



Introducción

Supuesto de automatización

- Imaginemos que queremos que la luz de entrada de nuestra casa se encienda de forma automática al abrir la puerta.
- ¿Qué necesitamos?
 - Un sensor que detecte que la puerta se abra o se cierre.
 - Una lámpara
 - Algo que tome la decisión de encender y apagar la luz.
 - Esto hay que desarrollarlo mediante un programa.

¿Por qué microcontroladores?

- Alternativas:
 - Un ordenador conectado al sensor y a la luz.
- Problemas:
 - Demasiada capacidad para algo simple
 - Demasiado coste
 - Demasiado espacio
- Un microcontrolador
 - Mucho más pequeño y con capacidad para desempeñar esa y otras funciones.
 - Mucho más barato.

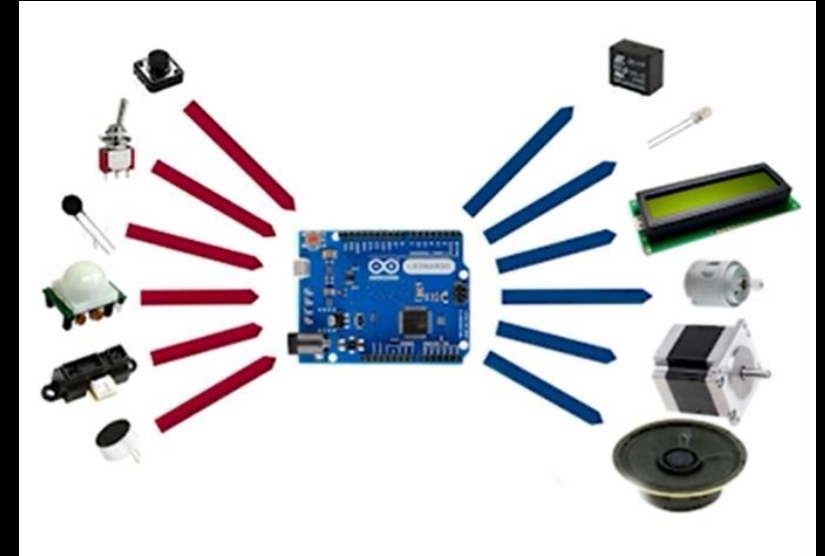


Evolución de los microcontroladores

- Antes el hardware no era estándar. Cada marca tenía su firmware o sistema.
- Se programaba en un lenguaje de bajo nivel (assembler).
- Muy complicado y cercano al código máquina.
- En el año 2005, unos docentes en Italia trataron de solucionar este problema.
- Con el objetivo de facilitar el desarrollo de este tipo de proyectos creando un estándar de desarrollo.
- Sin preocuparse demasiado por la electrónica.
- Con filosofía *Open-Source*.
- Así nació el proyecto ARDUINO.

¿Qué es Arduino?

- Hardware
 - Placa con sensores, motores, luces, Bluetooth, motores...
 - Lo más importante es la placa. Una placa estándar donde va conectado el resto de los periféricos.
 - Al ser un estándar funciona como esperamos que funcione, sin sorpresas.
- Software
 - Se implementó un sistema integral de desarrollo con un lenguaje de alto nivel.
- Una comunidad mundial detrás del proyecto



```
Blink | Arduino 1.8.5

Blink 5

This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Open hardware

- El hardware no es propietario. Cualquiera lo puede replicar.
- En China se fabricó muchísima placa.
- Consecuencias:
 - Más barato
 - Más público
 - Más difusión al proyecto

Placas Arduino

- No hay una única placa.
- Hay diferentes según las necesidades
 - Con Wifi
 - Con Bluetooth
 - Con interfaz de juegos
- Todo se programa con el mismo lenguaje y con la misma plataforma.
- No hay que aprender nada nuevo ni instalar nada nuevo.

Otras ventajas

- No es un producto para especialistas electrónicos.
- La placa ya viene encastrada y finalizada a punto para desarrollar.
- Si queremos complementos hardware, simplemente los encastramos a la placa principal.
- Estos complementos se llaman *SHIELDS*.
- Por ejemplo, si necesitamos utilizar conexión en red, encastramos el *shield* de ethernet y listo.
- Hay shields que son pantallas LCD si queremos poner una interfaz gráfica, motores, Wifi...

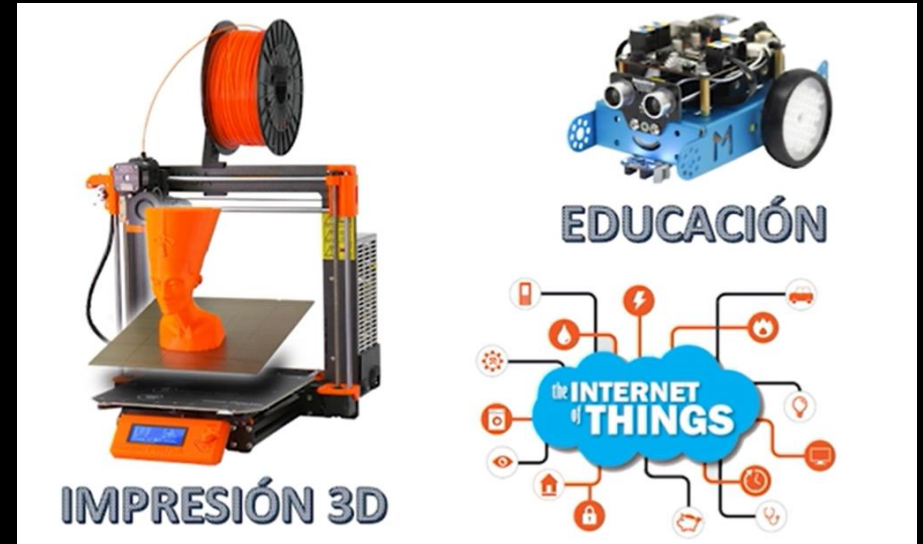


Lenguajes

- Al extenderse tanto el uso de Arduino, se sumaron a la comunidad grandes cantidades de programadores.
- Se fue creando un ecosistema de desarrolladores heterogéneo.
- Demandaron programar en sus lenguajes.
- Se comenzó a aumentar el abanico de lenguajes con la creación de diferentes scripts, si bien su origen está en C y este es el “oficial”.

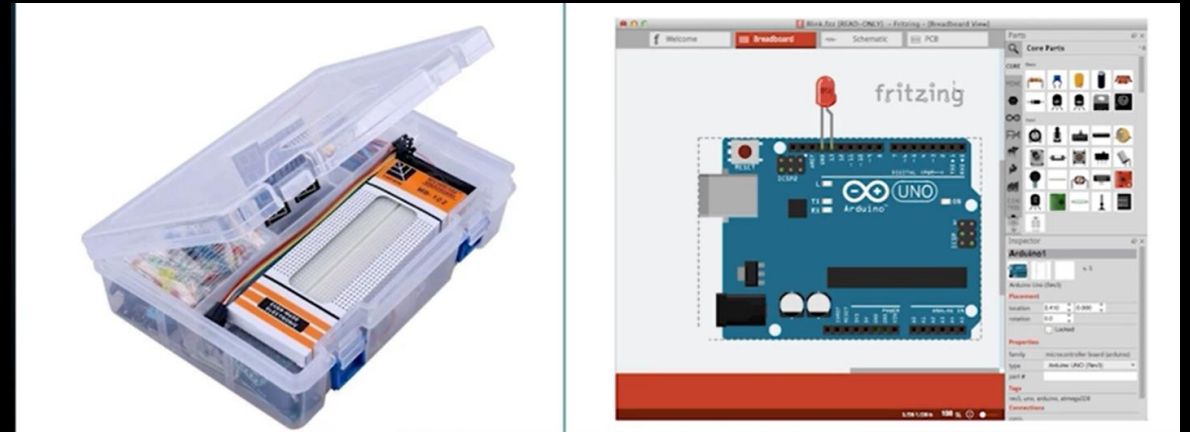
Gracias a Arduino

- Nació el mundo de la impresión 3D.
 - La lógica era estándar.
 - El hardware era barato.
- La educación en programación se ha impulsado mucho.
 - Los niños aprenden Scratch y robótica educativa.
 - Hacen sus programas y los orientan a, por ejemplo, juegos.
- El IOT también se ha impulsado bastante.
- Aceleró mucho el prototipado de productos porque las placas y shields son estándar.
 - Una empresa o usuario puede desarrollar sus aplicaciones sin apenas costes.



Simulador vs Kit

- En el mercado hay placas y kits.
- También tenemos simuladores on line.
- El simulador funciona muy bien, pero...
- ¿Qué kit podría comprar?
- Cualquier kit básico con placa, leds y protoboard ya nos vale.



Simuladores

- Tenemos los instalables
 - Fritzing
 - Autodesk Eagle
- Los disponibles online.
 - Tinkercad
 - EasyEDA
 - Permite diseñar placas y mandarlas a fabricar.



¿Por qué simuladores?

- Ventajas
 - No podemos quemar ni romper nada.
 - Es gratuito. Podemos poner en marcha decenas de luces sin comprar ninguna.
 - Más fácil de usar y de detectar errores.
- Inconvenientes
 - No es lo mismo verlo de forma física.
 - Puede que no tengan todos los sensores o módulos.

Tinkercad

- Herramienta online libre y gratuita.
- No es necesaria descarga. Se trabaja en la nube.
- Nos permitirá probar nuestros diseños con sus componentes electrónicos conectados, introducir el código de programación y finalmente simular su funcionamiento antes de pasar a su implementación física real.

