

# Introducción a Arduino

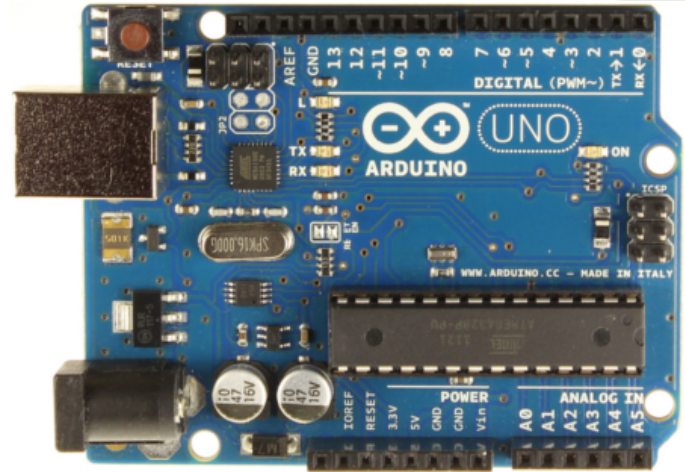
*14 noviembre 2013*

*Amaia Castro*

*@amaiac*

# La placa

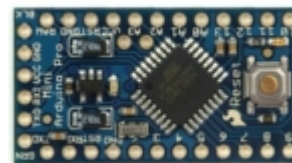
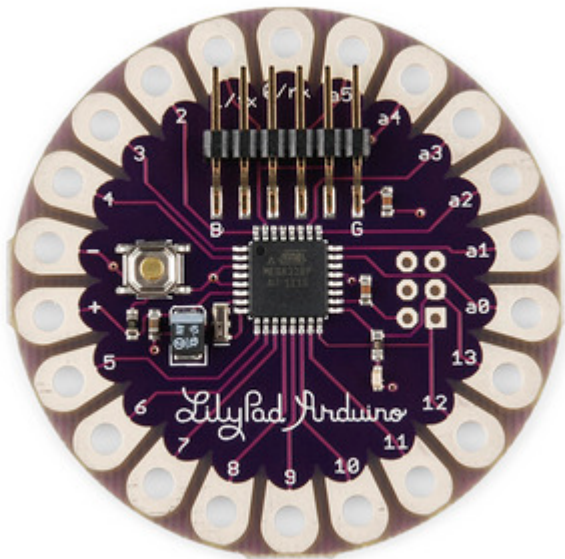
- Hardware libre
- Diseñado para ser fácil de manejar por todo tipo de usuarios
- Microcontrolador ATmega328
- 6 entradas analógicas
- 14 entradas/salidas digitales



# Shields (Escudos)

Extensiones de Arduino montadas en un placa que se pincha directamente sobre la de Arduino

# Otros modelos de Arduino



<http://arduino.cc/en/Main/Products>

# Cosas curiosas que la gente hace con Arduino



# Lo básico de arduino

- Leer datos de los sensores conectados a las entradas (botón, potenciómetro, LDR, sensor temperatura, etc)
- Controlar los actuadores conectados a las salidas (LED, altavoz, motor, etc)

# El entorno de desarrollo

Entorno sencillo en el que escribir los programas, compilarlos y cargarlos en la placa.

El lenguaje utilizado es Wiring, muy parecido a C.

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, opening, and other functions. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeat
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);          // wait for a second
}
```

The bottom status bar shows "1" on the left and "Arduino Uno on /dev/ttyACM0" on the right.

# El entorno de desarrollo

El programa básico consta de:

- Función de **setup**, que se ejecuta al conectar o resetear la placa.
- Función de **loop** que se ejecuta constantemente mientras la placa tenga alimentación.

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and other functions. The main text area contains the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeat  
 *  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

The bottom status bar shows "1" and "Arduino Uno on /dev/ttyACM0".



# Setup

- Declarar los pines que vayamos a usar como entrada o como salida

```
pinMode(13, OUTPUT); // salida digital
pinMode(12, INPUT);  // entrada digital
pinMode(A1, INPUT);  // entrada analógica
```

- Inicializar librerías, por ejemplo la de comunicación Serie

```
Serial.begin(9600)
```

# Entradas y salidas digitales

Sólo dos posibles valores:

0V - 5V

# Entradas analógicas

Arduino sólo maneja valores digitales, estas entradas tienen un conversor analógico-digital que convierte el voltaje de entrada (entre 0 y 5 voltios) en un valor numérico entre 0 y 1023.

Las entradas analógicas son las que están marcadas como A0 - A5.

# Salidas “analógicas” PWM

Arduino no puede generar salidas analógicas, sólo puede poner la salida a 0v (LOW) o a 5v (HIGH). Para simular una salida analógica usa PWM (Pulse Width Modulation).

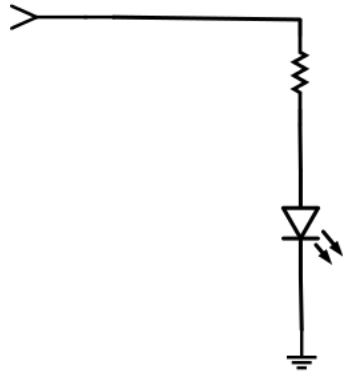
Se puede usar PWM en los pines marcados con ~.

Código:

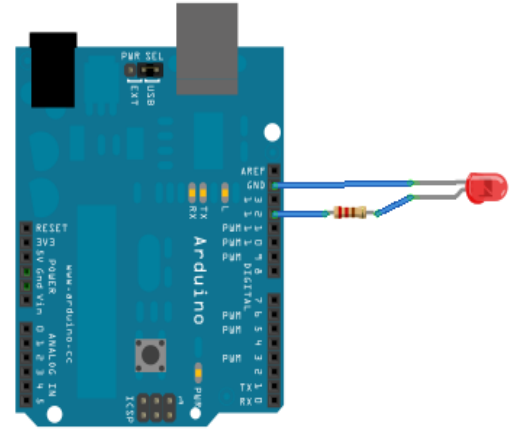
```
analogWrite(pin, value); // value 0-255
```

# Actuadores: LED

Esquema de conexión del LED  
pin - resistencia - LED - gnd



Made with  Fritzing.org



# Actuadores: buzzer o altavoz

Esquema de conexión:

```
pin -- (+ -) -- gnd
```

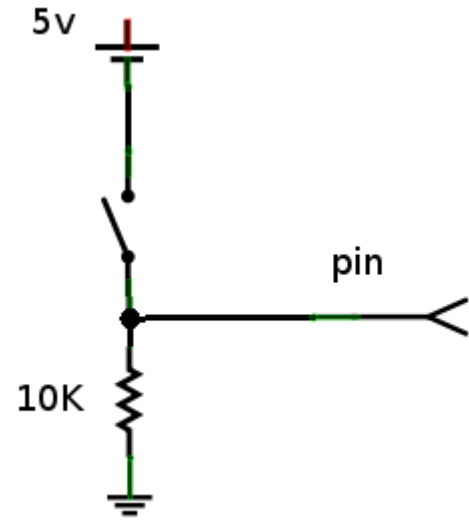
Código:

```
tone(pin, freq, duration);
```

# Sensores: Botón

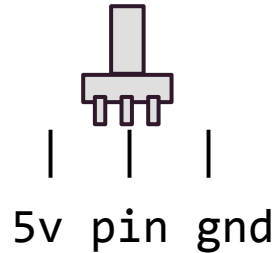
Esquema de conexión del botón:

```
digitalRead(pin);
```



# Sensores: Potenciómetro

Esquema de conexión del potenciómetro:



Código:

```
analogRead(pin);
```



# Sensores: LDR

## Esquema de conexión del LDR

5v - R - LDR - gnd  
|  
pin