

Arduino IDE

Arduino IDE

Iván H. Pérez-Tavera ^a

Abstract:

The advancement of technology has given rise to the generation of tools that help us to carry out technological solutions to solve some problem in the social environment, this type of tool has been very accessible to be applied today in the classroom, where students develop their capacities for innovation and creativity, capturing them in the realization of technological projects that are capable of opening a panorama towards high-demand innovation projects and one of these tools is the ARDUINO board, which, in order to be used, has its development software called IDE that allows you to program the necessary instructions so that the components used can work and perform specific tasks.

Keywords:

Arduino, IDE, Programming, Technological Solutions

Resumen:

El avance de la tecnología ha dado pie a la generación de herramientas que nos ayuden a realizar soluciones tecnológicas para solventar algún problema de entorno social, este tipo de herramientas han sido muy accesibles para poder aplicarlo hoy día dentro de las aulas de clases, en donde los alumnos desarrollan sus capacidades de innovación y creatividad, plasmándolas en la realización de proyectos tecnológicos que son capaz de abrir un panorama hacia proyectos de innovación de alta demanda y una de estas herramientas es la placa de ARDUINO, misma que para poder ser utilizada cuenta con su software de desarrollo denominado IDE que permite programar las instrucciones necesarias para que los componentes utilizados puedan funcionar y realizar tareas específicas.

Palabras Clave:

Arduino, IDE, Programación, Soluciones Tecnológicas

Arduino IDE

El entorno de programación de la placa de Arduino se denomina *Integrated Development Environment* (IDE) el cual permite llevar a cabo la escritura de las sentencias para el funcionamiento de los elementos físicos de la placa de Arduino.

Este software tiene por sí solo un conjunto de herramientas que permite, editar el código, compilar y depurar todo a través de una interfaz gráfica, así mismo, nos da la oportunidad de interactuar con el microcontrolador almacenando los programas realizados en su memoria interna para poner en marcha todo el hardware.

El aplicativo de IDE es un software libre, ya que está disponible su código fuente mismo que está alojado en GitHub y ofrece instrucciones de compilado.

Interfaz Arduino IDE

Este entorno de desarrollo que incorpora Arduino se conforma por un editor de código, un área de consola de mensajes y una barra que permite compilar y cargar los archivos a la memoria del microcontrolador.

Dentro de esta interfaz es muy sencillo comenzar a programar, lo único que debemos hacer siempre antes de iniciar es indicarle al software que placa de Arduino se estará utilizando y el puerto de comunicación. Esta se divide en tres secciones **Opciones del IDE (1)** el cual se encuentra en la parte superior, **el área de edición (2)** en el cuál plasmaremos el código de programación y **el centro de mensajes (3)** que nos muestra la consola. Figura 1

^a Iván Horacio Pérez Tavera, Universidad Autónoma del Estado de Hidalgo, Escuela Preparatoria Número 4, <https://orcid.org/0000-0001-7708-6912>, Email: itavera@uaeh.edu.mx

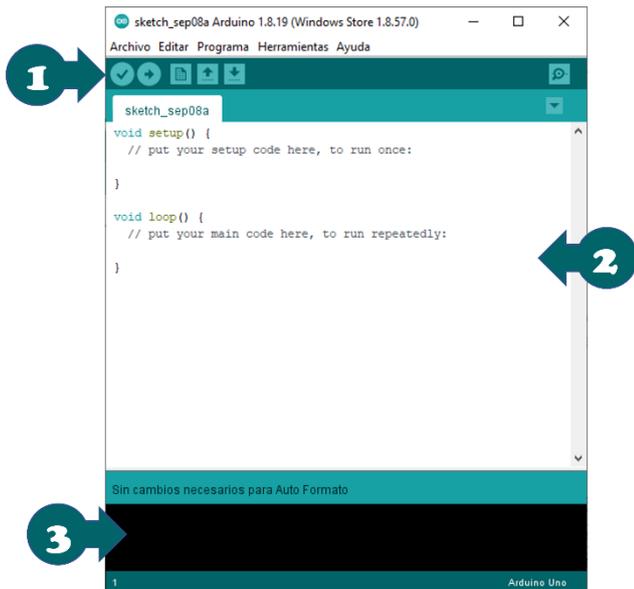


Figura 1 Pantalla principal Arduino IDE

Una de las funciones principales que nos brinda es la validación de nuestro código a nivel de sintaxis, ya que al compilar hace una revisión línea por línea y en el centro de mensajes nos muestra los errores encontrados en cada una de ellas, una vez verificado y que no contenga errores lo compila y lo carga al microcontrolador, todo ello de manera simple pulsando el botón de acceso rápido que se encuentra en la parte superior de la interfaz.

Otra de las opciones que nos brinda el IDE es su **monitor serie** el cual permite visualizar los valores que va leyendo nuestra placa como datos de entrada, así como también nos muestra todos los valores de salida para una mejor comprensión de los objetos utilizados y programados.

El lenguaje que utiliza el IDE de Arduino está basado en **C/C++** de una manera más simplificada, ofreciendo la oportunidad de cargar librerías que sean necesarias para el buen funcionamiento de nuestros proyectos. Así mismo nos da la oportunidad de cargar varios proyectos dentro de una misma carpeta, cada uno de ellos se guardan con la extensión **ino** los cuales se conocen con el nombre de **“sketches”**

Para poder iniciar a programar se deben de considerar dos funciones de manera obligatoria **void setup** y **void loop**, mismas que se observaran de manera predeterminada cada que iniciemos el IDE de Arduino.

Void setup(): es la primera función que encontraremos y que ejecutara el programa, en ella prácticamente deberemos inicializar nuestras funciones que se ejecutarán en el microcontrolador, se establecen criterios

que se considerarán por única ocasión, así como los pines que se utilizarán como parámetros de salida con la instrucción **OUTPUT**

Void loop(): la función denominada como bucle se ejecuta infinitamente hasta que apaga el microcontrolador o se reinicia, todas las sentencias escritas dentro de esta, se activan en orden, es decir, de arriba hacia abajo, al llegar a la última instrucción regresa y vuelve a iniciar.

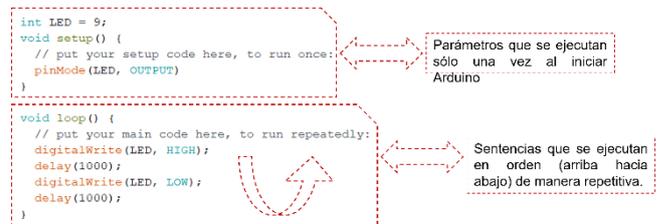


Figura 2 Descripción de funciones

Dentro del IDE de Arduino podemos hacer uso de diferentes funciones para controlar la tarjeta de Arduino y realizar diferentes cálculos:

Entrada y Salida digital: digitalRead(), digitalWrite() y pinMode().

Entrada y Salida análoga: analogRead(), analogReference() y analogWrite().

Matemáticas: abs(), constrain(), map(), max(), min(), pow(), sq() y sqrt().

Entrada y Salida avanzado: noTone(), pulseIn(), pulseInLong(), shiftIn(), shiftOut() y tone().

Tiempo: delay(), delayMicroseconds(), micros() y millis()

Conclusiones

A pesar de que el IDE de Arduino es una interfaz de lo más sencilla de utilizar, está muy completa para lograr proyectos muy significativos, así también, no debemos perder de vista que no es el único editor de código que podemos encontrar para poder programar nuestro microcontrolador, tenemos algunos otros que incorporan más funciones y herramientas, sin embargo, estamos hablando que ya es necesario un conocimiento más especializado para explotar su potencial.

Referencias

[1] Aguayo, P. (2022, 8 julio). Software de Arduino. Arduino.cl - Compra tu Arduino en Línea. Recuperado 9 de septiembre de 2022, de <https://arduino.cl/programacion/#:%7E:text=EI%20IDE%20de%20Arduino%20nos,de%20Arduino%2C%20a%20su%20accesibilidad.>

- [2] Arduino Integrated Development Environment (IDE) v1. (s. f.). Arduino Documentation. Recuperado 9 de septiembre de 2022, de <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>.
- [3] Crespo, J. (2016, 11 diciembre). IDE Arduino. Aprendiendo Arduino. Recuperado 9 de septiembre de 2022, de <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- [4] Guía de Referencia de Arduino - Guía de Referencia de Arduino. (s. f.). Recuperado 9 de septiembre de 2022, de <https://www.arduino.cc/reference/es>.