

Creación de un simulador de circuitos electrónicos para usar con placas Arduino

Cristian González García
Escuela de Ingeniería Informática
Universidad de Oviedo
gonzalezcristian@uniovi.es

Ana I. Gómez
Escuela de Ingeniería Informática
Universidad Rey Juan Carlos
ana.gomez.perez@urjc.es

Domingo Gómez
Facultad de Ciencias
Universidad de Cantabria
gomezd@unican.es

Resumen

Los simuladores son atractivos para la formación de los estudiantes, al mostrar un modelo de un sistema sin los riesgos que puede presentar el uso de plataformas de hardware real. Aunque existan soluciones comerciales, no siempre cubren todos los sensores, actuadores y bibliotecas utilizados en clase. Por lo tanto, es el docente que tiene que adaptar su material para incorporar los ejercicios propuestos y asumir posibles costes de licencia en el caso de software de pago.

Para evitar los inconvenientes mencionados, presentamos un simulador de código libre y modular que incluye el montaje de circuitos sobre una placa Arduino, permitiendo así la construcción de circuitos con los dispositivos electrónicos utilizados en la asignatura «Software para Robots». El desarrollo es una colaboración entre varias universidades para permitir su uso genérico, y es fácilmente ampliable con nuevos ejercicios y dispositivos a conectar. Aunque la aplicación sigue evolucionando y se van añadiendo más funcionalidades, está lista para ser usada. Esta afirmación está avalada por los comentarios positivos y un buen grado de satisfacción de estudiantes y profesorado.

Este simulador pretende ofrecer un recurso para los docentes, siendo abierto y ampliable para el mundo académico, y ayudando a profesores y estudiantes para sacar el máximo partido de las tecnologías existentes. Además, permite al alumnado la participación semi-presencial o virtual, favoreciendo al trabajo autónomo.

Abstract

Simulators are an appealing training tool for students, allowing them to interact with complex systems without the risks associated with real hardware platforms. While commercial solutions exist, they usually cover partially the necessary sensors, actuators, and libraries. As a result, adaptations are often needed, incurring in additional costs when using licensed software. To address this, we introduce a free and modular si-

mulator that enables circuit assembly on an Arduino board. With this simulator, students construct circuits using electronic devices in “Software for Robots.” The development of this simulator is a collaborative effort among several universities, making it suitable for generic use. It is also easily expandable with new exercises and additional devices for connection.

Despite ongoing enhancements and added functionality, the application is stable enough for daily use, as evidenced by positive feedback from both students and faculty. The primary goals of this simulator are to provide a valuable resource for teachers, foster openness, and support academic endeavors. Additionally, it allows students to participate in a semi-presential or virtual manner, promoting autonomous learning.

Palabras clave

Arduino, simulador, software, robótica, recurso docente, docencia en prácticas.

1. Introducción

El microcontrolador Arduino Uno ¹ se utiliza en muchos institutos y universidades para enseñar a los estudiantes a crear y programar sus propios proyectos siguiendo el concepto de «Do It Yourself» (DIY). Esta popularidad es, en gran parte, debida al abaratamiento de los componentes electrónicos y a su condición de hardware abierto. Por esto, su uso se ha extendido a asignaturas de electrónica, programación y robótica, así como a proyectos del Internet de las Cosas (IoT) y construcción de robots.

Actualmente, investigadores y docentes [2, 8, 9] utilizan microcontroladores Arduino como base de sus clases o investigaciones. Generalmente, Arduino se incorpora a la enseñanza mediante un enfoque de aprendizaje basado en problemas en materias STEM como ciencias, tecnología, ingeniería y matemáticas, y

¹<https://store.arduino.cc/products/arduino-uno-rev3>

posibilita el desarrollo del pensamiento computacional cuando se emplea conjuntamente con componentes electrónicos y entornos como Scratch [4].

El hardware robótico real presenta costes y riesgos en el aula, por lo que es habitual el uso de simuladores para reforzar su aprendizaje [10] fuera del horario lectivo, sin necesidad de acudir al laboratorio. Existen en el mercado simuladores de microcontroladores Arduino Uno como «Circuits²», que es gratuito y permite trabajar sobre el Arduino Uno R3. Cuenta con varios sensores y actuadores emulados, sin embargo, estos no son modificables y requieren el uso de versiones exactas. Sus ventajas incluyen la programación directa en Arduino o mediante piezas de puzzle y simulación de cortocircuitos. Bitbloq³ es una alternativa con funcionalidad similar a Circuits, que es gratuita solamente para el microcontrolador Zum y con un coste mensual o anual para el uso de Arduino Uno. También Horosim⁴ es un simulador de Robots con un motor físico muy completo, pero abstrae el conexionado de circuitos.

Otros ejemplos son Scratch for Lego Mindstorm⁵, MiniBloq⁶, Scratch for Arduino⁷, AppInventor⁸, y la plataforma Midgar con MOCSL [5, 6]. Estos se basan en lenguajes de dominio específico (DSL) gráficos con los que los usuarios pueden crear la lógica mediante la composición visual. Esto otorga una abstracción sobre el código fuente, que no es recomendable en grados de informática en los que los estudiantes requieren familiarizarse y desarrollar destreza en programación. Este hecho hace inviable utilizar estas herramientas en este contexto.

En este trabajo proponemos un recurso software educativo abierto, que permita la simulación de circuitos sobre un microcontrolador Arduino y que pueda ser modificado y utilizado libre y gratuitamente por la comunidad educativa o por otros usuarios. También, con este simulador pretendemos ayudar al alumnado para que pueda preparar las clases en casa, promoviendo su trabajo autónomo, y ayudando a los estudiantes de evaluación diferenciada o dispensa académica. De esta forma, podrán aprovechar mucho mejor el tiempo en clase con el profesor para probar el montaje del hardware y el funcionamiento de su algoritmo en un entorno real. Para el profesor la ventaja buscada es que reduzca su trabajo de preparación, teniendo un entorno controlado donde puedan experimentar nuevos trabajos y poder dedicar más tiempo a comprobar el código fuente, principal prioridad en estos grados.

²<https://www.tinkercad.com/circuits>

³<https://bitbloq.cc/>

⁴<https://bitbucket.org/afaina/horosim/>

⁵<https://scratch.mit.edu/ev3>

⁶<http://blog.minibloq.org/>

⁷<http://s4a.cat/>

⁸<https://appinventor.mit.edu/>

2. Descripción de la solución propuesta

La figura 1 muestra la pantalla principal del simulador, anteriormente presentada [7]. Este simulador soportaba dos tipos de robots: un robot móvil y un actuador lineal. En el caso del robot móvil, cuenta con un total de 6 circuitos y 3 configuraciones diferentes (2, 3 y 4 infrarrojos). En este artículo, se presenta la mejora del simulador añadiendo soporte a la placa Arduino Uno y a diferentes componentes electrónicos.

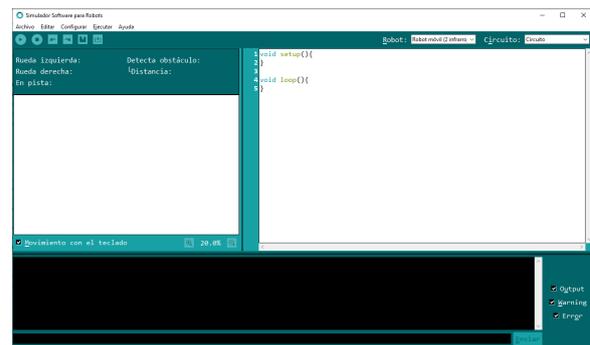


Figura 1: Pantalla principal del simulador

La interfaz de control permanece similar, manteniendo en la parte superior izquierda las listas desplegables y los botones. En la parte superior derecha se encuentran los «combobox». Estos nos permiten seleccionar el robot o placa a utilizar, y si estos tienen diferentes versiones o circuitos, nos aparece un segundo «combobox» con dichas opciones. El primer «combobox» permite añadir la placa Arduino Uno, y si esta está seleccionada, el segundo «combobox» permite elegir entre una opción de uso libre y 5 desafíos.

En la parte derecha del simulador se muestra el editor de código fuente y los números de línea. El simulador soporta «Wiring», que es el lenguaje utilizado para programar el microcontrolador Arduino, simulándose diferentes bibliotecas. Este área ofrece resaltado de sintaxis y, por defecto, emula la estructura del IDE oficial de Arduino con los métodos *Setup* y *Loop*.

En la figura 2 se puede ver el simulador de circuitos en la parte izquierda como principal novedad que hemos integrado en el simulador. La zona de trabajo se representa como la placa Arduino Uno sobre un fondo blanco. En la parte superior hay un menú que permite añadir diferentes sensores y actuadores a la zona de trabajo (ver figura 3). Este menú permite seleccionar entre tres modelos de resistencias mediante un elemento «pop-up», un botón, un potenciómetro, un LED con 3 colores diferentes a elegir, un «buzzer», un LED RGB, un sensor de luz o fotoresistor, un sensor PIR, un sensor de vibración, un sensor de infrarrojos, un sensor de ultrasonidos, un teclado matricial, una pantalla

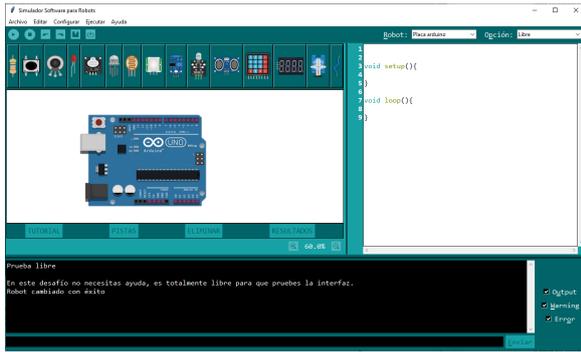


Figura 2: Pantalla de la simulación de la placa Arduino Uno

de 4 dígitos de 7 segmentos, y un servomotor. La última opción mostrada en el menú permite conectar dos componentes dibujando el cable libremente. Todos estos componentes son utilizados en clase.

En la figura 4 se puede ver un ejemplo de conexión; en particular, donde se muestra un LED conectado a una resistencia y al Arduino Uno. Queremos remarcar que se ha tenido en cuenta la conexión de pines con la placa Arduino Uno manteniendo las diferencias y usos reales entre componentes: positivo, negativo, digital, analógico, etc.

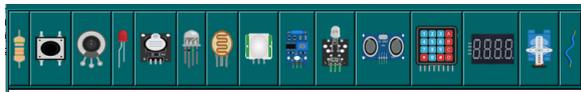


Figura 3: Menú con los sensores y actuadores disponibles

Finalmente, el paso de mensajes se realiza mediante la consola de salida en el área negra inferior. A su derecha se permite seleccionar el tipo de información a filtrar en la consola entre los siguientes: «output», «warnings» y errores.

Un ejemplo de ejercicio realizado en clase es la implementación de un semáforo en una puerta que se abre con teclado numérico (ver figura 5). El enunciado es el siguiente: «La puerta se abre cuando el usuario pulsa el botón A del teclado y permanece abierta durante 5 segundos, encendiendo un LED de color rojo. Pasado ese tiempo se cierra. En otros casos, el LED verde estará iluminando para indicar que se puede pasar. Si el usuario vuelve a pulsar el botón A mientras la puerta está abierta esa pulsación se ignora. Si el usuario pulsa cualquier otra tecla, no debe realizar ninguna otra acción» .

Este simulador puede ser utilizado libremente, pues cuenta con licencia GNU General Public License v3.0. El código fuente del programa está desarrollado en

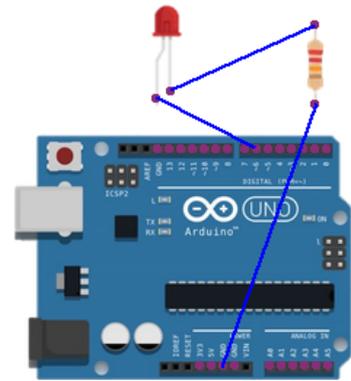
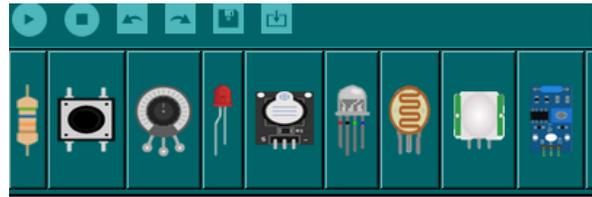


Figura 4: Circuito con un LED y una resistencia montados

Python⁹, y el simulador se encuentra disponible en GitHub¹⁰. Está acompañado con un manual del usuario en «manual-usuario.pdf». Actualmente, cuenta con soporte para Windows y tiene un script para generar el ejecutable para dicho sistema operativo.

3. Conclusiones

En este trabajo presentamos la ampliación de una herramienta previa con nuevo soporte para la creación

⁹<https://www.python.org/>

¹⁰<https://github.com/gonzalezgarciaacristian/simulator-robotic-software>

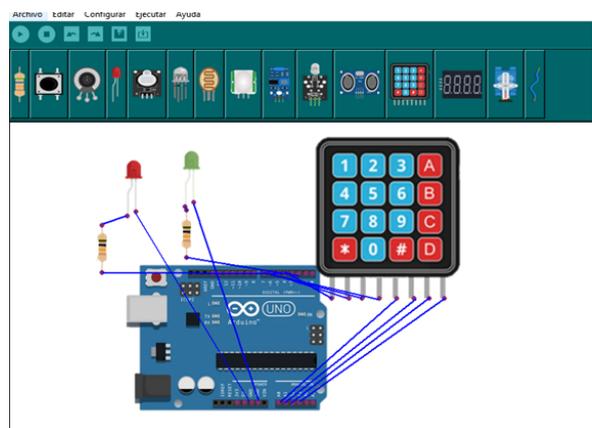


Figura 5: Desafío: semáforo para apertura y cierre de puerta

de circuitos electrónicos basados en Arduino. Este nuevo simulador extiende la funcionalidad del anterior, permitiendo que todas las prácticas estén centralizadas e integren la simulaciones en una única aplicación. Aunque este simulador no es el único disponible actualmente, sí es novedoso por su licencia libre y publicación en abierto. Esto permite la inclusión de nuevos sensores y actuadores que se requieran en clase. Actualmente, las herramientas solo soportan unos pocos y determinados sensores y actuadores y no son flexibles con respecto a las necesidades docentes.

Adicionalmente, esta herramienta permite a los estudiantes trabajar de forma más autónoma con un horario flexible, sin depender de la disponibilidad del hardware, de los horarios del centro o de sus laboratorios. Queremos destacar que esta herramienta está pensada en ser extensible, es decir, permite su uso, ampliación y mejora por parte de cualquier persona. Nuestra experiencia es que posibilita la mejora de las clases en robótica, algoritmia y otras afines.

Tras una primera experiencia con los estudiantes del curso 2023/2024 se recogieron las opiniones mediante un formulario consensuado basado en medidas de usabilidad como el SUS y otros adaptados de la literatura científica, y comentarios en clase. Gracias a esto, se han detectado diferentes fallos que se han corregido. Debido a que en otras asignaturas no utilizan este simulador y hay prácticas que las realizan directamente con el hardware, se pudo hacer una comparativa sobre si el simulador era útil o no. Los estudiantes indicaron que no les pareció complejo, que no necesitaban de personal técnico para usarlo, que podría ser fácil de aprender a utilizarlo ya que no han tenido que aprender muchas cosas para utilizarlo. Por otro lado, destacaron algunas faltas de integración y otras inconsistencias en la herramienta, así como algunos errores y mejoras reportadas.

Como trabajo futuro, queremos seguir avanzando en la usabilidad y ampliación de la herramienta. Por ejemplo, el guardado de los circuitos, retroalimentación utilizando colores en el uso de los pines y otros componentes, y la detección de cortocircuitos. La creación de un Lenguaje de Dominio Específico (DSL) puede facilitar añadir nuevos sensores y actuadores a la herramienta para su uso por parte de los alumnos. Finalmente, este simulador hará posible la inclusión de técnicas de gamificación para la motivación del alumnado a realizar prácticas fuera de clase, siendo una oportunidad para que el estudiante mejore su aprendizaje y que se plantea incluir próximamente.

4. Agradecimientos

Este proyecto forma parte de la convocatoria 2023/2024 de certificación de proyectos de innovación

docente de UniOvi. A Diego Fernández Suárez y a María Suárez Hevia por sus TFGs con los prototipos iniciales, dirigidos por Cristian González García. Al dpto. de informática de UniOvi por la inscripción. Finalmente, pero no menos importante, queremos agradecer a los revisores los comentarios realizados que han mejorado substancialmente el documento.

Referencias

- [1] Julio Cabero-Almenara y Jesús Costas. La utilización de simuladores para la formación de los alumnos. *Prisma Social*, 2016.
- [2] Dalibor Dobrilovic, Milan Malic y Dusan Malic. Learning platform for smart city application development. *Interdisciplinary Description of Complex Systems*, 17:430–437, 2019.
- [3] Andres Faina. Horosim, a holistic robot simulator: arduino code, electronic circuits and physics. En *Robotics in Education: RiE 2021 12*, pp. 256–267. Springer, 2022.
- [4] Aamir Fidai, Mary Margaret Capraro y Robert M Capraro. “scratch”-ing computational thinking with arduino: A meta-analysis. *Thinking Skills and Creativity*, 38:100726, 2020.
- [5] Cristian Gonzalez Garcia, Daniel Meana-Llorian, Vicente Garcia-Diaz, Andres Camilo Jimenez y John Petearson Anzola. Midgar: Creation of a graphic domain-specific language to generate smart objects for internet of things scenarios using model-driven engineering. *IEEE Access*, 8:141872–141894, 2020.
- [6] Cristian González García, Jordán Pascual Espada, Edward Rolando Nunez Valdez y Vicente García-Díaz. Midgar: Domain-specific language to generate smart objects for an internet of things platform. pp. 352–357. *IEEE*, 7 2014.
- [7] Cristian González García, Ana I. Gómez y Domingo Gómez. Simulador para robots basados en arduino. pp. 323–326. *AENUI*, 7 2023.
- [8] Ana I Gómez, Domingo Gómez y Lara González de Cos. Cnosos: una biblioteca para enrutamiento en arduino car kits. En *Actas de las XXVIII Jornadas sobre la Enseñanza Universitaria de la Informática*, pp. 367–370, 2022.
- [9] Aleksandra Labus, Miloš Radenković, Stefan Nešković, Snežana Popović y Svetlana Mitrović. A smart city iot crowdsensing system based on data streaming architecture. pp. 319–328, 2022.
- [10] Fang Chuan Ou Yang, Hui Min Lai y Yen Wen Wang. Effect of augmented reality-based virtual educational robotics on programming students’ enjoyment of learning, computational thinking skills, and academic achievement. *Computers and Education*, 195, 4 2023.