

Modulo_3:

Arduino, que es y como funciona

FORO: <https://nac-arduino.herokuapp.com/>

REPOSITORIO: <https://github.com/rody7val/nac-arduino/>

Contenidos:

1. [Que es Arduino?.](#)
2. [Por que Arduino?.](#)
3. [Para que sirve?.](#)
4. [Hardware.](#)
5. [Software.](#)

Que es Arduino?:

Arduino es una **plataforma** de prototipos electrónicos.

Está basada en hardware y software **flexible** y **fácil de usar** para cualquier persona y en cualquier proyecto electrónico.

Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede **sentir el entorno** mediante la recepción de entradas desde una variedad de sensores y puede **afectar a su alrededor** mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el **Lenguaje de programación de Arduino** (basado en [Wiring](#)) y el **Entorno de desarrollo Arduino** (basado en [Processing](#)).



Que es Arduino?:

Los proyectos de Arduino pueden ser **autónomos** o se pueden **comunicar** con software en ejecución en un ordenador.

Arduino no necesita una pieza separada de hardware para cargar nuevo código (o para decirle que hacer) al microcontrolador, simplemente se usa un cable USB.

El Entorno de Desarrollo Integrado, el Lenguaje de Programación de Arduino y las placas en las que se ejecutan han sido desarrollados de la mano, lo cual facilita el **desarrollo y la compatibilidad**.

El IDE de Arduino utiliza una versión simplificada de C++, por lo que es más fácil aprender a programar.



Por qué Arduino?:

Ofrece algunas ventajas para profesores, estudiantes y a aficionados interesados:

- **Barato:** Las placas Arduino son relativamente baratas comparadas con otras plataformas. La versión menos cara del modulo Arduino puede ser ensamblada a mano.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también.
- **Código abierto y software extensible:** El software Arduino está publicado como herramienta de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.



Para qué sirve?:

Podemos catalogar los usos que podemos darle a Arduino en dos grandes grupos:

- Aquellos en los que el Arduino es utilizado como **microcontrolador autónomo**, tiene un programa descargado desde un ordenador y funciona de forma independiente. Controla y alimenta determinados dispositivos, toma decisiones de acuerdo al programa descargado e interactúa con el mundo físico gracias a sensores y actuadores.
- Y aquellos en los que la placa Arduino **hace de interfaz** entre un ordenador y el mundo físico. Ejecuta una determinada tarea solo cuando ocurre un evento en el ordenador, como hacer click en un botón, para traducir dicha tarea en una acción (actuadores). Y viceversa, gracias a sensores que están conectados a la placa Arduino podemos hacer que el ordenador ejecute determinadas acciones. Por ejemplo mostrar un grafico lineal de muestras de temperatura diarias en un celular.



Hardware:

El hardware Arduino más sencillo consiste en una placa con un microcontrolador y una serie de puertos de entrada y salida.

Los microcontroladores que utilizan las placas Arduino están basados en un tipo de arquitectura llamados **RISC**, Computador con Conjunto de Instrucciones Reducidas por sus siglas en inglés.

Por nombrar algunos ejemplos de microcontroladores los más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños.

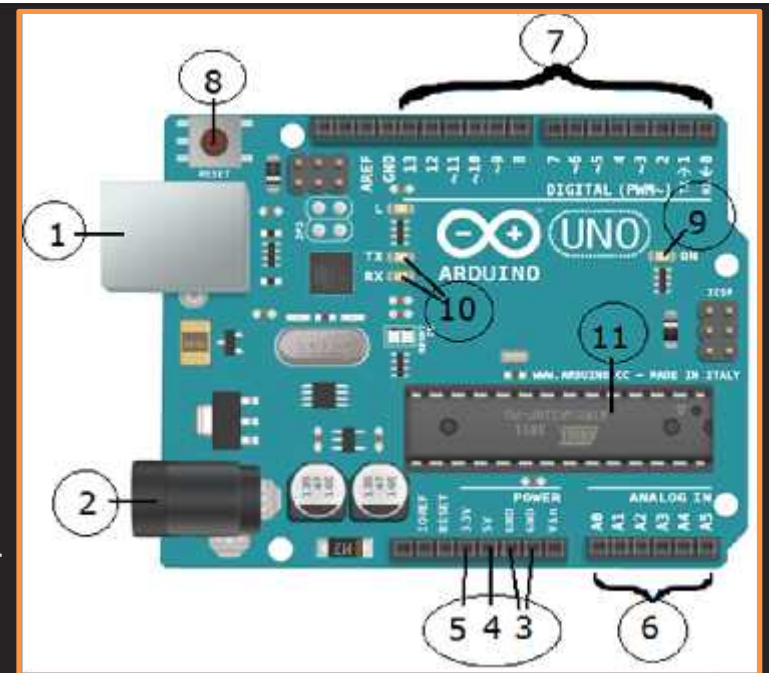
La diferencia entre los distintos Arduino la encontraremos por un lado en la tensión utilizada en las placas. Algunas tienen un voltaje de 3,3 voltios, mientras que otras utilizan una tensión de 5 voltios. También hay placas que pueden conmutar el voltaje.

Hay placas que incluso no necesitan drivers, siendo detectado por los ordenadores como un dispositivo periférico más.



Hardware:

- (1) USB: Es la conexión por donde vamos a cargar código de programación. También sirve como alimentación eléctrica.
- (2) Adaptador de corriente: Otra forma de alimentar nuestro Arduino en caso de no estar conectado a una PC.
- (3) GND: Pines para conectar a tierra el circuito.
- (4, 5) 5V y 3.3V: Son los suministros de 5 voltios y 3.3 voltios de energía.
- (6) Analógico: Pines que pueden leer la señal de un sensor analógico, como un sensor de temperatura, y convertirla en un valor digital que podemos leer. Pines etiquetados del A0 a A5.
- (7) Digital: Pines que se pueden utilizar tanto para la entrada digital (como oprimir un botón) y salida digital (como encender un LED).
- (8) Botón de Reinicio: Este botón reinicia cualquier código que se cargue en el Arduino.
- (9) LED de Alimentación: Este LED debe encenderse cada vez que conecte la placa Arduino a una toma eléctrica.
- (10) LED TX y RX: Estos LEDs nos indican visualmente siempre que nuestro Arduino está transmitiendo (TX) o recibiendo (RX) datos (como cuando subimos un nuevo programa desde el IDE).
- (11) Microcontrolador: Lo negro con todas las patas de metal es un circuito integrado. El cerebro de nuestro Arduino. Por lo general son de la línea ATmega, de la empresa ATMEL.



Software:

Si nos centramos en el software, contamos con un IDE para casi todas las plataformas (Windows, Linux, Mac).

¿Y qué es un **IDE**? Son las siglas de Integrated Desktop Development Environment, o **Entorno de Desarrollo Integrado**.

El IDE es un programa informático donde podemos escribir nuestras aplicaciones (sketch), descargarlas al Arduino y ejecutarlas o depurarlas desde allí. El IDE es gratuito y descargable desde el sitio oficial: <https://www.arduino.cc/en/main/software>

El proceso para descargar e instalar el IDE lo podemos encontrar en el tutorial Instalación del software de Arduino (IDE) en Windows de nuestro foro.

Link: <https://nac-arduino.herokuapp.com/topic/6/instalación-del-software-de-arduino-ide-en-windows>

Aprender el lenguaje de programación Arduino es sencillo, sobre todo si ya tienes experiencia en otros lenguajes de programación como C o Java ya que Wiring / Processing para su programación se basa en ellos.



Software:

El IDE de Arduino:

- (1) Sección para escribir nuestros programas (Sketch).
- (2) Una barra de estado donde se muestra la placa que se está utilizando, el tipo de Microcontrolador y el puerto COM al cuál está conectada.
- (3) Botón para compilar el código.
- (4) Botón para transferir el Sketch a la placa.
- (5) Botón para crear nuevo Sketch.
- (6) Botón para abrir un Sketch guardado.
- (7) Botón para guardar un Sketch.
- (8) También cuenta con un acceso al Monitor Serial y ver registros de eventos.
- (9) Consola de registro de las acciones que lleva a cabo el compilador.
- (10) Y una barra destinada a informar el estado de la compilación y la transferencia hacia la placa.



Fin del tema.

Muchas gracias!



FORO: <https://nac-arduino.herokuapp.com/>

REPOSITORIO: <https://github.com/rody7val/nac-arduino/>