Título: "Sistema de Accionamiento Electrónico con Interruptor *reed*, utilizando *Arduino y Simulink*".
Title: "System Working with reed switch, using Arduino and Simulink"

Geovanny Estuardo vallejo vallejo[1] Hugo Marcelo Velastegui [2], y Juan Carlos Cruz Hurtado[3].

1 Escuela Superior Politécnica del Chimborazo, Doctor en Informática Aplicada, Magister en Docencia Universitaria, Lic. En Informática Aplicada, Tecnólogo en Informática Aplicada, Profesor Principal, Ecuador, g_vallejo@espoch.edu.ec.

2 Escuela Superior Politécnica del Chimborazo, Ingeniero Docente, hugo.velastegui@espoch.edu.ec.

3 Universidad Tecnológica de la Habana, Ingeniero Eléctrico, PhD. Profesor Titular, Habana, Cuba, juankacruzhurtado@gmail.com. Autor encargado de la comunicación del trabajo.

RESUMEN

Normalmente las aplicaciones diseñadas con la plataforma "Arduino" se realizan empleando el lenguaje ***Processing*** utilizando su entorno de desarrollo, lo que requiere el conocimiento de dicho lenguaje. En este trabajo se brinda una alternativa muy simple, y que es muy conveniente para personas que se inician en el diseño electrónico asociado a esta plataforma hardware. Se presenta el diseño usando la herramienta ***"Simulink"*** de MATLAB lo que simplifica, acelera y dinamiza el diseño. Además, esta herramienta facilita la comprensión de la funcionalidad y la visibilidad de topología del diseño, siendo esto muy apropiado para facilitar el aprendizaje de la electrónica. En el artículo se presenta el accionamiento de un servomotor a través de la introducción de una combinación de dos dígitos que se suministra por medio de un interruptor ***"reed"***. El empleo de esta herramienta favorece, también, la simulación del modelo en ***"Simulink"***, lo que permite la precisión y comprobación del diseño, de forma dinámica y simple.

Palabras claves: Plataforma ***Arduino***, Hardware Libre, ***Simulink***, Llaves electrónicas, Interruptor Magnético.

ABSTRACT

Usually the applications designed with the platform "Arduino" are carried out using the language Processing using their development environment, what requires the knowledge of this language. In this work we offer a very simple alternative, and very convenient for people that begin in the electronic design associated to this platform hardware. In this article we show the design using the tool "Simulink" of MATLAB what simplifies, it accelerates and it intensifies the design. Besides, this tool facilitates the understanding of the functionality and the visibility of the design topology, being this very appropriate to facilitate the learning of the electronics. In the article we show up the servomotor working through the combination of two digits that it is given by means of a switch "reed." The using of this tool favors, also,

the model simulation in "Simulink", it beside allows the precision and confirmation of the design, in a dynamic and simple way.

Key Words: Arduino Platform, Free hardware, Simulink, Electronic keys, Magnetic Switch.

1. INTRODUCCIÓN

La plataforma "Arduino" surge en el 2005 en Italia dada la falta de un hardware de bajo coste y de fácil prototipado. Las pocas placas realizadas en los inicios, se hicieron por estudiantes de un instituto de diseño. La aceptación de dicho hardware aumentó en la medida de que varios empresarios visualizaron sus ventajas económicas y de la sencilla, y rápida, implementación circuital Ya hoy día se cuenta en el mercado con varias versiones de esta plataforma. Las versiones se diferencian en: el tipo de procesador que utilizan, el tamaño de la memoria flash y la cantidad de puertos entrada/salida que tienen (Evans M., 2013).

La familia de dispositivos "Arduino" está constituida por hardware y software libre, con un ambiente de desarrollo realizado en lenguaje ***Processing***. Los prototipos realizados con esta plataforma se pueden utilizar de forma independiente interactuando con sensores y actuadores, para relacionarse con el entorno, o pueden conectarse a un ordenador a través de software con entornos esclavos. Esta plataforma puede ser adquirida completamente ensamblada, u obtener sus componentes y ser construida por el usuario.

Regularmente en el entorno de desarrollo de esta plataforma es donde se escribe el código que constituyen los programas para las aplicaciones que se realizan con estos dispositivos. Luego este programa se introduce en el "Arduino". También se tienen otras herramientas de programación gráfica para "Aduino" que se comentarán posteriormente. Algunas de estas herramientas presentan un entorno poco descriptivo, que dificultan la comprensión de la topología del circuito de forma rápida y fácil.

Existen otras herramientas, además de las que se comentarán, que son muy útiles para programar el "Arduino", como es el caso del ***"Simulink"***. Ésta última es una herramienta divulgada en el entorno académico y profesional, con la que no sería necesario dominar el lenguaje de programación de esta plataforma, fundamentalmente en aplicaciones de determinada simplicidad. Posteriormente se describirán las bondades que tiene este software en la programación de este dispositivo. Esta herramienta cuenta con dos librerías específicas para "Arduino", las que se presentarán en materiales y métodos (T., 2006)**,** (Mathworks, 2013).

En este trabajo se propone la programación del dispositivo con una herramienta que proporciona la comprensión de la funcionabilidad de la aplicación en cuestión, así como su análisis circuital de forma fácil y rápida, muy apropiada para diseñadores de poca experiencia. Esto además apoyará el aprendizaje de aspectos elementales de electrónica, tanto digital como analógica. Todo lo anteriormente planteado constituye la dificultad que se pretende resolver.

Es por ello que el objetivo del trabajo lo constituye la presentación del diseño de una aplicación que usa la plataforma "Arduino" programada con la herramienta ***"Simulink"*** permitiendo, además de obtener un diseño ágil y dinámico, realizar la simulación del modelo de la aplicación sobre la misma placa de "Arduino". Esto último posibilitará precisar, adecuar y modificar el diseño de una forma rápida y muy simple.

Para proporcionar esta facilidad se tiene que la **hipótesis** que se propone es que si se tiene la herramienta ***"Simulink"*** y la placa "Arduino", es posible obtener diseños de modelos en ***Simulink*** de fácil comprensión, de bajo coste y que no requirieran de bibliotecas muy especializadas de "Arduino", aunque permitiendo utilizar los complejos módulos que presenta ***"Simulink"*** para el procesamiento lógico y analógico. Además de que la herramienta permite la obtención de códigos en lenguaje "C", también muy útiles para la programación de microcontroladores.

Por último, en el artículo se describe la placa "Arduino Uno", la herramienta ***"Simulink"***, algunas otras herramientas gráficas de programación y, los dispositivos utilizados en el diseño como: el servomotor, el interruptor tipo ***"reed"*** y una lámpara siete segmentos.

2. MATERIAL Y MÉTODO

A continuación se describen la placa "Arduino", los módulos que se conectan a dicha plataforma, algunas de las herramientas gráficas que se utilizan en la programación de este microcontrolador y la herramienta ***"Simulink"***, que es la que se utiliza. Se describen los elementos de esta herramienta que se utilizan en el diseño. También se exponen las características de los diferentes módulos que se usan.

La plataforma *Arduino* Uno

Como se conoce, hay diferentes versiones de placas "Arduino" que se comercializan. Todas están constituidas por procesadores de 8 bit Atmel AVR y se diferencian en el tipo específico de microcontrolador, el tamaño de la memoria de datos, la frecuencia de reloj y la cantidad de puertos de entrada/salida con las que cuentan. Dentro de las versiones de placas se tienen: el ***Arduino*** NG +, el

Decimilia, ***Duemilianove***, los ***Arduino Uno***, el ***Mega*** 1280 y el ***Mega*** 2560 (Evans M., 2013). Actualmente se ha desarrollado una nueva y potente plataforma llamada "Netduino" basada en Microsoft. Net Micro Framework, compatible pin a pin con la plataforma "Arduino" (Carlos, 2016)

La plataforma "Arduino Uno" es una placa de propósito general que tiene 14 pines de entrada/salida, de donde seis de estas se utilizan para señales de modulación de ancho de pulso (PMW) coincidiendo con los pines 3, 5, 6, 9, 10 y 11. Además cuenta con 6 pines de entradas analógicas. La placa presenta un interruptor de "***reset"***, un conector USB para su interconexión con la PC y un conector para alimentar la placa externamente, ya sea a través de una fuente de tensión o por medio de baterías. En la figura 1 se muestra la imagen de la placa "Arduino Uno" con su distribución de pines (Evans M., 2013), (Martín B., 2013), (S., 2012), (F., 2012).

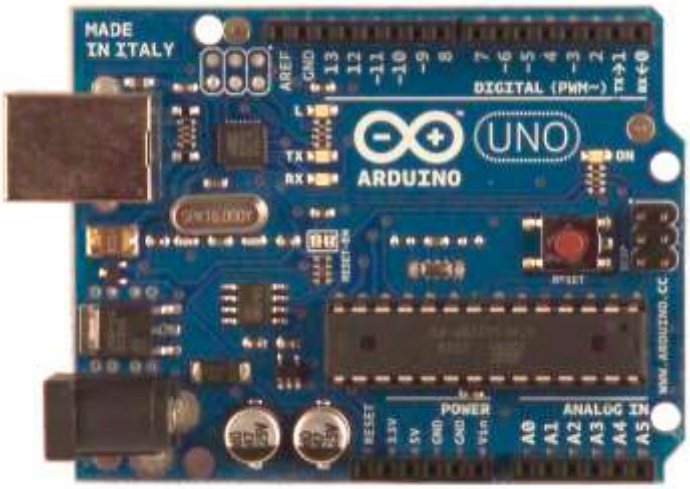

Figura 1. Esquema del ***layout*** de la placa "Arduino Uno" con su distribución de pines.
Fuente: (Evans M., 2013).

A continuación se pasa a describir los elementos que se utilizan en la aplicación que se propone.

Servomotor

Es un dispositivo que de aplicarse una tensión de CD o un tren de pulso por su entrada de control, se logra que su eje gire un ángulo determinado, permaneciendo en dicha posición hasta que no cambie dicha señal de control. Además de la entrada de control, tiene un pin para la alimentación y otro para la tierra (GND) (Future Electronics.), (Lucas.). Se clasifican por su tamaño o el torque puedan suministrar en: mini y los estándares. Unos y otros, pueden ser manipulados por las diferentes versiones de plataformas de "Arduino", así como por Raspberry pi, Lego, etc.

El “Arduino” puede controlar los servomotores en régimen de pulsos, a través de sus pines de entrada/salida PWM. En éste régimen, el ángulo de giro del eje del servo depende del ancho del pulso en cuestión. En la figura 2 se muestran algunos anchos de pulso de una señal PWM y su correspondiente ángulo del eje del servo.

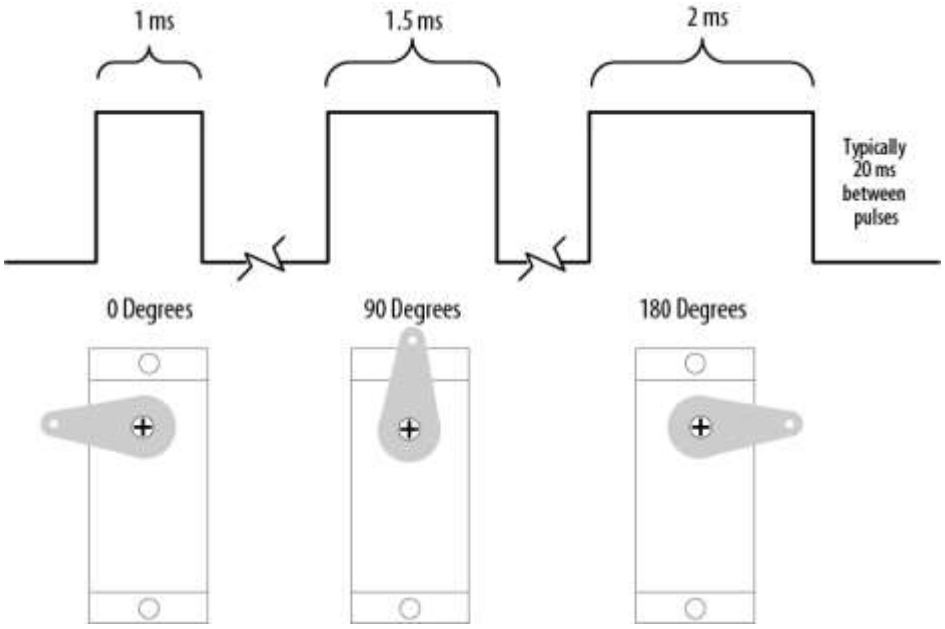

Figura 2. Carta de tensión de la señal de control con los diferentes anchos de pulso y sus correspondientes ángulos y posiciones del eje del servo.
Fuente: (Future Electronics.).

Dentro de las aplicaciones más comunes en las que se utilizan los servos se tienen: en el control de giro de las ruedas de un carro con radiocontrol, control de alas de los aviones no tripulados, control de los rotores de las aspas de un helicóptero, movimiento de las articulaciones de un robot, en el control del timón de un barco de radiocontrol, etc.

Interruptor *reed* (Manual SCHMERSAL), (Manual Satel, 2009).
Presentan dos elementos: el interruptor propiamente y el imán que lo acciona. Están contenidos dentro de un empaquetamiento termoplástico y su forma está en dependencia de su aplicación. Los contactos del interruptor pueden ser: normalmente abierto (NA), normalmente cerrado (NC) o biestable. El acercamiento del imán al sensor puede ser frontal o lateral y con diferentes distancias de accionamiento. El que se utiliza en el trabajo es del tipo NA y la distancia mínima de accionamiento es de 2.5 cm.

El interruptor ***“reed”*** se utilizó para proporcionar la combinación necesaria que produce el accionamiento del servomotor, además de la señalización del evento ocurrido. En resumen es el que se

encarga de suministrar el dígito que compone la combinación que da lugar al accionamiento del servo. En la figura 3 se muestra un esquema funcional del interruptor "***reed***".

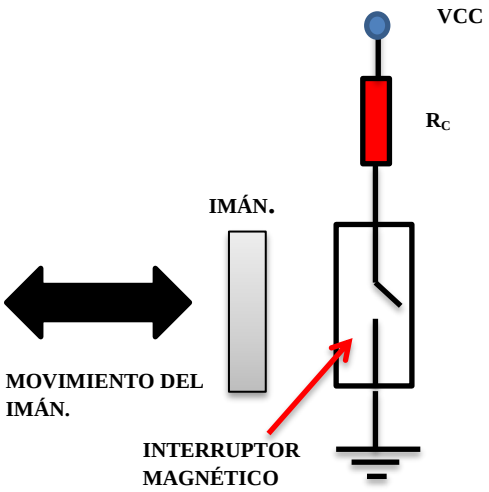

Figura 3. Esquema funcional y eléctrico del interruptor magnético tipo ***reed***.

Fuente: Los autores.

Lámparas Siete Segmentos

Son utilizados para presentar dígitos y caracteres en los equipos electrónicos. Aunque los ***display*** de cristal líquido LCD son muy utilizados en la actualidad debido a su bajo consumo y otras bondades, los de siete segmentos no necesitan de iluminación externa para poder ser leídos y son muy utilizados en estos casos. Como se conoce, están formadas por 7 u ocho leds en diferentes posición y forma. Siete segmentos para se utilizan para formar los caracteres alfanuméricos y el octavo para el punto decimal, de ser necesario. Con la combinación de los segmentos se logra representar los dígitos del cero al nueve. Los colores de leds más utilizados en estas lámparas son los de color rojo y amarillo, aunque se encuentran, también, las de color verde. En la figura 4 se muestra un tipo de lámpara siete segmentos.

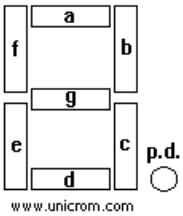

Figura 4. Lámpara siete segmentos con la asignación de letras para cada segmento.

Fuente: manual Motorola.

Existen dos tipos de lámparas en dependencia de la forma en que se alimentan. Se tienen las lámparas de ánodo común y las de cátodo común. Las primeras tiene los ánodos de los diodos de cada segmento

conectados entre sí e irán al terminal positivo de la fuente de CD. En este caso para activar un segmento determinado hay que conectar el cátodo a tierra, a través de una resistencia limitadora. En el caso de las lámparas de cátodo común, son ahora los cátodos los que van unidos y conectados a tierra. Para poder activar el segmento correspondiente, se necesitaría conectar el ánodo a un potencial positivo, a través de una resistencia que garantice la corriente nominal por el diodo. En el caso de éste trabajo se utiliza la lámpara de ánodo común.

Herramientas Gráficas de Programación para *Arduino* (M., 2014)

Se tienen tres tipos generales de herramientas para la programación gráfica que pueden interactuar con las placas de "Arduino" y son las siguientes:

Las de entornos autónomos: utilizadas para la programación del hardware en cuestión y que garantizan el trabajo autónomo del dispositivo. Dentro de estas se encuentran las siguientes: el ***Minibloq***, el ***Ardubloq***, el ***Amici***, el ***Mind+***, el ***ModKit***, el ***VirtualBreadBoard*** y el ***VBB-JARVIS*** entre otros.

Las de entornos esclavos: son las que se utilizan para el monitoreo y supervisión de la aplicación en cuestión. Las herramientas de programación gráfica que se encuentran dentro de este grupo son: el ***Etoys (Squeak)***, el ***S4A (Scratch)***, el ***Snap!, Scratch 2.0***, el ***Pure Data***, el ***Labview***, el ***Firefly (Rinho-Grashoper)***, el ***MyOpenLab***, entre otros.

Existen **otros entornos** dentro de este tipo de herramientas que son los usados, fundamentalmente, para la realización de circuitos impresos (PCB) y dentro de estos se destaca el "***Fritzing"***.

Descripción de la Herramienta Simulink (Karris, 2006)

Esta herramienta es utilizada en diferentes áreas de las ciencias exactas y técnicas. Se usa para el diseño, la simulación y el análisis de aplicaciones lineales y no lineales. Presenta un gran número de librerías asociadas a las ingenierías, las matemáticas, lógicas y especiales. También cuanta con dos librerías asociadas a la programación de las plataformas "Arduino". Cuenta además, con interfaces gráficas de usuario (GUI, por sus siglas en inglés).

A continuación se exponen, de forma breve, los bloques de "Arduino" que tiene esta herramienta y que se utilizan en la aplicación que se trata en este trabajo.

Bloque *"Digital Read"*

Se utiliza para realizar la lectura digital de un pin determinado del "Arduino". Dicho valor se obtiene en la salida de este bloque dentro del modelo de la herramienta ***"Simulink"***. En la figura 5 se muestra este bloque y su caja de diálogo donde se realiza la configuración del bloque.

Figura 5. Bloque funcional ***Digital Read*** (izquierda) y caja de diálogo para configurar el bloque.
Fuente: Herramienta ***Simulink*** de ***MATLAB***.

El primer parámetro que aparece en la caja de diálogo, se utiliza para definir el tipo de "Arduino" de la familia. El segundo parámetro que aparece es el número de pin que se utilizaría para la adquirir la tensión digital de entrada (pines del 2 al 13 para el "Arduino Uno"). Como se observa en la figura 6, el número del pin seleccionado aparece dentro del bloque funcional que aparece a la izquierda.

Bloque Arduino *"Digital Write"*

Este bloque tiene la función de escribir el valor digital ("1" o "0") en el pin de salida que se seleccione. El valor digital deseado se ubica en la entrada del bloque funcional dentro del modelo ***"Simulink"***. En la figura 6 se muestran el bloque funcional y la caja de diálogo de este bloque.

Figura 6. Bloque ***Digital Write*** (izquierda) y caja de diálogo para configuración el bloque de escritura digital.

Fuente: Herramienta ***Simulink*** de ***MATLAB***.

Al igual que en bloque anterior, el primer parámetro que aparece en la caja de diálogo es para seleccionar el tipo de "Arduino" de la familia con que se desea trabajar. El siguiente parámetro que aparece, es el pin de salida donde se quiere ubicar la tensión lógica en la placa.

Bloque Contador

Éste bloque se designa para realizar el conteo ascendente, o descendente, hasta una determinada cantidad que viene dada por el parámetro llamado: rango de conteo. La entrada de incremento (Inc.) que presenta el bloque, se habilita cuando se configura para conteo ascendente. En caso de que se configure en corrida libre, se deshabilitarían las entradas Inc. y Dec. Para esta configuración se realizaría un conteo libre en un tiempo determinado. Para el resto de las configuraciones del parámetro "evento", el bloque realizará un conteo en incremento, o en decremento, cada vez que ocurra un disparo en la entrada Inc. o Dec. respectivamente. En caso de que se aplique un pulso positivo en el puerto opcional "R_{ST}", se resetea el contador e iría a su estado inicial.

Los parámetros del bloque que se pueden configurar a través de su caja de diálogo son:

- Dirección de conteo: ascendente o descendente
- Tipo de evento de conteo: por flanco de subida, por flanco de caída, por flanco de caída o de subida.
- Tamaño del contador: definida por el usuario, de 8, 16 o 32 bits y especificar por puerto de entrada.
- Conteo máximo: es el número máximo de conteo del contador antes de pasar a su estado inicial.
- Conteo inicial: es el valor por donde inicia el conteo de eventos.

- Salida: puede ser de conteo, de ***Hit*** o de ambos.
- Valores del ***Hit***: cuando el contador llega a este valor, la salida ***Hit*** pasa a un "1" lógico.
- Tipo de dato del contador: especifica el tipo de dato en el puerto de conteo Cnt. Este parámetro aparece solamente cuando se configura el parámetro de salida del contador a ***Count*** o a ***Count and Hit.***

En la figura 7 se muestra el bloque contador y la caja de diálogo de parametrización del contador.

Figura 7. Bloque contador (izquierda) y caja de diálogo con parámetros del bloque.
Fuente: Herramienta ***Simulink*** de ***MATLAB***.

Bloque de Operador Lógico

Este bloque está constituido por las compuertas lógicas conocidas para diseño combinacional, como son: NOT, AND, OR, NAND, NOR, XOR y NXOR. En la figura 8 se muestra el bloque lógico NOT, que es uno de los utilizados en el modelo del diseño. El otro bloque lógico utilizado es el AND.

Figura 8. Bloque funcional del operador lógico y su caja de dialogo con los parámetros del bloque.
Fuente: Herramienta ***Simulink*** de ***MATLAB***.

Bloque Constante

Este es otro de los bloques utilizados en el diseño. El bloque se usa para generar un valor real o complejo. También se pueden generar escalares, vectores o matrices de salida lo que dependerá de: la dimensión del parámetro, del parámetro de valor constante y del ajuste del parámetro del vector. En la figura 9 se muestra el bloque funcional y la caja de diálogo para su parametrización.

Figura 9. Bloque constante y su caja de dialogo con los parámetros del bloque.

Fuente: Herramienta ***Simulink*** de ***MATLAB***.

Bloque Comparador

Obviamente es un bloque designado para comparar los valores de las variables de entrada, respecto a un valor constante que se asigna por el diseñador. Dicha constante se especifica en el parámetro "valor constante". La comparación se realiza teniendo en cuenta el parámetro llamado "Operador". A continuación se muestran las opciones de comparación que tiene el bloque con el significado del operador que parece en la caja de diálogo para su configuración:

== Determina si la variable de entrada es igual a la constante especificada.

~= Determina si la variable de entrada no es igual a la constante especificada.

< Determina si la variable de entrada es menor que la constante especificada.

<= Determina si la variable de entrada es menor o igual a la constante especificada.

> Determina si la variable de entrada es mayor que la constante especificada.

>= Determina si la variable de entrada es mayor o igual a la constante especificada.

En la figura 10 se muestran el bloque funcional del comparador en cuestión y la caja de diálogo para la parametrización del bloque.

Figura 10. Bloque funcional del comparador y su caja de diálogo para su parametrización.
Fuente: Herramienta ***Simulink*** de ***MATLAB***.

Bloque *Filp-Flop* tipo D

El bloque tiene tres entradas: la entrada de datos "D", la entrada de reloj "CLK" y la entrada que habilita la señal de entrada "!CLR". En caso que el bloque esté habilitado (!CLR si está en "1" lógico) y se recibe un pulso ascendente en la señal de reloj, esto hará que la señal de salida "Q" tome el mismo valor lógico que tenga "D". En la figura 11 se muestra el bloque.

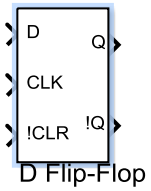

Figura 11. Bloque Filp-Flop tipo D.
Fuente: Herramienta ***Simulink*** de ***MATLAB***.

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

Este bloque se usa para accionar el eje de un servomotor y hacerlo girar desde 0 a 180 grados. Para realizar esta rotación el bloque deberá proporcionar, en el pin de salida seleccionado, valores comprendidos en este rango. Estos valores se ubican en la entrada del bloque para accionar el servo que estaría conectado en el puerto de salida en cuestión. También se pueden utilizar pulsos, como se mostró anteriormente, para seleccionar el ángulo deseado.

Son necesarios tener presente algunas observaciones para utilizar este bloque de ***"Simulink"*** con la placa "Arduino" y, son las siguientes:

- No utilizar el bloque en modo de simulación externo.
- 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 de ***Simulink*** es de 12 para el "***Arduino Uno***", y 48 para "***Arduino Mega 2560***".
- Si se utiliza el ***Arduino*** Uno en el modelo, el bloque ***PWM*** no puede utilizar los pines 9 y 10 cuando el modelo contiene bloques servo.
- En el caso de utilizar el ***Arduino Mega 2560***, el bloque ***PWM*** no podrá utilizar los pines 11 o 12 cuando el modelo contenga más de 12 bloques servo.

En la figura 12 se muestran el bloque y la caja de dialogo correspondiente al bloque de accionamiento del servomotor.

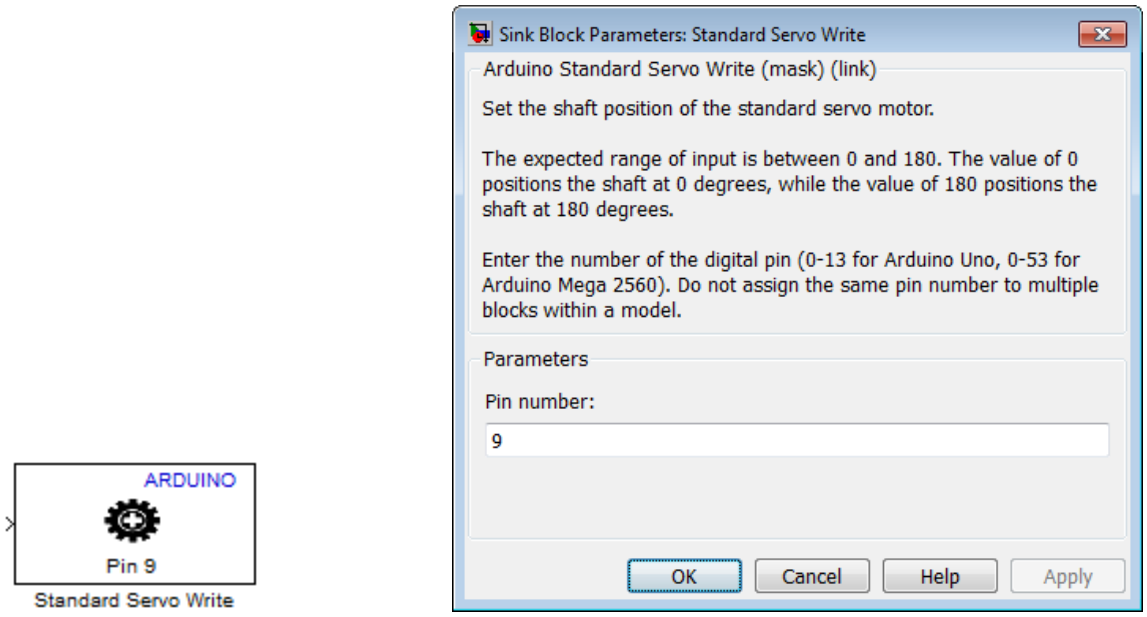

Figura 12. Bloque ***Arduino Standard Servo Write*** y su caja de diálogo.

Como se había Fuente: Software ***Simulink*** de "***MATLAB***".

Accionamiento del servomotor y señalización con combinación de dos dígitos

A continuación se describe la aplicación propuesta en el trabajo utilizando la herramienta ***"Simulink"*** y la placa de "Arduino Uno", como se había comentado. Se utiliza, además del servomotor, una lámpara 7 segmentos para señalizar el evento ocurrido y un interruptor "***reed"*** como ya se conoce.

La combinación para que se produzca el accionamiento del servo, se proporciona con el número de veces que se acerca el imán del interruptor "***reed"*** a este, lo que suministra los dos dígitos que forman la combinación de accionamiento. En la figura 13 se muestra el diagrama en bloques del sistema electrónico de accionamiento.

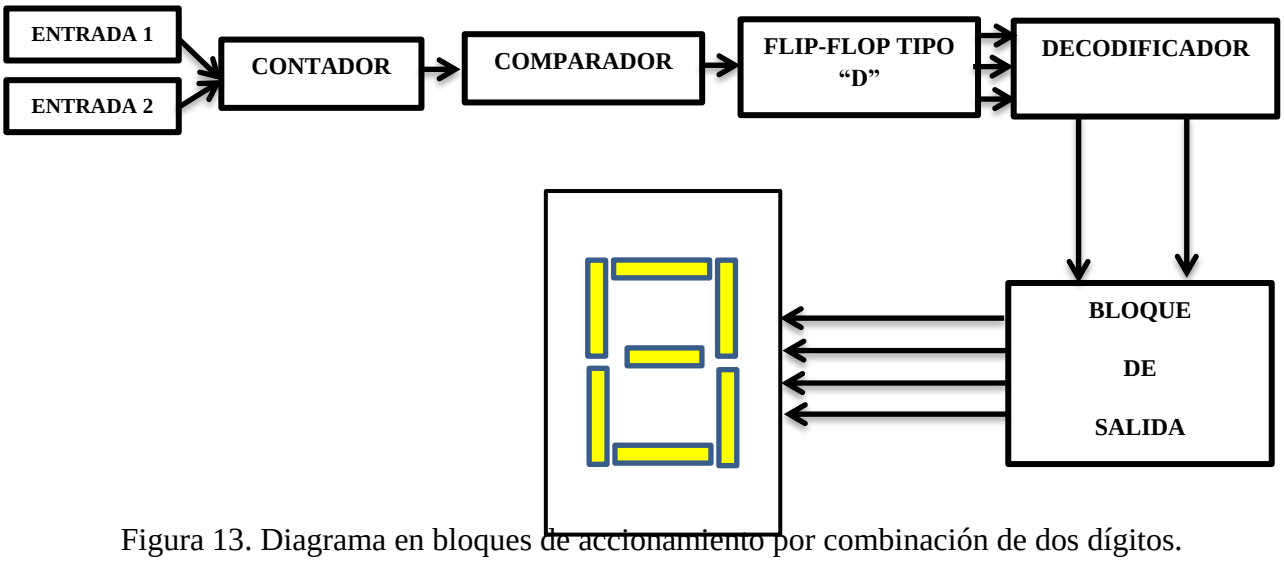

Figura 13. Diagrama en bloques de accionamiento por combinación de dos dígitos.

Fuente: los autores.

Los bloques de **entrada** 1 y 2 lo constituye el interruptor magnético y a una tecla, respectivamente. Se ha dedicado el pin 2 del ***Arduino*** al interruptor y el 8 a la tecla. Los tres contadores se conectan al interruptor y a la tecla. La salida de estos se acoplan a los comparadores, donde se sitúan los dígitos de la combinación, y la salida de estos últimos se conectan a los F-Fs. Los F-Fs. Se conectan al circuito combinacional encargado del accionamiento y la señalización. Luego la salida del decodificador se conecta al servo y a la lámpara 7 segmentos. En la figura 14 se muestra el modelo en ***Simulink*** de la aplicación propuesta.

La **entrada** correspondiente al bloque ***Arduino Digital Read*** del pin 2 conectada al interruptor, se acopla a las entradas de incremento de los 3 contadores. La otra entrada que va a la tecla unida al pin 8, se acopla a las entradas de ***reset*** de los dos primeros contadores y a las entradas de reloj de los dos F-Fs.

Los tres **contadores** se encuentran dispuestos de la siguiente forma: los primeros dos contadores se encuentran conectadas, sus entradas Inc. y ***reset*** a las entradas del esquema, como ya se dijo, y su salida a los comparadores donde se ubican los dos dígitos de la secuencia de accionamiento. La salida de conteo del primer contador se conecta a un comparador con operador "≥" con constante 6, en este caso (ver figura 14). El segundo contador, se conecta su salida de conteo a un comparador con operador "=" con constante 9. Mientras que el tercer contador, se conecta su salida "Hit" a la entrada del

decodificador directamente. Se observa que este último no tiene conectada su entrada de ***reset*** a las entradas del esquema (ver figura 14).

Los **contadores** deben de tener ciertas configuraciones para que respondan a las características de funcionamiento. El máximo conteo del contador 1 debe ser mayor o igual que la constante del comparador 2 más 1, en este caso mayor que 9 y por tanto se le designará el valor de 10. En el caso del contador 2 el máximo conteo debe ser igual a la constante del comparador 2. Por último, el valor máximo de conteo del contador 3 deberá ser la suma de la constante 1 y la constante 2, en este caso 15 (ver figura 14).

Esta variante utiliza dos **comparadores**, uno conectado a la salida del contador 1 y el otro al dos. El primer comparador (el de constante 6) conecta su salida a la entrada "D" del primer F-F y el segundo comparador (el de constante 9) a la entrada "D" del otro F-F. El tercer contador tiene conectada su salida "Hit" al inversor que va al bloque de salida acoplado al pin7 y su entrada Inc. a la entrada del interruptor magnético.

Las entradas de los dos **F-Fs.** se conectan a las salidas de sus correspondientes comparadores y sus entradas de reloj CLK, se conectan al bloque de entrada conectado a la tecla. Las salidas de estos dispositivos se acoplan a la entrada del circuito combinacional.

El bloque **decodificador** está constituido por una compuerta AND con la que se implementa la combinación, junto con los comparadores que contienen los dígitos que se combinan.

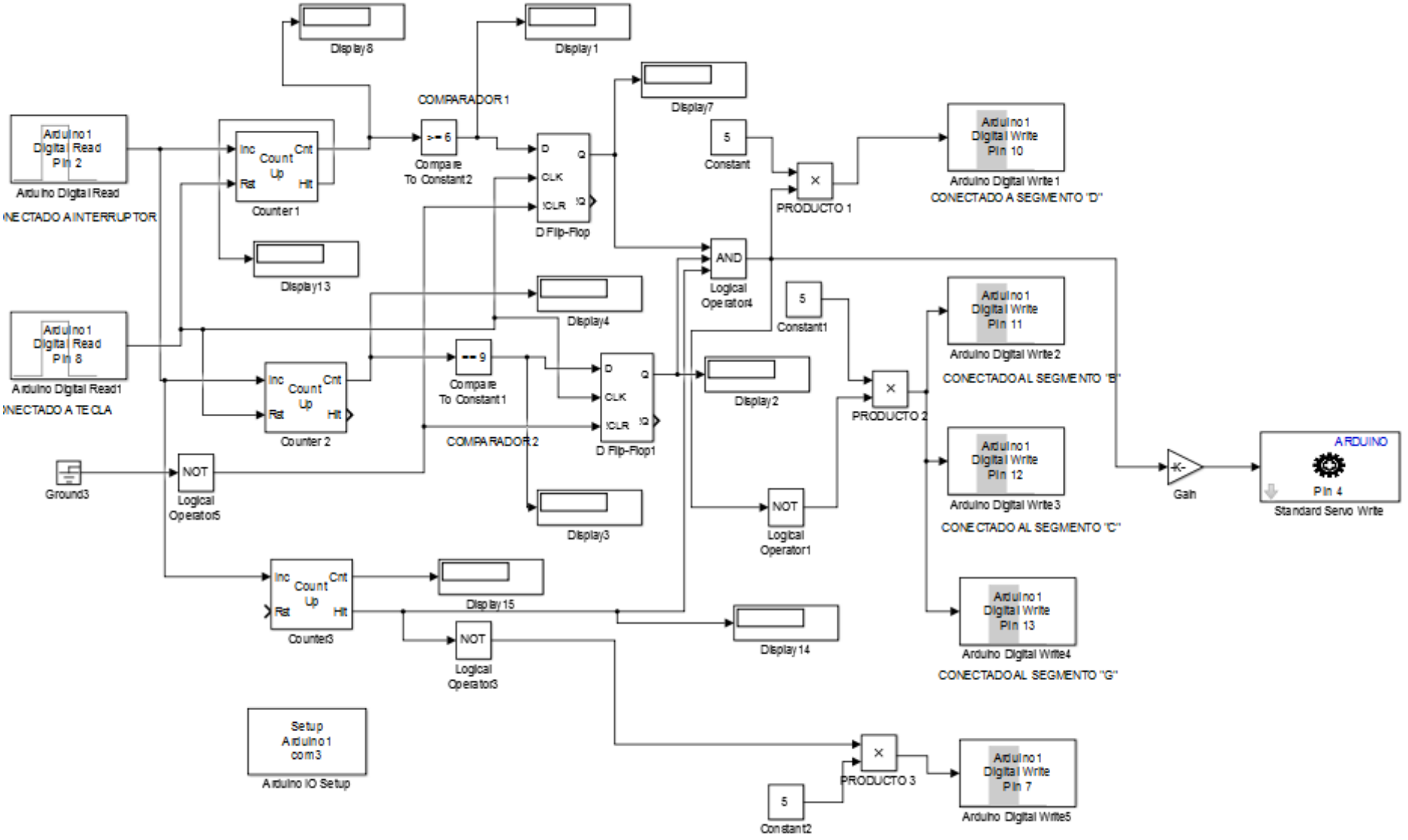
Display8
Display1
COMPARADOR1
Display7
Arduino1 Digital Read Pin 2
Arduino Digital Read
NECTADO A INTERRUPTOR
Inc Count Cnt Up Rst Hit
Counter1
>= 6
Compare To Constant2
D Q CLK IClR IQ
D Flip-Flop
5
Constant
PRODUCTO 1
Arduino1 Digital Write Pin 10
Arduino Digital Write1
CONECTADO A SEGMENTO "D"
Display13
AND
Logical Operator4
Display4
5
Constant1
Arduino1 Digital Read Pin 8
Arduino Digital Read1
NECTADO A TECLA
Inc Count Cnt Up Rst Hit
Counter 2
-- 9
Compare To Constant1
COMPARADOR2
D Q CLK IClR IQ
D Flip-Flop1
Display2
PRODUCTO 2
Arduino1 Digital Write Pin 11
Arduino Digital Write2
CONECTADO AL SEGMENTO "B"
Ground3
NOT
Logical Operator5
Display3
NOT
Logical Operator1
Arduino1 Digital Write Pin 12
Arduino Digital Write3
CONECTADO AL SEGMENTO "C"
-K-
Gain
ARDUINO
Pin 4
Standard Servo Write
Inc Count Cnt Up Rst Hit
Counter3
Display15
Display14
NOT
Logical Operator3
Arduino1 Digital Write Pin 13
Arduino Digital Write4
CONECTADO AL SEGMENTO "G"
Setup Arduino1 com3
Arduino IO Setup
PRODUCTO 3
5
Constant2
Arduino1 Digital Write Pin 7
Arduino Digital Write5

Figura 14. Modelo en ***Simulink*** del circuito de accionamiento del servo con combinación de dos dígitos. Fuente: Herramienta ***Simulink***.

El **bloque de salida**, formado por los elementos ***"Digital Write"*** de ***"Simulink"*** se encuentran conectado a los segmentos de la lámpara 7 segmentos. El bloque conectado al pin 10 se conecta al segmento "d" y los bloques conectados a los pines del 11 al 13, se conectan a los segmentos "b", c" y "g" respectivamente. La salida "Hit" del tercer contador se conectó al "punto" de la lámpara, a través del bloque conectado al pin 7 del ***Arduino***.

Explicación del funcionamiento del sistema de accionamiento y señalización para dos dígitos

Para realizar la explicación de esta variante, se tomará como ejemplo la combinación 69 como verdadera para el accionamiento y señalización de evento válido. Habrá que introducir al esquema los dígitos 6 y 9 por medio de la cantidad de acercamientos del imán al interruptor.

Primeramente se realizan los 6 acercamientos del imán al interruptor y el primer contador efectuará el conteo correspondiente presentando este dígito en la entrada del comparador 1. A su vez, el segundo y tercer contador están realizando el conteo de forma simultánea. En este momento la salida del comparador 1 pasa a "1" lógico el que se ubica en la entrada de dato del 1er. F-F. Seguidamente se pulsa la tecla por lo que se dará un pulso de reloj a ambos F-Fs. y a las entradas de ***reset*** de los contadores 1 y 2, no así al tercero, como se había comentado. Esto último hará que se limpien estos contadores y que pase el "1" a la salida del 1er F-F. El tercer contador habrá efectuado un conteo de 6 eventos y este es el dígito que tendrá a su salida.

Para introducir el dígito 9, se realiza esta cantidad de acercamientos del imán al interruptor. Esta cantidad de eventos también serán computados por todos los contadores. En el caso del primer contador también tendrá este digito en su salida, ya que el número máximo de conteo que tiene configurado es de 10. Esto hará que a la salida del comparador 1 de nuevo aparezca un "1", ya que el operador de este es de "≥" 6 y, que se tenga este valor lógico a la entrada del 1er. F-F. En el caso del 2do. contador se tiene el dígito 9 en su salida, lo que hace que la salida del comparador 2 pase a "1".En este momento, el 3er. contador ha llegado a la cuenta máxima de 15 (suma de las constantes de los comparadores 1 y 2) y su salida "Hit" pasa a "1", lo que completará la combinación correcta en la entrada del decodificador. Posteriormente cuando se pulsa la tecla por segunda vez, en las entradas de los F-Fs 1ro. y 2do. se encontrarán los "1" lógicos de los comparadores 1 y 2, respectivamente. Esto hará que se tenga la

combinación “11” a la salida de los F-Fs y, en las entradas de la compuerta AND. Es en este momento que se escribirá la “A” en la lámpara 7 segmentos y se accionará el servo.

En caso que se quiera cambiar la combinación de accionamiento solamente se tendrán que cambiar los dígitos de las constantes de los comparadores, observando las relaciones que tienen que existir entre: las constantes de los comparadores 1 y 2, el valor del máximo “Hit” y el valor máximo de conteo del 1er. contador respecto a la constante del comparador 2.

Segunda combinación del esquema de accionamiento dos dígitos

El diseño obtenido tiene la particularidad de poseer otra combinación para el accionamiento del servomotor. En esta variante de accionamiento y señalización de dos dígitos, existe otra combinación al introducir los mismos dígitos pero de forma invertida. Ahora la combinación de accionamiento es 96. Primero se acercaría el imán el número de veces del dígito mayor (9) y posteriormente un número de veces correspondiente al dígito menor (6). Se debe recordar que la condición establecida es que la constante del comparador 1 tiene que ser menor que la del comparador 2.

Al acercarse el imán al interruptor 9 veces, el 2do. contador presentaría este dígito en su salida, por lo que el comparador 2 ubica un “1” en su salida y en la entrada del 2do. F-F (ver figura 14). Cuando se pulsa la tecla, este “1” pasará a la salida del 2do. F-F y se resetean los contadores 1ro. y segundo. El tercer contador no tiene conexión en su entrada de ***reset***. Posteriormente se realizan los 6 acercamientos correspondientes al segundo dígito. En este momento tanto la salida del comparador 1 como el segundo tendrán en sus salidas un “1” y este nivel lógico también está en las entradas de ambos F-Fs. A continuación se pulsa la tecla por segunda y vez ambos valores lógicos pasarán a las salidas de los F-Fs. y, estarán en las correspondientes entradas de la AND. En el caso del 3er. contador, ya ha cumplimentado el conteo igual a la sumatoria de ambos dígitos, en este caso es 15, lo que hace que la salida “Hit” del contador pase a “1”. Todo lo anterior hará que las entradas de la AND estén en valor lógico de “1”, ocurriendo el accionamiento del servomotor y la señalización de la combinación correcta con una “A” en la lámpara 7 segmentos. La conexión de los bloques de salida con la lámpara 7 segmentos continúa siendo la misma.

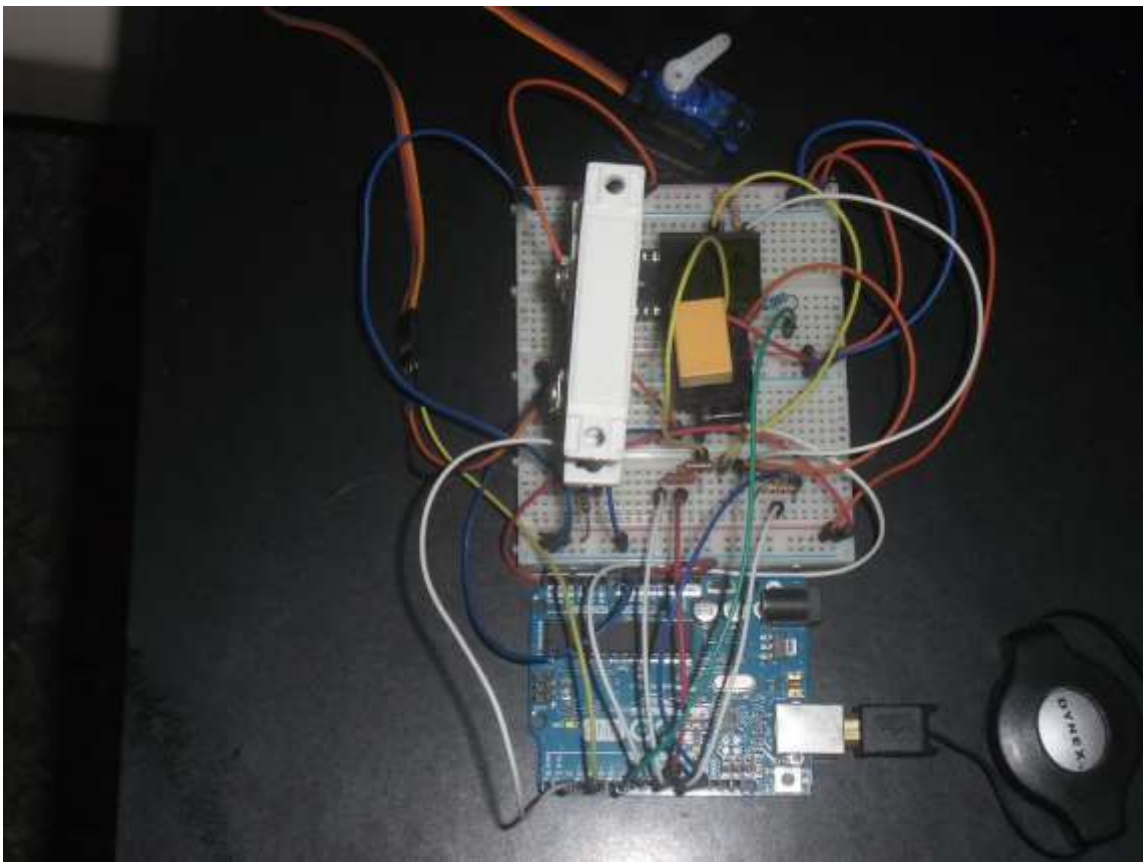

Figura 15. Imagen del montaje en ***board*** del circuito de accionamiento propuesto conectado al ***Arduino***.
Fuente: Los autores.

3. RESULTADOS

Para realizar la comprobación del funcionamiento del esquema de **accionamiento y señalización con dos dígitos**, se realizan varias maniobras para dicha comprobación, que se describen a continuación:

- Realizando el procedimiento correcto de introducción de los dígitos, utilizando la tecla, se introdujeron varios dígitos diferentes a los correctos en la secuencia, no ocurriendo el accionamiento del servo,
- Se introdujo la secuencia invertida de la combinación (96) utilizando la tecla, comprobándose su efectividad,
- Realizando el procedimiento descrito para la introducción de la combinación invertida de los dígitos, se hicieron acercamientos diferentes a los dígitos de la combinación, sin ningún resultado,
- Se realizaron varias secuencias de números acercamientos sin la utilización de la tecla, sin ocurrencia del accionamiento del sevomotor.

Luego de las comprobaciones anteriores, se pudo corroborar que solamente con las combinaciones correctas de ambos dígitos ocurría el accionamiento del servomotor y la señalización, en la lámpara 7 segmentos, de la combinación válida.

4. CONCLUSIONES Y RECOMENDACIONES

En el trabajo se presentó, de forma detallada, el diseño de un esquema de accionamiento de un servomotor y su señalización en una lámpara de 7 segmentos. El diseño se implementó utilizando una placa de ***Arduino Uno***, programada con la herramienta ***Simulink*** lo que permitió simularlo en la misma placa, pudiéndose comprobar y precisar el diseño de manera dinámica y de forma muy simple. Esto también contribuyó a la realización muy rápida de dicho diseño.

El esquema consistió en una especie de llave electromagnética de dos dígitos que puede accionar el servo, y su señalización en una lámpara 7 segmentos, con la introducción de dos combinaciones solo cambiando los mismos dígitos de lugar en la secuencia. Este esquema se pudiera emplear en otras aplicaciones. Esta implementación es de muy bajo costo y con un consumo mínimo de recursos de la placa.

En el trabajo también se mencionó, de forma muy breve, las diferentes herramientas que se utilizan para la programación gráfica del ***Arduino***.

En la explicación de la implementación del esquema, se brinda una guía que muestra la forma de programar la placa ***Arduino*** con la herramienta S***imulink*** y obtener un modelo, simple en este caso. También muestra la posibilidad de su simulación.

Al trabajo se le pudieran sugerir las siguientes **recomendaciones:**

Comprobar con qué rapidez el esquema capta la introducción de la combinación, concibiendo una forma de aumentar la velocidad de dicha introducción de los dígitos.

Programar el ***Arduino***, empleando su lenguaje de programación en su entorno, así como de otras herramientas gráficas de programación que evalúe, en este caso: la velocidad máxima de introducción de la secuencia de ambos dígitos, el consumo de recursos y el consumo de potencia. Y que después se compare con el esquema propuesto en este trabajo.

REFERENCIAS

Banzi, M. (2011). *Getting Started with Arduino.* Sebastopol: O'Reilly books.

Carlos, R. N. (2016). *El Primer Libro de Netduino 2 en español.* USA : Middletown, DE.

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

F., F. (2012). *Estudio, diseño e implementación de módulos deventrenamiento sobre plataforma Arduino para el Laboratorio de Microcontroladores de la Facultad de Ingeniería Electrónica de la Universidad Tecnológica Israel.* San Francisco, California, USA: Universidad de Israel.

Future Electronics. (n.d.). *Servo Motors Control & Arduino.* Future Electronics.

Karris, S. T. (2006). *Introduction to Simulink with Engineering Applications.* USA: Orchard Publications.

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

M., R. (2014). Herramientas Gráficas de Programación para Arduino. *II Jornadas Murcia Lan Party.* (p. 99). Murcia, España.: Centro Social Universitario.

Manual Satel. (2009). *Detector de Vibración con Contacto Magnético.* Gdansk, Polonia: Gdansk.

Manual SCHMERSAL. (n.d.). *Interruptores Magnétios Reed.*

Martín B., D. R. (2013). *Control de Posición de un Balancín con Arduino.* Valladolid, España.: Universidad de Valladolid. .

Mathworks. (2013, Agosto 13). "MATLAB" 2013b (8.2.0701). EEUU.

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.