

Entorno docente con Arduino y Python para Educación Robótica en Secundaria

Julio Vega
Colegio Ntra. S.^a Sagrado Corazón
Grandeza Española, 89, Madrid
julio.vega@sagradocorazonfranciscanas.es

J. M.^a Cañas
Universidad Rey Juan Carlos
Fuenlabrada, Madrid
jmplaza@gsyc.urjc.es

RESUMEN

La habilidad de programar robots está cada vez más demandada y la robótica se percibe cada vez más como una herramienta útil para introducir a los niños conceptos tecnológicos de manera atractiva. La Comunidad de Madrid introdujo el pasado curso académico unas nociones básicas de Programación y Robótica en el currículum oficial de la Educación Secundaria Obligatoria (ESO). Según nuestra experiencia previa en actividades extraescolares con plataformas robóticas educativas como Scratch, Lego Mindstorms, Lego WeDo y otras, la demanda de los estudiantes de la última etapa de enseñanza obligatoria (3º, 4º de ESO) sobrepasa cada vez más la capacidad de desarrollo de tales plataformas aunque sean buenas para niveles académicos inferiores. Presentamos aquí una plataforma educativa basada en microprocesador Arduino y lenguaje Python que permite adentrarse en el mundo de la Informática, la Electrónica y la Robótica y por su potencia escala en complejidad a aplicaciones más interesantes para los estudiantes de esos cursos.

Palabras Clave

Education, Robotics, Teaching, Arduino, Python

1. INTRODUCCIÓN

La educación en robótica en secundaria está cobrando una importancia creciente. Tiene poder de motivación en los estudiantes y eso permite acercar la tecnología a los niños, usando la robótica como herramienta para exponerles a conceptos básicos de ciencias, tecnología, ingeniería y matemáticas (STEM *Science, Technology, Engineering and Math*). Los alumnos aprenden, casi mediante un juego, nociones complejas difíciles de explicar o asimilar mediante la clásica clase magistral. La Comunidad de Madrid (Decreto 48/2015) ha introducido recientemente la asignatura *Tecnología, programación y robótica* en el currículum oficial de Educación Secundaria Obligatoria. Por otro lado, la aparición de aplicaciones robóticas en el mercado masivo como las aspiradoras robóticas, los coches autónomos o los drones hacen más visible la utilidad de esta tecnología y refuerzan su atractivo.

En la praxis de la enseñanza robótica en educación secundaria son frecuentes plataformas como los robots LEGO (RCX, NXT, Ev3, WeDo) u otros robots como mBot o Zowie de BQ que incluyen placas con procesadores Ar-

duino a los que se conectan sensores de bajo coste y servos [Mubin *et al.*, 2013]. Se enseña el funcionamiento básico de sensores, actuadores y los rudimentos de la programación. Se usan lenguajes sencillos que facilitan su programación por niños, como los lenguajes gráficos RCX-code y recientemente Scratch o Blockly [Jiménez *et al.*, 2010; Cerezo y Sastrón, 2015]. No sólo hacen énfasis en la construcción mecánica del robot. Cada vez es más patente que la funcionalidad reside fundamentalmente en la programación [Jamieson, 2012]). Además, los alumnos se divierten más con robots programables, que pueden hacer varias cosas según ellos los programen.

En nuestra experiencia, la impartición con Scratch y plataformas Lego Mindstorms resulta ideal en los primeros cursos de Educación Secundaria, para iniciarse en el mundo de la Programación en general, y en Robótica en particular. El aprendizaje inicial es casi inmediato, lo que conlleva un gran entusiasmo por hacer, diseñar y programar a toda velocidad. Es una plataforma muy versátil en cuanto al diseño mecánico, aunque no tanto en cuanto a la programación. En los últimos cursos, dado que ya tienen cierta soltura en programación adquirida en los años anteriores, ese entusiasmo por hacer cosas nuevas disminuye considerablemente. Percibimos ciertas carencias y limitaciones en cuanto a desarrollo se refiere. Los alumnos de cursos altos se acaban aburriendo de un lenguaje tan limitado como Scratch.

Como continuación de nuestra experiencia con el uso de Lego Mindstorms en Educación Secundaria [Vega y Cañas, 2014] en este artículo presentamos un entorno docente y el contenido de un curso concreto que se han diseñado para superar esas limitaciones. Este entorno utiliza Arduino como plataforma hardware y lenguaje Python como ejes principales. Arduino se ha usado en otros muchos cursos de enseñanza preuniversitaria [Araujo *et al.*, 2015]. Hemos elaborado un programa educativo concreto (sección 3) que consta de varias prácticas organizadas en diferentes fases: desde la familiarización con la programación y el lenguaje Python hasta las prácticas con un robot Arduino real, sus sensores, actuadores y comportamientos autónomos.

El programa propuesto se ha utilizado en el Colegio Nuestra Señora del Sagrado Corazón con 27 alumnos de edades comprendidas entre los 15 y 16 años. En la sala de informática del colegio se cuenta con treinta ordenadores con Sistema Operativo Linux Edubuntu. Tenemos cinco unidades completas (kits) de Arduino. Éstos incluyen la placa de Arduino, dos servos, sensores de luz, infrarrojos, ultrasonidos, así como numerosos componentes enfocados a la construcción de un robot móvil totalmente equipado.

2. DISEÑO Y PLATAFORMA DE DESARROLLO

El entorno docente diseñado se muestra en la Figura 1. La plataforma hardware con la que se realizan las prácticas consta de un robot móvil con procesador Arduino al que se conectan varios sensores (de luz, ultrasonidos...) y actuadores (motores, luces, zumbador). Este robot se conecta mediante un cable USB a un ordenador personal, en el cual ejecuta el programa escrito por el alumno en lenguaje Python.

Este diseño ofrece a los alumnos la experiencia con dispositivos reales, más enriquecedora que con simuladores. Además las prácticas fomentan el paradigma de *aprender haciendo*, de aprendizaje activo. La interactividad de los alumnos con los robots facilita el aprendizaje y asentamiento de los conceptos teóricos, los algoritmos y técnicas.

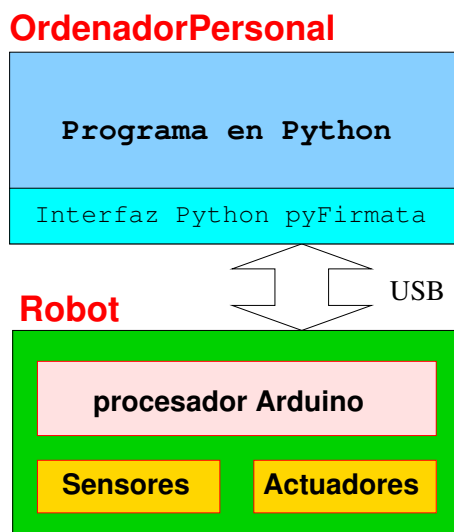


Figure 1: Diseño del entorno de prácticas

Los motivos para elegir Arduino como plataforma hardware robótica son varios: es un entorno sencillo a la par que versátil y más avanzado que otros entornos como Lego Mindstorms, es barato y de hardware libre. Es un entorno completo, flexible, de complejidad acotada y relativamente potente.

En cuanto al lenguaje de programación, Arduino utiliza una versión simplificada de C, que es un lenguaje muy popular, aunque difícil de aprender por los estudiantes que se inician en la programación de ordenadores. Por ello y su rigidez de sintaxis lo descartamos. Hemos elegido el lenguaje Python para que los estudiantes escriban sus programas robóticos. Python es más intuitivo, menos estricto, lo que resulta de gran ayuda a los estudiantes que se están iniciando en el mundo de la programación. Además es rápido de aprender, fácil de escribir y tiene una gran potencia expresiva. Por otro lado, es un lenguaje que se utiliza cada vez más a nivel universitario, con lo que la formación académica de aquellos alumnos que deseen continuar sus estudios y decantarse por alguna carrera técnica se verá altamente enriquecida. Además, prescinde de compilador, dado que es un lenguaje interpretado.

2.1 Plataforma hardware Arduino

Hay infinidad de placas basadas en Arduino ya que, al ser *open-source*, cualquiera que quiera hacer una placa puede hacerlo. Por ello existen multitud de modelos Arduino disponibles, lo que puede ser un poco confuso para los nuevos usuarios. A la hora de elegir lo normal es que nos fijemos en la cantidad de entradas y salidas que tiene, especialmente las analógicas dado que son las que habitualmente restringen la complejidad de la aplicación posible. En la Figura 2 se muestran los modelos de Arduino más habituales.



Figure 2: Placas de Arduino más habituales.

La placa más empleada es la Arduino UNO r3. Ésta es la que usamos nosotros en nuestro programa, ya que, para nuestros proyectos, que no son muy grandes y requieren manejar sólo algunos servos y sensores sencillos. Tenemos suficiente con ella. Una posible mejora de cara a posibles proyectos futuros, más complejos, con más servos y sensores, puede ser pasar a un modelo con mayor número de salidas, especialmente analógicas, como la Arduino MEGA r3.

Los sensores que utilizamos en nuestro programa educativo son: infrarrojos, ultrasonidos y luz. Asimismo, en cuanto a actuadores, empleamos un total de tres servos, y algunos complementos que nos reportan sobretodo información, tanto visual como acústica, como son los LEDs y los zumbadores. Todo ello conectado mediante cableado electrónico a una *protoboard* que facilita el trabajo con estos elementos de tamaño reducido. El cerebro de todo este entramado es la placa de Arduino UNO r3. Y todo está montado sobre un chasis que se asienta en un par de ruedas y una bola loca. El montaje completo se muestra en la Figura 3.

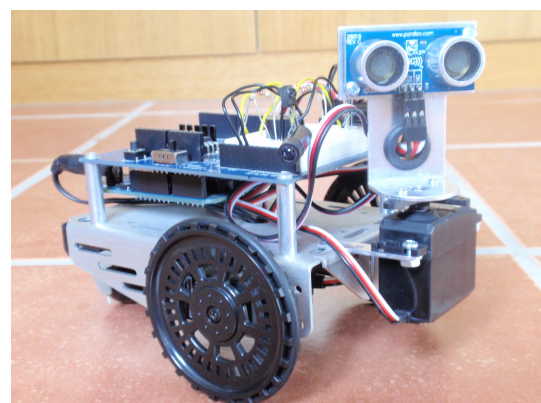


Figure 3: Robot completo con placa de Arduino.

2.2 Kit software Arduino IDE

La manera habitual de programar un robot Arduino es usando el Arduino IDE, que se muestra en la Figura 4. Éste se puede descargar gratuitamente desde su página oficial¹, y es multiplataforma, está disponible para distintos sistemas operativos: Windows, Linux o MacOS. Para cargar el programa en la placa es necesario conectarlo al PC mediante el puerto USB, mientras que la ejecución del algoritmo puede efectuarse con el robot conectado al PC, para así obtener los 5V que necesita para funcionar, o proporcionarle la energía mediante batería (cinco pilas AA en un soporte existente).

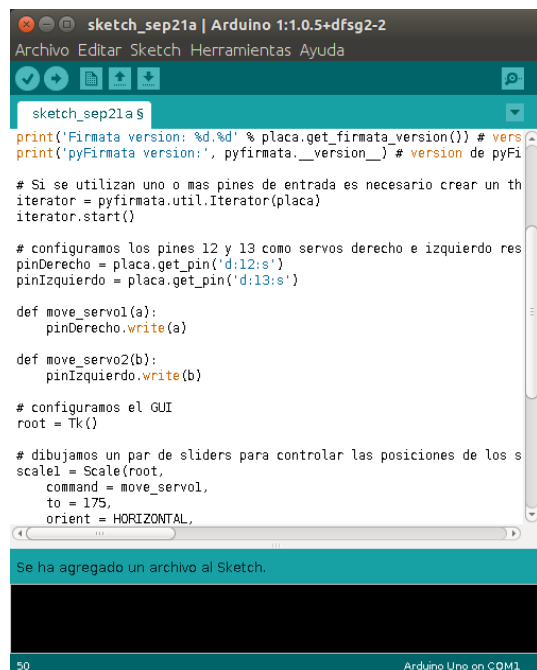


Figure 4: Entorno de programación Arduino IDE.

El IDE de Arduino es una versión reducida del lenguaje C/C++, que no es lo más indicado para iniciarse en la programación de instrucciones de código. Observamos un gran salto de los coloridos bloques de programación de Lego Mindstorms, más propios de edades tempranas; a las clásicas líneas de código de Arduino (C, C++), con sus formalismos y sus estrictas normas de escritura. Las características que ofrece este entorno son:

- Tiene un lenguaje simple, basado en C/C++.
- Permite desde un primer momento estar programando directamente el hardware.
- Es un proyecto *open-source*, por lo que debido a su precio podemos probar y experimentar sobre la misma tarjeta.
- Tiene una comunidad de desarrollo alrededor que permite un acceso a referencias, ejemplos y proyectos de gran ayuda.

¹<https://www.arduino.cc/en/Main/Software>

Para poder usar los desarrollos hechos en lenguaje Python con la placa de Arduino hacemos uso de la librería **pyFirmata**, cuyo módulo es facilitado por el propio IDE de Arduino, que incluye un amplio repertorio de bibliotecas como ésta. Cuando esta biblioteca está cargada en Arduino, ese microprocesador admite órdenes por el puerto USB, que proceden precisamente de otro programa, en Python, ejecutando en el ordenador personal (PC). El programa en Python se escribe en el PC con cualquier editor de texto.

Los pasos para programar una placa Arduino mediante código Python son:

1. Abrir con el IDE de Arduino la librería **pyFirmata**, incluída en el mismo.
2. Conectar la placa de Arduino por USB al PC y cargar dicha librería en el procesador Arduino.
3. Ejecutar nuestro código Python en el PC como se haría normalmente.

3. PROGRAMA EDUCATIVO

Se ha diseñado y llevado a cabo un plan de actividades para la asignatura de *Programación, Robótica y Tecnología* en el 4º curso de Educación Secundaria. Como los estudiantes con los que tratamos no tienen todavía ninguna noción de Programación de ordenadores, hemos de partir de un nivel muy básico de programación hasta llegar en última instancia al desarrollo de un proyecto complejo consistente en alguna tarea clásica robótica. Lo hemos dividido en varias fases de aprendizaje:

1. Nociones básicas teóricas de la Programación de ordenadores. Bucles, condiciones, variables, funciones, etc. Familiarización con la Programación mediante lenguaje de Pseudocódigo.
2. Conceptos y claves teóricas del lenguaje Python. Paso de Pseudocódigo a lenguaje de programación. Desarrollo de prácticas básicas en lenguaje Python.
3. Desarrollo de prácticas robóticas en lenguaje Python.
4. Proyecto completo robótico: tarea clásica de un robot.

Cada una de ellas se describe a continuación, indicando qué comprenden, qué labores prácticas desarrollan los alumnos a medida que va avanzando el programa, y qué carencias encontramos al subir cada peldaño.

3.1 Nociones básicas de programación

En la primera parte del curso damos nociones básicas de Programación de ordenadores. Así los alumnos comprenden la forma de trabajar de un ordenador a nivel interno; sólo así más adelante entienden el por qué del uso de variables o funciones. Por otro lado, conceptos como bucles o condicionales son totalmente nuevos para ellos. De ahí que esta primera toma de contacto sea tan importante. Dependiendo del alumnado, esto suele durar unas cinco sesiones.

Después se dedican otras cinco sesiones a materializar las nociones básicas aprendidas en un lenguaje cercano a ellos, un lenguaje intuitivo, como es el Lenguaje de Pseudocódigo. Aquí matizamos algunos aspectos sintácticos a grandes rasgos, así como conceptos novedosos que siguen apareciendo, como el concepto de un contador, o del uso de conjuntos o

vectores. Entienden también el por qué de la definición de una variable, y algunos otros flecos menores. Es una fase muy importante donde los alumnos interiorizan la estructura, organización y restricciones de un lenguaje de programación.

3.2 Lenguaje Python

En la segunda fase ya concretamos el lenguaje, en nuestro caso Python. Nuevamente dedicamos varias clases, unas cuatro sesiones, a que conozcan palabras clave del lenguaje en cuestión, así como matizar cuestiones meramente sintácticas y propias de este lenguaje. Llegados a este punto, los alumnos ya están preparados para realizar algunos ejercicios clásicos de iniciación a la programación como estos:

1. Programa que imprima los números del 1 al 100.
2. Programa que imprima los números del 100 al 0.
3. Programa que imprima los números pares entre 0 y 100.
4. Programa que imprima la suma de los 100 primeros números.
5. Programa que imprima los números impares hasta el 100 y que imprima cuántos impares hay.
6. Programa que imprima todos los números naturales que hay desde la unidad hasta un número que introducimos por teclado.

La realización, corrección y explicación (en ese orden) de esos ejercicios llevan seis sesiones.

3.3 Prácticas con Arduino: sensores y actuadores

En la tercera fase los alumnos ya llevan a cabo prácticas más complejas y directamente relacionadas con la Robótica. Comienzan construyendo el kit de Arduino y repasando algunos conceptos básicos de electrónica para que, a la hora de conectar los distintos dispositivos, no tengan ningún problema. Poco a poco comienzan por la instalación de componentes sencillos, como un zumbador, o LEDs, y sus correspondientes desarrollos software, para pasar a otros más complejos como sensores de luz, infrarrojo o ultrasonido. Una vez tengan dominados los sensores, pasan a dominar el control de servos. Todo ello por separado, durante 10 sesiones aproximadamente, del siguiente modo:

1. Tratamiento de entrada/salida de fichero. Lectura de imagen estática, usando OpenCV.
2. Imagen dinámica. Lectura desde WebCam.
3. Uso de LEDs y zumbador. Prácticas varias de combinación de los mismos.
4. Lectura de sensores complejos: luz, infrarrojos, ultrasonidos.
5. Control de servos.

3.4 Prácticas con Arduino: comportamientos

El último peldaño de esta pirámide de aprendizaje consiste en la elaboración de un proyecto completo de robótica, donde combinan todo lo aprendido anteriormente. Por ejemplo, en este último curso el proyecto consistió en diseñar y programar un robot que navegara siguiendo la luz proyectada por el flash de un teléfono móvil y que, a la vez, esquivara los posibles obstáculos que encontrara por el camino. Les lleva otras 10 sesiones acabar este proyecto final.

4. CONCLUSIONES

La Robótica Educativa es un área en auge. La sociedad demanda cada vez más habilidades en cuanto a diseño, construcción y programación de robots. Aunque el diseño mecánico y construcción de una máquina autónoma puedan resultar asequibles para un adolescente, no lo es tanto su programación.

Existen en el mercado multitud de plataformas robóticas educativas cuyo propósito es que el estudiante pueda trabajar con el artefacto robótico inmediatamente, lo cual es ideal. No obstante, esa fácil iniciación tiene un límite de aprendizaje. Suelen ser kits que en un periodo de tiempo no muy largo dejan de entusiasmar a los jóvenes aprendices. Por ello, integrar algún lenguaje de programación más versátil y maduro, como es Python, en una plataforma *open-source* muy rica en posibilidades de desarrollo, como es Arduino, resulta ser una combinación perfecta: aprendizaje inicial asequible, fácil manejo del hardware, y amplias posibilidades de desarrollos robóticos.

Como líneas futuras, estamos trabajando en ampliar el repertorio de prácticas. También en conseguir una conexión inalámbrica entre PC y Arduino (*Bluetooth*, *Wifi*...) y explorando la posibilidad de ejecutar el código Python directamente embarcado en Arduino.

5. BIBLIOGRAFÍA

- [Araujo *et al.*, 2015] Andre Araujo, David Portugal, Micael S. Couceiro, y Rui P. Rocha. Integrating arduino-based educational mobile robots in ros. *J Intell Robot Syst* (2015) 77:281–298, 2015.
- [Cerezo y Sastrón, 2015] F. Cerezo y F. Sastrón. Laboratorios virtuales y docencia de la automática en la formación tecnológica de base de alumnos preuniversitarios. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 12(4):419–431, 2015.
- [Jamieson, 2012] Peter Jamieson. Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat? *Miami University, Oxford, OH*, 45056, 2012.
- [Jiménez *et al.*, 2010] E. Jiménez, E. Bravo, y E. Bacca. Tool for experimenting with concepts of mobile robotics as applied to children education. *IEEE Trans. Education*, 53(1):88–95, 2010.
- [Mubin *et al.*, 2013] Omar Mubin, Catherine J. Stevens, y Suleman Shahid. A review of the applicability of robots in education. *Technology for Education and Learning*, 2013, 2013.
- [Vega y Cañas, 2014] Julio Vega y J.M. Cañas. Curso de robótica en educación secundaria usando constructivismo pedagógico. *JITICE 4th Workshop, Educational Innovation and ICT, U. Rey Juan Carlos, November 26th 2014. ISSN: 2172-6620*, 2014.