

# CORTO CIRCUITO

Un blog dedicado al área de electrónica digital. PIC, PICAXE y ARDUINO

[Artículos principales](#)[Puertos](#)[PICAXE](#)[Arduino](#)[ABC Electrónica](#)

## Introducción a Arduino

Arduino es una plataforma de electrónica abierta (open Hardware) para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en **Wiring**) y el entorno de desarrollo Arduino (basado en **Processing**). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.



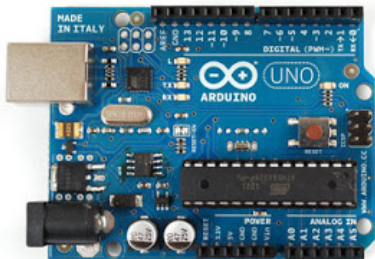
**¿Qué es Processing?** Es un lenguaje de programación de código abierto, para las personas que quieran crear imágenes, animaciones e interacciones. Inicialmente desarrollado para servir como un software como block de dibujo y para enseñar los fundamentos de la programación de computadora en un contexto visual. Processing también ha estado involucrado como una herramienta para generar trabajos finales profesionales (visita <http://www.processing.org/>).

**¿Qué es Wiring?** Wiring es una plataforma de abierta de prototipos electrónicos compuestos de un entorno de programación, una tarjeta de prototipo, documentación (creada por ingeniero y artistas y una comunidad de expertos), personas con experiencia tanto principiante como nivel intermedio comparten sus ideas, conocimiento y su experiencia colectiva (visita <http://wiring.org.co/>).

Las placas pueden ser hechas a mano o compradas las cuales fueron hechas en una fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues eres libre de adaptarlos

a tus necesidades.

Te invito a que mires el siguiente documental sobre Arduino y sus fundadores:



Síguenos por E-mail



Buscar este blog



Archivo del blog

diciembre (4) ▼

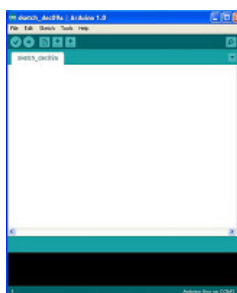
28:17



**Un Hardware abierto.** Para que todos en vez de “reparalo” podamos diseñar y aprender. Arduino es un Hardware con el cual podremos hacer nuestras implementaciones fácilmente. Arduino ha sido diseñado para ser fácilmente extensible, estando basado en estándar y poderosos componentes.

El Hardware de Arduino está basado en un microcontrolador AVR, en particular el ATmega8, ATmega 168, ATmega328 y el ATmega1280. Los programas de Arduino están basados en C/C++ y compilados con el compilador de código abierto avr-gcc y enlazado con la Libc de AVR de código abierto. El lenguaje de Arduino proviene de Wiring. El entorno de Arduino está basado en Processing e incluye modificaciones hechas por Wiring.

**El entorno de Arduino.** El entorno de desarrollo de Arduino contiene un editor de texto para escribir el código, un área de mensaje, una consola de texto, barra de herramientas, etc. Los programas escritos usando Arduino son llamados en inglés “sketches”. Cuando cargas un sketch (proyecto), estas usando el bootloader de Arduino, el cual es un pequeño programa que ha sido precargado al microcontrolador de tu tarjeta. Esto te permite cargar un código sin usar ningún hardware adicional. El bootloader es activado durante unos pocos segundos cuando se reinicia el microcontrolador; después de eso inicia el programa que le hayas cargado.



**¿Qué sucede cuando le doy en el botón “build”?** El software de Arduino realiza algunas pequeñas transformaciones para verificar que el código en C o C++ este correcto. Éste programa pasa por el compilador avr-gcc, el cual pasa el código de alto nivel a lenguaje de máquina. Luego el código es combinado con librerías estándar de Arduino que proveen las funciones. Como resultado se obtiene un archivo .hex, el cual contiene los bytes necesarios para programar la memoria del microcontrolador que se encuentra presente en la placa de Arduino y luego enviado por el puerto USB o serial a través del bootloader que se encuentra en el microcontrolador.

**¿Solo se puede usar Arduino en su circuito impreso?** No, también uno puede pasar el Atmega328 a protoboard o hacer tu propia placa para algún diseño (para mayor información visita: <http://arduino.cc/en/Tutorial/ArduinoToBreadboard>)

#### Pros:

- Arduino está basado en dos sistemas totalmente abiertos (Wiring y Processing), por lo que nos da la total libertad de entender el Hardware y Software, abriéndonos las posibilidades de desarrollo de sistemas electrónicos.
- Su entorno y modo de programación permite que muchas personas que son novatas escojan Arduino como herramienta de aprendizaje.
- Posee todo su circuito base ya ensamblado así que solo es de armar el nuestro y programar, ahorrándonos espacio en el protoboard para un IC el cual simplemente lo conectamos con Jumpers o cables.
- Posee su programador interno y no es necesario comprar una programadora extra.
- Puedes programar otros Atmel328 con la placa de Arduino, cargándoles primero el Bootloader y luego el programa (cosa que no se puede hacer con PICAXE, con PICAXE hay que comprar integrados con ese firmware precargado).
- El bootloader es un programa que se ejecuta solamente una vez durante el arranque del microcontrolador y se apaga a los pocos segundos si no recibe datos por medio de la comunicación serial de la PC.



#### Contras:

- Su firmware (bootloader) que lo hace tan potente, a mi modo de ver también le da una desventaja, ya que el MCU empieza a trabajar al segundo de ser alimentado y en diseños donde se necesita monitorear constantemente una señal, le da una desventaja ya que si sucede algún fallo o suceso que pase en ese periodo podría ser perjudicial al equipo (si se tiene un diseño de protección), o un sistema de alarma, etc. Para ejemplificar mejor: yo diseñe un detector de transientes el cual protege al sistema de transientes al conmutar equipo industrial a la red eléctrica (velo acá: <http://cortoc.blogspot.com/2011/10/proyecto-desarrollar-sistema.html>) así que, ¿qué pasaría si sucede ese transiente durante el segundo de arranque del bootloader? El circuito no protegerá y pasara el transiente a la red "protegida" y causará daños a los sistemas.
- Su costo es elevado comparándolo con el precio de los PICS, PICAXE e incluso los integrados de AVR.
- Si estas implementando un diseño donde no necesitas un microcontrolador de 28 pines, no hay una amplia gama de IC Arduino, aunque algunas personas han realizado un trabajo para programar Atiny45 y Atiny85 (MCU de 8 pines), aunque no se pueden usar todos los comandos de Arduino (próximamente una entrada sobre este tema).

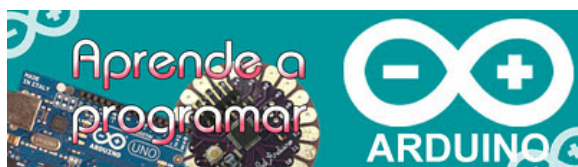


Lo que incluye en el paquete de Arduino.

Para más información visita [www.arduino.cc](http://www.arduino.cc)

Descarga el software de Arduino: **software**

Ingresa a los laboratorios de Arduino y aprende a implementar hardware para tus proyectos: **Arduino**



Publicado por **Julio Roberto Letrán Cardona**

 +6 Recomendar esto en Google

Etiquetas: **Arduino**

## 1 comentario



Añadir un comentario como Javier Martín Alonso

## Mejores comentarios



hace 4 años

Me encanta la versatilidad del arduino y eso de poder compartir ideas, casualmente yo lo descubri por medio de un amigo que lo utilizo para automatizar un sistema de riego. Felicitaciones por el proyecto arduino!

[Entrada más reciente](#)[Página principal](#)[Entrada antigua](#)Suscribirse a: [Enviar comentarios \(Atom\)](#)