

**ARDUINO COMO UNA HERRAMIENTA PARA MEJORAR EL PROCESO DE ENSEÑANZA –
APRENDIZAJE DE LAS CIENCIAS, TECNOLOGÍAS E INGENIERÍAS EN
LA UNIVERSIDAD POLITÉCNICA DE TAPACHULA**

**ARDUINO AS A TOOL FOR THE IMPROVEMENT OF THE TEACHING –
LEARNING PROCESS OF SCIENCES, TECHNOLOGIES AND ENGINEERING AT
THE UNIVERSIDAD POLITECNICA DE TAPACHULA**

(Recibido el 19-01-2015 - Aprobado el 20-02-2015)

**Mg. Carlos Alberto
Gómez Moreno**
**Universidad Politécnica
de Tapachula,**
*Coordinador de Ingeniería de
Sistemas Automotrices,
Cuerpo Académico Ingeniería
Aplicada a las Organizaciones.
Chiapas-México.*
carlos.gomez@uptapachula.edu.mx

Mg. Alfredo Castillo Solís
**Universidad Politécnica
de Tapachula,**
*Coordinador de Ingeniería en
Software, Cuerpo Académico
Ingeniería Aplicada
a las Organizaciones.
Chiapas-México*
alfredo.castillo@uptapachula.edu.mx

Mg. Alfredo Gómez Meoño
**Universidad Politécnica
de Tapachula,**
*Coordinador de Ingeniería en
Logística y Transporte,
Cuerpo Académico Ingeniería
Aplicada a las Organizaciones.
Chiapas-México*
alfredo.gomez@uptapachula.edu.mx

Resumen. Arduino es una plataforma abierta diseñada para crear prototipos de objetos o ambientes interactivos usando electrónica libre. Consiste, tanto en hardware como en software, en una tarjeta de circuito impreso que puede ser adquirida a bajo costo o ensamblarse siguiendo planos disponibles de forma gratuita, así como un ambiente de desarrollo de fuente abierta con librerías para escribir códigos para controlar la tarjeta. Enriquece el proceso de enseñanza-aprendizaje mediante la experimentación, además de que provee un soporte asequible y flexible para estudiantes, profesores e investigadores, de tal manera que puedan contar con una base para desarrollar múltiples y diversos proyectos en ciencias, tecnologías e ingeniería, ofreciendo así la oportunidad de construir un amplio portafolio de innovadoras aplicaciones y prototipos que actuarán como vectores para detonar la creatividad y aumentar las habilidades y capacidades del estudiante, a fin de brindar soluciones tecnológicas tanto para propósitos educativos, como para organizaciones públicas o privadas, industrias de diversos sectores, entre otros. Este artículo presenta una descripción esencial de Arduino y un análisis del impacto que dicha plataforma tiene como herramienta didáctica en la educación de nivel superior, específicamente en la enseñanza de las ingenierías, así como algunas experiencias en aplicaciones y prototipos desarrollados a través de su uso en la Universidad Politécnica de Tapachula.

Palabras clave: Arduino, ciencia, enseñanza–aprendizaje, ingeniería, tecnología

Abstract. Arduino is an open platform designed to create prototypes of objects and interactive environments using open electronics. It consists of both hardware and software: a printed circuit board that can be acquired for a low cost or assembled following free available diagrams, as well as an open-source development environment with libraries for writing the codes to control the board. It enhances the teaching-learning process through experimentation and also provides an affordable and flexible support for students, professors and researchers, in a sort that they can count on a basis to develop multiple and diverse projects in sciences, technologies and engineering, settling that way the opportunity to build a wide portfolio of innovating applications and prototypes that will act as vectors to detonate creativity and improve the student's skills and capabilities, in order to bring technological solutions for educational purposes, public or private organizations and industries from diverse sectors. This paper presents an essential

description of Arduino and an analysis of the impact that this platform has as a didactic tool in superior level education, specifically in teaching engineering, as well as some experiences in applications and prototypes developed through its use at the Universidad Politecnica de Tapachula.

Keywords: Arduino, engineering, science, teaching–learning, technology

1. INTRODUCCIÓN

En el ámbito de la educación a nivel superior, específicamente en la enseñanza de las Ingenierías, siempre debe tenerse en mente el lenguaje utilizado para comunicarse y la profundidad de los contenidos (Cabrero, 2007). En este sentido, el uso de Arduino obedece a esta filosofía y evita quitar la complejidad inherente a la electrónica, como lo hacen otras plataformas de alto nivel. La electrónica consiste de componentes, los cuales son dispositivos físicos que representan funciones lógicas, y es esencial que esos conceptos permanezcan dentro del proceso de enseñanza-aprendizaje (Gallegos, 2013).

Se trata de hacer las cosas lo suficientemente fáciles para que los estudiantes entiendan cómo funcionan las cosas a través de la experimentación. En vez de intentar esconder la complejidad de la electrónica a los estudiantes principiantes, es preferible simplificarla hasta el punto en el que puedan lidiar con ella directamente. Eventualmente, conforme el estudiante avanza en su carrera, tendrá la oportunidad de profundizar sus conocimientos mediante el acceso a otros recursos existentes en diferentes disciplinas, tales como Ingeniería de Sistemas Automotrices, Software, Logística y Transporte, Agroindustrial, Animación y Efectos Visuales, por mencionar algunos casos en la Universidad Politécnica de Tapachula.

Arduino puede enriquecer el proceso educativo, dado que los estudiantes, profesores e investigadores generan sus propios materiales, talleres, ejemplos y tutoriales, así mismo, tienen la oportunidad de retroalimentar y ser retroalimentados por miembros o comunidades de otras universidades de todo el mundo.

Genera mayores habilidades y capacidades en el alumno gracias a la experimentación, a través del desarrollo de aplicaciones y prototipos interactivos, incrementando las posibilidades de llevar a cabo proyectos multidisciplinarios aun cuando no se cuente con un laboratorio, o cuando no hay presupuesto para adquirir equipo de alto costo, como actualmente sucede en la Universidad Politécnica de Tapachula.

2. DESCRIPCIÓN DE ARDUINO Y SU POTENCIAL USO COMO HERRAMIENTA PARA LA EDUCACIÓN

Existen muchas herramientas para crear aplicaciones y prototipos usando electrónica, los cuales se emplean prácticamente en todo. Desde habitaciones inteligentes hasta instrumentos musicales o juguetes, desde pequeños robots hasta drones, que pueden incluir dispositivos personalizados e interactivos.

Estas herramientas pretenden reducir la dificultad que en ocasiones representa para el alumno promedio el trabajar con sistemas y componentes electrónicos, facilitando su uso en cualquier rama o nivel, poniéndose al alcance de quienes estén ávidos de experiencias e innovación, favoreciendo en gran medida a estudiantes, profesores e investigadores (Ruiz Velasco, 2007).

Sin embargo, muchas de estas herramientas, son productos comerciales, los cuales son caros y de plataforma cerrada, difícilmente asequibles como equipamiento para aquellas Instituciones de Educación Superior del sector público, que cuentan con recursos limitados, en especial si son de reciente creación, como es el caso de la Universidad Politécnica de Tapachula; o derivan de proyectos de investigación que no están disponibles para el uso de la mayoría de las personas. Otras, consisten únicamente en tarjetas de circuito impreso, que en la mayoría de los casos no proporcionan una plataforma para simplificar su programación.

Mientras tanto, el movimiento *open-source*, conocido en español como libre o de fuente abierta, ha mostrado que se puede crear software útil y robusto a través de un equipo distribuido de voluntarios dispuestos a compartir libremente los resultados de sus desarrollos y proyectos. Dichos proyectos a menudo reúnen grandes y fuertes comunidades de personas que trabajan de manera concurrente en varios niveles: Algunos trabajan sobre el código del núcleo, otros contribuyen en pequeñas extensiones, otros escriben la documentación u ofrecen soporte, aunque la mayoría simplemente colabora haciendo uso de un producto de calidad en diversas aplicaciones y prototipos (Arduino, 2014).

¿Podemos aplicar los principios de fuente abierta o libre en el hardware y la electrónica? ¿Qué significado o impacto tendría la creación de una tarjeta de circuito impreso que sea abierta y extensible, y a la vez fácil de usar? ¿Cómo podríamos hacer para trabajar con electrónica de forma fácil, rápida, barata y de buena calidad? Estas son algunas de las preguntas que llevaron a la creación de la plataforma Arduino (Banzi y Shiloh, 2014).

En la actualidad, existen muchas herramientas de desarrollo de microcontroladores disponibles para la enseñanza y la creación de prototipos. Las más populares fuera de la comunidad de ingeniería eléctrica, fueron desarrolladas pensando en ofrecer un equilibrio entre el costo, la capacidad de expansión y la facilidad de su uso. Arduino nació de la combinación de varias culturas de prototipado; se ha esforzado en encontrar un balance entre los factores mencionados anteriormente, e incluso ha logrado superar las carencias o debilidades de otras plataformas existentes. Algunas como Phidgets (Phidgets Inc., 2014), que son herramientas del más alto nivel de abstracción y que generalmente no son programables por el usuario final, solamente se pueden configurar y operar a través de una computadora, por lo que no son dispositivos independientes, o bien, las herramientas de software y hardware desarrolladas por algunas universidades y centros de investigación de avanzada, que aunque pueden ser más robustos y flexibles que las anteriores, la plataforma no está disponible comercialmente.

También existen las plataformas de nivel medio, que consisten en módulos provistos de un microcontrolador y un soporte electrónico para que el usuario incorpore sus propios circuitos de entrada y salida, los cuales se programa usualmente en BASIC o alguna variación de C. Un buen ejemplo de este tipo de plataformas es Parallax (Parallax Inc., 2014), siendo sus principales ventajas la programabilidad en lenguaje de alto nivel y la sencilla interfaz de programación; sin embargo, por lo general los lenguajes de programación son limitados, prácticamente todos disponibles únicamente para sistemas operativos de Windows, además de que los niveles más bajos del controlador no son accesibles para el usuario. Aunque el costo es menor comparado con los controladores de alto nivel, éste sigue siendo alto. Considerando que especialmente los estudiantes tratan de evitar el uso de múltiples módulos en sus proyectos, ya que esto multiplicaría sus gastos, sin mencionar que la probabilidad de cometer un error en el cableado de los circuitos que dañe el procesador es bastante alta.

En el nivel más bajo, están los microcontroladores por sí mismos, siendo la opción más económica, pero no apta para principiantes, dado que requieren que el estudiante tenga mayores conocimientos en electrónica.

3. APLICACIONES DE ARDUINO PARA ESTUDIANTES

Son varios los pasos requeridos para que el estudiante pueda llevar a cabo la más básica de las tareas con un microcontrolador:

- Seleccionar un microcontrolador en particular y buscar el circuito necesario para usarlo.
- Ensamblar las partes necesarias
- Descargar el software necesario para programar el microcontrolador.
- Encontrar la manera de comunicar el microcontrolador con la computadora, instalar todos los drivers necesarios, comprar o construir un dispositivo externo para programar el microcontrolador.
- Aprender cómo escribir el código para el microcontrolador, que implica la lectura de las hojas de especificación, pudiendo ser cientos de ellas.
- Escribir el código, trabajar sobre los comandos para compilar y subir los códigos, etc.

Arduino pretende eliminar o facilitar la mayor cantidad de pasos posible con una combinación de hardware y software (Barret, 2013).

El hecho de que el hardware en la plataforma Arduino consista en partes que un estudiante principiante pueda usar, obedece también a otro propósito: significa que los estudiantes con cierta experiencia pueden construir determinado módulo en una tarjeta de prototipado si así lo desean. Esto puede acelerar el desarrollo y reducir los costos de un proyecto que utiliza múltiples procesadores. Los esquemas y los diagramas de los circuitos tienen una licencia abierta, por lo que cualquiera puede usarlos y compartir sus trabajos con el resto del mundo (McRoberts, 2013).

Todas las versiones de la tarjeta Arduino están diseñadas para trabajar con componentes electrónicos estándar. La tarjeta provee una plataforma base pero no limita a los estudiantes al uso de sensores o actuadores prefabricados. Esto significa que se pueden usar componentes nuevos o inusuales sin necesidad de esperar a que salgan versiones especiales de Arduino. Existen extensiones denominadas shields,

las cuales consisten en tarjetas de circuitos que pueden ser montadas en la tarjeta Arduino para proveer mayores funcionalidades, por ejemplo, circuitos para soportar dispositivos de alta potencia, tales como motores, sensores RFID, o bien interfaces bluetooth (Yarnold, 2015).

El software utilizado en la plataforma Arduino suma los esfuerzos por simplificar el proceso de escritura de código sin limitar la flexibilidad. Se constituyó de otros varios proyectos de fuente abierta, adaptados al hardware Arduino y escondiendo sus complejidades innecesarias; consiste en 2 partes principales: el ambiente de desarrollo y una librería núcleo, ambas de fuente abierta (Blum, 2013).

El ambiente de desarrollo en Arduino es mínimo pero proporciona un editor de fuente completo. Es una aplicación transversal escrita en Java y utilizable en Windows, Mac OS X y en Linux. Dentro de él, los alumnos pueden administrar, editar, compilar y subir sus programas, llamados sketches. Todas las funciones pueden ser llamadas desde un conjunto de 7 botones de la barra de herramientas o desde unos cuantos menús. El alumno no tiene que perder el tiempo con argumentos de línea de comandos, los cuales pueden representar obstáculos para el principiante. El ambiente incluye un monitor serial, permitiendo al usuario enviar y recibir datos hacia y desde la tarjeta, facilitando la depuración sin requerir un software adicional. De hecho, todos los programas requeridos para el desarrollo son incluidos en un solo archivo que se descarga desde el sitio web de Arduino, a excepción de Linux, ya que sus usuarios necesitarían instalar algunos paquetes.

La librería núcleo de Arduino consiste en funciones AVR C/C++, que son compiladas junto con el sketch del usuario. El archivo binario combinado puede cargarse en la tarjeta Arduino. A través del uso de una Interfaz de Programación de Aplicaciones - API (por sus siglas en inglés), que es compatible con Wiring (lenguaje de programación de fuente abierta para microcontroladores), el núcleo de Arduino encapsula aspectos de la programación de bajo nivel de microcontroladores, tales como la manipulación de registros, permitiendo a los usuarios concentrarse en una tarea en particular (Timmis, 2011). Esto evita que el estudiante tenga que leer unas 150 páginas o más de especificación del microcontrolador, la única fuente de información confiable respecto a su funcionalidad a bajo nivel.

En Arduino, los estudiantes siguen programando en estándar C/C++, de tal forma que el conocimiento en programación que van adquiriendo, puede ser transferido y aplicado en muchas otras situaciones.

Dada la limitada capacidad del microcontrolador, parte del código es dividido en librerías separadas, las cuales pueden ser incluidas específicamente para cuando se requieran en un sketch en particular. Cualquier alumno puede escribir una librería adicional e instalarla, simplemente con moverla al directorio correcto.

4. PROTOTIPOS CREADOS EN ARDUINO POR ESTUDIANTES: CASO UPTap

Algunas aplicaciones y prototipos desarrollados usando Arduino en la Universidad Politécnica de Tapachula han sido dos prototipos, de una Casa Remota y un Robot Recolector de Basura.

4.1 Prototipo de Casa Remota

La idea que persigue este prototipo es automatizar el uso de las luces y otros dispositivos en el hogar, dando como posibilidad que este proceso pueda ser realizado vía remota bajo el control directo del usuario, a través de su teléfono móvil. Ya que hoy por hoy es difícil imaginar que una persona ciudadana esté separada de su celular. Actualmente, el sistema operativo para teléfonos móviles inteligentes usado por las firmas que los fabrican, es Android, mismo que está diseñado para dispositivos interactivos, es decir, que posean dispositivos como una pantalla táctil y una conexión inalámbrica, tanto Wi-Fi como Bluetooth. Aprovechando estas tecnologías, el prototipo utiliza la terminal Android de un teléfono móvil inteligente a manera de control remoto de los dispositivos de la casa, logrando controlar la energía del circuito gracias a la tarjeta Arduino (Karvinen y Karvinen, 2011).

Para poder desarrollar este prototipo, se utilizó una aplicación destinada para Android que provee acceso al mismo, ya sea por botones, y/o por control de voz a una lista de acciones predefinidas, que permiten el control de aparatos conectados a la tarjeta Arduino, cuya función principal en este caso es controlar el flujo de la energía, y a través de esto encender los actuadores o LED que sean requeridos para cumplir dichas acciones. La comunicación entre Android y Arduino ocurre gracias a una terminal Bluetooth con la cual los 2 aparatos serán sincronizados y trabajarían en conjunto. Los materiales utilizados para los 2 aparatos mencionados son: Arduino Starter Kit (con todos sus componentes), Modulo Bluetooth Hc-06, cableado flexible con conectores DUPONT, cableado UTP, sensor PIR, Terminal Android (teléfono móvil inteligente).

Se realiza una maqueta de la casa, utilizando cartón batería, cúter, pegamento instantáneo, papel contact temático, regla, lápiz HB. Una vez cortados los pliegos de la casa, se procedió a pegarlos en la base de la estructura, y posteriormente se forraron con papel temático, formando un área de 55.5 cm de largo por 40 cm de alto. Los cortes fueron pegados a la base usando pegamento instantáneo. Como se puede apreciar en la figura 1.



Figura 1. Prototipo de casa remota terminado

En la figura 2 se muestra que una vez instalada la aplicación en Android, se crea un icono alusivo a Arduino (izquierda). Cuando se accede a la aplicación se muestra un menú con las opciones disponibles para el control, mismas que se encontrarán desactivadas mientras la terminal no esté conectada con la tarjeta Arduino (derecha). Para poder realizar la conexión solo basta con hacer clic en símbolo de conectar. Los comandos de voz puede ser solicitados apretando el botón de en medio con forma de micrófono o agitando el teléfono.



Figura 2. Interface de la aplicación en Terminal Android

La programación consta de tres partes: la primera es la declaración de variables, la segunda es la configuración de los pines, la tercera parte es un ciclo infinito, en la cual se ejecutan ciclos, y condiciones que son necesarias para el reconocimiento de los sectores solicitados por la terminal Android.

4.2 Prototipo de un Robot Recolector de Basura

La idea de este prototipo es aprovechar los avances tecnológicos en electrónica y robótica para facilitar ciertas tareas de la vida cotidiana, como por ejemplo la recolección de basura. Este prototipo de robot recolector de basura fue construido a partir de un robot minisumo de plataforma Arduino (Warren, Adams y Molle, 2011), muy popular en el ámbito de la robótica didáctica, puesto que actualmente existen muchas competencias tanto regionales como internacionales para esta categoría.

El robot debe moverse en distintos sentidos, de manera aleatoria sin salirse de su plataforma o área delimitada. Esto se logró usando sensores de orilla o TES (por sus siglas en inglés), además de implementarle algunos dispositivos adicionales como una pantalla LCD para mostrar mensajes, así como un mecanismo de enfriamiento utilizando un sensor de temperatura, esto último para evitar daños en la tarjeta Arduino. El robot recolector puede seleccionar un objeto de acuerdo a su tamaño y así saber si puede levantarlos y depositarlos en su contenedor (Margolis, 2012).

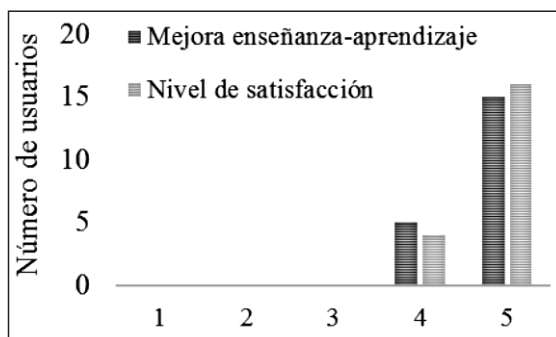
Los materiales utilizados fueron: tarjeta Arduino uno, pantalla LCD, 4 resistencias de 240 Ω , 2 resistencias de 10k Ω , 5 resistencias de 555 Ω , 2 fotorresistencias, 2 potenciómetros de 20k Ω , 4 interruptores, 2 sensores de orilla (TES), 4 sensores infrarrojos, 4 motores, 1 servomotor, 3 LED, 1 protoboard, 4 llantas, batería de 9V, batería de 5V, 2 metros de cables UTP, 1 placa metálica, 1 sensor de calor, 1 puente H, 1 microcontrolador 555.

5. VALORACIÓN EN EL PROCESO DE ENSEÑANZA-APRENDIZAJE

Los prototipos y aplicaciones que se exponen en el presente trabajo (casa remota, robot recolector de basura), permiten vislumbrar, con una pequeña muestra, la amplia gama de posibilidades que los estudiantes, profesores e investigadores tienen para llevar a cabo todo tipo de proyectos multidisciplinarios en las áreas de ciencias, tecnologías e ingenierías y, ya sea para fines académicos, o bien para uso en otros sectores de la sociedad, aprovechando la flexibilidad, bajo costo y relativa facilidad que implica el uso de la plataforma de electrónica libre Arduino.

A través de una encuesta aplicada a cada alumno y profesor involucrados en este trabajo, siendo un total de 20, siendo 10 participantes de la carrera de

Ingeniería en Software, y 10 participantes de la carrera de Ingeniería de Sistemas Automotrices en el cuatrimestre Septiembre – Diciembre 2014, se les preguntó si percibían alguna mejora en su aprendizaje o enseñanza, según fuera el caso, y qué tan satisfactoria había sido esta experiencia, utilizando una escala de Likert, del 1 al 5, en donde 1 representa “Totalmente en desacuerdo”, el 2 “En desacuerdo”, el 3 “Ni de acuerdo ni en desacuerdo”, 4 “De acuerdo” y el 5, “Totalmente de acuerdo”.



Gráfica 1. Percepción de estudiantes y profesores usuarios de Arduino en la UPTap

El desarrollo de los prototipos se tradujo también como un impacto positivo en el proceso de enseñanza-aprendizaje de los estudiantes y profesores de la Universidad Politécnica de Tapachula involucrados, ya que éstos expresaron que les permitió ejercitar su creatividad, además de que observaron actitudes y valores deseables como responsabilidad y organización.

6. CONCLUSIONES

De la gráfica 1 se aprecia que el 75% de los participantes tuvo una alta percepción en cuanto a mejoras en el proceso de enseñanza-aprendizaje, mientras que el 80% de los participantes tuvo una alta satisfacción en su experiencia usando Arduino. En ningún caso hubo una percepción media, baja o muy baja sobre el proceso de enseñanza-aprendizaje, tampoco lo estuvo en el nivel de satisfacción.

Respecto a los beneficios obtenidos, los profesores involucrados hicieron hincapié en el desarrollo de competencias de trabajo en equipo y solución de problemas aplicando el razonamiento lógico deductivo y diversos conceptos y principios físicos. Los alumnos comentaron que la realización de sus prototipos les ayudó a fortalecer principalmente las áreas de circuitos eléctricos, electrónica y programación.

La aceptación de Arduino en la comunidad de la Universidad Politécnica de Tapachula ha sido bastante buena, y en este sentido su uso ha generado varias experiencias positivas. Desde el punto de vista didáctico, la incorporación de esta plataforma en el aula y/o laboratorio se adapta sin problemas al modelo educativo que siguen las IES Politécnicas: educación basada en competencias, dado que desarrolla en los estudiantes competencias no solo específicas sino también genéricas y transversales, al poder tomar elementos de las metodologías de aprendizaje colaborativo, aprendizaje basado en problemas, así como aprendizaje orientado a proyectos.

Se pretende que para los siguientes periodos escolares enero – abril y mayo – agosto 2015, más profesores y estudiantes de todas las ingenierías de la UPTap incorporen Arduino para la realización de proyectos, aplicaciones y prototipos multidisciplinares. Quienes ya han acumulado cierta experiencia participarán en el desarrollo de proyectos más ambiciosos, como es el caso de un cuadracóptero (helicóptero de 4 hélices) para vigilancia aérea, un AGV (Vehículo Guiado Automáticamente) y un sistema ARS (Almacén Automatizado).

RECONOCIMIENTO

Al Consejo de Ciencia y Tecnología del Estado de Chiapas (COCYTECH), por el valioso apoyo brindado para la adquisición de las tarjetas Arduino utilizadas en los prototipos que se incluyen en el presente trabajo.

A los alumnos y profesores de la Universidad Politécnica de Tapachula que participaron en la realización de los proyectos basados en Arduino, y que brindaron su retroalimentación para la obtención de las conclusiones del presente trabajo.

REFERENCIAS

- Arduino. (2014). *What is Arduino*. Recuperado de <http://arduino.cc/>
- Banzi, M., y Shiloh, M. (2014). *Make: Getting Started with Arduino: The Open Source Electronics Prototyping Platform*. Make Media, Inc.
- Barret, S. (Ed. 3) (2013). *Arduino Microcontroller Processing for Everyone!* Estados Unidos: Morgan & Claypool.

- Blum, J. (2013). *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. Estados Unidos: John Wiley & Sons.
- Cabrero, J. (2007). *Nuevas Tecnologías Aplicadas a la Educación*. España: McGraw Hill.
- Gallegos, J. (2013). *Aplicación de la robótica educativa en la enseñanza de las ciencias: Propuesta de un módulo electrónico programable*. España: Editorial Académica Española.
- Karvinen, T., y Karvinen, K. (2011). *Make: Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software*. O'Reilly Media, Inc.
- Margolis, M. (2012). *Make and Arduino-Controlled Robot*. O'Reilly Media, Inc.
- McRoberts, M. (2013). *Beginning Arduino*. Apress.
- Parallax Inc. (2014). *Microcontrollers overview*. Recuperado de <http://www.parallax.com/microcontrollers/microcontrollers-overview>
- Phidgets Inc. (2014). *Products for USB sensing and control*. Recuperado de <http://www.phidgets.com/index.php>
- Ruiz Velasco, E. (2007). *Educatrónica: Innovación en el aprendizaje de las ciencias y la tecnología*. Madrid, España: Ediciones Díaz de Santos.
- Timmis, H. (2011). *Practical Arduino Engineering*. Apress.
- Warren, J.-D., Adams, J., y Molle, H. (2011). *Arduino Robotics*. Apress.
- Yarnold, S. (2015). *Arduino in easy steps*. In Easy Steps.