

Ciencias técnicas y aplicadas

Artículo original

## **Manipulación de un servomotor con un módulo acelerómetro de 3 ejes MMA 7361 empleando “Arduino” y “Simulink”**

*Manipulation of a servo motor with a MCA 7361 3-axis accelerometer module using "Arduino" and "Simulink"*

*Manipulação de um servomotor com 3 eixos módulo acelerómetro 7361 usando MMA "Arduino" e "Simulink"*

Edwin V. Altamirano-Santillán <sup>III</sup>  
[ealtamirano@epoch.edu.ec](mailto:ealtamirano@epoch.edu.ec)

Geovanny E. Vallejo-Vallejo <sup>IV</sup>  
[g\\_vallejo@epoch.edu.ec](mailto:g_vallejo@epoch.edu.ec)

Juan C. Cruz-Hurtado <sup>III</sup>  
[juankacruzhurtado@gmail.com](mailto:juankacruzhurtado@gmail.com)

**Recibido:** 16 de septiembre de 2016 \* **Corregido:** 20 de diciembre de 2016 \* **Aceptado:** 28 de enero 2017

<sup>I</sup> Magister en Gestión Académica Universitaria, Diplomado Superior en Docencia Universitaria, Ingeniero en Electrónica y Computación, Docente en Escuela Politécnica de Chimborazo ESPOCH.

<sup>II</sup> Magister en Docencia Universitaria e Investigación Educativa, Doctor en Ciencias de la Educación Mención Informática Educativa, Licenciado en Ciencias de la Educación Mención Informática Educativa, Docente de Escuela Superior Politécnica de Chimborazo ESPOCH.

<sup>III</sup> Ingeniero Eléctrico, PhD. Profesor Titular, Instituto Superior Politécnico José A. Echeverría, Habana, Cuba.

## Resumen

Generalmente cuando se usa el “Arduino” para el diseño, la programación del mismo se realiza empleando el lenguaje Processing, lo que requiere de un conocimiento previo de un set de instrucciones y de programación justamente. El objetivo fundamental del trabajo es brindar una alternativa simple y conveniente para personas con bajo conocimiento, o ninguno, del diseño electrónico ni de programación; simplemente conociendo los bloques de la herramienta “Simulink” de Matlab y tener clara la idea gráfica de su diseño, sería capaz de implementarla. Esto también ofrece la posibilidad de realizar el diseño fácil, rápido simple y preciso, probando paso a paso la funcionalidad adecuada del mismo. En el trabajo se brinda una explicación detallada de cómo se haría esto, a través de la implementación del accionamiento de un servomotor con un acelerómetro de 3 ejes, y se muestra la correspondencia entre la posición angular del eje del servomotor con la tensión de salida de dos de sus ejes; lo que, posteriormente, daría la posibilidad de realizar aplicaciones más complejas, mencionadas en el trabajo y que usaran estos dos dispositivos.

**Palabras clave:** plataforma “Arduino”; hardware libre; “Simulink”; “MATLAB”; módulo acelerómetro; servomotor.

## Abstract

Generally when "Arduino" is used for the design, the programming of the same one is carried out using the language “Processing”, it is requires of a previous knowledge of an instructions Set and of programming exactly. The fundamental objective of this work is to offer both, a simple and convenient alternative for people with low knowledge, or none, of the electronic design or programming; simply with the knowing of the blocks of the "Simulink" tool of Matlab and to have clear the graphic idea of its design, it would be able to implement it. This also offers the possibility to carry out the easy, quick easy and the precise design, proving step-by-step the appropriate functionality of the same one. In the work we offer a detailed explanation of how this would be made, through the implementation of the servomotor working using an accelerometer of 3 axes, and we show the correspondence among the angular position of the servomotor axis with the output voltage of two of its axes; this would give the

possibility later, to make more complex applications, mentioned in the work and that they used these devices.

**Keywords:** "Arduino" platform; free hardware; "Simulink"; "MATLAB"; accelerometer module; servomotor.

## Resumo

Normalmente, quando o "Arduino" é usado para projetar, a programação é feita usando o mesmo processamento de linguagem, o que exige conhecimento prévio de um conjunto de instruções com precisão e programação. O principal objetivo deste trabalho é fornecer uma alternativa simples e conveniente para as pessoas com pouco conhecimento ou nenhum projeto eletrônico ou de programação; basta saber os blocos da "Simulink" ferramenta Matlab e ter uma ideia clara do design gráfico, seria capaz de implementá-lo. Isto também oferece a possibilidade de fazer o projeto rápido e preciso fácil, simples, testando passo a funcionalidade adequada. Neste papel é fornecida uma explicação detalhada de como fazer isso, através da aplicação da unidade de um servomotor com aceleretro de 3 eixos, e a correspondência entre a posição angular do eixo do servomotor com a tensão de saída mostrada dois dos seus eixos; O que, então, teria a capacidade de executar aplicativos mais complexos mencionados no trabalho e usar esses dois dispositivos.

**Palavras chave:** plataforma "Arduino"; hardware livre; "Simulink"; "MATLAB"; módulo acelerómetro; servomotor.

## Introducción

El “Simulink”, es una herramienta asociada a la plataforma “MATLAB”. Es muy útil para modelar, simular y analizar sistemas, tanto lineales como no lineales. Permite, desarrollar diagramas en bloques que proporcionan la realización de un modelo que se identifica con un sistema determinado. Esta herramienta además de contar con un número importante de bibliotecas, tiene dos asociadas a “Arduino” que permiten trabajar con esta plataforma. En la literatura consultada por los autores no fue frecuente encontrar aplicaciones donde la herramienta de programación del “Arduino” fuera el “Simulink”. Es por esto que se quiso describir un diseño en el que se utilizara el “Simulink” para

programar un “Arduino Uno” en este caso. En el trabajo se da una breve explicación sobre algunos de los bloques de “Arduino” que se utilizan en la implementación que se propone (T., 2006), (Mathworks, 2013). El uso de esta herramienta para programar la placa, facilita, agiliza y dinamiza este proceso. La programación se hace completamente gráfica interconectando módulos parametrizables, además de que se puede simular el modelo que resulta de dicha programación. Esta simulación es de forma interactiva y sobre la placa. Esto último propicia precisar, corregir, adecuar y modificar de forma dinámica y simple el modelo.

La plataforma de Hardware libre “Arduino” tuvo sus inicios en el Instituto “Interaction Design Institute” de la ciudad italiana de Ivrea en el 2005, cuando se decidió realizar el diseño de un microcontrolador para utilizarlo en proyectos estudiantiles. La popularidad de esta plataforma creció rápidamente cuando varios empresarios se percataron de que la nueva placa constituía un sistema de muy bajo costo y fácil de usar. (Evans M., 2013).

En el mercado actual existen varias versiones de “Arduino”, todas basadas en procesadores de 8 bit AVR de Atmel. Las versiones se diferencian en el tipo de procesador, la frecuencia de reloj, el tamaño de la memoria flash que utilizan y la cantidad de pines de entrada salida que presentan (Evans M., 2013). La placa que se utilizó fue la placa “Arduino Uno”, una de las de complejidad media y de la más barata de las que existen.

Los Acelerómetros son dispositivos que miden la aceleración en los 3 ejes de un objeto al que se encuentren sujetos. Esta se mide en metros por segundo en cada segundo, o en fuerzas G(g). Sus aplicaciones fundamentales son: inclinación y censado de movimiento, detección de caída libre, antirrobo de PC, estabilidad de imágenes en teléfonos celulares, brújulas, compensación de inclinación, censado de movimiento en robótica, etc. Luego se precisa el funcionamiento a través del dispositivo MMA7361 (MMA7361 3-Axis Accelerometer Module. (2013), (Guarnizo, 2013) y (Freescale Semiconductor., 2008).

Los servomotores son módulos con los que se puede posicionar el eje del motor en una posición angular específica utilizando una señal de control. Son muy utilizados para aplicaciones robóticas,

como control desde la superficie de aviones no tripulados, o cualquier objeto que se desee mover a un cierto ángulo y permanecer en este. Más adelante se precisarán aspectos de su funcionamiento (Future Electronics.), (Lucas).

En el trabajo se muestra el accionamiento de un servomotor a través de un acelerómetro de 3 ejes soportado en la plataforma “Arduino”, el cual ha sido programado con la herramienta “Simulink” de MATLAB. A través del “Arduino” se adquiere y procesa la tensión proveniente de dos de los ejes del acelerómetro, el eje Y y X y en dependencia de estos voltajes se rota el eje del servomotor, utilizando las salidas de la placa “Arduino” para accionarlo.

Dentro de las aplicaciones que se pudieran implementar con la combinación de estos dispositivos (acelerómetro y servomotor), que fueron los escogidos para describir la utilización del “Simulink” junto con la plataforma “Arduino”; se pudieran mencionar por ejemplo: las brújulas electrónicas, el control de nivel de una plataforma, el control de movimiento de un brazo robótico (con una combinación de servomotores), etc.

Dentro de los objetivos del trabajo se encuentran el brindar una forma de programar diferente a la tradicionalmente conocida asociada a la placa “Arduino”, a través del diseño, simulación e implementación de un hardware asociado a un acelerómetro y un servomotor. Dicha implementación podría estar asociada a cualquiera de las aplicaciones mencionadas.

Del objetivo de trabajo declarado se persigue resolver el problema que pudiera constituir para el grupo de personas que sin tener conocimiento profundos, o ninguno, de la plataforma “Arduino”, ni de electrónica, pudieran proyectar de forma gráfica una aplicación determinada, sólo conociendo los bloques de “Simulink” asociados a la plataforma y en la que estuvieran involucrados un acelerómetro y un servomotor.

## Desarrollo

### Placa “Arduino Uno”

Existen varias versiones en el mercado de plataformas “Arduino”, como ya se mencionaba, todas basadas en procesadores de 8 bit Atmel AVR. La primera placa contó con un Atmega 8 con un reloj de 16 MHz y con 8 KB de memoria flash. Luego aparecieron las placas “Arduino NG” plus y el “Decimilia” que utilizaban el procesador Atmega 168 con 16 KB de memoria flash. Se tienen versiones más recientes, como el “Duemilanove”, el Uno y el “Arduino Due”, que usan el Atmega 328 con 32 KB de memoria flash. Posterior a estas, se han desarrollado plataformas “Arduino” de más entradas y salidas y de mayor memoria como es el caso de los “Arduinos” Mega 1280 y 2560 de 128 y 256 KB de memoria, respectivamente. (Evans M., 2013).

La placa “Arduino Uno” presenta 14 pines de entrada/salida y 6 de ellas se utilizan como pines de modulación de ancho de pulso (PMW) que son los pines 3, 5, 6, 9, 10 y 11. También presenta 6 pines de entradas analógicas, un interruptor de reset, un conector USB para su conexión con la PC fundamentalmente y un conector para alimentarla externamente, ya sea a través de una fuente de tensión o por medio de baterías. En la figura 1 se muestra el layout y distribución de pines de dicha placa. Esta versión presenta compatibilidad de pines con respecto a versiones previas. (Evans M., 2013), (Martín B., 2013), (S., 2012).

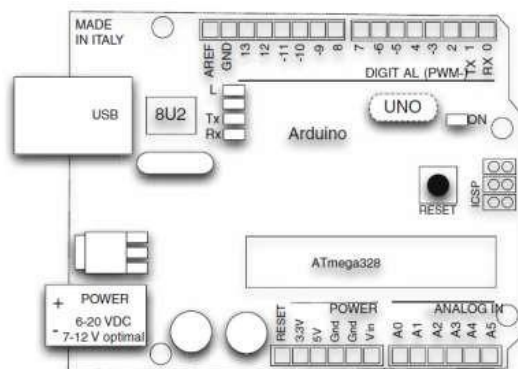


Figura 1. Esquema del layout de la placa “Arduino Uno” con su distribución de pines.

Fuente: (Evans M., 2013).

### Módulo del acelerómetro MMA7361

Es un dispositivo que mide la aceleración, generalmente en los 3 ejes, del objeto al que se encuentre asociado. Están constituidos por capas capacitivas internas que se mueven relativamente unas respecto a las otras, dando lugar a cambios capacitivos y su salida es proporcional a la aceleración. Son útiles como detectores de vibraciones, medidores de inclinación, en aplicaciones de orientación, medición de movimiento, caída libre, brújulas, etc. (Guarnizo, 2013).

El dispositivo que se utiliza (MMA7361) es de baja potencia, acondicionamiento interno de la señal, con un filtro pasa bajo de un polo, compensación de temperatura, auto comprobación, detección de caída libre y posibilidad de selección de dos sensibilidades. Incluye además un modo de reposo (sleep) que lo hace ideal para alimentarlo por baterías (Freescale Semiconductor., 2008).

En la figura 2 se muestra una vista superior de este módulo con su distribución de pines y la disposición relativa de sus ejes. Se observa que tiene dos entradas de tensión, de 5 y de 3.3 V, así como una entrada de tierra GND. En realidad el dispositivo trabaja con 3.3 V de tensión de alimentación. En caso de alimentarse por la entrada de 5 V se activa el regulador de 3.3, aunque admite un rango de tensión de alimentación de 5 a 7 Voltios. Cuenta también con 3 pines de salida (ejes X, Y y Z), que presentarán una tensión relativa a la posición de los ejes del dispositivo en el espacio y a su aceleración.

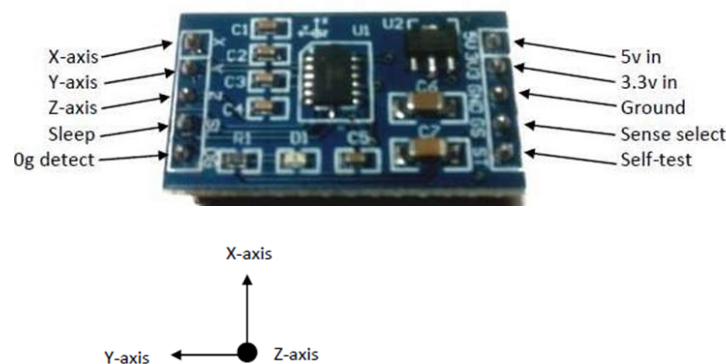


Figura 2. Vista superior del módulo del acelerómetro MMA7361 y su distribución de pines.

Fuente: (Guarnizo, 2013).

De la figura 2 se observa que el dispositivo cuenta con 4 pines de control que se describen a continuación:

**Pin ST:** Self Test: brinda la posibilidad de tener una función de auto verificación que permite la comprobación de la integridad mecánica y eléctrica del dispositivo.

**Pin OG:** g-Detect: un nivel alto en este pin señala la caída libre del dispositivo (0g en los 3 ejes).

**Pin SG:** g-Select: este pin permite el ajuste de sensibilidad del sensor incluso durante la operación del dispositivo. En la tabla 1 se muestra la tabla de ajuste desensibilidad.

Tabla 1. Tabla de ajuste de sensibilidad mediante Pin SG. Fuente: Los autores y, los datos del manual (Freescale Semiconductor., 2008).

Pin SG	Rango de ajuste de g	Sensibilidad
“0”	1.5g	800mV/g
“1”	6g	206mv/g

**Pin SL:** Sleep Mode: un nivel bajo en este pin apaga las salidas y se reduce la corriente de consumo del sensor a 3 $\mu$ A típico, ideal para aplicaciones portátiles.

### Servomotor

En estos dispositivos el eje se ubica en una posición específica usando una señal de control. Constituido por un motor de CD y una caja reductora y 3 cables para alimentarlos y controlar posición en 180° (Future Electronics.), (Lucas).

Los servomotores se pueden controlar a través de circuitos simples, como por ejemplo timer NE555, o controlarlos por microcontroladores como “Arduino”, Raspberry, Lego, etc. Se pueden clasificar de acuerdo al tamaño o torque que pueden soportar, pudiendo ser mini-servo o servos gigantes. Generalmente los mini y estándar servos, pueden ser alimentados por las placas “Arduino”



directamente sin necesidad de fuentes de potencia externas. En la figura 3 se muestra la imagen de un servomotor tipo mini con sus cables de conexión y los diferentes tipos de hélices que tienen (Future Electronics), (Lucas).

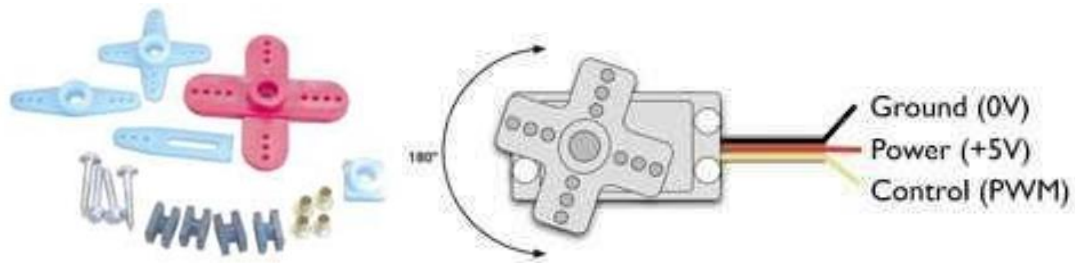


Figura 3. Vista superior de la imagen de un servomotor y sus cables de conexión y, diferentes tipos de hélices con que cuentan.

Fuente: (Future Electronics)

El pin de control puede aceptar una señal modulada por ancho de pulso (PWM) fácilmente obtenidas de microcontroladores y placas de “Arduino”. La longitud del pulso se corresponde con el ángulo que debe girar el servomotor. También pueden ser manejados por tensiones analógicas. En la figura 4 se muestran los diferentes anchos de pulso de una señal PWM y su correspondiente ángulo del eje del servo.

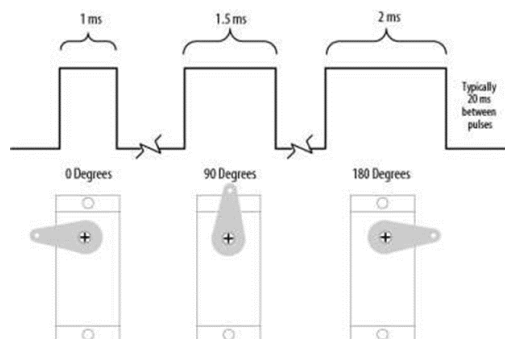


Figura 4. Carta de tensión de la señal de control con los diferentes anchos de pulso y sus correspondientes ángulos y posiciones del eje del servo.

Fuente: (Future Electronics)

Dentro de las aplicaciones más comunes de los servomotores se encuentran las siguientes:

- Control de giro de las ruedas de carro de radio control,
- Control de las alas de los aviones no tripulados,
- Control de las aspas de un helicóptero,
- Movimiento de las articulaciones de un robot,
- Movimientos de impulsos,
- Control del timón de un barco de radio control.

### **Descripción del software Simulink**

En este epígrafe se hará una descripción muy breve de esta herramienta de “MATLAB” y de los bloques que se utilizan en la aplicación propuesta. Es muy utilizada en diferentes áreas de diseño y análisis de ingeniería, entre otras. Se utiliza para crear modelos, simularlos y analizarlos. Presenta una interfaz gráfica que facilita mucho el desarrollo de los diagramas de bloque de las diferentes aplicaciones que se deseen instrumentar. Cuenta con un gran número de bibliotecas, además de que se le añadieron dos librerías para el trabajo con las plataformas de “Arduino”.

### **Bloques de Librerías de “Arduino”**

Los bloques de las librerías de “Arduino” descritos se utilizan en el diseño de la implementación propuesta en este trabajo.

### **Bloque de Salida digital de “Arduino”**

Este bloque pertenece a la Librería: Paquete de Soporte de Simulink para el Hardware de “Arduino” (Simulink Support Package for Arduino® Hardware). El mismo ubica un valor digital en un pin digital del hardware de “Arduino”. Si se ubica el valor lógico de “1”, la tensión en el pin digital es alto de 5 V y si se coloca un valor bajo, en el pin se tendrá 0 V. En el bloque es necesario situar el número del pin donde se desea ubicar el valor lógico. En el caso del “Arduino Uno” los pines serían del 0 al 13 y para el “Arduino Mega 2560” serían del 0 al 53. En la figura 5 se muestran el bloque de salida digital y la caja de dialogo del mismo para su configuración.

No se debe asignar el mismo número de pin a diferentes bloques que se encuentren en el modelo de “Simulink”. Una vez que se fije un número de pin a un tipo de placa asignada, para cambiar esta última deberá hacerlo en: Tools > Run on Target Hardware > Options de “Simulink”.

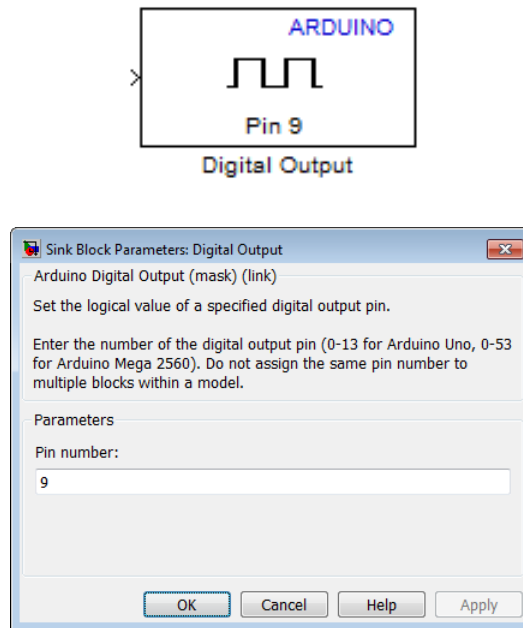


Figura 5. Imágenes del bloque “Digital Output” y la caja de dialogo de sus parámetros.

Fuente: Software Simulink de “MATLAB”

### **Bloque de Escritura de Servo Estándar (Arduino Standard Servo Write)**

Este bloque acciona el eje de un servomotor estándar desde 0 a 180 grados. Para realizar esta rotación se sitúan valores desde 0 a 180 en la entrada del bloque. También se pueden utilizar pulsos para ubicar el eje del servo en el ángulo deseado, como ya se explicó.

Se mencionarán algunas limitaciones que presenta la utilización de este bloque en un modelo en Simulink”:

- No utilizar el bloque Servo con modo Externo en la herramienta “Simulink”.

- Si se utiliza este bloque en modelos con los bloques “Serial Receive y Serial Transmit”, usar números de muestra mayores para evitar “overruns”
- El máximo número de estos bloques en un modelo es de 12 para el “Arduino Uno”, y 48 para “Arduino Mega2560”
- Con “Arduino Uno”, el bloque “Arduino PWM” no puede utilizar los pines 9 y 10 cuando el modelo contiene bloques Servo.
- Con el “Arduino Mega 2560”, el bloque “Arduino PWM” no puede usar los pines 11 o 12 cuando el modelo contiene más de 12 bloques Servo.

En la figura 6 se muestra este bloque y la caja de dialogo correspondiente a dicho bloque.

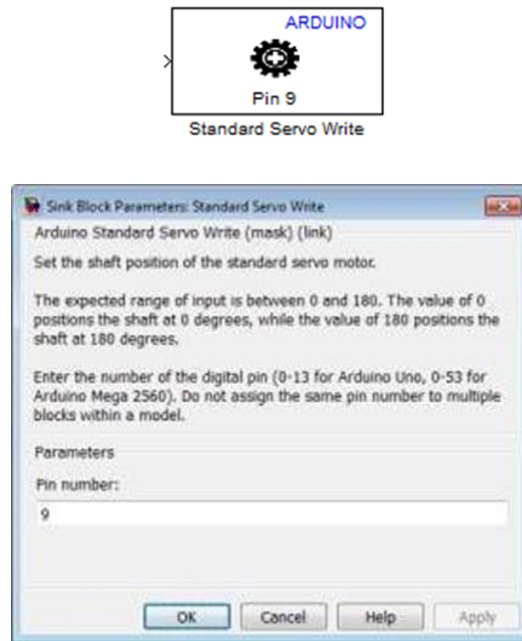


Figura 6. Bloque “Arduino Standard Servo Write” y su caja de diálogo.

Fuente: Software Simulink de “MATLAB”.

### **Bloque Entrada Analógico de “Arduino” (Arduino Analog Input)**

Este bloque también pertenece a la Librería del Paquete de Soporte de “Simulink” para “Arduino”.

El bloque adquiere la tensión de un pin analógico relativo a la entrada analógica de la plataforma “Arduino Uno”. La salida del bloque se corresponde con un valor digital de 10 bit en el rango de 0 a 1023, representando el valor de tensión en la entrada del pin. En caso de que se capture la tensión de tierra, la salida del bloque es de 0 volt y, si fuera la tensión de referencia, en este caso 5 volt, la salida del bloque sería 1023. Para cambiar el parámetro de la Tensión de Referencia en los Parámetros de Configuración del modelo, se deberá seleccionar: Tools > Run on Target Hardware > Options....En la figura 7 se muestra la imagen de dicho bloque y la caja de diálogo que lo acompaña.

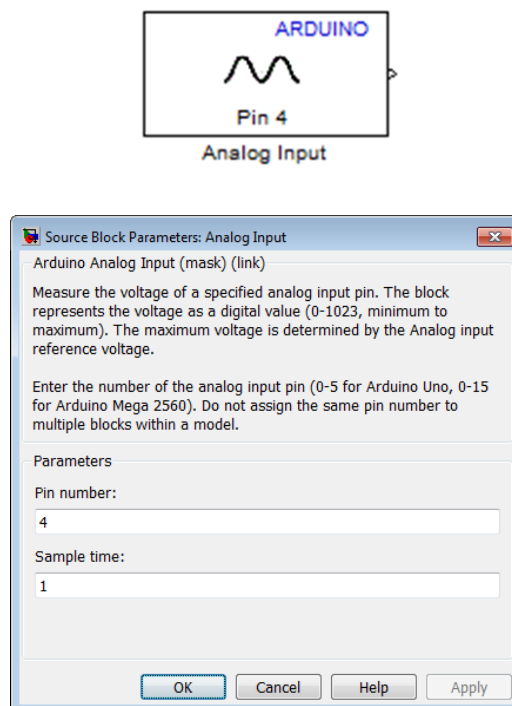


Figura 7. Bloque de entrada analógica y la caja de dialogo asociada.

Fuente: Software Simulink de “MATLAB”.

En el modelo de la implementación aparecen otros bloques más simples y comunes como:

- El Bloque producto: que multiplica o divide las entradas.
- El Bloque constante: que ubica un valor determinado en su salida.

- El Bloque sustractor: que realiza la diferencia o suma de las entradas. Puede tener varias entradas.
- El Bloque comparador: este compara la entrada con una constante que se designe, o compara las entradas del bloque. Presenta varios operadores de comparación que pueden ser escogidos.
- El Bloque Operador Lógico: este realiza la operación lógica que se designe a las entradas que también se seleccione.

A continuación se explica el diseño de la implementación que se propone para describir los pasos y bloques “Simulink” que la integraron, junto a la plataforma “Arduino”. El diseño se dividió en: el bloque de adquisición asociado al acelerómetro, el bloque de procesamiento matemático y lógico y el bloque actuador relacionado con el accionamiento del servomotor.

### **El Bloque de adquisición del acelerómetro**

En este subsistema de adquisición se encuentran los bloques de entradas analógicas 1, 2 y 3 del “Arduino Uno” provenientes de las salidas X, Y y Z del acelerómetro, respectivamente. En la figura 8 se muestra una imagen tomada del modelo “Simulink”, que constituye este subsistema. Como se muestra en la figura, además de contener los tres bloques de entrada analógica, se tienen los 3 bloques divisores y bloques de constantes. Los bloques divisores son necesarios para llevar los valores binarios de la salida de los bloques de entrada analógicos, a los correspondientes valores de tensión de los ejes del acelerómetro. En este caso se dividen por el valor constante 204 en cada eje. Hasta aquí, se tienen los valores de tensión de las salidas de los ejes del módulo acelerómetro MMA 7361. En la figura también aparecen los bloques de salida digital de “Arduino Uno” con sus correspondientes valores constantes en sus entradas, para ubicar los niveles lógicos “0” (0 volt) y “1” (5 volt), correspondientes a los pines de control SG y SL del acelerómetro, respectivamente. Estos niveles se obtienen en los pines 11 y 13 del “Arduino Uno”.

Manipulación de un servomotor con un módulo acelerómetro de 3 ejes MMA 7361 empleando “Arduino” y “Simulink”

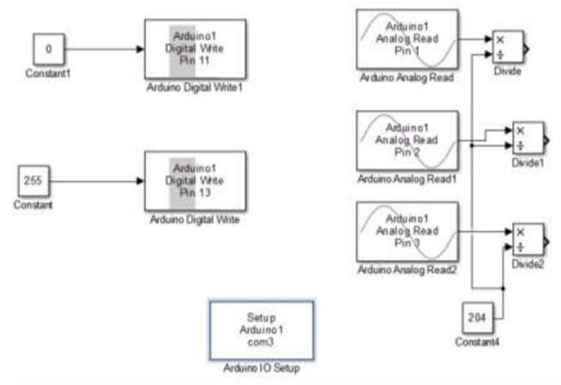


Figura 8. Modelo parcial de “Simulink” que contiene los bloques correspondientes al de adquisición del acelerómetro.

Fuente: Los autores, extraído del modelo “Simulink”.

En correspondencia con el modelo de “Simulink” de la figura 8 se presenta la interconexión del “Arduino” con el acelerómetro, como se muestra en la figura 9. De esta figura se tiene que las entradas analógicas físicas 1, 2 y 3 del “Arduino Uno” se encuentran conectadas a los ejes X, Y y Z, respectivamente como ya se había comentado. También se observa la conexión de los pines 11 y 13 con las entradas de control del acelerómetro mencionadas.

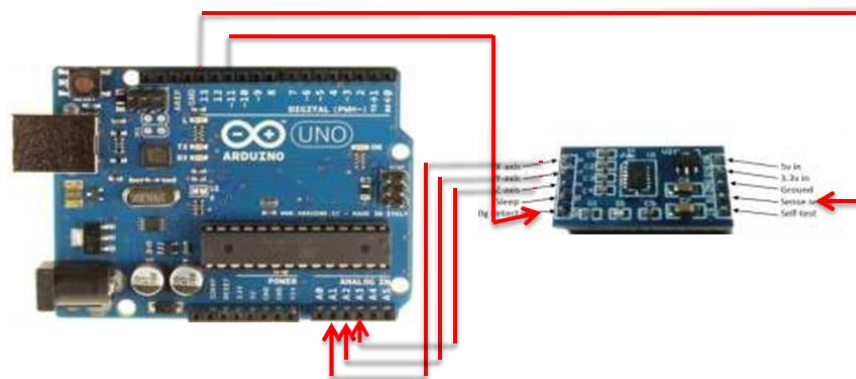


Figura 9. Se muestra el diagrama de interconexión del “Arduino Uno” con el módulo acelerómetro MMA 7361.

Fuente: los autores.

## El Bloque de procesamiento matemático y lógico

Aquí es donde se realiza el procesamiento a las señales de los ejes del acelerómetro, luego de llevarlos a sus correspondientes valores de tensión. Es el encargado de realizar la conversión de tensión de los ejes del dispositivo MMA7361, a ángulo en grados. En la figura 10 se muestra el diagrama en bloques del modelo en “Simulink” que representa a este subsistema.

Del modelo de la figura 10 se observa que el diagrama de este bloque de procesamiento presenta dos elementos comparadores: uno compara las tensiones de salida X y Y y otro compara la suma X+Y respecto al valor “3”, por lo que es necesario un elemento sumador de estas tensiones. Por último le sigue un circuito combinacional sencillo formado por compuertas AND y NOT, que será explicado en breve.

Las entradas al subsistema son las tensiones de los ejes X, Y del acelerómetro, como se señala en la figura 10. En este caso no es necesario medir la tensión del eje Z ya que la rotación del acelerómetro se produce en el plano X-Y, siendo el eje Z el de rotación. Solamente existirá cambio angular en los ejes X y Y.

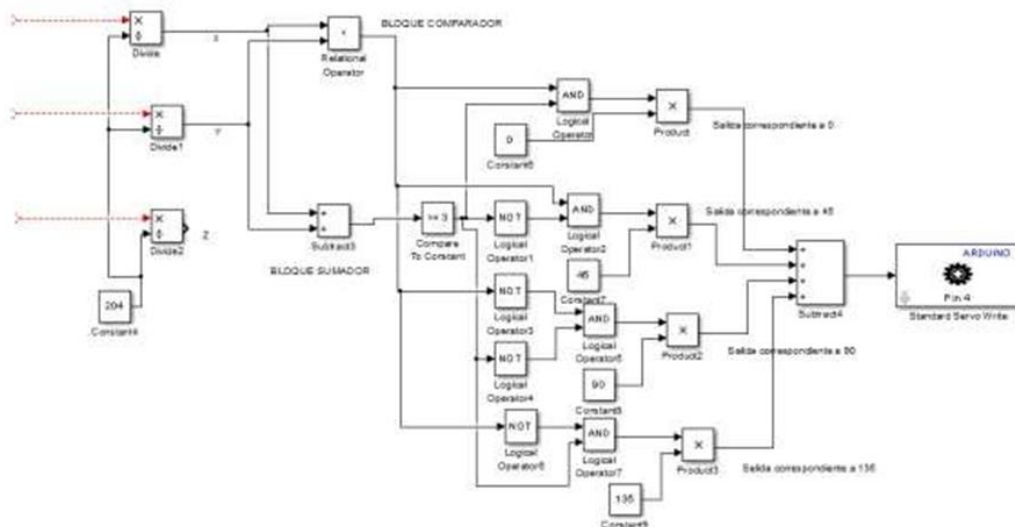


Figura 10. Diagrama del modelo en “Simulink” del bloque de procesamiento matemático y lógico.

Fuente: Los autores, extraído del modelo “Simulink”.



La aplicación consiste en rotar el eje del servomotor en dependencia del giro del acelerómetro. Se coloca el dispositivo MMA7361 de forma vertical y se hace girar sobre el eje Z, como ya se comentó, y hacer rotar el rotor del servo 450 por cada ángulo de 900 que gire el acelerómetro. En la figura 11 se muestra la secuencia de giro (1, 2, 3 y 4) del acelerómetro con las respectivas tensiones de sus ejes que brinda el manual.

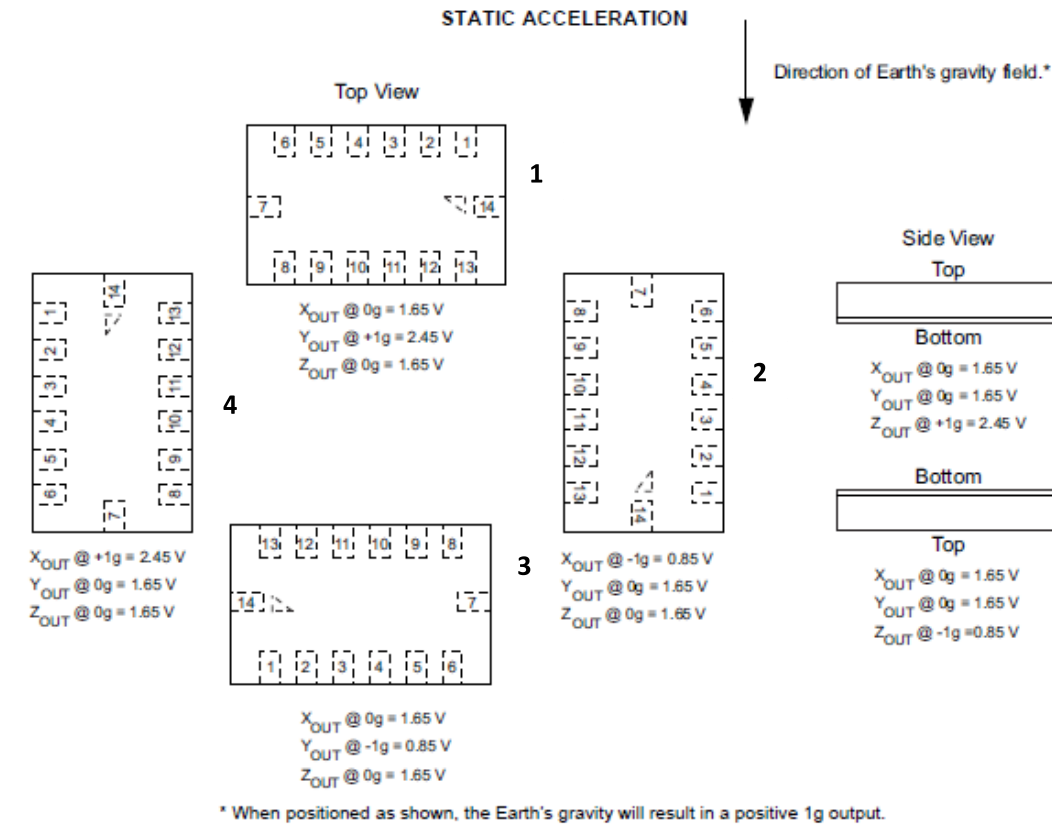


Figura 11. Diferentes posiciones espaciales del acelerómetro con sus correspondientes valores de tensión de salida de cada eje.

Fuente: (Freescale Semiconductor., 2008)

En la tabla 2 se muestran los valores de tensión X y Y de cada una de las posiciones de la secuencia de rotación del acelerómetro.

Tabla 2. Valores de tensión de los ejes X, Y y Z según las posiciones de la secuencia de rotación del acelerómetro mostradas en la figura 11.

POSICIÓN	Tensión en eje	Tensión en eje	Tensión X +
1	1.65 V	2.45 V	4.1 V
2	0.85 V	1.65 V	2.5 V
3	1.65 V	0.85 V	2.5 V
4	2.45 V	1.65 V	4.1 V

De la tabla anterior se pudieran identificar las diferentes posiciones de forma más compactas si se observa lo siguiente:

- Para la posición 1 se tiene que  $X < Y$  y  $X + Y \geq 3$ .
- Para la posición 2 se tiene que  $X < Y$  y  $X + Y < 3$ .
- Para la posición 3 se tiene que  $X > Y$  y  $X + Y < 3$ .
- Para la posición 4 se tiene que  $X > Y$  y  $X + Y \geq 3$ .

Las posiciones del acelerómetro anteriores se pudieran representar por combinaciones digitales de variables dadas por el cumplimiento, o no, de las desigualdades anteriores, tanto de las desigualdades entre  $X-Y$ , como de su sumatoria. Por ejemplo de ser verdadera la desigualdad “ $X < Y$ ” se dice que esta variable tomaría el valor lógico de “1” y de ser falsa tomaría el valor de “0”. Lo mismo ocurriría con la desigualdad de la suma “ $X + Y \geq 3$ ”, que de ser verdadera esta variable tomaría el valor de “1” lógico y, de ser falsa tomaría el valor de “0” lógico.

Además, las posiciones también pudieran identificarse con los ángulos de giro del eje del servo. La posición 1 se va a considerar como la de ángulo 00 o referencia, la 2 con un ángulo de 450, la posición 3 con un ángulo de 900 y la posición 4 con un ángulo de 1350. Con las representaciones digitales de las desigualdades y la correspondencia angular explicadas anteriormente, se pudiera construir la tabla de “verdad 3” siguiente:

Tabla 3. Tabla de la verdad con las variables combinacionales de las desigualdades y los ángulos del eje del servo.

POSICIÓN	X <	X +Y >	ÁNGUL
1	1	1	00
2	1	0	10
3	0	0	00
4	0	1	10

El circuito combinacional de la figura 10 se obtiene de la tabla de la verdad 3 anterior. Se debe observar que las entradas al mismo proceden de los comparadores mencionados. Debido a que la salida de los comparadores tiene una representación digital, como ya se comentó, los mismos, junto al sumador, constituyen un conversor de tensión a binario, con una representación binaria para cada posición del acelerómetro correspondiente a un ángulo, como muestra la tabla de la verdad 3. Se pudiera obtener una mayor resolución angular teniendo en cuenta otras operaciones matemáticas entre las salidas Y y X, que fueran recurrentes y constituyeran otra entrada binaria dicho conversor. Las salidas de los comparadores constituirán las entradas del circuito combinacional, como se indica en la figura 10.

A la salida del circuito combinacional se colocan bloques productos para obtener a la salida el valor de ángulo correspondiente a la combinación binaria asociada con la posición espacial del acelerómetro. Se utiliza un bloque sumador al final para la obtención del ángulo correspondiente.

### **El Bloque actuador del servomotor**

Este es un Bloque de Escritura de Servo Estándar (Arduino Standard Servo Write), ya explicado, por lo que la implementación es muy simple, como se observa en la figura 10. A la entrada de dicho bloque aparecerán los ángulos correspondientes a la posición del acelerómetro en el espacio.

En la figura 12 se muestra el modelo completo de la aplicación implementada.

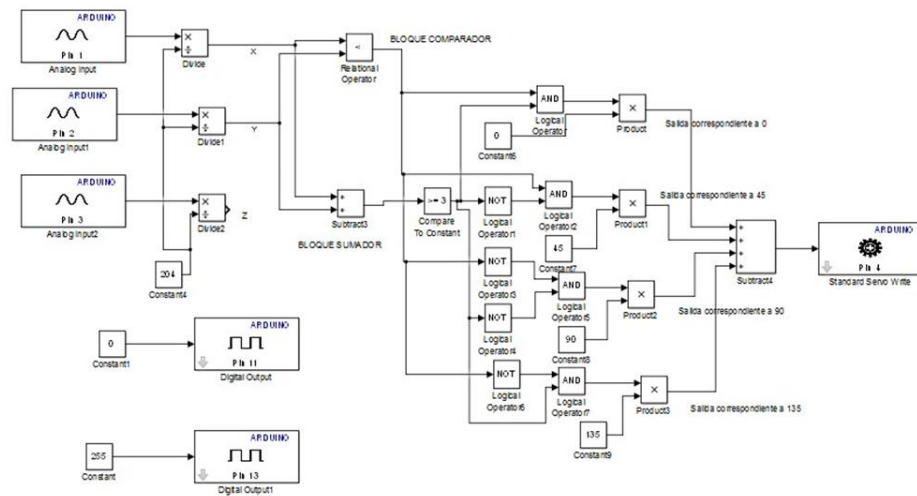


Figura 12. Muestra el modelo completo de “Simulink” de la aplicación implementada.

## Resultados

En este apartado se mostrarán:

- Los valores de tensión de las salidas X y Y del acelerómetro obtenidas en la práctica y las que brinda el manual, para las cuatro posiciones espaciales del acelerómetro descritas,
- Las figuras que representan las cuatro posiciones espaciales del dispositivo MMA7361 con sus valores de tensión correspondientes y la posición angular del eje del servomotor,
- La foto del circuito implementado, mostrando el sentido de giro del acelerómetro para obtener las cuatro posiciones espaciales explicadas.

En la tabla 4 siguiente se muestran los valores prácticos y los valores que brinda el manual, para las cuatro posiciones espaciales. Tabla 4. Muestra los valores reales obtenidos y los brindados por el manual, para cada una de las cuatro posiciones descritas.

Manipulación de un servomotor con un módulo acelerómetro de 3 ejes MMA 7361 empleando “Arduino” y “Simulink”

POSICIÓN	TENSIÓN REAL DE X	TENSIÓN DEL MANUAL DE X	TENSIÓN REAL DE Y	TENSIÓN DEL MANUAL DE Y
1	1.6 V	1.65 V	2.5 V	2.45 V
2	0.8 V	0.85 V	1.6 V	1.65 V
3	1.6 V	1.65 V	0.87 V	0.85 V
4	2.4 V	2.45V	1.6 V	1.65 V

El valor de tensión del eje Z es constante e igual a 1.7 V, ya que este no cambia de posición.

En la figura 13 se muestra la posición relativa del dispositivo MMA7361, los valores de tensión de manual, los ejes de coordenada que representan dicha posición y la posición angular del eje del servomotor para la posición 1. Esta constituirá la referencia del eje. Cada una de las posiciones difiere en 450 de la posición anterior.

En la tabla 5 se muestran los errores relativos a las tensiones del manual, de los resultados de las tensiones de los ejes X y Y del acelerómetro para las diferentes posiciones de los ejes del rotor deservomotor.

Tabla 5. Muestra los errores relativos de las tensiones de salida de los ejes del acelerómetro respecto a las tensiones nominales, para cada posición.

POSICIÓN	ERROR RELATIVO EN X	ERROR RELATIVO EN Y
1	3	2.04
2	5.8	3
3	3	2.35
4	2.04	3

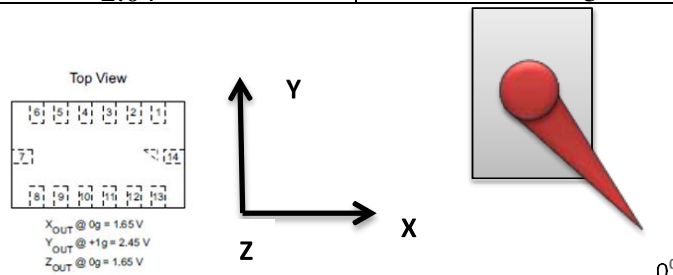


Figura 13. Representa la posición espacial 1 del acelerómetro con su correspondiente posición del eje del servo.

Fuente: los autores.

En la figura 14 siguiente se muestra la posición del acelerómetro, las tensiones correspondientes y la posición del servo dos.

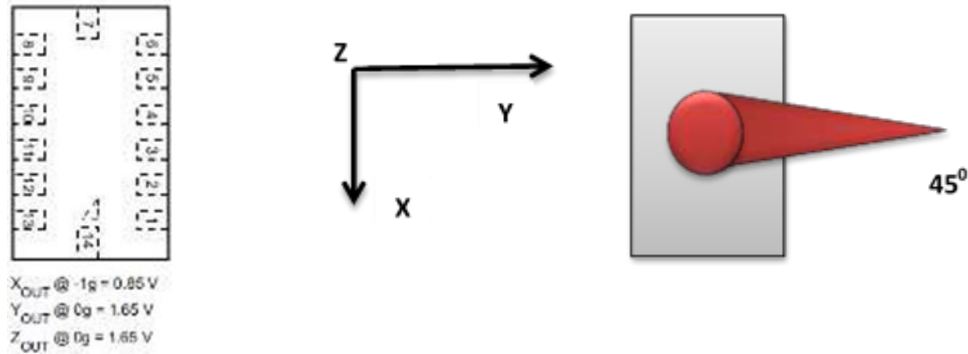


Figura 14. Muestra la posición espacial 2 del acelerómetro con su correspondiente posición del eje del servo y las tensiones de cada eje.

Fuente: los autores.

En la figura 15 siguiente, se muestra la posición del acelerómetro para la posición número 3, y sus correspondientes valores de tensión y posición angular del eje del servomotor.

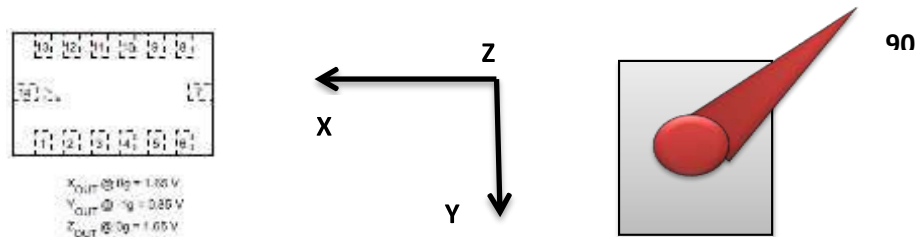


Figura 15. Muestra la posición del acelerómetro, sus valores de tensión y la posición del eje del servo para la posición 3.

Fuente: los autores.

En la figura en la figura 16 siguiente se muestra la posición número cuatro del acelerómetro junto con sus valores correspondientes de tensión de los ejes de salida y a la posición angular del eje del servomotor.

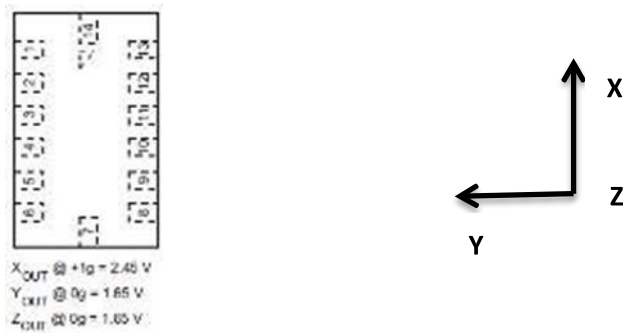


Figura 16. Muestra la posición del acelerómetro, sus valores de tensión y la posición del eje del servo para la posición 4.

Fuente: los autores.

Por último, en la figura 17 se muestra una imagen del circuito real implementado y el sentido de rotación realizado al board donde se encuentra conectado el acelerómetro, con el que se fueron obtenidas cada una de las cuatro posiciones ya documentadas.

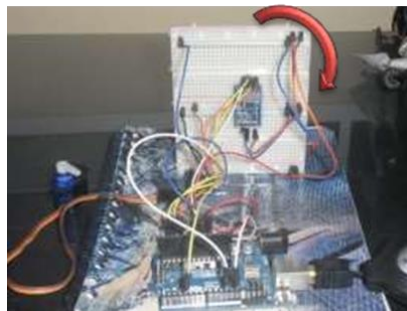


Figura 17. Se muestra la imagen del circuito de la implementación propuesta con el sentido de giro del acelerómetro.

## Conclusiones

En el trabajo se mostró, de forma didáctica, cómo interactuar con la herramienta “Simulink” y el “Arduino”, describiendo una forma diferente de programación poco documentada en la literatura consultada, utilizando bloques específicos de la herramienta asociados a la plataforma hardware, cubriéndose el objetivo fundamental del trabajo.

Esta forma de hacer, además de facilitar, agilizar y dinamizar la programación, lo que se hace de una forma completamente gráfica a través de la interconexión de módulos parametrizables, también hace posible realizar la simulación del modelo que resulta de forma interactiva y sobre la misma placa, lo que ofrece la oportunidad de precisar, adecuar y modificar el diseño de forma muy dinámica y simple.

Para ilustrar esta forma de hacer, se implementó un circuito actuador de un servomotor estándar por medio de un acelerómetro de tres ejes tipo MMA7361, lo que constituye una de las posibles aplicaciones realizables con estos dos dispositivos.

Destacar que esta forma de programación gráfica tan simple, podría ser apropiada para el grupo de personas que no tienen conocimientos profundos de electrónica ni de la plataforma hardware.

Por último, es importante señalar que, observando la tabla 5, se tiene que, en general, los errores son adecuados para este tipo de aplicación preliminar y simple.

## Referencias bibliográficas

Evans M., N. J. (2013). *Arduino in Action*. Shelter Island, New York.: Manning Publications Co.

Freescale Semiconductor. (2008).  $\pm 1.5g$ ,  $\pm 6g$  Three Axis Low-g. Arizona, EEUU: Freescale Semiconductor. Future Electronics. (s.f.). *Servo Motors Control & Arduino*. FutureElectronics.

Guarnizo. (2013). *Manual de Usuario. Acelerómetro MMA7361*. Moviltronics, Ltda.

Lucas, L. M. (s.f.). *CONTROL DE UN SERVO CON ARDUINO*. Sevilla: IES Los Viveros Dpto. Electrónica.

Martín B., D. R. (2013). *Control de Posición de un Balancín con Arduino*. Valladolid, España.: Universidad de Valladolid. . Mathworks. (13 de Agosto de 2013). "MATLAB" 2013b (8.2.0701). EEUU.

MMA7361 3-Axis Accelerometer Module. (2013). Obtenido de <http://www.apexelectrix.com>



---

Manipulación de un servomotor con un módulo acelerómetro de 3 ejes MMA 7361 empleando “Arduino” y “Simulink”

---

S., L. (2012). Diseño de un sistema de control domótico basado en la plataforma Arduino. Valencia, España.: Escuela Técnica superior de Ingeniería Informática, Universidad Politécnica de Valencia.

T., S. (2006). Introduction to Simulink with Engineering Applications. Orchard Publications.