

Salomón Obed Pazmiño Chávez

Programación en Arduino vs. circuitos tradicionales

El uso educativo de los circuitos tradicionales que en electrónica se armaban y se ponían en funcionamiento implicaban una inversión considerable en tiempo y dinero (Figura 1). Esta forma de hacer electrónica se ha ido quedando atrás con la última tecnología de programación en tarjetas microcontroladoras, como Arduino (Figura 2).



Figura 1. Tarjeta de pruebas (protoboard) con algunos componentes electrónicos tradicionales

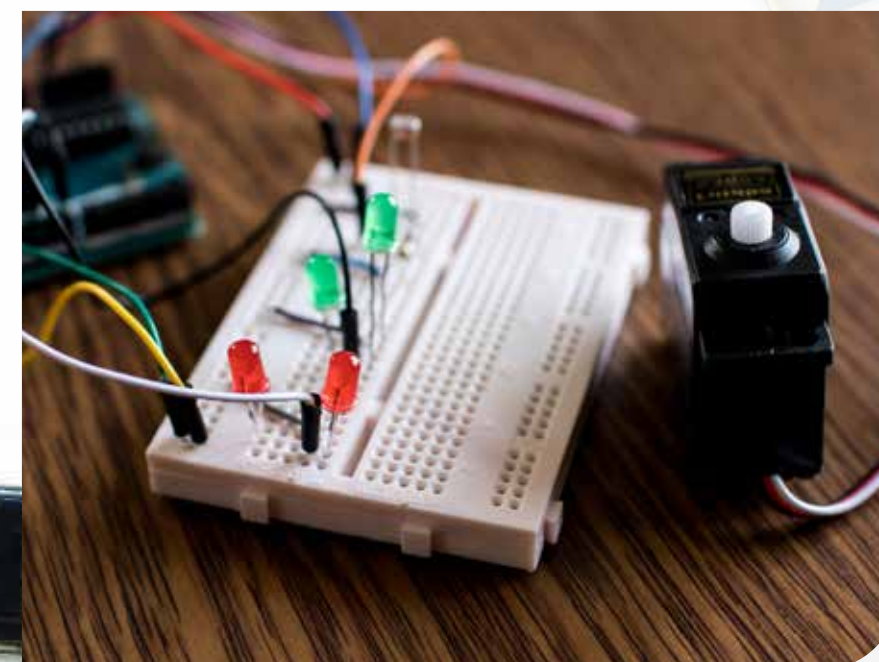


Figura 2. Tarjeta microcontroladora Arduino

Sin intención de subestimar los aprendizajes obtenidos con el uso de los circuitos tradicionales, que sustentaban de mejor manera el modelo constructivista, pero sin alejarnos de este modelo pedagógico, se puede afirmar que los códigos de los circuitos integrados mediante Arduino están revolucionando los aprendizajes significativos y está dando paso también a experiencias educativas con la robótica.

En años anteriores se dio mucha importancia a los aprendizajes significativos, recuerdo los ladrillos legos que tenían en el Departamento de Consejería Estudiantil (DECE), que solo se utilizaban para los estudiantes con problemas de aprendizaje, y actualmente creo que hubiera sido mejor que todos los alumnos de primaria y secundaria los usaran y desarrollaran las destrezas requeridas.

Actualmente se está implementando la nueva educación a distancia, se está programando en Arduino y de la mano con la robótica aunque, lamentablemente, algunos creadores solo lo ven con un buen negocio.

El lenguaje de programación C++

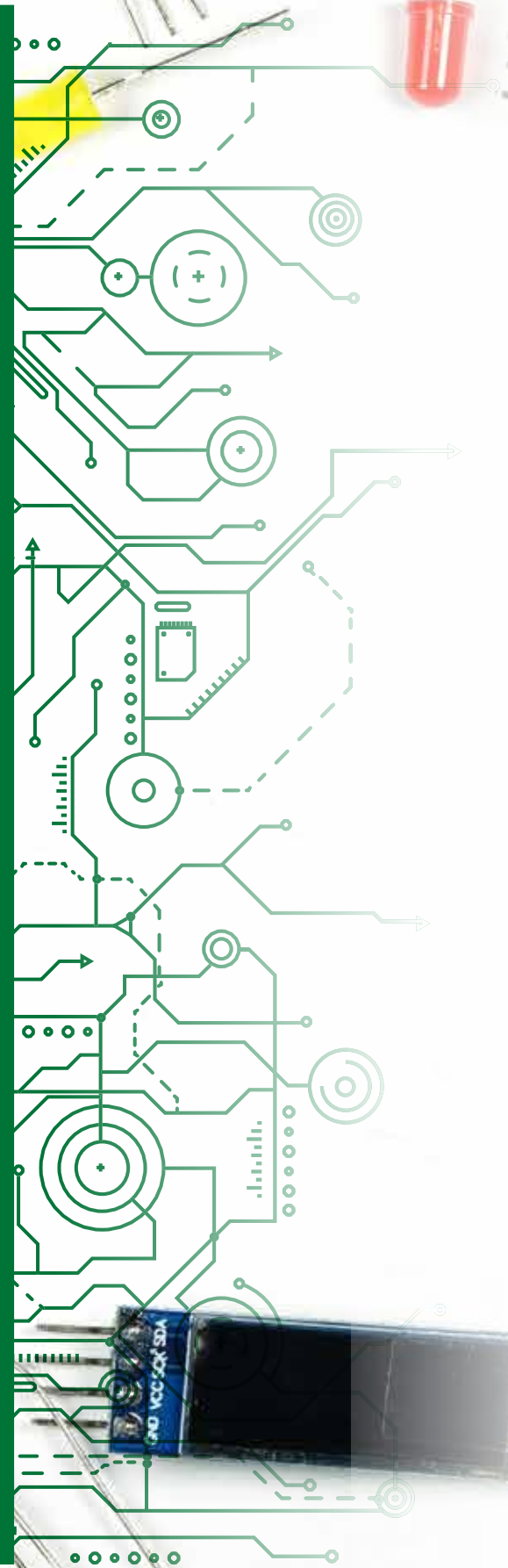
El lenguaje de programación de Arduino está basado en C++ y la referencia la encontramos en la siguiente dirección: <http://arduino.cc/en/Reference/HomePage>, también es posible usar comandos estándar de C++ en la programación de Arduino.

C++ es un lenguaje de programación que fue diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue extender el lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente, se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

Actualmente existe un estándar, denominado ISO C++ y se puede consultar en:

<https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>



El lenguaje que interpretan los micros controladores, pic's y microprocesadores es el código binario, es por esa razón que se utilizaban algunos lenguajes de programación como C++, Visual Basic, Java, Java script, etcétera. Sin embargo, para algunos usuarios de programación era demasiado desgastante y requería de conocimientos previos.

Hoy por hoy en pleno siglo XXI hasta un niño o niña podría experimentar programando comandos para movimientos de un robot, por ejemplo.

De todas maneras no deja de ser importante el uso del protoboard, por que mediante ensayo error los estudiantes y profesores van adquiriendo la experiencia necesaria y no es otra cosa que se resume en lo siguiente: programo un circuito, funciona o no funciona, así de simple, también se sigue usando las tablillas de baquelita, los pcb y los protoboard que combina la programación de comandos, se integra un circuito o un protoboard para experimentar los resultados propuestos, por ejemplo, el efecto de una fotocelda, el control de voltaje por ancho de pulso, un sensor de movimiento, etcétera.

Soluciones sencillas

Para elaborar los circuitos tradicionales, por ejemplo, las luces intermitentes, se tenía que armar el circuito para obtener como resultado final que uno o varios diodos led se enciendan intermitentemente (Figura 3).

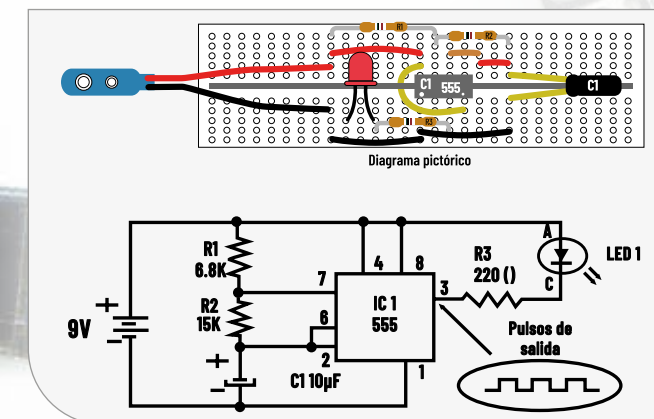


Figura 3. Esquema para instalar un circuito integrado digital en una tarjeta de prueba

Hoy por hoy, se puede obtener el mismo resultado con una simple programación de códigos en Arduino, como en el ejemplo de programación de la Figura 4.



Figura 4. Programación para ejecutar en una tarjeta Arduino

“El Arduino es una tarjeta programable en lenguajes de alto nivel que contiene un micro controlador ARM que en esencia es prácticamente una computadora. Los creadores de Arduino han tenido mucho éxito con su sistema y probablemente una de las razones es que tanto el hardware como el software son abiertos, es decir, los desarrolladores han creado una plataforma que cualquiera puede copiar y en algún caso mejorar, sin tener que pagarles regalías. Tal vez el truco esté en que Arduino hace una serie de tarjetas -como la popular Arduino UNO- que resulta más práctico y barato, que armar de cero la propia. El hecho de poner todo en un entorno abierto da la posibilidad que mucha gente se involucre y los productos mejoren significativamente.”

(López, 2016).

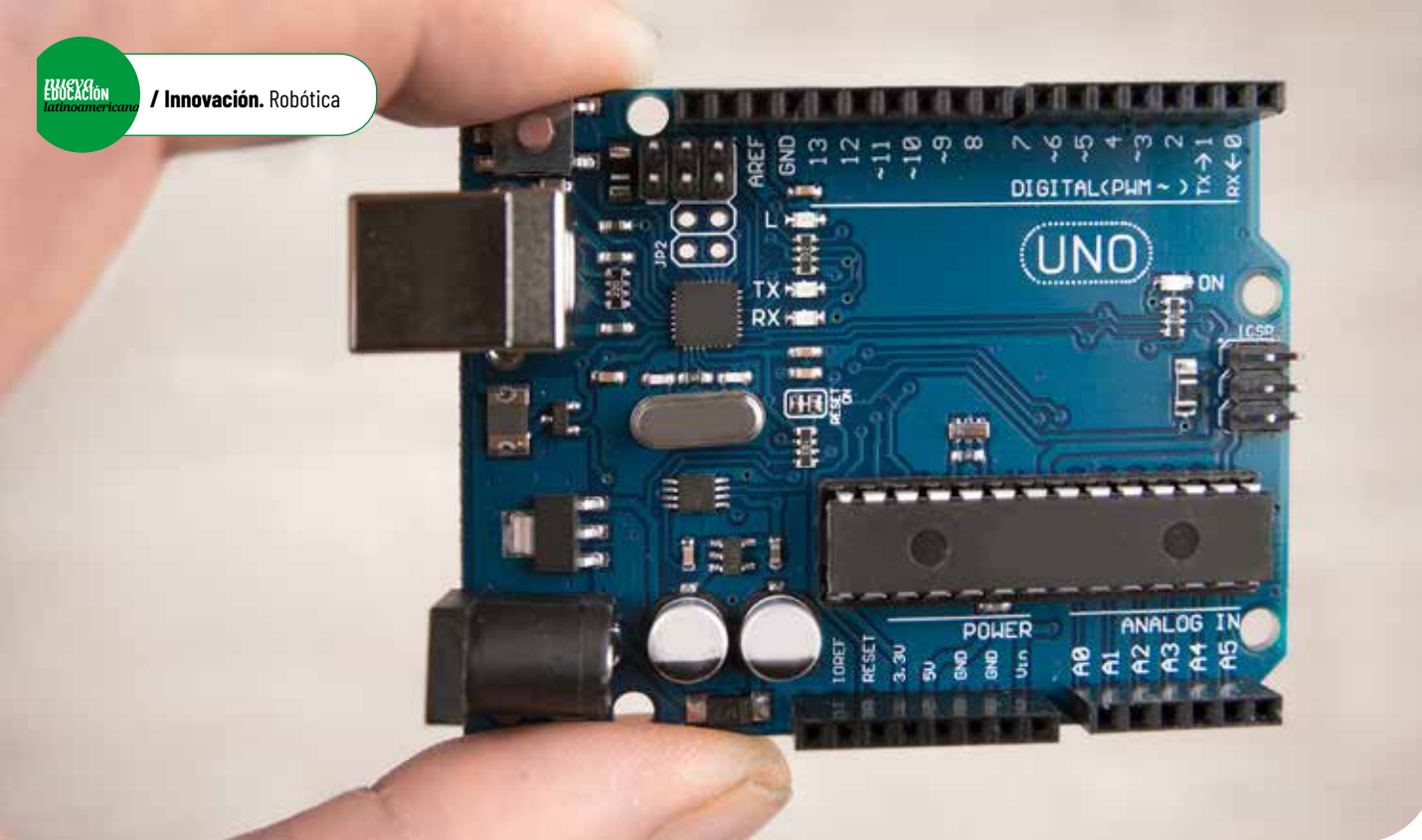


Figura 5. Un modelo de tarjeta Arduino

El ingreso de la programación será entonces la función: `Void - Setup ()`
`Delay (ms);` será entonces el tiempo que el diodo led cambia de estado de *off* a *on*.
 Causando el mismo efecto eléctrico de un tren de pulsos cuadrados.

Conclusiones

Concluyendo con lo que hemos visto hasta aquí, podríamos decir sin temor a equivocarnos que la programación de nuestro circuito integrado con el software Arduino UNO es una forma excelente de sustituir los circuitos tradicionales por una sola programación en Arduino, ahorrando recursos y tiempo, además de eso, la tecnología está dando pasos agigantados. Por cual es necesario que los niños ya demuestren su interés por la robótica, que será la mejor manera de motivarlos al aprendizaje significativo y constructivista. De todos modos, no quiero decir que la forma en que aprendimos electrónica en nuestras protoboard no haya servido.

Como señalan Vidal *et al* (2019) "Arduino facilita el desarrollo de soluciones con una simbiosis de electrónica y computación para estudiantes de una baja o nula formación previa en electrónica y computación[...]" En resumen, la experiencia en aula muestra que este recurso tecnológico es viable para apoyar a los estudiantes en el desarrollo de competencias para la programación, el diseño de sistemas y la electrónica aplicada a la solución de problemas, y no requiere nociones complejas de programación o electrónica.

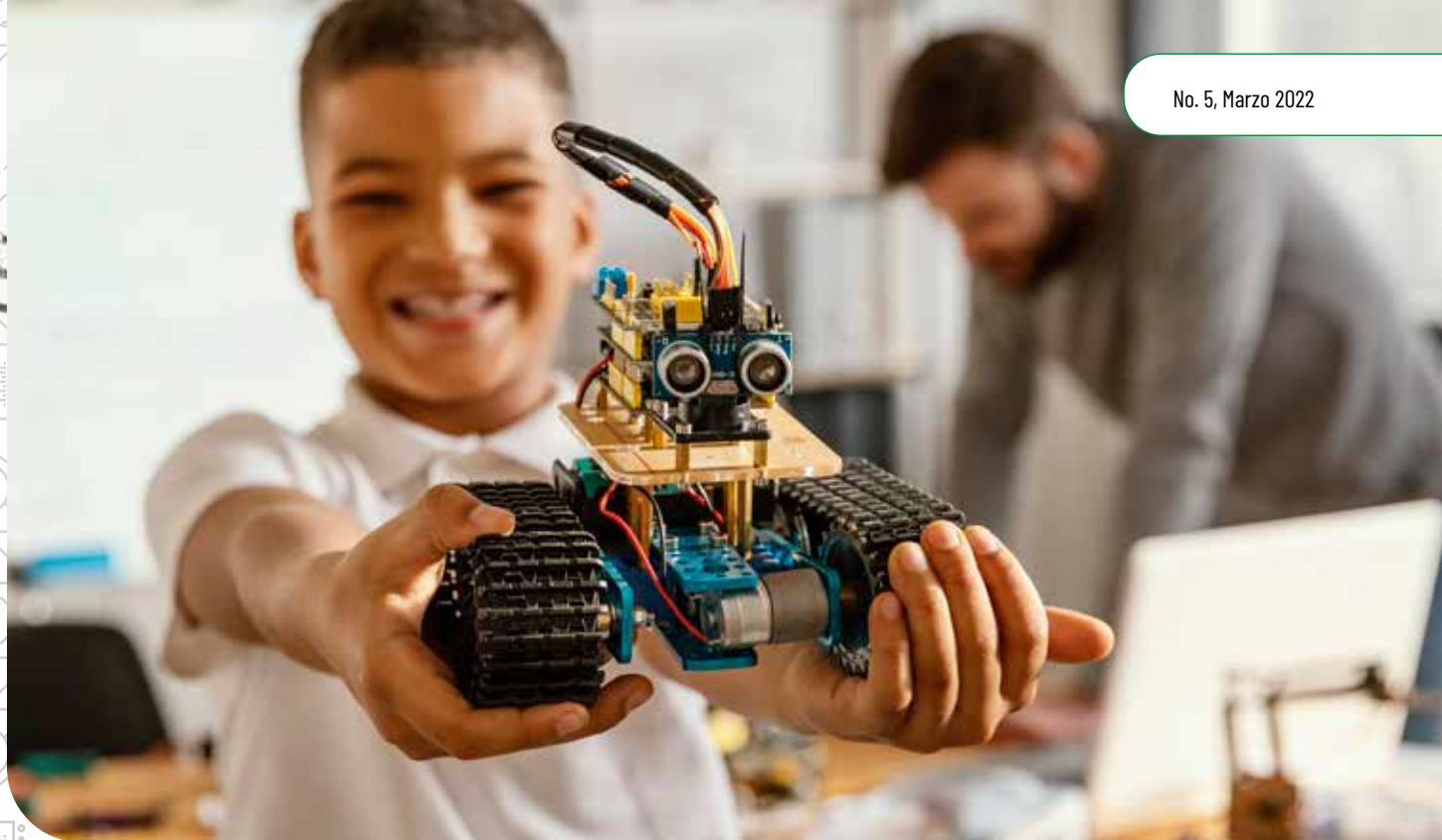
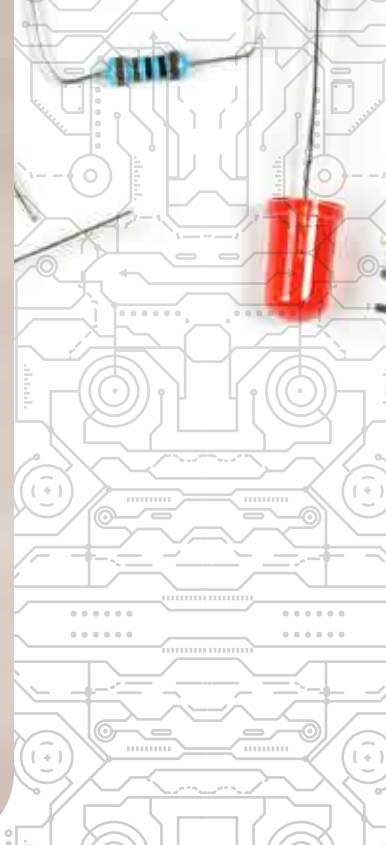


Figura 6. Estudiantes trabajan con Arduino

Referencias

- Arduino (2022) *Language Reference*. <http://arduino.cc/en/Reference/HomePage>
- Crespo, Enrique (2015) *Lenguaje de programación C++*. Aprendiendo Arduino. Wordpress. <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>
- López, Manuel (2016, 20 de abril) Lenguajes para usarse en la plataforma Arduino. Unocero. <https://www.unocero.com/noticias/lenguajes-para-usarse-en-la-plataforma-arduino/>
- Vidal, Cristian; Lineros, Matías; Uribe, Guillermo y César Olmos (2019) *Electrónica para todos con el uso de Arduino: experiencias positivas en la implementación de soluciones Hardware-Software*, en *Información tecnológica* vol.30 no.6 La Serena dic. 2019. <https://www.scielo.cl/scielo.php?script=sci-arttext&pid=S0718-07642019000600377&lang=pt>

Ficha del autor

Salomón Obed Pazmiño Chávez: eldurodeldibujo@gmail.com

Licenciado en Ciencias de la Educación, Mención: Matemática. Es Docente de educación secundaria en Pichincha, Ecuador, para las asignaturas de Dibujo técnico, Educación artística, Matemática, Física y cálculo. Cursa la Maestría en Ciencias de la Educación.