dc);
    delay(3);}
}
```

Figura 5: PWM por las 6 salidas con ciclo útil variable

Con el objetivo de lograr lo anterior se utiliza la misma función *analogWrite* declarando una variable, llamada dc, que tendrá el valor del ciclo útil que se va a imponer en todos los pines. El valor de la variable dc cambiará cada 3 ms, incrementando su valor en 1.

**Simulación**

Cuando se simuló con la biblioteca ARDUINO TEP todas las salidas respondieron según lo previsto al escribir el código analizado anteriormente (Figura 5). En los pines 5 y 6 se obtuvo una señal con un período de 1,02 ms (frecuencia de 980 Hz). En el resto de las salidas PWM el período obtenido fue de 2,04 ms (frecuencia de 490 Hz).

En la Figura 6 se muestran seis intervalos que se corresponden con los valores de 0%, 20%, 40%, 60%, 80% y 100% para el ciclo útil. Los valores marcados en el eje temporal se corresponden con los momentos de transición de la señal. Se aprecia claramente como el ciclo útil de la señal varía de un intervalo al otro. Igualmente se aprecia que la frecuencia de las salidas 5 y 6 es el doble del resto.

El tiempo en “1” varió desde 8  $\mu$ s hasta más de 2.04 ms (al alcanzar el valor 255), dando una variación del ciclo útil desde 0.39 % hasta 100 %. El intervalo de tiempo antes del primer pulso es mayor que el período de la señal, confirmando un ciclo útil de 0%.

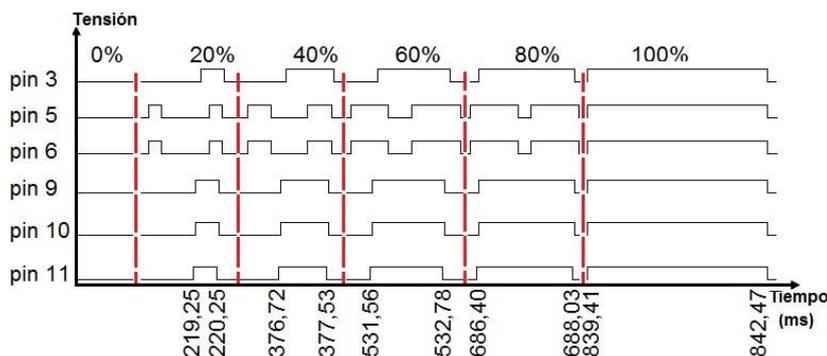


Figura 6: Simulación de PWM por las seis salidas de ARDUINO UNO con la Biblioteca ARDUINO TEP

Cuando se realizó la simulación con el modelo de la biblioteca ARDUINO REV3 en las salidas 9, 10 y 11 se obtuvieron salidas con una frecuencia y ciclo útil variable iguales a las obtenidas anteriormente.

En las salidas 3, 5 y 6 se obtuvo un “0” para todos los valores de dc, excepto 255, para el cual apareció un pulso en “1” hasta se reinició el valor de dc.

En la Figura 7 se muestra parte del barrido de ciclo útil descrito anteriormente. Fueron seleccionados solo algunos valores. En la misma se aprecian nuevamente los intervalos que se corresponden con los valores de 0%, 20%, 40%, 60%, 80% y 100% para el ciclo útil. Los valores marcados en el eje temporal se corresponden con los momentos de transición de la señal.

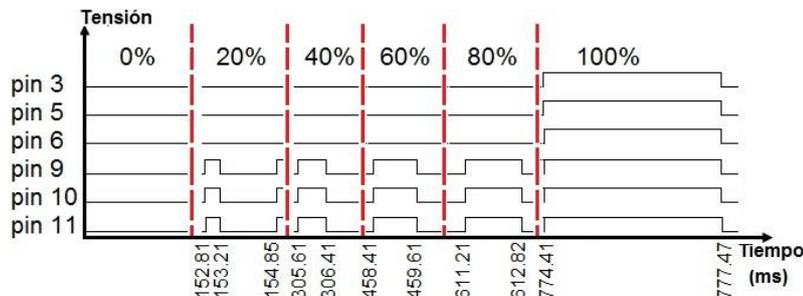


Figura 7: Simulación de PWM por las seis salidas de ARDUINO UNO con la Biblioteca ARDUINO REV3

## 6. MEDICIONES EXPERIMENTALES

Debido a las contradicciones encontradas entre los resultados de la simulación, se realizaron experimentales con la plataforma Arduino UNO en los laboratorios docentes del Centro de Investigaciones de Microelectrónica (CIME).

Las condiciones para las mediciones experimentales se prepararon considerando dos aspectos fundamentales:

- poder contrastarse con los resultados de la simulación,
- adecuar las condiciones experimentales a un equipamiento que no puede reproducir exactamente las mismas condiciones de la simulación.

Por las razones antes expuestas, para las mediciones experimentales en lugar de variar en ciclo útil cada 3 ms, se programó un ciclo útil fijo con valores diferentes para cada salida, como se muestra en la Figura 9. Se emplearon valores tomados distribuidos por todo el rango posible. Para la selección de los valores del ciclo útil se tuvo en cuenta que permitieran una mejor visualización de los pulsos que se esperaban obtener en la pantalla del osciloscopio.

## Evaluación por simulación en Proteus de Salidas PWM de Arduino Uno

```

void setup() {
}

void loop() {
  analogWrite(3, 25) ;
  analogWrite(5, 50) ;
  analogWrite(6, 75) ;
  analogWrite(9, 100) ;
  analogWrite(10, 150);
  analogWrite(11, 200);
  delay(10);}

```

Figura 8: Código utilizado para las mediciones experimentales

Para realizar las mediciones se utilizó un osciloscopio digital Rigol. Los resultados se muestran en la Figura 9.

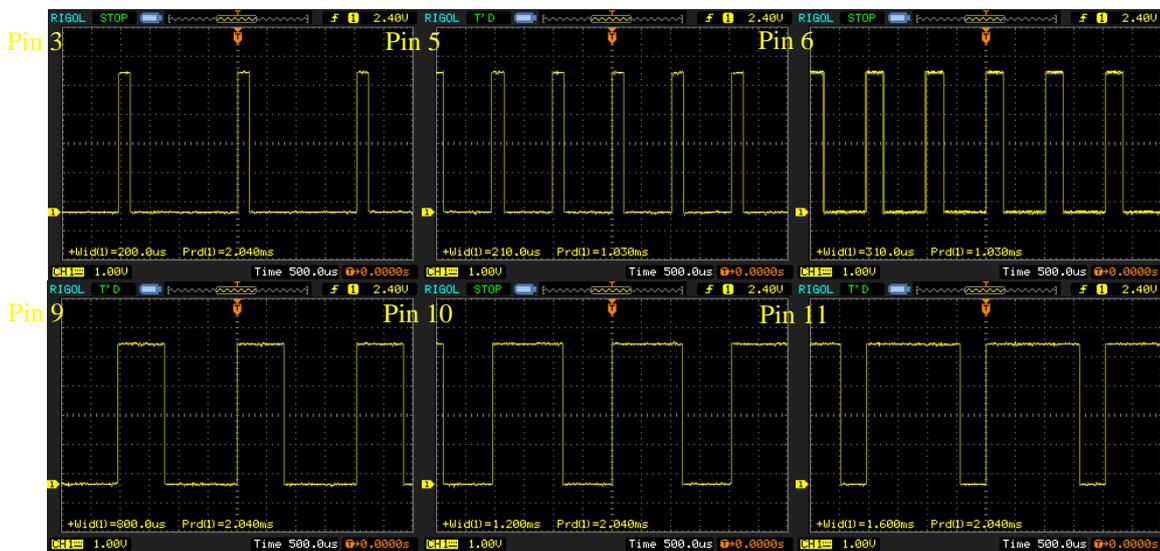


Figura 9: Resultados experimentales

Como puede observarse se obtuvieron las 6 salidas moduladas en ancho de pulso, con un ciclo útil diferente en cada una de ellas. En las salidas 3, 9, 10 y 11 se tienen señales a 490.2 Hz, mientras que en las salidas 5 y 6 la frecuencia es de 970.87 Hz. Los valores de los ciclos útiles obtenidos, comparados con los esperados, así como el error en cada caso se resumen en la Tabla 1.

Tabla 1: Resumen de las mediciones realizadas

Terminal de salida	Ciclo útil esperado	Ciclo útil obtenido	% error
3	9.8039%	9.8039%	0.0000%
5	19.6078%	20.3883%	3.8282%
6	29.4118%	30.0971%	2.2770%
9	39.2157%	39.2157%	0.0000%
10	58.8235%	58.8235%	0.0000%
11	78.4314%	78.4314%	0.0000%

El bajo error obtenido evidencia la correspondencia de los resultados prácticos con lo descrito por el fabricante sobre las salidas PWM en Arduino UNO. Este resultado contradice nuevamente los valores obtenidos durante las simulaciones empleando la biblioteca ARDUINO REV3, mientras confirma la concordancia con el comportamiento del modelo brindado por ARDUINO TEP.

## 7. CONCLUSIONES

En este trabajo se desarrollaron dos ejemplos de simulación para las salidas PWM de la plataforma Arduino UNO utilizando las bibliotecas ARDUINO REV3 y ARDUINO TEP, así como la comprobación experimental de estas salidas.

La comparación de los tres resultados permite concluir que los resultados experimentales y la simulación del comportamiento de las salidas PWM con la biblioteca ARDUINO TEM coinciden con la descripción del fabricante. Los resultados de simulación con la biblioteca ARDUINO REV3 están en contradicción tanto con los resultados experimentales, como con la descripción del fabricante en el comportamiento de los pines de salida PWM 3, 5 y 6.

Estos resultados evidencian que se hace necesario realizar pruebas de comprobación a los modelos empleados antes de confiar plenamente en los resultados obtenidos, ya que no todos los autores de software libre se comprometen con la fiabilidad de sus productos. Esto hace necesario introducir en el flujo de diseño un paso adicional para corroborar la calidad de la herramienta de simulación empleada.

Se recomienda darle continuidad a este trabajo, caracterizando la respuesta de estas bibliotecas ante otras prestaciones que brindan las plataformas Arduino. Los resultados de este trabajo no invalidan de ninguna manera la utilización de la biblioteca ARDUINO REV3 con otros modelos o aplicaciones, solo se previene sobre el error en algunas de las salidas analógicas utilizadas como PWM.

## RECONOCIMIENTOS

Los autores desean agradecer a todos los que de una forma u otra han ayudado a la realización de este artículo.

## REFERENCIAS

- [1] A. B. Carlson, *Communication System: An introduction to signals and noise in electrical communication*, 4a ed. New York: McGraw-Hill Companies, 2002.
- [2] S.-C. Hsia, Y.-C. Lin, W.-K. Wong, y R.-C. Hsu, «Asynchronous control and driver for high-speed LED display with local scanning approach», *IET Circuits, Devices & Systems*, vol. 14, n.º 1, pp. 1-6, jul. 2019.
- [3] M. H. Ahmed, A. Nabih, F. C. Lee, y Q. Li, «Low-Loss Integrated Inductor and Transformer Structure and Application in Regulated LLC Converter for 48-V Bus Converter», *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 8, n.º 1, pp. 589-600, mar. 2020.
- [4] M. Vujacic, O. Dordevic, y G. Grandi, «Evaluation of DC-Link Voltage Switching Ripple in Multiphase PWM Voltage Source Inverters», *IEEE Transactions on Power Electronics*, vol. 35, n.º 4, pp. 3478-3490, abr. 2020.
- [5] N. M. Safyannikov y O. Bureneva, «Time-to-Voltage Converters Based on the Time-Sharing Principle», *IEEE Access*, vol. 8, pp. 17442-17453, 2020.
- [6] P. Szczepankowski, T. Bajdecki, y R. Strzelecki, «Direct Modulation for Conventional Matrix Converters Using Analytical Signals and Barycentric Coordinates», *IEEE Access*, vol. 8, pp. 22592-22616, 2020.
- [7] «Arduino - Introduction», 15-sep-2017. [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>. [Accedido: 15-sep-2017].
- [8] J. A. Cortés, J. A. Chaves, y E. A. Quintero, «GENERACIÓN DE SEÑALES SENOIDALES MEDIANTE PWM Y FILTROS ACTIVOS DE SEGUNDO ORDEN», *Scientia Et Technica*, vol. XVII, n.º 47, 2011.
- [9] E. Sanchis, *Fundamentos y electrónica de las comunicaciones*. UNIVERSITAT DE VALENCIA., 2004.
- [10] W. Tomasi, *Sistemas de comunicaciones electrónicas*, 4.ª ed. PRENTICE HALL MEXICO, 2003.
- [11] C. Pérez Vega, «Modulación de Amplitud Mediante Modulación por Duración de Pulsos (PWM o PDM)». Dpto. de Ingeniería de Comunicaciones, Universidad de Cantabria., 2008.
- [12] Steven F. Barrett, *Arduino Microcontroller Processing for Everyone*, Third Edition. Mitchell A. Thornton, Southern Methodist University, 2013.
- [13] «Microcontrolandos». [En línea]. Disponible en: <http://microcontrolandos.blogspot.com/>. [Accedido: 30-oct-2016].

## Evaluación por simulación en Proteus de Salidas PWM de Arduino Uno

[14] «the engineering projects», 19-sep-2017. [En línea]. Disponible en: [www.theengineeringprojects.com](http://www.theengineeringprojects.com). [Accedido: 19-mar-2017].