

Monitoreo volcánico usando plataformas Arduino y Simulink

Volcanoes monitoring using Arduino and Simulink platforms

Edwin Vinicio Altamirano-Santillán¹
Geovanny Estuardo Vallejo-Vallejo²
Juan Carlos Cruz-Hurtado³

Recibido: mayo 31 de 2016
Aceptado: septiembre 21 de 2016

Resumen

El monitoreo de la actividad volcánica es una preocupación importante en aquellos países con alto grado de sismicidad. El principal inconveniente es el alto costo de la instrumentación requerida para vigilar el comportamiento de los volcanes, lo cual limita el acceso a información actualizada que permita realizar actividades de prevención. En este trabajo se presenta la implementación de la adquisición de dos variables asociadas al monitoreo volcánico, usando un Arduino y la herramienta Simulink de Matlab. Se implementó un medidor de inclinación o movimiento, que permite medir las deformaciones del edificio volcánico y las posibles sacudidas que pudieran existir de su cono, entre otras aplicaciones. Igualmente se implementó un medidor de temperatura con una termoresistencia PT100. La experimentación permitió comprobar la funcionalidad de los medidores de inclinación y temperatura, arrojando resultados satisfactorios y con errores relativos adecuados. Lo anterior permite concluir que el empleo de Simulink para la programación del Arduino posibilita la simulación y evaluación de los modelos, directamente sobre la placa de Arduino durante el proceso de diseño, con las ventajas que brinda esta alternativa en el ahorro de tiempo y simplicidad en la programación del dispositivo.

Palabras clave: arduino, hardware libre, Simulink, sistemas de monitoreo volcánico, sismología.

Abstract

The monitoring of volcanic activity is a major concern in those countries with high degree of seismicity. The main drawback is the high cost of instrumentation required to monitor the behavior of volcanoes, which limits access to updated information that allows carrying out prevention activities. This work presents the implementation of the acquisition of two variables associated to volcanic monitoring, using an Arduino and the Simulink tool of Matlab. A tilt or movement meter was implemented, which allows measuring the deformations of the volcanic building and the possible jolts that could exist of its cone, among other applications. In the same way, a temperature meter with a heater-resistance PT100 was also implemented. The experimentation allowed verifying the functionality of both, the inclination and temperature meters, throwing satisfactory results and with adequate relative errors. This allows concluding that the use of Simulink for the programming of the Arduino allows both the simulation and evaluation of the models, directly on the Arduino board during the design process, with the advantages offered by this alternative in saving time and simplicity in the device programming.

Keywords: arduino, free hardware, Simulink, volcanic monitoring systems, seismology.

¹ Ingeniero en Electrónica y Computación, Máster en Ciencias, Escuela Superior Politécnica del Chimborazo, Ecuador. E-mail: edwin.altamirano@esPOCH.edu.ec

² Licenciado en Informática Aplicada, Máster en Ciencias, Escuela Superior Politécnica del Chimborazo, Ecuador. E-mail: geesvava@gmail.com

³ Ingeniero Eléctrico, Doctor en Ciencias Técnicas, Instituto Superior Politécnico José A. Echeverría, Habana, Cuba. E-mail: juankacruzhurtado@gmail.com.

1. Introducción

El hombre siempre se ha interesado por explicarse y predecir los fenómenos de su entorno, fundamentalmente los que constituyen peligro para su vida y para los recursos materiales que lo benefician, como: la concentración de gases tóxicos, el caudal de los ríos, la concentración de material radioactivo y el comportamiento volcánico, entre otros (Carreño-Bodensiek, 2010; Fernández-Morales, & Duarte, 2012; Martínez-Ovalle, Reyes-Caballero, & González-Puín, 2013). En este hemisferio existen varios países donde el evento volcánico es una verdadera amenaza, como son los casos de EEUU, México, Chile, Haití, el Salvador, Ecuador, Nicaragua y Colombia, por mencionar los que han sido más castigados por las erupciones volcánicas.

La vulcanología es una ciencia bastante joven, solo hasta 1912 el científico Thomas Jaggar constituyó el primer observatorio vulcanológico en Estados Unidos en la isla de Hawai (López-Mendoza, Molina-Arévalo, & Vaides-Ramírez, 2007). El estudio de estos fenómenos implica la implementación de sistemas de monitoreo para los volcanes, los cuales requieren instrumentación altamente sofisticada y costosa, que no está al alcance de todos los países latinoamericanos.

Dentro de las magnitudes vulcanológicas a medir se encuentran: la concentración de gases volcánicos como CO₂ y gases ácidos, temperatura, sismicidad, deformación del edificio volcánico y el potencial espontáneo (Valdés-Gonzales, & Puente-Espinosa, 2008; López-Mendoza et al., 2007). Adicionalmente, dentro de las técnicas de vigilancia volcánica se encuentran: la vigilancia visual y el monitoreo: sísmico, geodésico, geoquímico y la medición de temperatura (Quaas-Weppen, González, Guerra, Ramos, & de la Cruz-Reyna, 1995; Valdés-Gonzales, Puente-Espinosa, & Osorio-Chong, 2008).

El monitoreo geodésico y la medición de temperatura son dos de las técnicas más utilizadas en vulcanología y fueron las seleccionadas para los

diseños propuestos en este trabajo. El monitoreo geodésico consiste en la medida de las deformaciones que la estructura volcánica sufre debido a los esfuerzos que se ejercen desde el interior del volcán, por el material magmático al salir. La técnica consiste en la instalación de varios sensores en diversos puntos de observación, especialmente sensores de inclinación (López-Mendoza et al., 2007; Alcaldía de Albacete, sf). La Medición de Temperatura en un volcán no solo se limita a las cercanías del mismo, sino que se realiza en lagos, rocas, suelo y lava. Para medir temperaturas relativamente altas como las del suelo y lava, se utilizan termopares y termo-resistencias, ya que son dispositivos con sondas protegidas que permiten realizar estas medidas sin dañarse (López-Mendoza et al., 2007).

Una alternativa interesante para desarrollar instrumentación vulcanológica la constituye el empleo de sistemas basados en microcontroladores, como la plataforma de hardware libre de Arduino y la herramienta Simulink de Matlab, que permiten el diseño y simulación de un sistema de monitoreo, en este caso volcánico. Sin embargo, la literatura especializada tiene pocos reportes de este tipo de sistema de adquisición con esta plataforma. Se presentan sistemas de detección y monitoreo sísmico en (Vargas, Rodríguez, & Otero, 2013). También existen implementaciones similares asociadas a plataformas parecidas al Arduino utilizando diferentes tipos de microcontroladores PIC (López-Mendoza et al., 2007) y usando tarjetas de adquisición de datos (Huamán, 2001). También se han desarrollado más recientemente las redes de sensores inalámbricos para el monitoreo volcánico (Werner, 2006; Ariosto, 2008; Peng, 2009; Lopes, 2014; Blanco-Velandia, & Pérez-Castillo, 2012).

El Arduino es una plataforma de hardware y software libre muy factible de usar en diseños electrónicos para muy diversas aplicaciones, fundamentalmente en las de adquisición y procesamiento de señales de sensores. A pesar de su potencial, no se tienen muchos reportes de esta plataforma programada con Simulink, asociadas

a variables que podrían estar relacionadas con el monitoreo volcánico (Vargas, Rodríguez, & Otero, 2013). Las implementaciones con Arduino reportadas en la literatura utilizan el lenguaje de programación de esta plataforma, que requiere tener conocimiento de este software, con la desventaja de que no permite tener una idea, a priori y de forma fácil, de la topología circuital del diseño completo.

Actualmente existen varias herramientas gráficas de programación para Arduino como son: el Minibloq, el Ardubloq, el Amici, el Modkit, el VirtualBreadBoard, el Fritzing, el Physical Etoys, el S4A (Scratch), el Labview, el Rhino+Firefly, el MyOpenLab, entre otras. Estas herramientas lo que persiguen es la programación, la monitorización del esquema implementado en microcontrolador y la obtención del prototipo físico. En el caso de las que tienen el objetivo de programar el microcontrolador, como el Arduino en este caso, presentan un entorno con una programación gráfica y descriptiva muy simple, adecuada para la enseñanza en los primeros niveles educativos. Estas herramientas de programación dificultan formarse una idea, a primera vista, del circuito topológico. Otra posibilidad la constituyen los ambientes de MATLAB y Simulink, integrados en una sola identidad, pues son instrumentos muy útiles para programar el Arduino sin necesidad de manejar su lenguaje de programación, así como para analizar, simular y revisar los modelos diseñados.

En vista de lo anterior, el objetivo del presente trabajo es describir el diseño e implementación de dos aplicaciones, que podrían estar asociadas a la adquisición de dos magnitudes físicas rela-

cionadas con el monitoreo volcánico, en las que se utiliza el hardware Arduino programado con la herramienta Simulink. También se presentan la simulación y las mediciones de cada uno de los esquemas propuestos, utilizando este mismo software de programación.

A continuación se realiza una breve descripción de dos técnicas de monitoreo de magnitudes volcánicas relacionadas con los dos diseños que se toman como ejemplo para explicar la programación y simulación del Arduino con el Simulink. Luego se muestran el diseño y la comprobación del funcionamiento de un medidor de inclinación y de uno de temperatura, utilizando el soporte Arduino-Simulink. Los resultados permiten concluir que el uso de esta combinación posibilita la simulación y verificación de los modelos en Simulink, directamente sobre la placa de Arduino durante el proceso de diseño, con las ventajas que brinda esta alternativa en el ahorro de tiempo y simplicidad en la programación del dispositivo.

2. Materiales y métodos

2.1 Diseño propuesto

Se implementaron dos aplicaciones que pueden asociarse a la adquisición de dos magnitudes físicas relacionadas al monitoreo volcánico, en las que se muestra el uso del Arduino programado con Simulink. En este caso se decidió desarrollar un Medidor de Inclinación y un Medidor de Temperatura con Dispositivo Termo-Resistivo PT 100. En la Figura 1 se muestra un diagrama en bloques descriptivo muy simple de la arquitectura de adquisición de las dos variables.

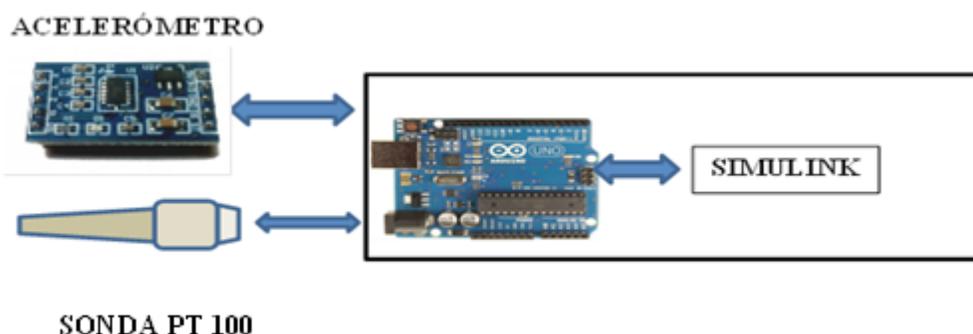


Figura 1. Esquema funcional en bloques que describe las implementaciones propuestas.

En el diagrama de la figura 1 se observa que, tanto el acelerómetro como la sonda PT100 usados para el medidor de inclinación y de temperatura, respectivamente, se conectaron de forma conveniente a los pines de entrada/salida de la placa Arduino y, esta última se acopló, a través del puerto USB, a la PC donde corre el Simulink.

Existen dos tipos de Hardware de Adquisición (HA) utilizables en el monitoreo volcánico destinadas a medir, procesar, almacenar y transmitir los datos. Estos HA dependen de: la cantidad y tipo de parámetros a monitorear, las condiciones del terreno, el nivel de actividad del volcán y de los recursos con que se cuente.

El HA más simple es el que tiene una topología cableada, constituida por: un grupo de sensores, un bloque amplificador, uno de acondicionamiento y con el de adquisición propiamente dicho. La otra variante consta de estos mismos bloques, además de un bloque de transmisión inalámbrica. En la actualidad esta última variante es la más utilizada en el monitoreo volcánico, implementada a través de una red de sensores, debido a las bondades que presenta al ser un sistema de los llamados desatendidos, ideal para estas aplicaciones, (Werner, 2006; Ariosto, 2008; Peng, 2009; Lopes, 2014; Blanco-Velandia, & Pérez-Castillo, 2012).

A continuación se hace una breve descripción de los bloques que constituyen el HA más simple:

Etapa de sensores: Encargada de convertir la variable física en una señal eléctrica de tensión o de corriente (López-Mendoza et al., 2007).

Etapa multiplexora y amplificadora: Generalmente constituida por un multiplexor y un amplificador para que, además de amplificar, se pueda conectar cada sensor con el bloque siguiente. El bloque tendría un único nivel de amplificación o con control automático de ganancia según el requerimiento de los sensores. Es la etapa encargada de proporcionar a la señal un nivel que asegure el trabajo del convertidor Análogo a Digital (CAD).

Bloque acondicionador de señal: encargado de desplazar el nivel de tensión de CD de la señal y los rangos que aseguren el trabajo del CAD.

Bloque de adquisición y procesamiento de datos: En este bloque se encuentra el CAD, por lo que es el encargado de brindar los datos en formato digital. Cuenta con dispositivos controladores, o una PC, con unidades de procesamiento y almacenamiento de datos. Estos datos pueden ser analizados por software especializados o transmitirse remotamente (López-Mendoza et al., 2007).

2.2 Metodología

Los métodos de investigación utilizados fueron el teórico y el empírico, fundamentalmente. En el caso del teórico, permitió establecer la sinergia existente entre el hardware Arduino y el software Simulink para su programación. En este sentido se recurrió al análisis y síntesis, específicamente al análisis hipotético-deductivo e histórico-lógico, para poder establecer las relaciones esenciales y cualidades fundamentales en el soporte de diseño Arduino-Simulink y poder generalizar los resultados. En el caso del método empírico permitió corroborar la hipótesis de las facilidades que reporta la utilización del Simulink frente a otras herramientas gráficas de programación de este tipo de plataformas hardware. En este caso se utilizó el método de la medición y la simulación.

Los materiales utilizados para realizar el diseño y la implementación fueron: una placa de Arduino Uno genérica, un módulo acelerómetro para Arduino tipo MMA 7361 y una sonda de temperatura tipo PT100. El software utilizado para realizar la programación de la placa, fue el Simulink asociado al MATLAB R2013b (8.2.0.701) de 32 bit de agosto 13 del 2013.

Las variables de estudio se encuentran asociadas a los valores de resistencia de una sonda PT100 y las tensiones de los ejes X, Y y Z del acelerómetro. El valor de las simulaciones de estas variables se contrastó con los valores que aparecen en los manuales de ambos dispositivos (Arian, 2015a; Guarnizo, 2013; Freescale Semiconductor, Technical Data., 2008).

2.3 Materiales

La plataforma arduino

Se trata de una plataforma de hardware y software flexible y fácil de utilizar en muy diversas aplicaciones. Esta puede conectarse al mundo exterior a través de sensores y actuadores, mediante las dife-

rentes entradas y salidas que presenta. El microcontrolador (Atmel AVR) de la placa se programa con el "Arduino Programming Language" (basado en el Wiring1) (Hay, & Wenbin, 2010) y el "Arduino Development Environment" (basado en Processing2). Otros lenguajes de programación que se han utilizado para este dispositivo son: Visual Basic, LabView y C++.

Entre las ventajas importantes del uso de Arduino se tienen: el precio de comercialización que es de unos 20 USD; el software multiplataforma. Además puede correr en sistemas operativos como: Windows, Macintosh OSX y GNU/Linux; debido a la arquitectura, tanto de software como de hardware, el conocimiento y el tiempo de aprendizaje de la plataforma son flexibles para los usuarios (Vargas, Rodríguez, & Otero, 2013).

Existe una variedad grande de modelos de placas, según la creatividad de los desarrolladores. Las mejoras de los modelos estriban en el aumento de la cantidad de memoria y número de puertos de entrada/salida, entre otras. En las aplicaciones que se describen a continuación se utilizó la placa de Arduino Uno, como soporte de hardware libre y el software Simulink de Matlab para el diseño, programación y simulación del microcontrolador.

La plataforma de desarrollo Arduino uno consiste en una placa que tiene 14 pines de entradas/salidas digitales y de estas, seis se utilizan como salidas de modulación de ancho de pulsos (PWM) y, 6 pines de entradas analógicas. Además presenta, un botón de "reset", una conexión USB para conectarse con la PC y un conector para su alimentación de forma externa. Esta plataforma se basa en el microcontrolador Atmega328 (Martín-Ballesteros, & Del Rio-Carbajo, 2013).

Herramienta Simulink de Matlab

Esta herramienta funciona mediante un entorno de programación gráfico, cuyas funciones se representan por bloques. Esto hace muy sencillo conformar sistemas sin necesidad de emplear

lenguajes complejos de programación y facilita la comprensión de la funcionalidad del esquema con un golpe de vista. Presenta un entorno de programación de más alto nivel que el propio Matlab. Esta herramienta genera archivos .mdl (modelo). Al ejecutarse el modelo se genera un código "C" que el ordenador reconoce y ejecuta. Además, la herramienta admite el diseño y simulación del sistema modelado y, la prueba y verificación del mismo en el hardware en cuestión. Esto ofrece la posibilidad de conectar el modelo con el hardware y comprobar, en tiempo real, su funcionamiento. Es esta cualidad la que se aprovecha en los diseños con el Arduino que aceleran el prototipado y la implementación del diseño en cuestión (Martín-Balasteros, & Del Rio-Carjajo, 2013).

Existen dos bibliotecas que permiten trabajar conjuntamente con Arduino y Simulink y que se instalan desde el sitio de Mathworks, gratuitamente, para poder utilizarse. En este trabajo se ha elegido la biblioteca "Arduino IO Library".

El bloque Setup, ver figura 2, es necesario para poder establecer la conexión con la placa a través del puerto USB. En este bloque se configura el tipo de placa de Arduino que se esté utilizando y el puerto a la que está conectada. Los bloques que se observan en las Figuras 2 y 7, se utilizarán para

los diseños propuestos. Una vez que se obtienen los resultados perseguidos con el diseño, los bloques son exportados a la placa, pudiendo funcionar de forma autónoma, alimentando la placa externamente.

3. Resultados y discusión

3.1 Implementación del medidor de inclinación

En esta sección se describe el diseño, simulación e implementación de un medidor de inclinación o movimiento, que sirve para medir las deformaciones del edificio volcánico y las posibles sacudidas que pudieran existir del cono volcánico, entre otras aplicaciones (Guarnizo, 2013).

En este diseño se utilizan los bloques funcionales de la biblioteca Arduino IO Library. Para seleccionar los pines de control del acelerómetro se usan los bloques de salida digitales del Arduino, en este caso: Arduino Digital Write (ADW). Se utilizan dos, uno para el Pin SL y otro para el Pin SG. El bloque ADW que se conecta al SL, se sitúa en un nivel alto para garantizar el trabajo normal del acelerómetro. El otro ADW se conectaría a SG para poder seleccionar la sensibilidad del dispositivo. El dispositivo se alimenta de la placa Arduino con la tensión de 3.3V.

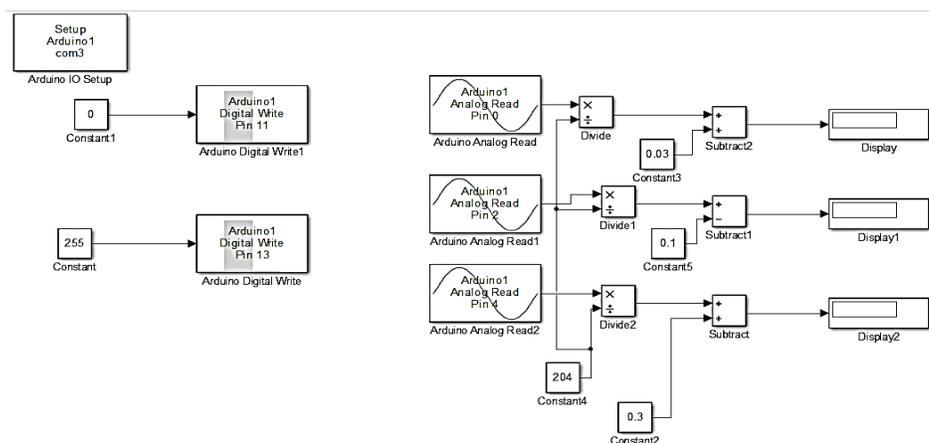


Figura 2. Diagrama en bloque del modelo en Simulink del medidor de inclinación.

En la figura 2 se muestra el diagrama en bloques del modelo de la implementación en Simulink. Como se observa, las salidas de los dos bloques ADW van a los pines de salida del Arduino 11 y 13 y colocaría un 0V y un nivel alto en los pines 11 y 13, respectivamente. Para la adquisición de las salidas X, Y y Z del acelerómetro se utilizaron los bloques Arduino Analog Read. En los bloques se han seleccionado las entradas analógicas de la placa Arduino: pin 0, pin 2 y pin 4, y en estas se

conectan las salidas X, Y, y Z respectivamente. En la Figura 3 se muestra el esquema físico de conexión del acelerómetro y el Arduino, que se corresponde con el esquema en bloques del modelo mostrado en la Figura 2. Se debe señalar que en el modelo desarrollado se tiene en cuenta la posibilidad de calibración de cada eje del acelerómetro, a través de los bloques "sustractores" y "constantes" que se conciben en el modelo.

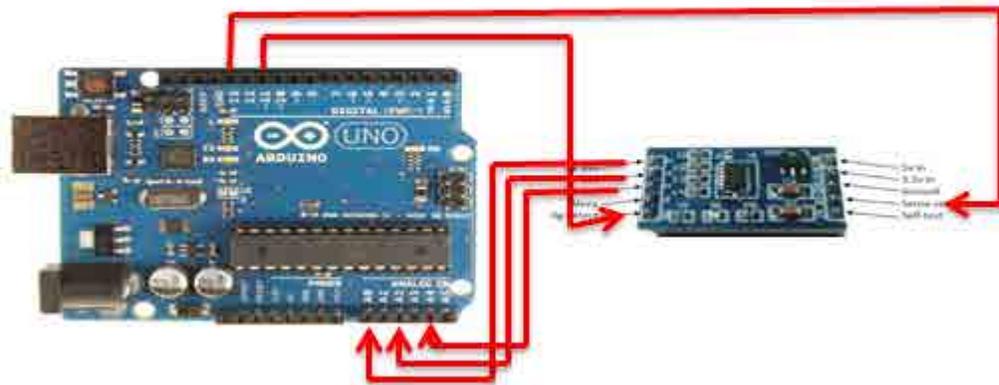


Figura 3. Esquema físico de conexión del "Arduino" y acelerómetro MMA7361.

La comprobación del funcionamiento del medidor de inclinación se realiza leyendo los valores de salida de los displays del modelo, para una posición horizontal del acelerómetro y una posición vertical del dispositivo, comprobándose la aceleración, estática en este caso (Guarnizo, 2013). Se debe tener en cuenta que el display superior se corresponde con el eje "X", el display 1 con "Y" y el display 2 con el eje "Z".

En la Figura 4 se muestra la captura de pantalla del modelo Simulink para el caso del acelerómetro en posición horizontal, sobre una mesa, con su cara superior hacia arriba. En dicha posición, las tensiones son: $X = 1.643 \text{ V}$, $Y = 1.64 \text{ V}$ y $Z = 2.501 \text{ V}$, lo que se corresponde con los valores dados por el fabricante (Guarnizo, 2013; Freescale Semiconductor, Technical Data, 2008).

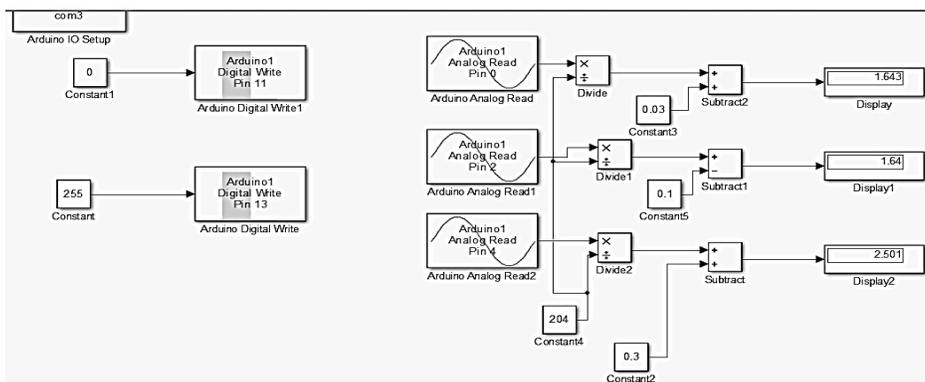


Figura 4. Captura de pantalla de la corrida de Simulink con acelerómetro horizontal.

En la Figura 5 se muestra la simulación del modelo de Simulink para el caso que el acelerómetro se encuentre en posición vertical, habiéndose girado sobre el eje X respecto a la posición anterior. En este caso se tiene que el eje $X = 1.633V$, $Y = 0.851$ y $Z = 1.687V$, lo cual es debido a que en la nueva posición solo existieron giros de 90° en los ejes Y y Z con respecto a la posición de la Figura 4. Es decir,

solo existió una rotación en el eje X. Este resultado se puede comprobar en el manual del MMA7361 tomado como referencia (Guarnizo, 2013). El acelerómetro se puede calibrar haciendo una serie de medidas de ángulo de inclinación de cada eje contra la tensión de salida correspondiente. En el modelo de Simulink se brinda la posibilidad de hacer dicha calibración.

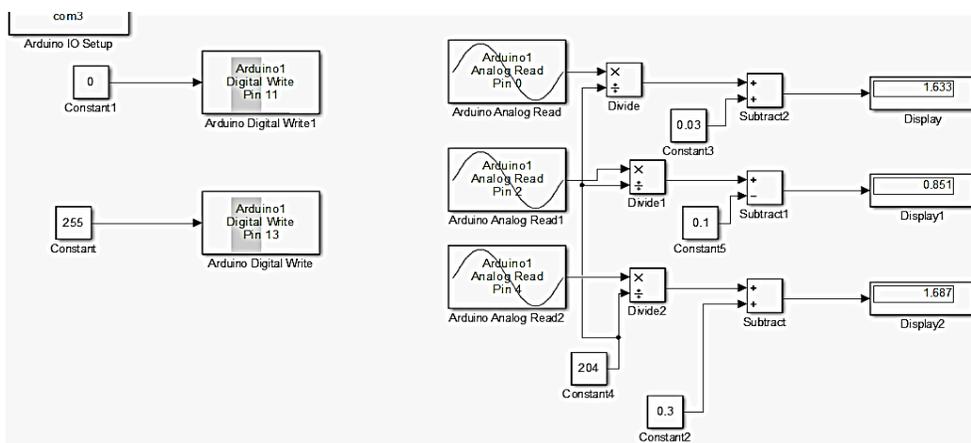


Figura 5. Captura de pantalla de la corrida de Simulink con acelerómetro vertical.

3.2 Medidor de temperatura con dispositivo PT100

En este caso se pretende mostrar la posibilidad de implementar un medidor de temperatura con una PT100 usando el Arduino, con un modelo simple en Simulink. Para implementar un medidor como tal, se tendría que calibrar la PT 100 o incorporar,

de alguna forma al modelo, la tabla de resistencia contra temperatura para poder brindar la lectura de temperatura (Arian, 2015b). La PT 100 es un sensor de temperatura que consiste en un alambre de platino que a $0^\circ C$ tiene una resistencia de 100 ohm, la cual aumenta con la temperatura. Este incremento es creciente pero no es lineal y, se ten-

dría que contar con las tablas de temperatura contra la resistencia $R(t)$ de la PT100 para poder tener la temperatura exacta (Arian, 2015b). Las PT100 industriales, como la utilizada, están encapsuladas dentro de un tubo de acero inoxidable o vaina, quedando el elemento sensible a la temperatura en un extremo mientras que en el otro están los cables de conexión. Este encapsulado lo hace conveniente para la aplicación de monitoreo volcánico de temperatura.

El esquema implementado es parecido a la conexión de 4 hilos (Arian, 2015B). Se utiliza el sencillo divisor de tensión, con una resistencia de valor conocido y la particularidad de la medición indirecta de la corriente que circula por dicho divisor y, con ella se determina la resistencia que presenta la PT100 para cada temperatura. En este caso se ha supuesto que la resistencia que compone el divisor tiene una variación muy pequeña con la temperatura y, que por las entradas analógicas del Arduino no circula una corriente apreciable. En la figura 6 se presenta el esquema de conexión de la sonda PT100 con el Arduino.

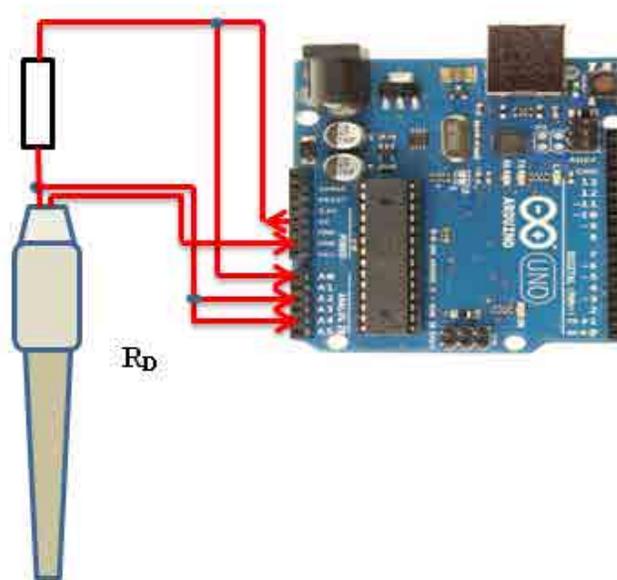


Figura 6. Esquema de conexión de la PT100 con el Arduino.

Se conectaron las entradas analógicas del Arduino a los extremos tanto de la resistencia conocida, como de la sonda PT100 ($R(t)$). La resistencia conocida R_D seleccionada es de un valor tal que se garantice la circulación de 1 mA por el divisor de tensión, siguiendo el rango recomendado por el fabricante (Arian, 2015b). El valor de R_D es de $4,6K\Omega$, lo que garantiza la gama de la corriente de excitación que aconseja el manual.

El modelo en Simulink que se corresponde con el esquema eléctrico de la Figura 6, se muestra en la Figura 7. Se tiene que a través de los bloques de lectura analógica del Arduino se obtienen las caídas de tensión de R_D y de $R(t)$. Por el Pin cero y el Pin 2 se obtiene la caída de tensión por R_D y entre Pin 4 y GND se obtiene la caída de tensión por $R(t)$. Luego la diferencia de tensión por R_D se divide entre la resistencia conocida R_D y se obtiene la corriente por el divisor. Posteriormente la tensión por $R(t)$ se divide entre la corriente anteriormente

determinada y, se tiene el valor de $R(t)$ para el valor de temperatura a que estuviera sometida la sonda.

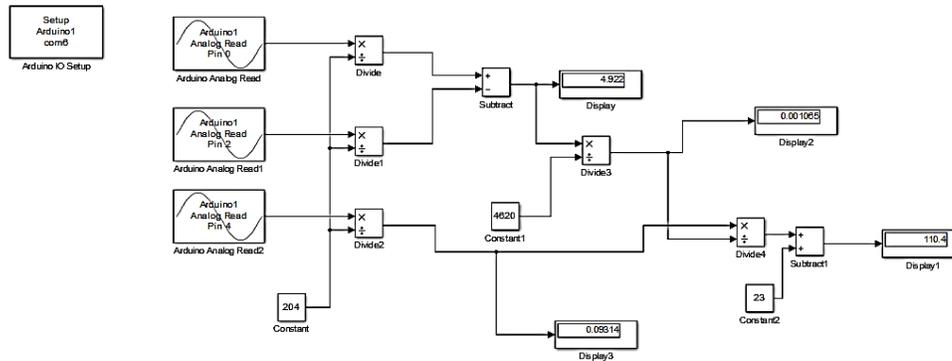


Figura 7. Diagrama en bloques del modelo en Simulink para el esquema circuital de la figura 6.

La funcionalidad del detector de temperatura se comprobó realizando variaciones discretas de temperatura, observando el cambio que se produce en el valor de la resistencia $R(t)$. En la figura 8 se muestra una captura de pantalla del modelo en Simulink cuando la sonda PT100 se encuentra

a temperatura ambiente (28°C). El valor en el display 2 se corresponde con la corriente por el divisor (1.064 mA) y, el del display 1 se corresponde con el valor de $R(t)$ de la PT100, que en este caso es de 110.5Ω .

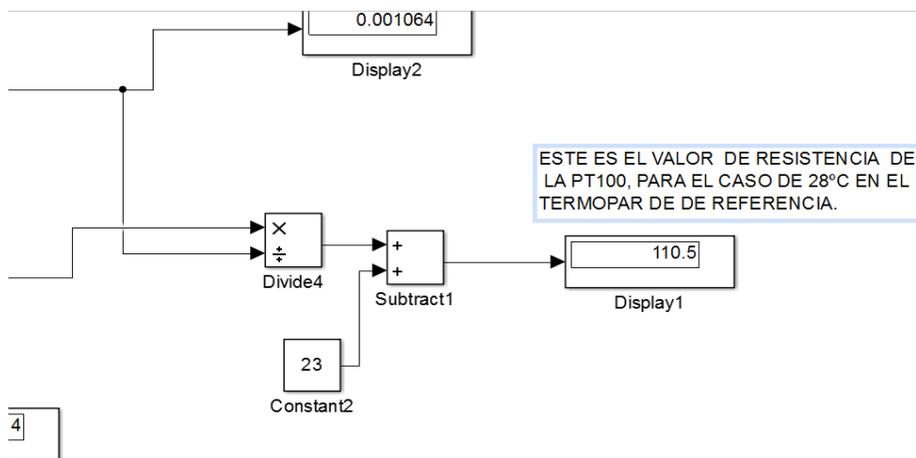


Figura 8. Captura de pantalla de modelo de la sonda en Simulink para temperatura ambiente.

En la figura 9 se muestra la captura de pantalla del modelo en Simulink del detector para el caso de un valor de temperatura superior al ambiente (41°C). Se observa que ahora el valor de $R(t)$ es su-

perior al de la Figura 8, en este caso de 115.1Ω . En este mismo esquema es posible mejorar la exactitud de la medida de $R(t)$, si se alimentara el divisor con una fuente de corriente conocida.

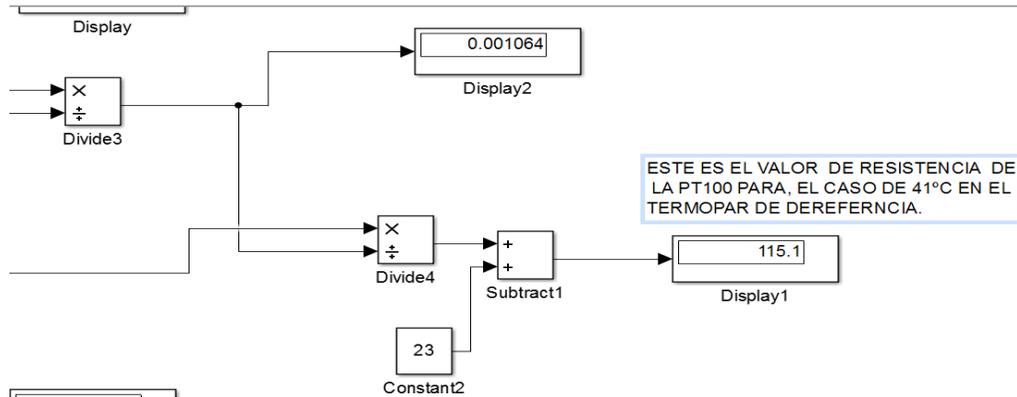


Figura 9. Modelo de la sonda en Simulink para temperatura superior al ambiente.

3.3 Resultado de los medidores de inclinación y temperatura

En el caso del medidor de inclinación se comparan los valores de tensión de cada eje, obtenidos en el modelo Simulink, corrido sobre el Arduino, con los correspondientes valores que muestran los manuales (Freescale Semiconductor, Technical Data, 2008; Guarnizo, 2013). En la Figura 10 se muestran las posiciones espaciales del módulo acelerómetro MMA7361, con los valores de tensión correspondientes de sus 3 ejes. Estos valores de tensión son los que se compararon con los de las Figuras 4 y 5. Los valores de la Figura 4 se corresponden con los de la imagen izquierda de la Figura 10 y, los de la Figura 5 con los de la posición espacial de la derecha.

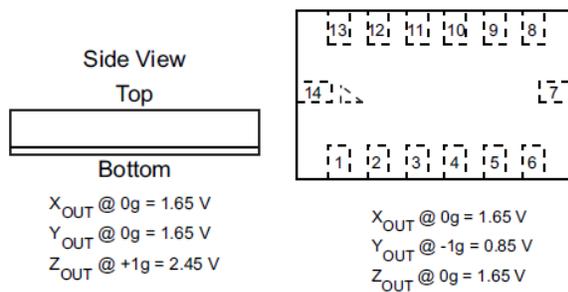


Figura 10. Posiciones relativas espaciales del acelerómetro MMA 7361 con los valores de tensión de sus ejes. (Freescale Semiconductor, Technical Data, 2008).

De la comparación de estas medidas se observa que el error relativo mayor ocurrido en la posición de la izquierda, es de 2.08% para el eje Z. En el caso de la posición de la derecha se tiene que el error relativo mayor es de 2.24%, al igual que para el eje Z. Se debe destacar que en esta implementación se brinda la posibilidad de calibrar el medidor, a través de los bloques sustractores y constantes, asociados a cada eje, que se observan en el modelo, ver figuras 4 y 5.

Para el medidor de temperatura, aunque el objetivo solo era el de describir su diseño e implementación con PT100 utilizando el soporte Arduino-Simulink, es importante comentar los resultados de las mediciones realizadas con este dispositivo. Los resultados de esta variable se compararon con la tabla temperatura en °C contra resistencia de la PT100 que aparece en el manual (Arian, 2015b).

Temperatura (° C)	R(t) Manual (Ω)	R(t) Simulink (Ω)	Error Relativo (%)
28	110.90	110.5	0.36
41	115.93	115.1	0.71

Tabla 1. Comparación de Resultados de medición de sonda PT100 y Simulink con su manual.

La comparación de la tabla de resistencias del manual de la PT100 con los valores resistivos obtenidos en el modelo de Simulink, ver figuras 8 y 9, para 28°C y 41°C, respectivamente, se indican en la tabla 1. Aunque el error relativo es bajo, este valor se puede reducir a través de los bloques "Constant 2" y "Subtract1" que aparecen en el modelo, los cuales brindan la posibilidad de la calibración del medidor.

4. Conclusiones

El trabajo presenta un enfoque interesante de la utilización del Simulink en la programación del Arduino, lo que permite también simular directamente sobre la placa de Arduino el modelo implementado. Esto además constituye una ventaja para el diseño y un ahorro de tiempo para la obtención del prototipo final.

La explicación de la metodología de las implementaciones propuestas se apoyaron en el diseño de un esquema de adquisición de variables físicas con Arduino, que pudieran concebirse para sistemas de monitoreo volcánico, lo que constituye una contribución de los sistemas de adquisición asociados a este tipo de monitoreo usando este soporte.

Se comprobó la funcionalidad de los medidores de inclinación y temperatura con PT100 contrastando las mediciones con los valores suministrados por los manuales de fabricante, arrojando resultados satisfactorios y con errores relativos adecuados. Dentro de las fuentes de error se encontrarían: la medición con dos hilos pues no hubo acceso físico a la PT100 para poder usar el resto de las topologías, de 3 o 4 hilos, y la no linealidad del incremento de la resistencia con la temperatura.

Sería importante la implementación de un medidor de temperatura con PT100 que incorpore uno o varios bloques dentro del modelo Simulink para

la linealización, que permita incorporar la tabla de resistencia que garantice obtener la resistencia correspondiente de la PT100 para cada temperatura. Esto es necesario ya que, como se comentó, la variación de resistencia con la temperatura no es una función lineal. Por otro lado, para disminuir los errores en la medición de temperatura sería recomendable alimentar el divisor con una fuente de corriente conocida, y del menor valor posible, lo que debe mejorar la exactitud de la medida de $R(t)$.

Los recursos del Arduino Uno utilizados en las implementaciones, es bajo aún. Esto permitiría añadir en la misma placa la adquisición y procesamiento de otras variables de interés, incluso para el propio monitoreo volcánico, como: la concentración de CO₂, la sismicidad y el potencial espontáneo. El caso de la medición de deformación se resolvería con la medida de inclinación del edificio volcánico, debidamente instrumentada.

Referencias

- Alcaldía de Albacete (sf). *Manual S.E.P.E.I. de Bomberos. Elementos de un Sismo: Erupciones Volcánicas*. Albacete, España.
- Amiya, N., & Stojmenovic, I. (2010). *Wireless Sensor and Actuator Networks Algorithms and Protocols for Scalable Coordination and Data Communication*: Wiley.
- Arian (2015a). Nota Técnica 4, rev. a. *Control & Instrumentación*. Recuperado de: <http://www.arian.cl>.
- Arian (2015b). Tablas de termocuplas y Pt100. Obtenido de Nota Técnica 3, rev. b. *Control & Instrumentación*. Recuperado de: <http://www.arian.cl>
- Ariosto, O. (2008). *Sistema de Monitoreo Volcánico en Tiempo Real*. Puebla, México: Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación.

- Blanco-Velandia, J. A., & Pérez-Castillo, J. N. (2012). Redes Inalámbricas de Geosensores Aplicadas en Sistemas de Observación y Monitoreo Ambiental. *Gerencia Tecnológica Informática*, 11 (29), 59-68.
- Carreño-Bodensiek, C. (2010). Sistema de control y monitoreo automatizado para gases en minas de carbón. *Revista de Investigación, Desarrollo e Innovación*, 1 (1), 61-69. Recuperado de: http://revistas.uptc.edu.co/revistas/index.php/investigacion_duitama/article/view/1294
- Fernández-Morales, F., & Duarte, J. E. (2012). Desarrollo de un caudalímetro digital para la medición de caudal de ríos. *Revista de Investigación, Desarrollo e Innovación*, 3 (1), 44-51. Recuperado de: http://revistas.uptc.edu.co/revistas/index.php/investigacion_duitama/article/view/2130
- Freescale Semiconductor, Technical Data. (2008). 1.5±g, 6±g Three Axis Low-g, Micromachined Accelerometer. Arizona, EEUU: Freescale.
- Guarnizo. (2013). *Manual de Usuario: Acelerómetro MMA7361*: Moviltronics, Ltda.
- Hay, J., & Wenbin, J. (2010). *From Principle to Practice*. Recuperado de: <http://www.cloudbus.org/papers/SensorWeb2010Chaper20.pdf>
- Huamán, C. A. (2001). *Implementación del Equipo de Adquisición de Datos de Prospección Sísmica*. Lima, Perú: Centro Peruano Japonés de Investigaciones Sísmicas y Mitigación de Desastres.
- Lopes, R. T. (2014). A wireless sensor Network for Monitoring Volcano-Seismic Signals. *Natural Hazards and Earth System Sciences*, 3123-3142.
- López-Mendoza, L. A., Molina-Arévalo, S. O., & Vaides-Ramírez, R. A. (2007). *Desarrollo de un Sistema de Monitoreo Volcánico* (Tesis de Pregrado). San Salvador, El Salvador: Universidad del Salvador.
- Martín-Ballesteros, A., & Del Rio-Carbajo, M. (2013). *Control de Posición de un Balancín con Arduino* (Tesis de pregrado). Valladolid, España: Universidad de Valladolid.
- Martínez-Ovalle, S., Reyes-Caballero, F., & González-Puin, L. X. (2013). Protección radiológica a trabajadores y público en instalaciones que operan radioisótopos industriales. *Revista de Investigación, Desarrollo e Innovación*, 3 (2), 120-124. doi: 10.19053/20278306.2166
- Peng, Y. L. (2009). Design Of Smart Sensing Component For Volcano Monitoring. *Sensorweb Research Laboratory*. Vancouver, USA: Washington State University.
- Quaas-Weppen, R., González, R., Guerra, E., Ramos, E., & dela Cruz-Reyna, S. (1995). *Monitoreo Volcánico: Instrumentación y Métodos de Vigilancia*. D.F, México: Centro Nacional de Prevención de Desastres.
- Quaas-Weppen, R. (1995). *Monitoreo Volcánico: Instrumentación y Métodos de Vigilancia*. D.F, México: Centro Nacional de Prevención de Desastres.
- Valdés, C. M., Puente-Espinosa, L. F., & Osorio-Chong, M. A. (2008). *Volcanes Peligro y Riesgo Volcánico en México*. D.F, México: CENAPRED.
- Valdés, O. Y. (2014). *Volcanes: Peligro riesgo volcánico en D.F, México*. México: Centro Nacional de Prevención de Desastres.
- Vargas, D., Rodríguez, E., & Otero, J. (2013). *Alternativas para la detección y monitoreo de amenazas sísmicas basadas en arduino*. *Ing. USBMed*, 45-54.
- Werner, G. L. (2006). Deploying a Wireless Sensor Network on an Active Volcano. *Ieee Internet Computing*, 18-25.