Microcontrollers, sensors & servers

14 November, 2019

e/s/d/ madrid

P.PORTO ESCOLA SUPERIOR DE MEDIA ARTES E DESIGN

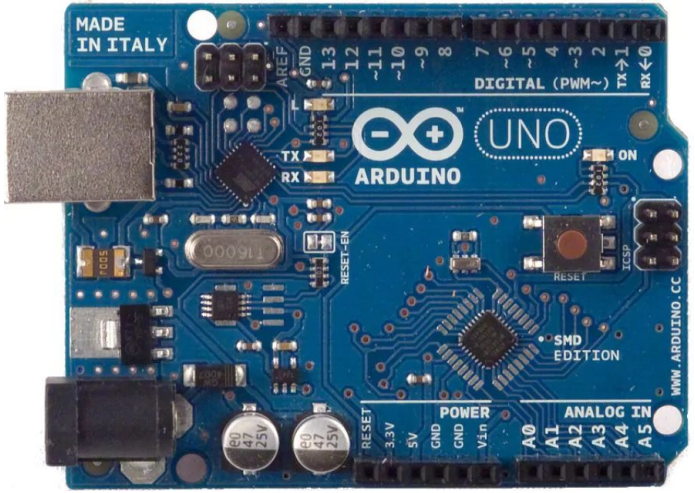

GGGGGG
GGGGGG
GGGGGG
GGGGGG
GGGGGG
GGGGGG

Arduino

Free-use electronics

Arduino is an **electronic creation platform** based on **open hardware and software**, flexible and easy to use for designers and developers.

- **Open hardware**: devices whose specifications and diagrams are publicly accessible, so anyone can replicate them.
- **Open software**: software whose code is publicly accessible, so that it can be used and modified by anyone.

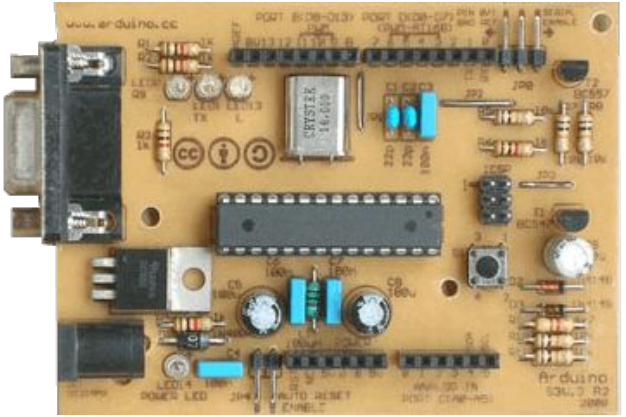


Arduino

The board and its creators

The project was born in **2003**. Several **students from** the Interactive Design Institute in Ivrea (**Italy**) set themselves the objective of facilitating access to and use of electronics and programming.

The result was Arduino, a **board** with all the necessary elements to connect peripherals to the **inputs and outputs** of a **microcontroller**, which can be programmed in Windows as well as macOS and Linux.

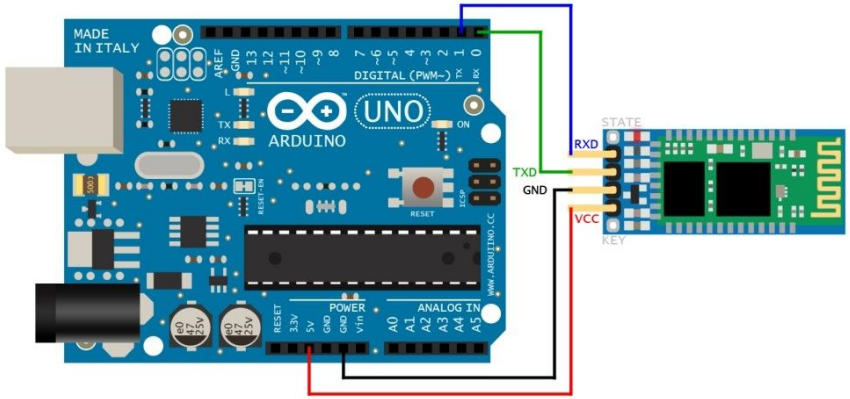


Arduino

Microcontroller

Microcontrollers are **integrated circuits** on which you can record **instructions**, which you write with the Arduino programming language (Arduino IDE). These instructions allow you to create **programs** that interact with the board's circuits.

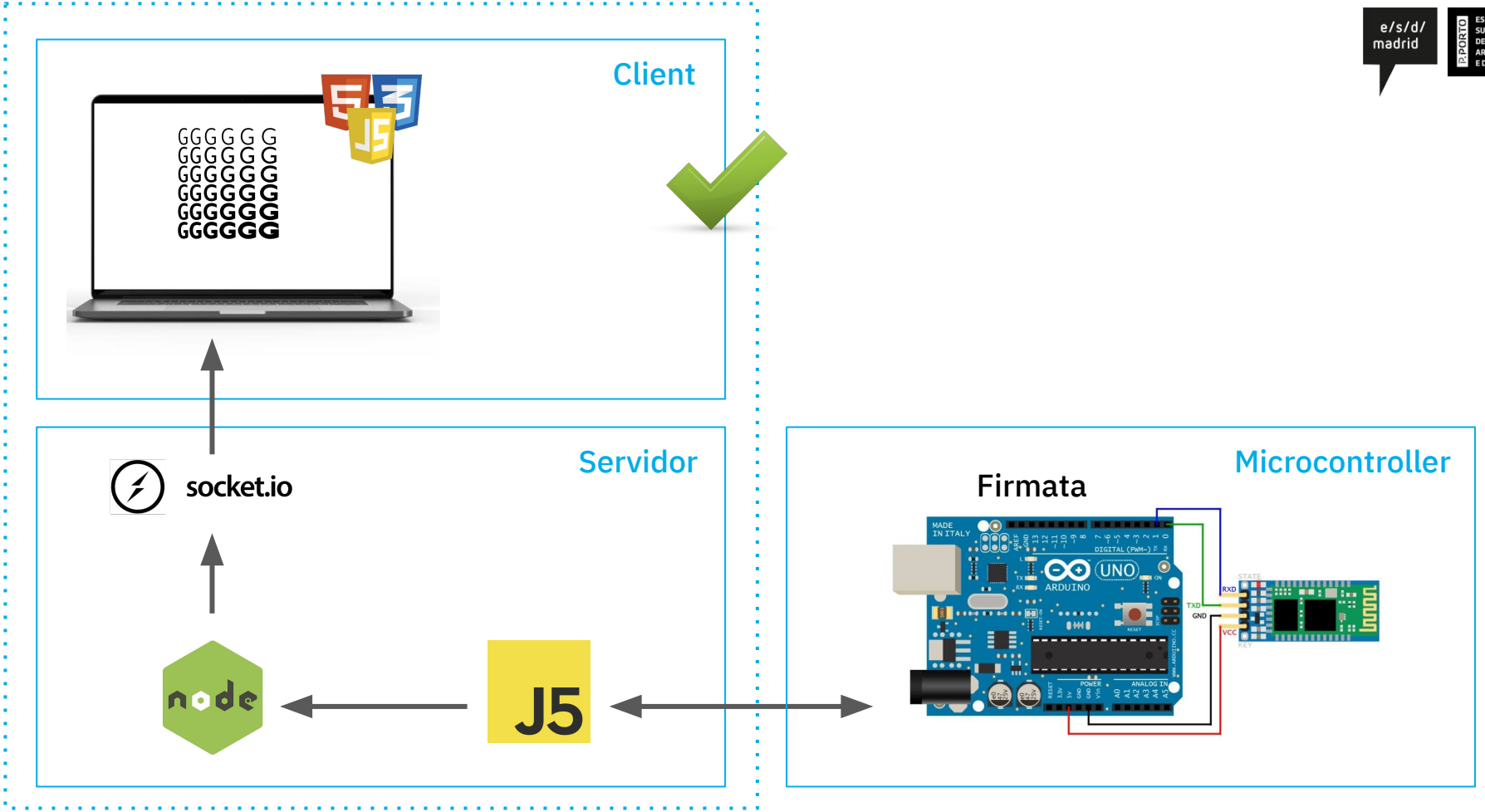
Through the arduino **input interface** we can connect on the board different types of **peripherals** (from a keyboard, to a camera or any other type of **sensor**).



Arduino

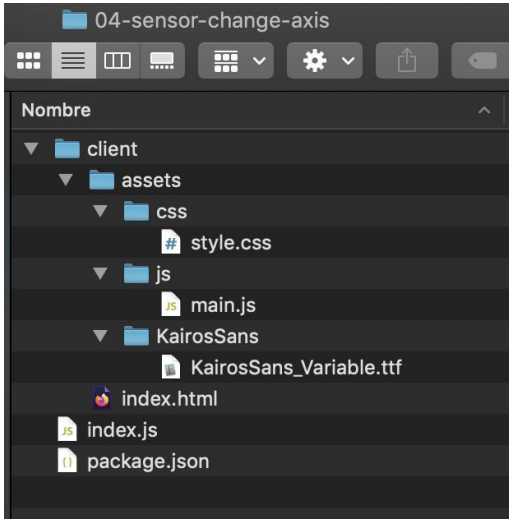
Sensors

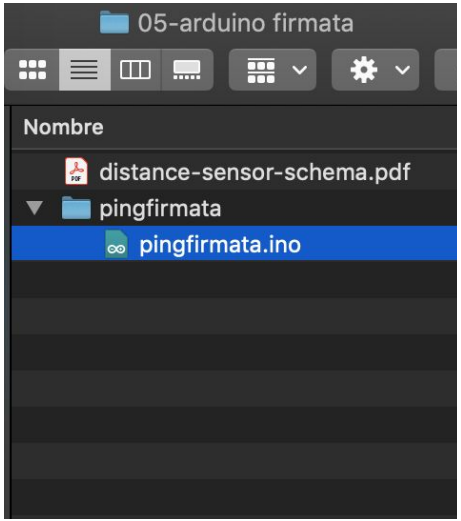
Arduino's great flexibility and openness mean that you can use this board for just about anything. Using sensors and peripherals, you can build robots, alarms, fingerprint readers, automations of any kind... the **possibilities are endless.**


Client
socket.io
Servidor
node
J5
Firmata
Microcontroller
RXD
TXD
GND
VCC

04-sensor-change-axis

05-arduino firmata





https://github.com/Lourdesmarco/sensor-variable-font-workshop/

Arduino IDE

Begin with Arduino

To write programs and load them onto the board, we will need the **Arduino IDE**, which can be found on the official Arduino website: https://www.arduino.cc/en/Main/Software

Download the Arduino IDE

ARDUINO 1.8.10

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

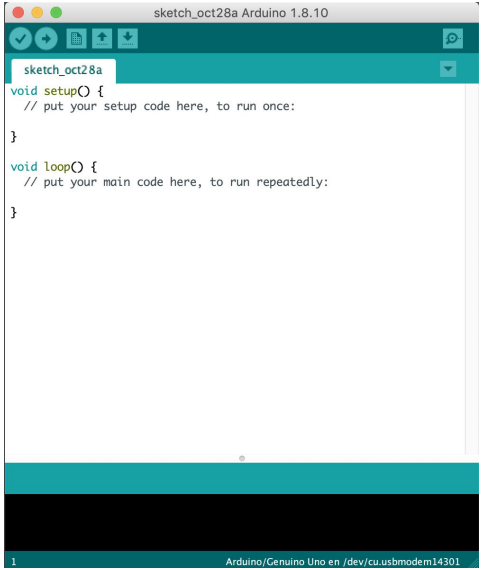
Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

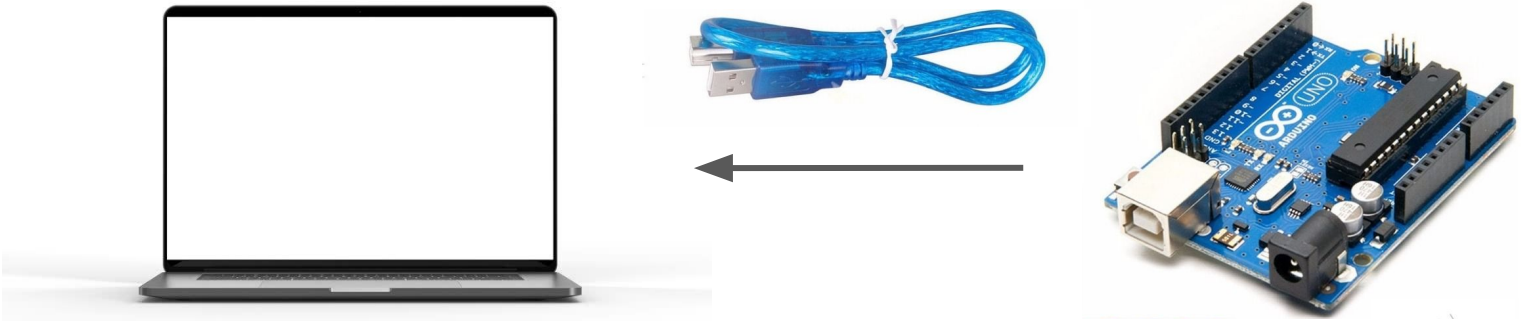
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)



Arduino board

Begin with Arduino



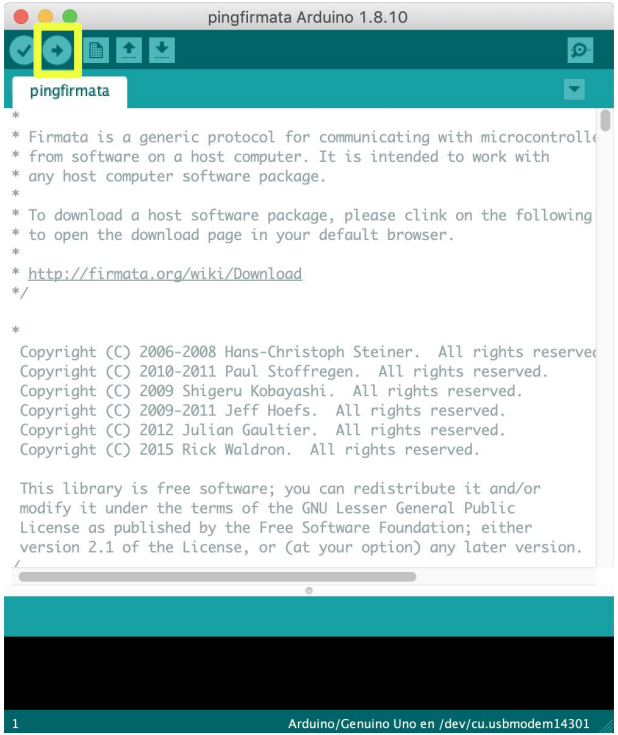
Arduino

Load firmata

As our goal is not to learn to program in Arduino, but to communicate with the board from the computer, we will need an **intermediary: firmata**.

Firmata is a **generic protocol** for **communicate microcontrollers** and computers that will allow access from external software, such as windows or MacOSX.

Double click on the file **pingfirmata.ino** in the example (05 - arduino firmata / pingfirmata / pingfirmata.ino) and having the board (USB) connected to the computer, click on the **upload button**.

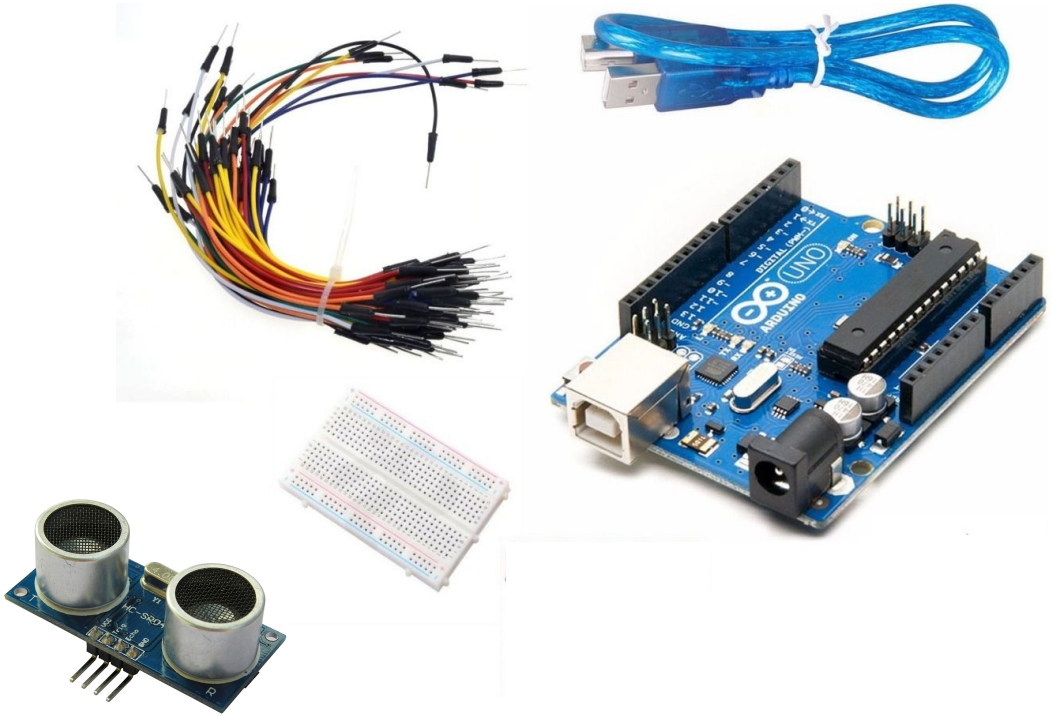


Arduino

Mounting sensors

To begin with the Arduino sensors we're going to need:

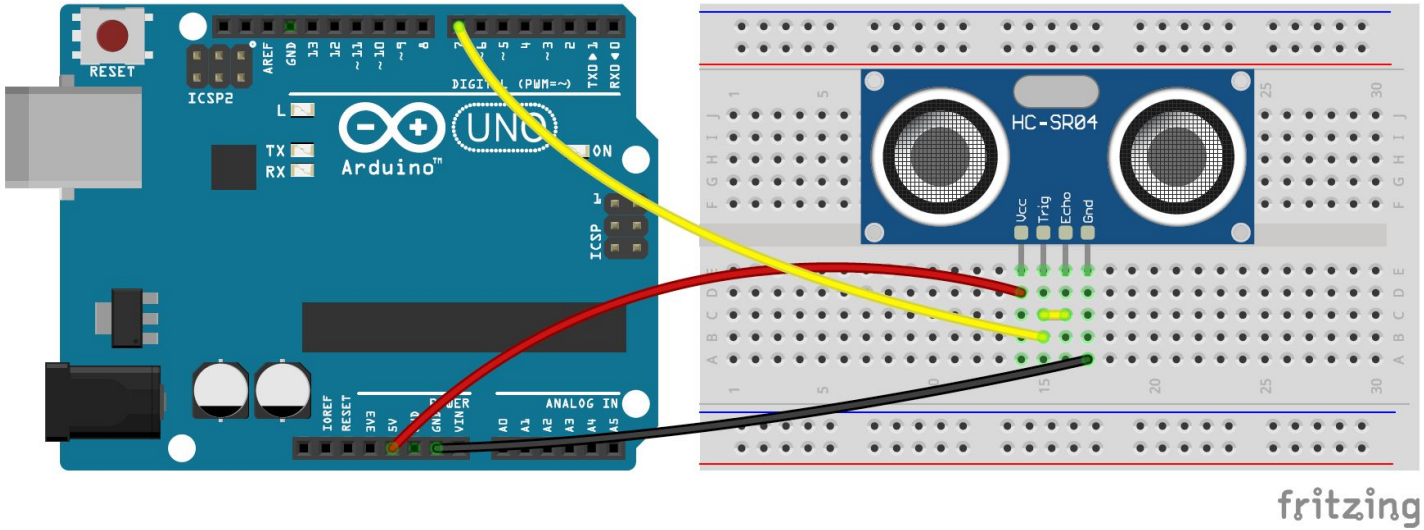
- An Arduino board
- A USB cord
- **A protoboard**
- **Wires**
- **Sensors** (distance for this example)

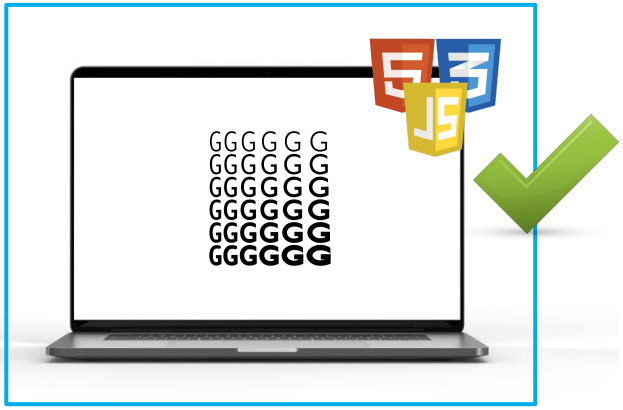


Arduino

Mounting distance sensor

In order to prepare the distance sensor on the protoboard, just follow the scheme of the example (05 - arduino firmata / distance-sensor-schema.pdf)



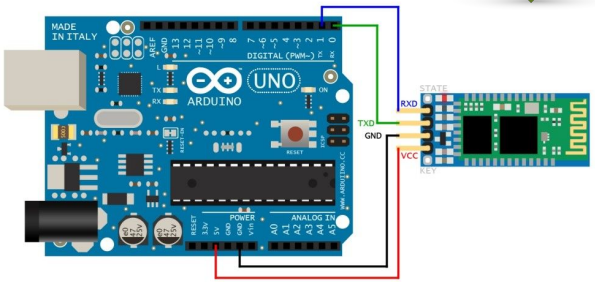


socket.io

node

J5

Firmata



Server

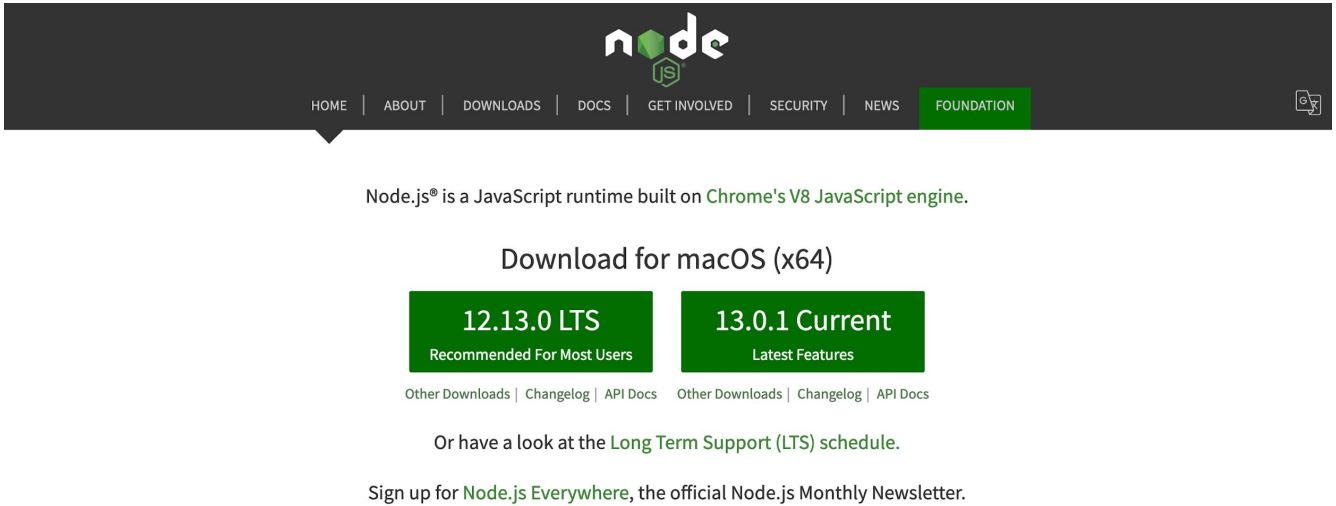
Node.js

Node.js is a **JavaScript environment** that allows us to run it on the **server**, asynchronously, with an event-oriented architecture and based on Google's V8 engine. The traditional languages of the server side had been php, ruby, ... with node.js comes an option in JS with all the ingredients to win market.

Server

Node.js

To install node, go to **https://nodejs.org/es/** and download the recommended version.



Server

Package manager

The idea of a package and dependency manager is that developers can **share small portions of code** that are reusable in different projects and avoid each programmer having to re-encode basic elements. The most used package managers with JS are **npm** and **yarn**. We will install yarn: https://yarnpkg.com/lang/en/

$ **sudo npm install -g yarn**

Server

Install packages and run server

Once node.js and yarn are installed, we can work with any project that uses server-side JS and dependency management, as is the case in the example "04-sensor-change-axis".

From the terminal, we will navigate to the folder of example 04 and execute the following commands:

// install all package.json dependencies

$ yarn

// run the server

$ yarn start

Thank you very much for your attention