

**FEDERACIÓ  
CATALANA  
D'AMICS DEL  
FERROCARRIL**

# ARDUINO

APROXIMACIÓ A LA UTILITZACIÓ DE L'ENTORN

ARDUINO I EL SEU IDE

# QUE ÉS ARDUINO?

És una plataforma de creació d'electrònica de codi obert.

Està basada en hardware i software lliure, flexible i fàcil d'usar per a creadors i desenvolupadors.

El hardware lliure són els dispositius amb especificacions i diagrames d'accés públic, de manera que qualsevol persona pot replicar-los.

Això vol dir que qualsevol persona o empresa pot crear les seves pròpies plaques, que poden ser diferents entre elles però totalment funcionals perquè parteixen de la mateixa base.

El software lliure té un codi accessible per qualsevol que vulgui utilitzar-lo o modificar-lo. Arduino ofereix la plataforma Arduino IDE (que és un entorn de desenvolupament integrat) en que qualsevol persona pot crear aplicacions per a les plaques Arduino per donar-li tot tipus d'utilitats.



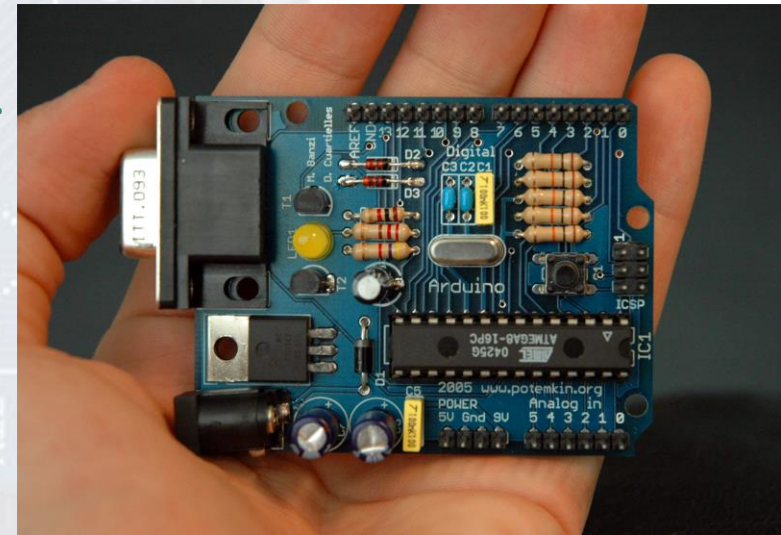
## El projecte va néixer en 2003

Diversos estudiants de l'Institut de Disseny Interactiu de Ivrea (Itàlia) van fer una placa electrònica que era una alternativa a les populars BASIC Stamp, que per llavors tenien un preu de 100 \$.

El projecte es va anomenar Wiring i contava amb la possibilitat de poder fer servir un entorn de programació de codi obert multi-plataforma, que es pot utilitzar en Linux, Windows, Mac OSX, GetApp i ARM.

El nom Arduino ve del lloc on es reunien els estudiants, un bar d'Ivrea dedicat a Arduino d'Ivrea, que va ser rei d'Itàlia entre 1002 i 1014.

El primer disseny que va veure la llum pública de forma comercial va ser introduït l'any 2005 al mercat oferint un baix cost i facilitat d'ús.



# TIPUS DE PLACA

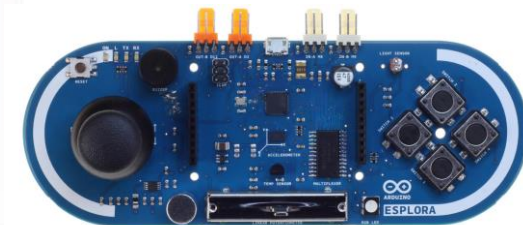
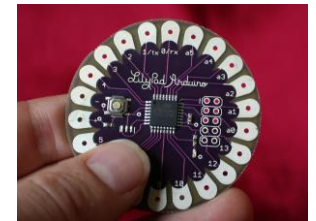
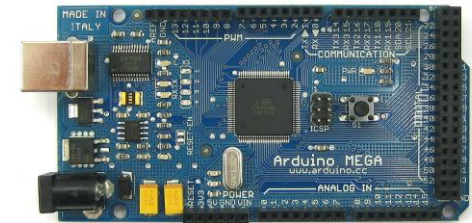
Al llarg de la seva vida útil ha tingut diferents tipus de processadors, Atmel AVR, ARM Cortex o Intel Quark. Encara que la filosofia és la mateixa el codi font per fer funcionar aquestes plaques difereix, per la qual cosa hi ha diferents versions del software.

La placa principal és el tipus Single Board Computer, conté memòria SRAM i una Flash EEPROM. Dependent de la seva forma de comunicacions pot tenir port sèrie, USB, Bluetooth, o de prototip.

Dependent del fabricant hi ha infinitat de plaques, però la més utilitzada és la Arduino Uno.

Una cosa important és que el seu bus a cada costat possibilita que pugui tenir extensions connectades i fa que pugui controlar des de robots a impressores 3D.

Lògicament també serveix per controlar tots els dispositius que hi ha en una maqueta...



# BUS SÈRIE

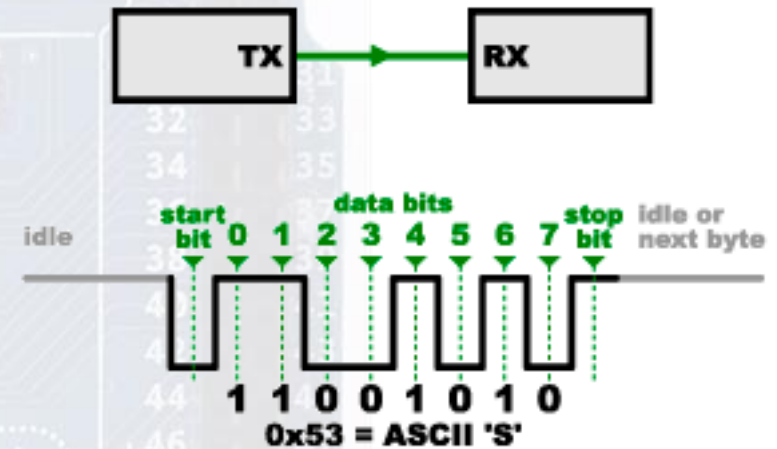
En les plaques Uno, Nano, Mini i Mega els pins 0 i 1 s'usen per comunicar-se amb l'ordinador.

Les comunicacions en els pins TX (transmissió)/RX (Recepció) utilitzen nivells lògics TTL de 5 volts o 3,3 volts depenent de la placa.

Aquests pins no es poden connectar directament a un port RS232, ja que aquests ports operen a 12 volts.

A banda dels dos pins també tenim un port (o més) que utilitza la comunicació sèrie (excepte l'Arduino Pro Mini), que és el USB que el connecta a l'ordinador, que serveix per introduir la programació a l'Arduino.

En aquest port és molt important que transmissor i receptor funcionin a la mateixa velocitat (de 9600 a 115.400 baudios).



# BUS SPI

El bus *Serial Peripheral Interface* va ser desenvolupat per Motorola l'any 1980, i és un estàndard de facto degut a les avantatges que té sobre altres sistemes.

Té una arquitectura mestre/esclau en que el mestre pot iniciar la comunicació amb un o diferents esclaus i enviar i rebre dades d'ells. Els dispositius esclau no poden iniciar la comunicació ni comunicar-se entre ells.

El tipus de comunicació és del tipus síncrona, en la que hi ha un senyal de rellotge que té tots els dispositius sincronitzats, evitant el problema de velocitat que té el BUS Sèrie.

Com a mínim té 3 línies:

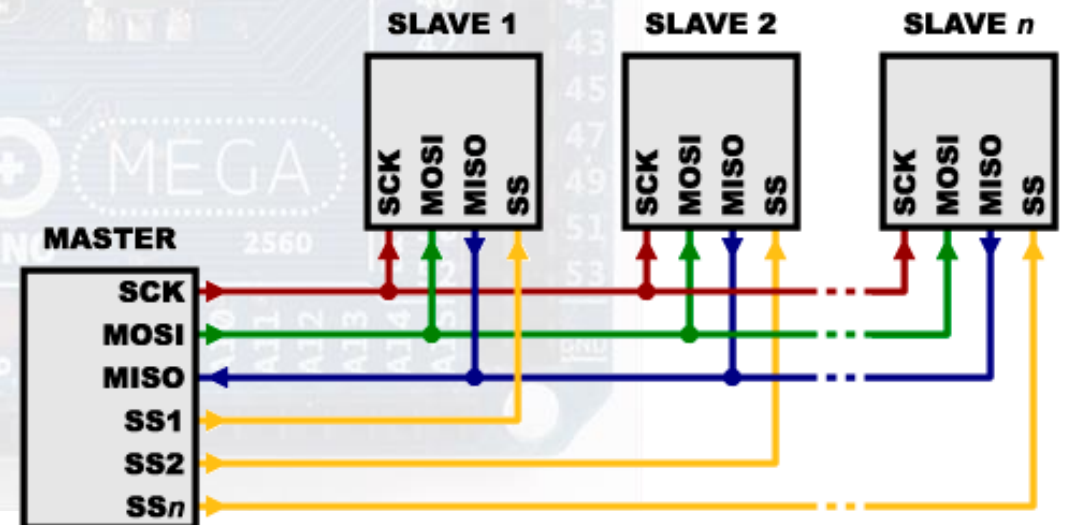
MOSI: comunicació mestre a esclau

MISO: comunicació esclau a mestre

SCK: Senyal de rellotge del mestre

SS: Slave-select (indica que el missatge és per ell)

Es possible fer una connexió en cascada en què cada Esclau transmet dades al següent.



# BUS I2C

El Inter-Integrated Circuit (I2C) es una comunicació estàndard. Va ser dissenyat per facilitar la comunicació entre diferents tipus de dispositius, entre els quals es troben els microcontroladors.

La comunicació a través del bus no pot ésser anàrquica, sinó que ha de seguir les normes que dictaminen protocol de comunicacions, que en aquest cas és el I2C.

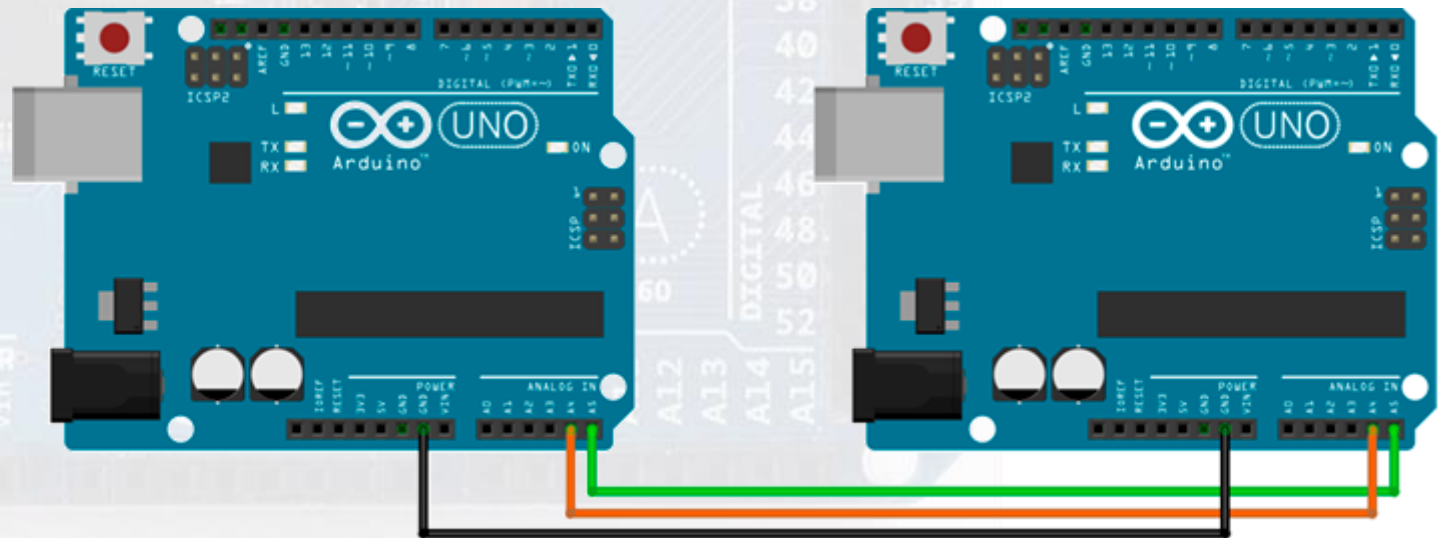
El tipus de comunicació és del tipus síncrona, en la que hi ha un senyal de rellotge que en l'Arduino pot arribar als 100 kbits per segon i té tres senyals:

SCL: rellotge del sistema

SDA: senyal de dades

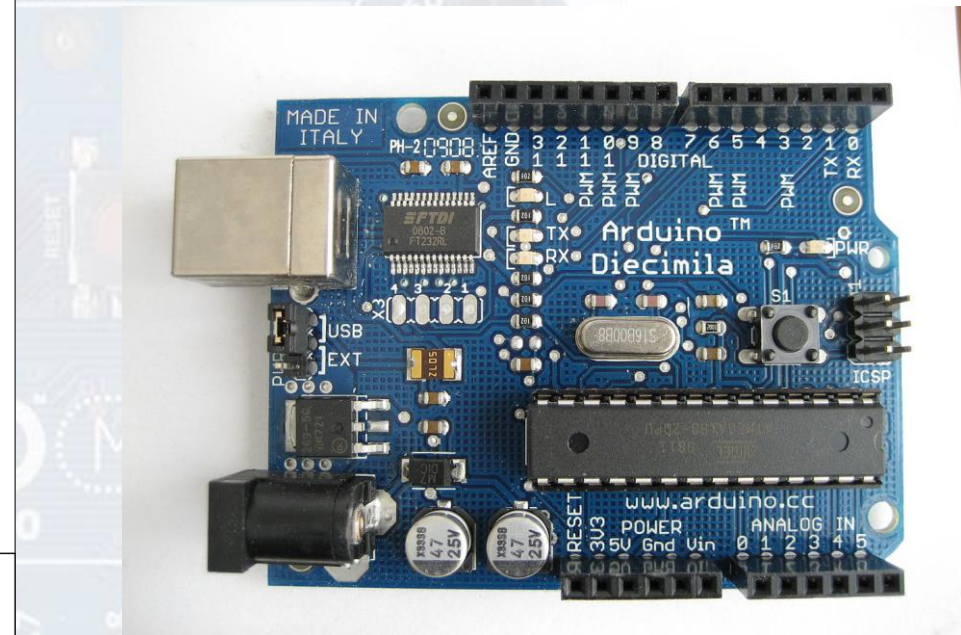
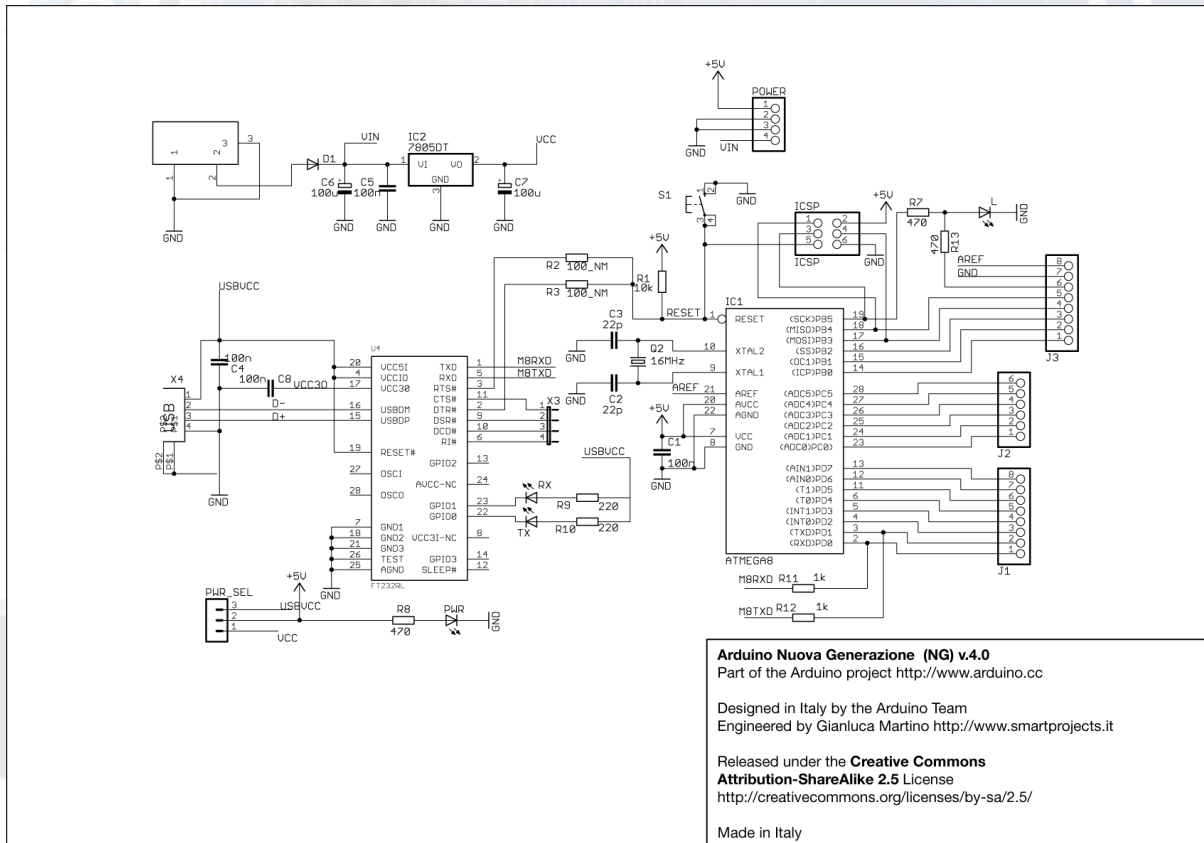
GND: masa del senyal

Aquests senyals s'han de compartir entre tots els dispositius que hi hagi al Bus. Un exemple, un Arduino mestre i un esclau, com a la foto.



# ARDUINO UNO

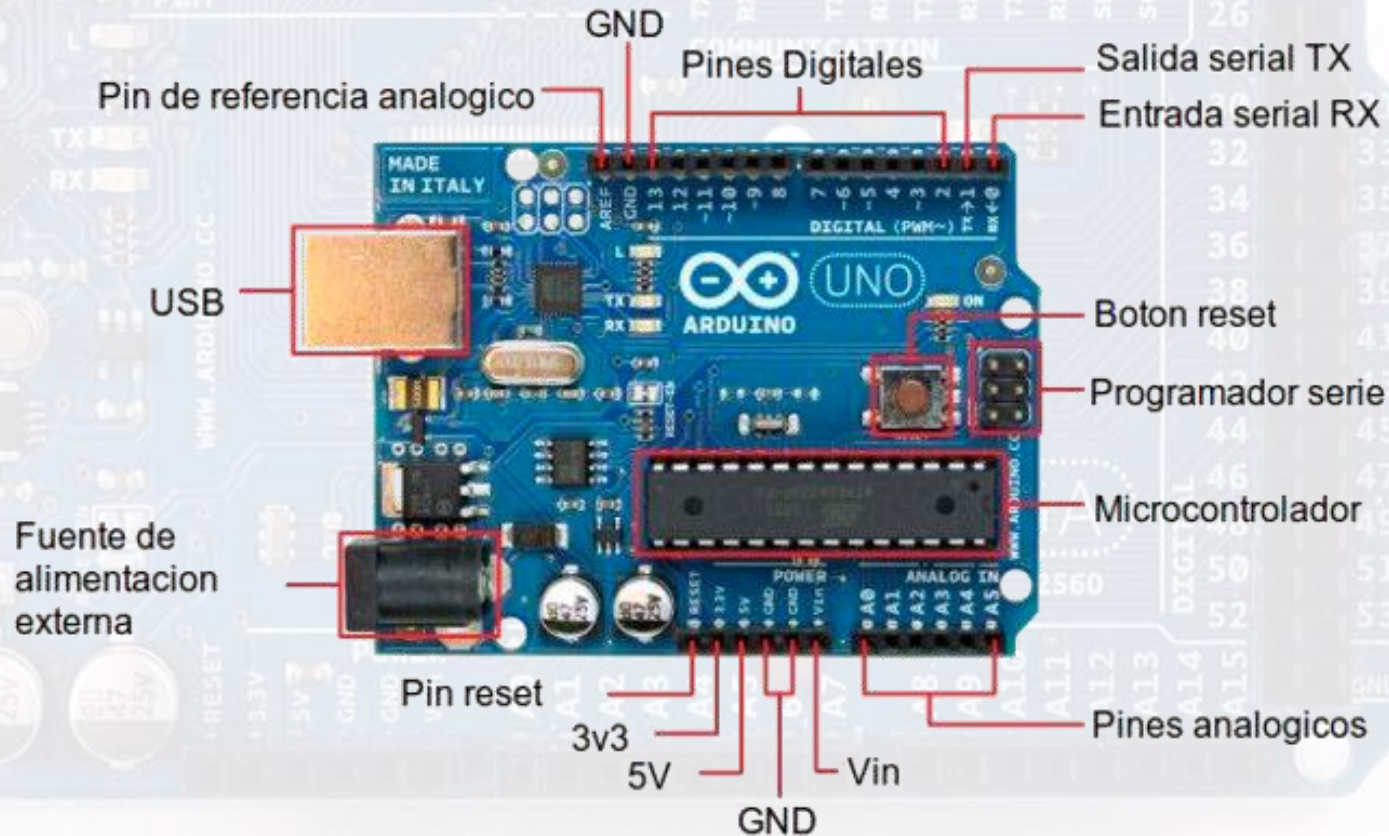
La més assequible de totes les plaques i la que pràcticament trobarem a tots les kits és la placa Arduino Uno. A la foto es veu l'esquema Creative Commons, i a la dreta una versió per USB.





# ARDUINO UNO

La placa electrònica Arduino Uno R3 pot ser alimentada de diferents formes, amb un cable USB del tipus B connectat a l'ordinador o amb una font d'alimentació externa (adaptador DC 7-12 volts).



# ARDUINO UNO

El microcontrolador és un ATmega328P.

És un microcontrolador d'arquitectura RISC avançat de Atmel, d'alt rendiment, baix consum i optimitzat per compiladors de llenguatge C.

L'arquitectura interna (figura de la dreta) és del tipus Hardward, amb memòries i busos separats per a programa i dades. La CPU utilitza una pipeline, en la que mentre s'executa una instrucció ja s'està buscant la propera des de la memòria del programari.

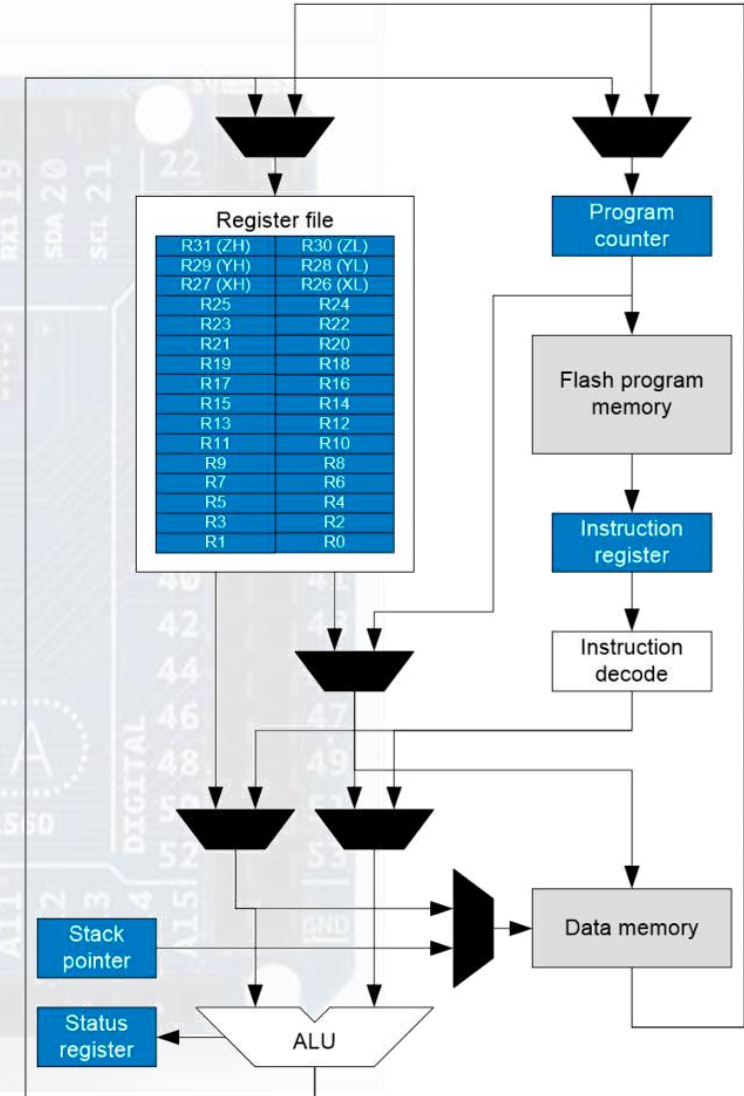
D'aquesta manera es poden executar les instruccions en un sol cicle de rellotge.

També posseeix un banc de registres de propòsit general de 32x8 bits (a la pràctica fa 16x16) que permet que les instruccions aritmètiques i lògiques es facin en un sol cicle de rellotge.

El microcontrolador té una memòria reprogramable Flash de 32 kBytes d'emmagatzematge.

La memòria es divideix en dos espais per seguretat:

- La secció d'inici de càrrega ( Boot Loader)
- La secció de programa de l'aplicació



# ARDUINO UNO

La placa està disposada en grups:

A sota del processador hi ha el grup d'alimentació en que es veuen una sèrie de voltatges i massa amb el pin del reset. Alguns voltatges poden ser d'entrada però també de sortida depenent de l'ús.

També a sota hi ha les entrades analògiques PC0 a PC5, que proporcionen 10 bits, anomenats de resolució.

A la capçalera del processador hi ha els pins de programació (ICSP) i el pin o el botó del reset, que permet a l'Arduino tornar al seu estat original.

A la banda contrària de les entrades analògiques es troben les entrades i/o sortides digitals en número de 14, que ofereixen una tensió de 5 volts.

Dos d'elles la 0 RX i la 1 TX serveixen per rebre i transmetre dades en sèrie.

Els fabricants tenen la possibilitat de canviar determinats components físics, però bàsicament encara que canviï la fesomia de la placa, per sota roman amb la mateixa filosofia inicial.



# SHIELD

Les plaques Shield son plaques modulars que es munten unes sobre altres per donar funcionalitat extra a un Arduino. Són apil·lables.

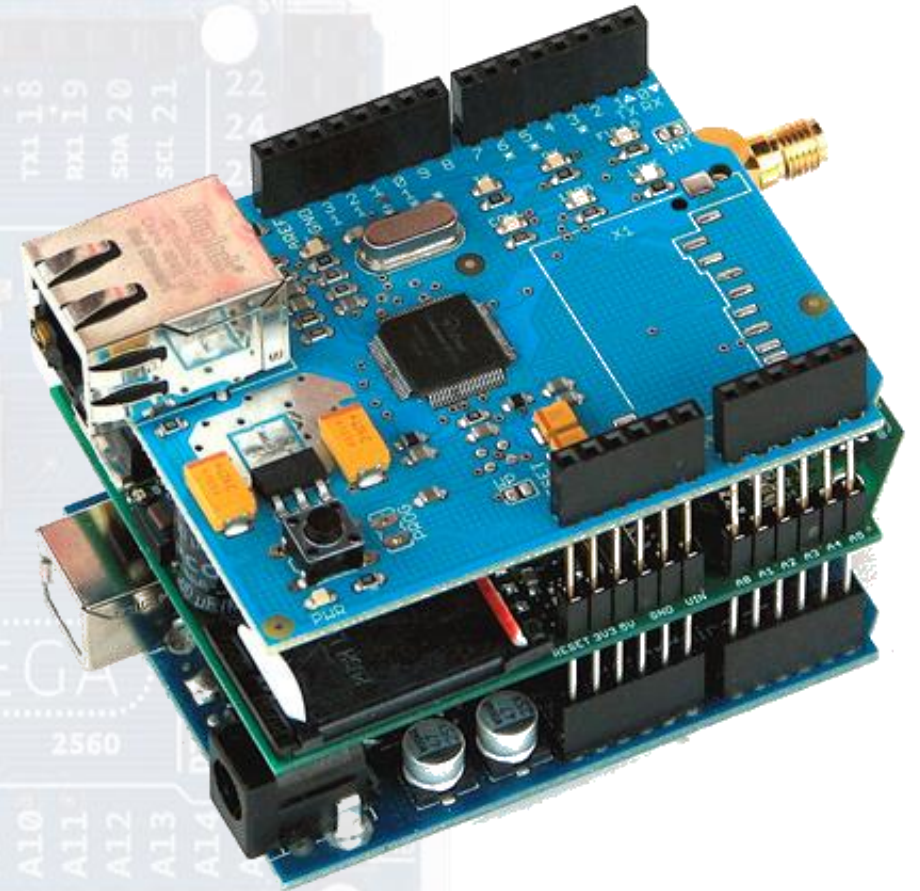
Un shield en Arduino és una placa que s'apila sobre l'Arduino sobre un altre shield de forma que ens permet ampliar el hardware o les capacitats de l'Arduino.

Los shield es poden comunicar amb l'Arduino bé per alguns dels pins digitals o analògics o per algun bus com el SPI, I2C o port sèrie, o utilitzar alguns pins per la interrupció.

Cada shield ha de tenir la mateixa forma que l'Arduino perquè només hi hagi una forma d'encaix.

Potser que es pugui apilar un altre shield a sobre o no, y això ve donat pels pins de connexió.

És molt important llegir la seva documentació per saber si utilitza un bus, inhabilita algunes entrades/sortides. Habitualment el shield ve amb una llibreria per la seva utilització, encara que pot utilitzar genèriques.



# PROGRAMACIÓ D'ARDUINOS

Tots els Arduinos poden utilitzar-se específicament per a una finalitat concreta. Ja sigui pel control d'un robot, impressora 3D, domòtica de la casa, etc.

Perquè faci coses concretes se l'ha d'introduir un programari a la seva memòria Flash per que respongui d'una determinada manera a impulsos d'entrada.

La manera més habitual de fer-lo servir és com a centraleta en que una sèrie de botons fan que determinats dispositius penjats a la sortida funcionin, i ho facin com està previst a la programació.

En teoria programar qualsevol element electrònic és introduir un llistat amb una sèrie de funcions i dades que funcionen amb una cadència, com hem dit abans a cop d'impuls.

El llenguatge de programació permet posar comentaris al desenvolupament del programari de tal manera que tenim la idea del que fan les línies successives(en neutre tenim les explicacions i en color les funcions):

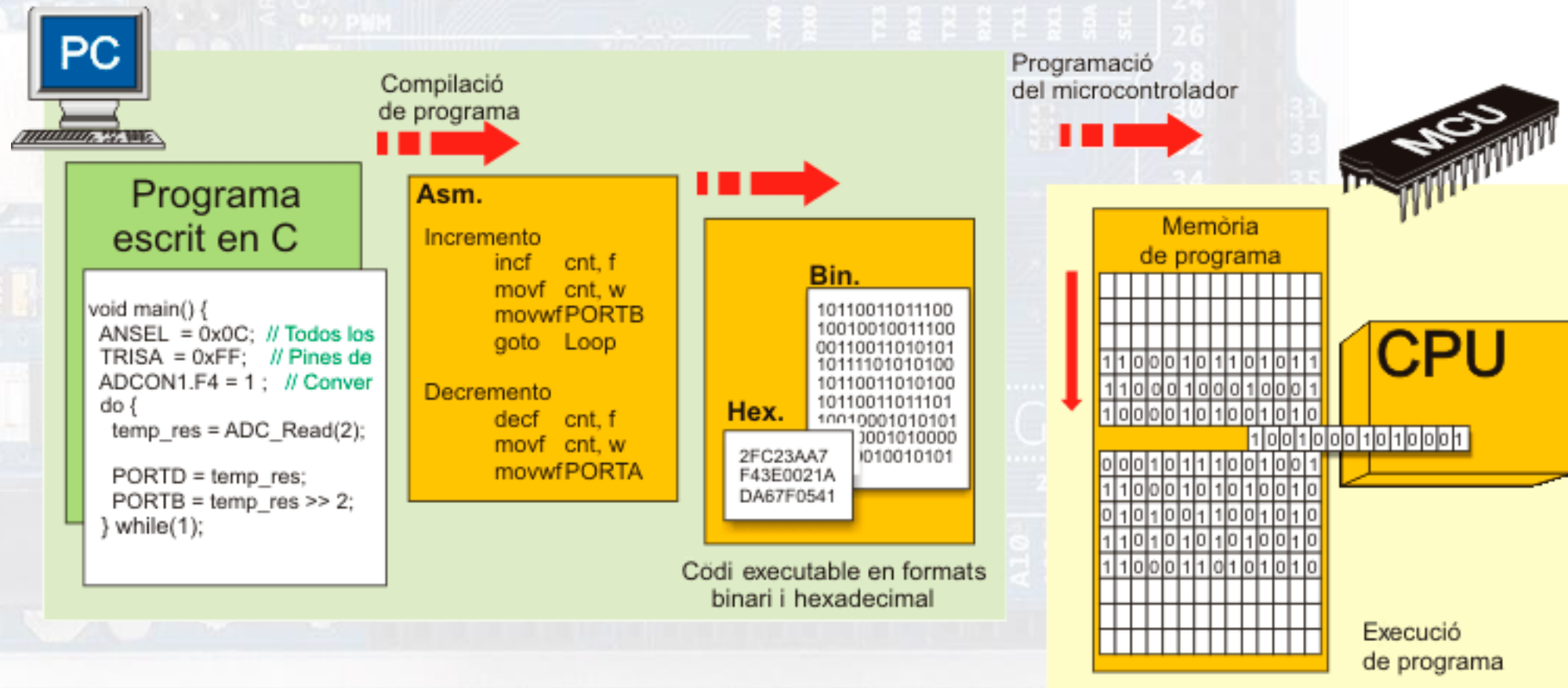
```
void store_log (void)
{
  /* we continuously update the "current" entry */
  int logno = get_minutes_since_midnight () / LOG_INTERVAL;
  if ((logno < 0) ||
      (logno >= (24*60/LOG_INTERVAL)))
    logno = 0;
  // datalog[logno].temperature = cached_temperature;
  // datalog[logno].humidity = cached_humidity;
  // datalog[logno].moisture = moisture_read();//cached_moisture * 100 / moisture_calib;
}

void setup(){

  reset_settings (config);
```

# COMPILAR I CARREGAR EL PROGRAMA

Per fer que el microcontrolador entengui que n'ha de fer, el programa s'ha de compilar en assemblar, binari i hexadecimal. Una vegada feta aquesta passa intermèdia ja es pot programar el microcontrolador.



# IDE DE ARDUINO

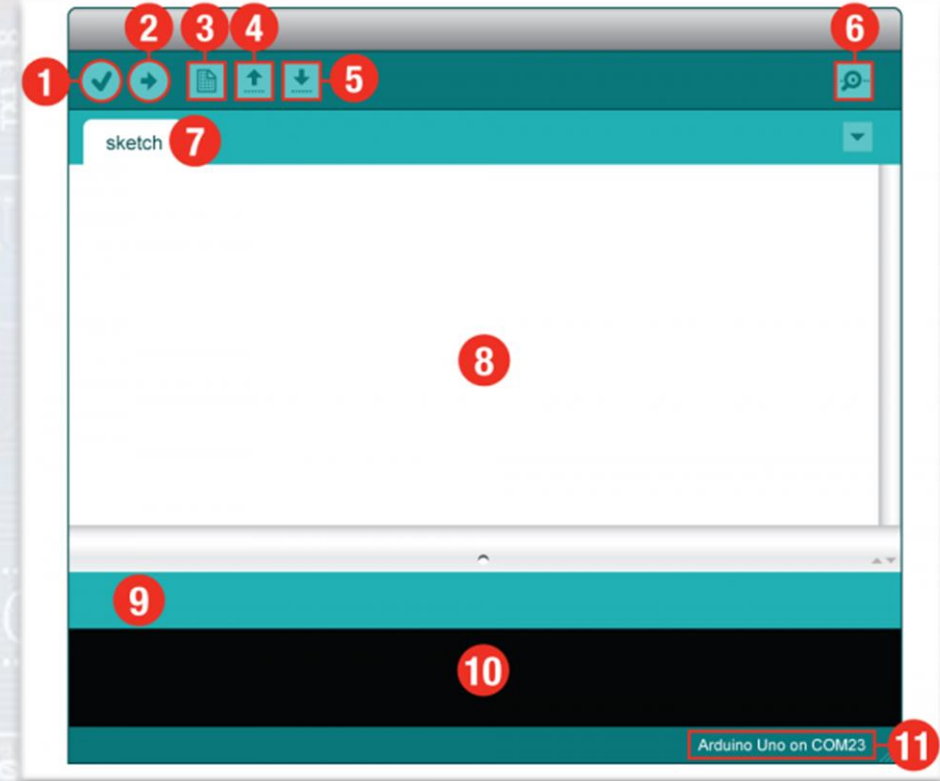
Com l'Arduino es connecta a l'ordinador, tenim una eina per programar el seu funcionament de forma ràpida i fàcil. El IDE.

El IDE és un programari que s'instal·la a l'ordinador (hi ha diferents versions segons sistema operatiu) i en ell es pot fer el programa que anirà a l'aparell. En el programa es poden copiar codis d'altres o fer el nostre propi.

**Recordeu la filosofia inicial de compartir-ho tot.**

La finestra te uns paràmetres bàsics sota el menú típic del programari:

1. Verificar el codi.
2. Carregar el codi a la placa.
3. Nou obre una nova pestanya de codi.
4. Obrir: Obre un arxiu existent.
5. Guarda l'arxiu que està actiu.
6. Monitor sèrie: obre una finestra d'informació sèrie. Útil per depurar.
7. Nom de l'arxiu existent.
8. Àrea de codi, es la que comprèn tot el codi de l'arxiu.
9. Àrea de missatges, on s'indica si hi ha errors de codi.
10. Consola de text, que mostra els missatges d'error complets. Útil per depurar.
11. Placa i port sèrie: mostra el tipus de placa i el port on està connectada.



# DESCARREGAR EL PROGRAMARI

A la web d'Arduino es pot descarregar el programari que ens faci falta, a la darrera versió.

La pàgina principal és: <https://www.arduino.cc/>

Malauradament només està en anglès o xinès

El IDE serveix per totes les plaques d'Arduino.



**ARDUINO 1.8.9**  
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 \(sha512\)](#)

# INSTAL·LAR EL PROGRAMARI



1.8.9 (Windows Store 1.8.21.0)

**ARDUINO** | **Genuino**

AN OPEN PROJECT WRITTEN, DEBUGGED,  
AND SUPPORTED BY ARDUINO.CC AND  
THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS  
OF **ARDUINO.CC** on [arduino.cc/credits](https://www.arduino.cc/credits)

Després de baixar el programa de la web principal, i després de demanar ajut (si vols). Per Windows antic i Linux és un programari, però a Windows 8.1-10 ja és una App, i s'instal·la i configura com a tal.

La ruta on s'instal·la és Program Files\WindowsApp, i normalment no podem tocar allà per problemes de permisos.

Si s'utilitza la placa Arduino Picaro seran necessaris els divers del chip FTDI.

Ja tenim el programari instal·lat i es pot fer servir per programar l'Arduino.



# PRIMERS PASSOS

El millor quan ens posem a manegar quelcom nou és fer provatures.  
Una bona solució seria comprar un kit amb una placa de Arduino Uno amb accessoris, especialment la placa “breadboard” que serà la base d’operacions per “punxar” tots els elements que farem servir en les nostres pràctiques.

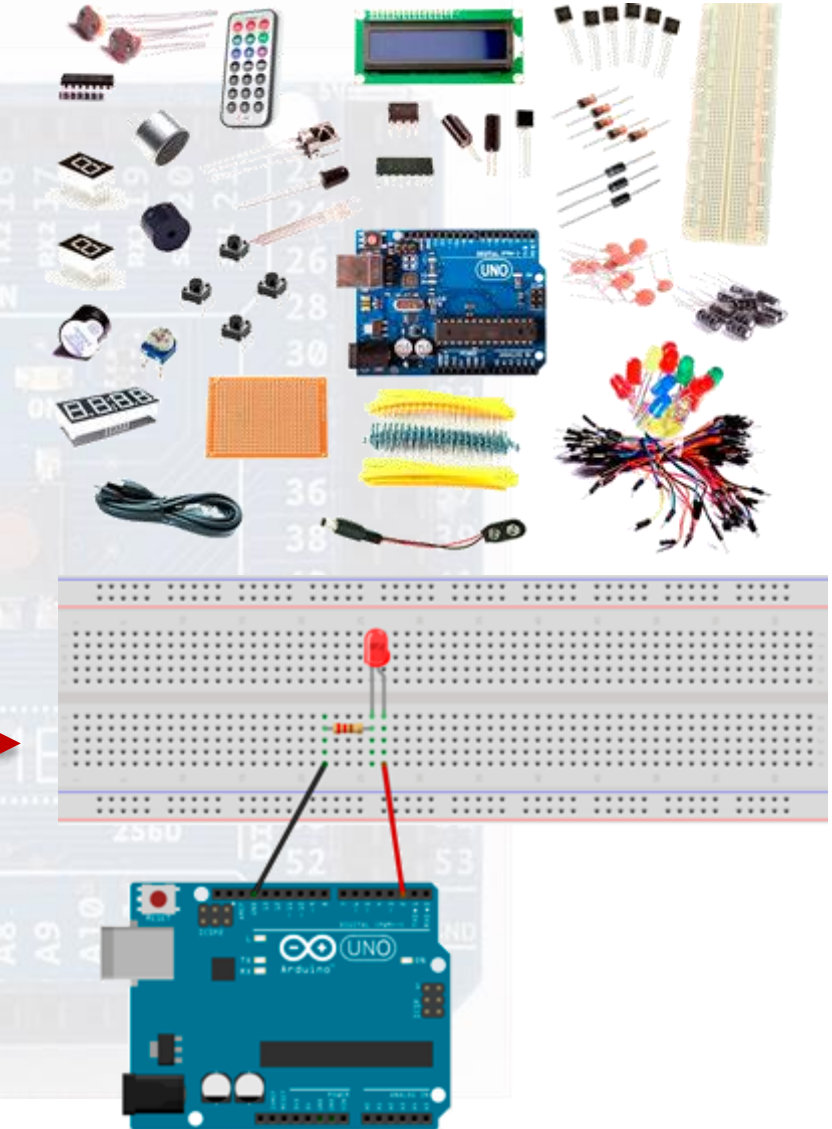
Encara que trobarem tots els esquemes a Internet, i gratis, pot ser factible comprar el kit d’inici amb manual i practiques.

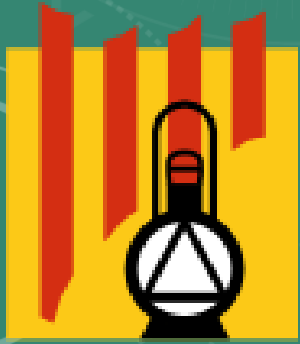
# PRIMERA PRÀCTICA

Descric aquí de forma ràpida el que hauria de ser la nostra primera pràctica amb Arduino (a la dreta el muntatge, a sota la programació)

- Connectar un LED a una de les sortides.
- Fer mitjançant la programació que faci intermitències.

```
1. void loop() {  
2.   digitalWrite(13, HIGH);  
3.   delay(500);  
4.   digitalWrite(13, LOW);  
5.   delay(500);  
6. }
```





**FEDERACIÓ  
CATALANA  
D'AMICS DEL  
FERROCARRIL**

# CONTROL D'ACCESSORIS AMB ARDUINO

PASSES BÀSIQUES PER UTILITZAR L'ARDUINO  
EN UNA MAQUETA DE TREN MINIATURA

# ESTRATÈGIA

Portar a lloc un projecte per Arduino implica una sèrie de paràmetres a tenir en compte:

1. Saber avaluar la nostra perícia, perquè no és el mateix començar sabent coses d'electrònica que des de zero. A vegades un bon manual pot servir.
2. Saber avaluar la complexitat del projecte propi, ja que posar en marxa petits gadgets de manual no és el mateix que iniciar un projecte propi que tingui certa complexitat i/o dificultat.
3. Saber avaluar les diferents fases de treball per acabar el projecte, tot i recordant que l'Arduino no farà allò que nosaltres ens oblidem.
4. Comparar les diferents solucions tecnològiques, tant en la part hardware com al software.
5. Hem de saber trobar la documentació adequada.
6. Re-copiar o inspirar-se en el treball d'altres.
7. Per a cada fase s'ha de treballar l'algoritme, el quadre sinòptic del que s'ha de fer en cada moment.
8. Avaluar els recursos i escollir l'Arduino adequat.
9. Escriure el codi en diferents versions ens pot ajudar a trobar el mètode més adient de control.
10. Establir un calendari i saber on trobar ajut.



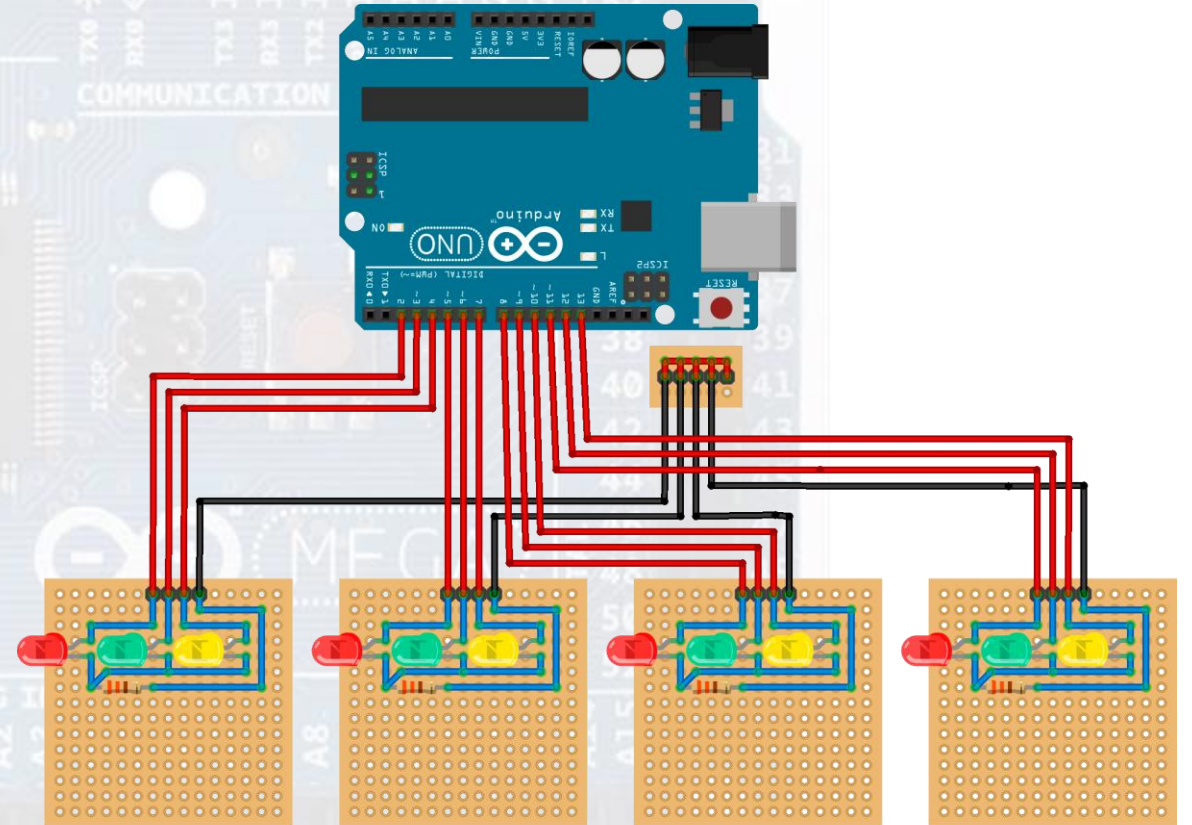
# CONTROL D'UN SEMÀFOR DE CRUÏLLA

Una de les aplicacions més senzilles que es poden fer amb Arduino són els semàfors, aquesta aplicació consisteix en crear una seqüència de LEDs de colors vermell, groc i verd, simulant el funcionament de quatre semàfors i controlar una cruïlla de carrers.

El codi es prou senzill, fa la seqüència corresponent a cada semàfor, en la secció de constants pots variar el temps de duració de cada llum, que està en milisegons.

La seqüència completa ja no és tan senzilla com la del LED intermitent, però és molt repetitiva i la forma seria donar unes constants en primer lloc, unes connexions en les següents línies i la inicialització, i posteriorment hi ha la seqüència per als quatre semàfors.

En total es fan 134 línies de codi.



# CONTROL D'UN SERVO

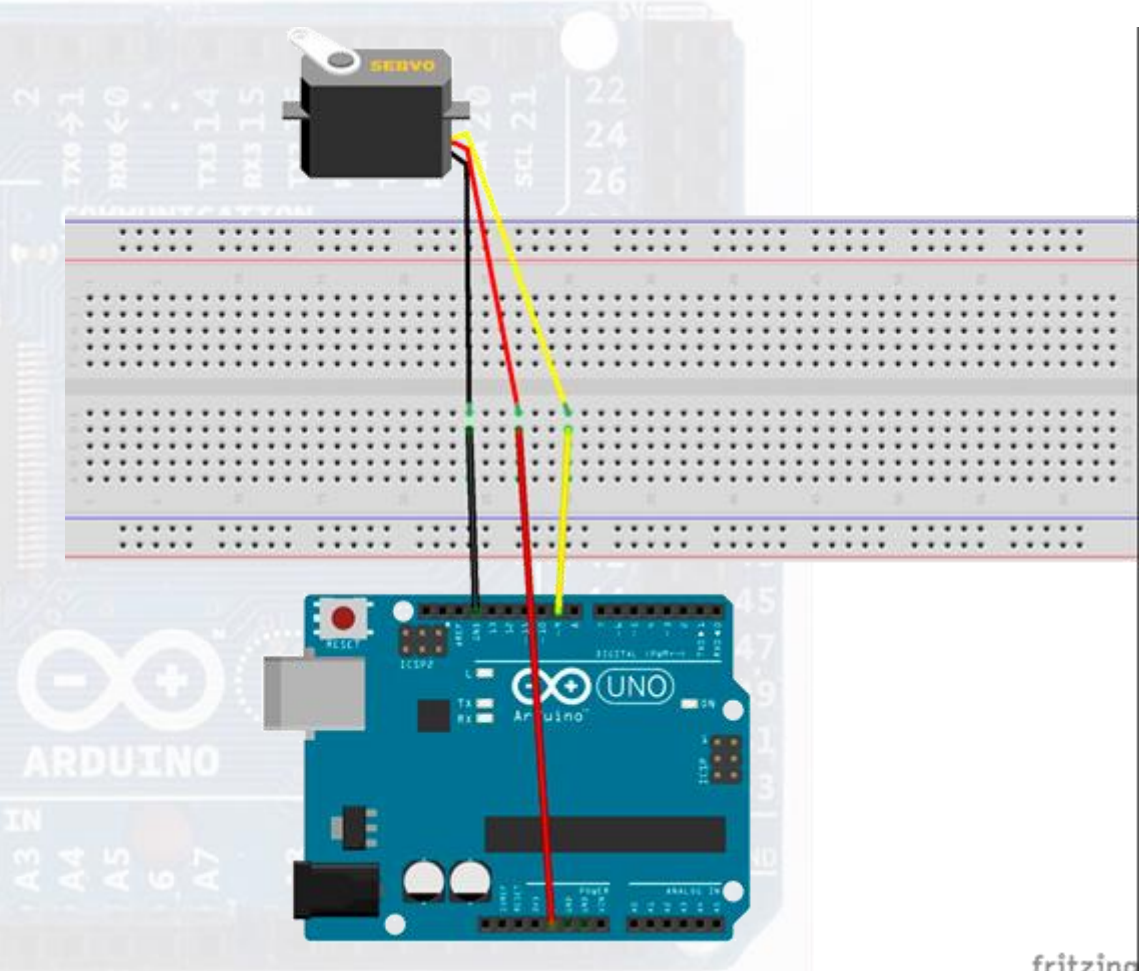
El servomotor és un dispositiu motoritzat que permet fer un angle de 180 graus d'angle de gir.

Es pot moure amb una resolució de més d'un grau, però aquest és el màxim de resolució que pot donar un Arduino UNO.

Aquests tipus de motors funcionen amb un senyal PWM amb un puls de treball entre 1 ms i 2 ms, amb un període de 20 ms (50 Hz). Això vol dir que és la velocitat màxima en la que podem moure el servos amb un Arduino.

Tots els motors servo han de tenir 3 cables, un anirà a terra o massa, l'altre és l'alimentació positiva a 5 volts i el tercer és un pin PWM que dona el senyal per moure el servo.

També es pot fer servir un shield per controlar els servomotors, si aquests no pogués ser controlats per l'Arduino.



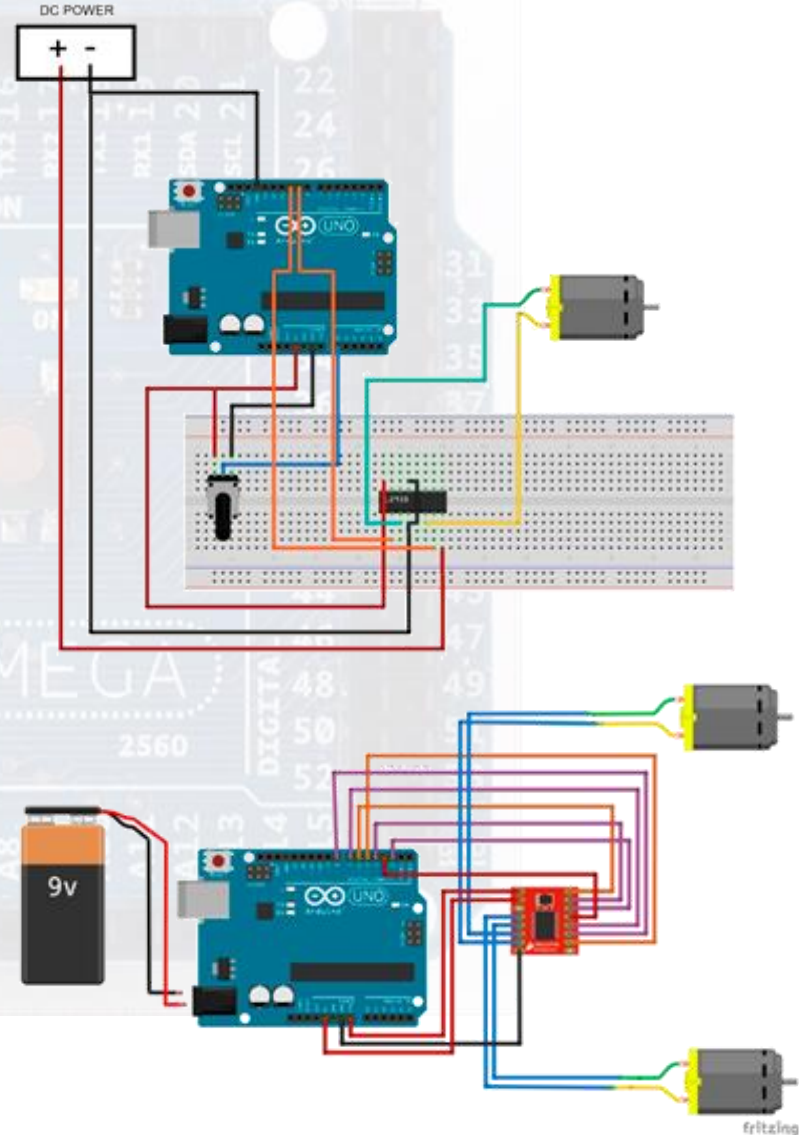
# CONTROL D'UN MOTOR CC

El motor de corrent continu es pot controlar directament per l'Arduino a través d'un muntatge.

El motiu és que l'Arduino no té potència suficient per moure motors, sinó que normalment dona l'ordre de moure's.

Es pot fer servir un potenciòmetre per graduar la velocitat i sentit del motor mitjançant un L293N, o si només és un un sentit, mitjançant un transistor de potència.

El més habitual és utilitzar un L298N que serveix per controlar motors de corrent continu (encara que també ho pot fer per motors pas a pas). El corrent que pot subministrar és de 2A i de 3 a 35 V. Però amb un inconvenient, que s'escalfa per poder perdre 3 volts fins arribar al motor.



# CONTROL D'UN MOTOR PAS A PAS

El motor pas a pas (també anomenat stepper) és un dispositiu que converteix impulsos elèctrics en moviments mecànics de rotació. Es a dir, es mouen un pas per cada impuls que reben.

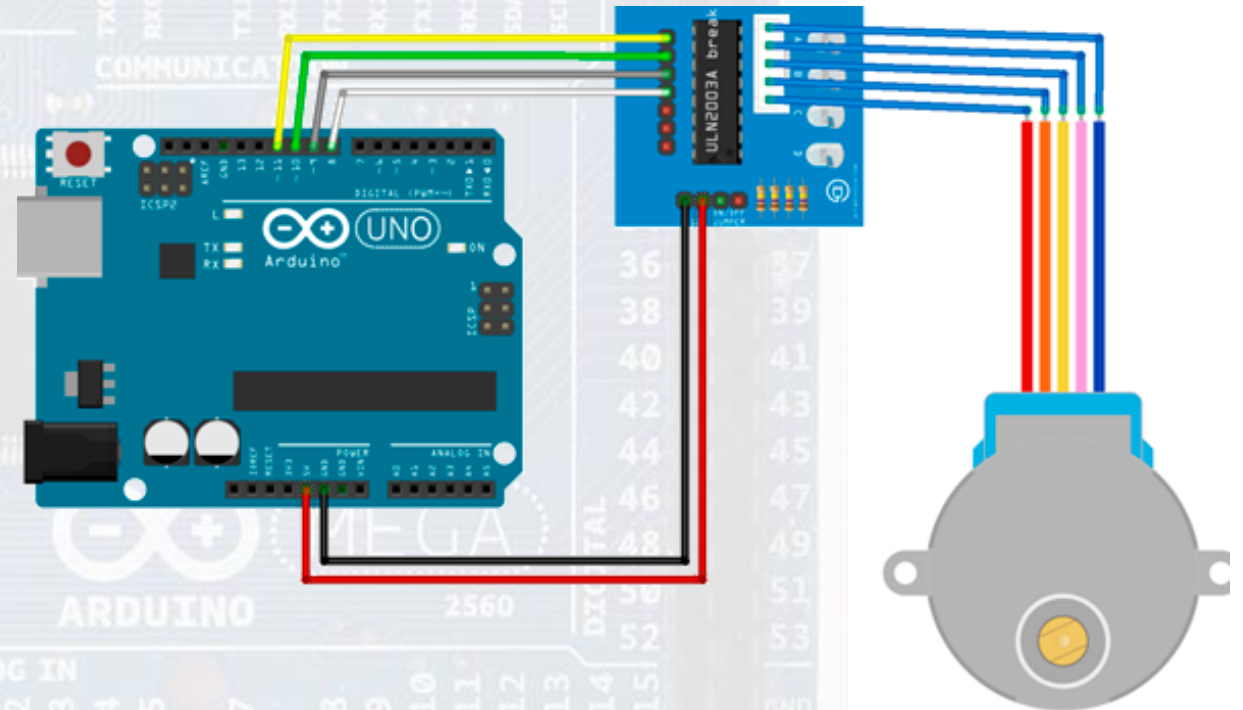
Hi ha dos tipus de motos pas a pas:

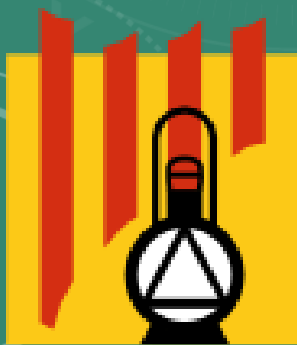
- Unipolar que conté 4 bobines amb comú entre parelles de bobines (6 terminals)
- Bipolar, que conté només 2 bobines (4 terminals)

Depenent del tipus de motor es fa un muntatge o un altre per fer el control del motor pas a pas.

El més habitual és utilitzar un ULN2803A (transistor Darlington) que serveix per donar potència als senyals provinents de l'Arduino 8uns 500 mA).

Hi ha uns Shields expressament concebuts pels motos pas a pas que s'utilitzen a les impresores 3D que tenen entre 4 i 5 sortides per controlar tots els motors a l'hora.





**FEDERACIÓ  
CATALANA  
D'AMICS DEL  
FERROCARRIL**

# ENLLAÇOS

PASSES BÀSIQUES PER UTILITZAR L'ARDUINO

EN UNA MAQUETA DE TREN MINIATURA



LOCODUINO L'Arduino per el tren miniatura, en francès: <https://www.locoduino.org/>

7 TRAIT PROJECTS <https://create.arduino.cc/projecthub/projects/tags/train>

ARDUINORAILWAYCONTROL <https://arduinorailwaycontrol.com>

APRENDIENDO ARDUINO <https://aprendiendoarduino.wordpress.com/>

FRITZING Open hardware Initiative: <http://fritzing.org/home/>

ARDUINO Pàgina oficial en anglès: <https://www.arduino.cc/>

EDUCALAB Taller de robòtica:  
<http://educalab.es/documents/10180/640047/TallerRoboticaLibreArduino.pdf/c77adbfd-606a-4fbe-acd4-11630927b5a4>