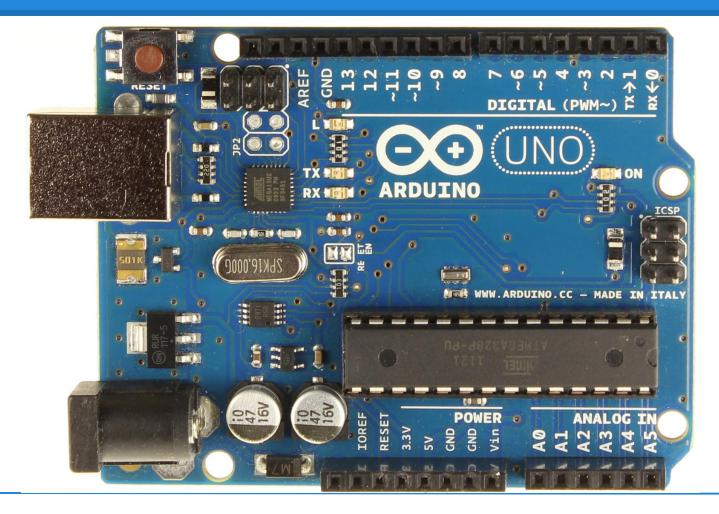
# Curso avanzado sobre Arduino

#### Arduino Avanzado





## **Arduino Intermedio: Presente**





### **Arduino Avanzado: Presente**



#### José Antonio Vacas Martínez



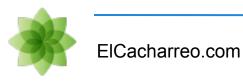


## Programando Arduino: Lenguaje

Una librería no es más que un conjunto de código empaquetado y al que podemos llamar desde nuestro programa

Facilita la tarea de desarrollar y nos abstrae y encapsula la dificultad de la tarea

Para usarlas solo tenemos que importarlas desde nuestro código con un #include <....>



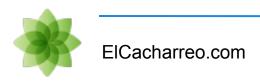
# Librerías: ¿cómo funcionan?

El include

inline

referencia

Linker



### Librerías: básicas

- EEPROM permite leer y escribir en almacenamiento duradero
- Ethernet para conectar a internet
- Firmata comunicaciones usando un protocolo concreto
- LiquidCrystal manipulación de LCD
- SD lectura y escritura en tarjetas SD
- Servo control de Servos
- SPI comunicaciones con dispositivos usando SPI
- SoftwareSerial permite comunicaciones serie por otros pines
- Stepper control de motores paso a paso
- Wire comunicaciones I2C



#### Librerías: otras

- Communication (networking and protocols):
- <u>Messenger</u> for processing text-based messages from the computer
- NewSoftSerial an improved version of the SoftwareSerial library
- OneWire control devices (from Dallas Semiconductor) that use the One Wire protocol.
- PS2Keyboard read characters from a PS2 keyboard.
- <u>Simple Message System</u> send messages between Arduino and the computer
- <u>SSerial2Mobile</u> send text messages or emails using a cell phone (via AT commands over software serial)
- <u>Webduino</u> extensible web server library (for use with the Arduino Ethernet Shield)
- <u>X10</u> Sending X10 signals over AC power lines
- XBee for communicating with XBees in API mode
- <u>SerialControl</u> Remote control other Arduinos over a serial connection

#### Sensing:

- <u>Capacitive Sensing</u> turn two or more pins into capacitive sensors
- <u>Debounce</u> for reading noisy digital inputs (e.g. from buttons) Displays and LEDs:
- <u>Improved LCD library</u> fixes LCD initialization bugs in official Arduino LCD library
- <u>GLCD</u> graphics routines for LCD based on the KS0108 or equivalent chipset.
- <u>LedControl</u> for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- LedControl an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
- LedDisplay control of a HCMS-29xx scrolling LED display.



# Librerías: y más

- These libraries are compatible Wiring versions, and the links below point to the (excellent) Wiring documentation.
- <u>Matrix</u> Basic LED Matrix display manipulation library
- Sprite Basic image sprite manipulation library for use in animations with an LED matrix Frequency Generation and Audio:
- <u>Tone</u> generate audio frequency square waves in the background on any microcontroller pin Motors and PWM:
- <u>TLC5940</u> 16 channel 12 bit PWM controller.

#### Timing:

- DateTime a library for keeping track of the current date and time in software.
- Metro help you time actions at regular intervals
- ullet MsTimer2 uses the timer 2 interrupt to trigger an action every N milliseconds.

#### **Utilities:**

- <u>PString</u> a lightweight class for printing to buffers
- Streaming a method to simplify print statements



#### Escribiendo nuestra librería

- Para crear nuestra librería tenemos que generar nuestro código en C++
- Crearemos una clase con nuestro código
- Usaremos un fichero "nuestralibreria.h" donde declararemos nuestro interface (obligatorio que exista el constructor)
- Un fichero "nuestralibreria.cpp" con nuestro código
- Incluiremos todos los ficheros en una carpeta "nuestralibreria" en la carpeta libraries del directorio de usuario
- Cerramos y abrimos el entorno arduino para que la recompile y ya está disponible.

## Conclusiones

## Gracias por vuestra atención

