

Curs d'Arduino CTM16-17

Mòdul 1

CONTINGUTS INICIALS

CONCEPTES INICIALS D'ARDUINO

1. *Què és Arduino?*
2. *Novetats en Arduino: Arduino.cc i Arduino.org*
3. *Parts de l'Arduino UNO:*
4. *Descàrrega i instal·lació de l'IDE d'Arduino*
5. *Comprovació de la instal·lació de l'IDE*
6. *L'IDE d'Arduino*
7. *Estructura bàsica dels codis*
8. *Sentències bàsiques i funcions*
9. *Llibreries*
10. *Tipus de memòria en Arduino: Flash, SRAM i EEPROM*
11. *Compilació i càrrega*
12. *Com espatllar l'Arduino*
13. *Programació Arduino*

SIMULADORS D'ARDUINO

1. *Fritzing*
2. *123D Circuit.io*

PROGRAMACIÓ GRÀFICA D'ARDUINO

1. *Scratch for Arduino (S4A)*
2. *Bitbloq*
3. *Altres entorns gràfics*

CONCEPTES INICIALS D'ARDUINO

1. Què és Arduino?

Arduino és:

- una placa controladora,
- amb una sèrie d'entrades i sortides
- i un entorn de programació

que permeten de manera ràpida i senzilla realitzar projectes d'electrònica, automatisme, control, domòtica, etc.

La plataforma Arduino es basa en prototips d'electrònica de codi obert (open-source) basada en maquinari i programari flexibles i fàcils d'utilitzar.

Hi ha diversos models d'Arduino com són UNO, Leonardo, Mega, Nano, ... encara que l'Arduino UNO, fins ara, és el més comú i econòmic.



Arduino Uno



Arduino Leonardo



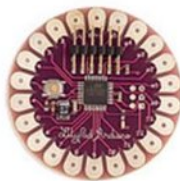
Arduino Ethernet



Arduino Pro



Arduino Mega 2560



Arduino LilyPad



Arduino BT



Arduino Nano



Arduino Mega ADK



Arduino Fio



USB/Serial Light Adapter



Arduino Mini

Per conèixer les característiques de cadascuna i altres models:

<http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>

El programari es pot descarregar gratuïtament. Els seus esquemes, especificacions i plànols estan disponibles al públic donant la possibilitat de la fabricació d'una còpia idèntica a l'original.

2. Novetats en Arduino: Arduino.cc i Arduino.org

A principis de 2015 es va produir una divisió dins d'Arduino i des de llavors ha aparegut una nova marca anomenada " Genuino " i una nova web oficial d'Arduino www.arduino.org . Però no només és que hi hagi dues pàgines web oficials d'Arduino, sinó que ara hi ha dos entorns de programació i han aparegut nous models de plaques Arduino però fabricades per diferents empreses.

En aquest moment hi ha dues pàgines oficials d'Arduino:

- www.arduino.cc
- www.arduino.org

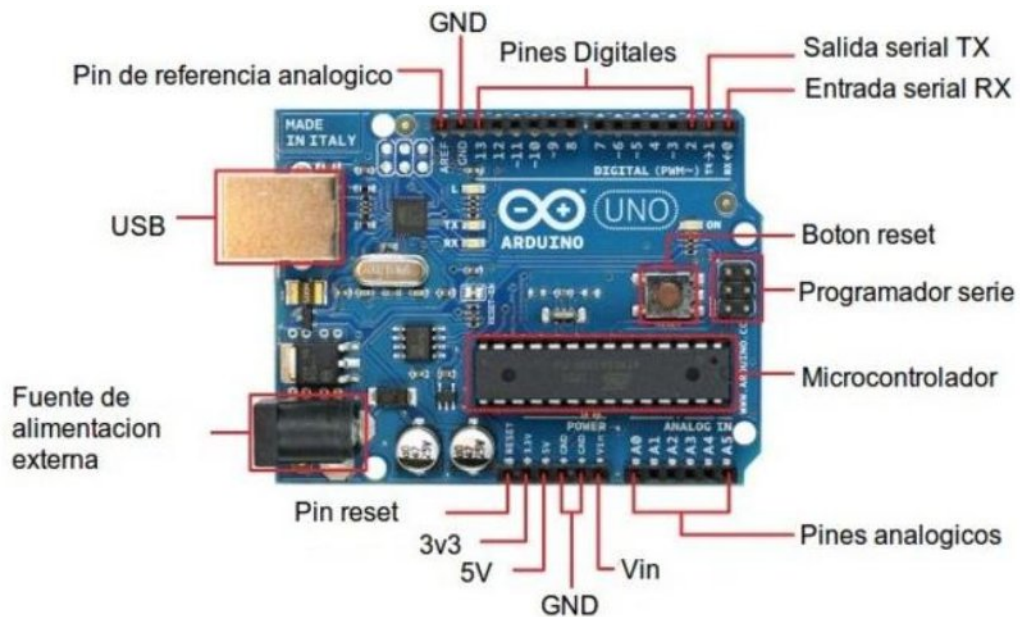
La primera és la que va néixer originalment i la segona és la que es va crear arran de la divisió entre l'equip creador d'Arduino i la empresa que fabricava el maquinari a Itàlia.

Cadascuna d'aquestes webs tenen plaques Arduino diferents, IDEs diferents i marques diferents amb l'aparició de Genuino a Europa.

Algunes notes a tenir en compte:

- Una conseqüència d'aquesta divisió és que les plaques d' [arduino.org](http://www.arduino.org) poden no funcionar amb l'IDE original d'[arduino.cc](http://www.arduino.cc) i al contrari.
- El més probable és que hi hagi problemes amb els Arduinos nous que vagin sortint, però amb els Arduinos anteriors a la separació d'Arduino seguiran sent suportats en tots dos IDEs.
- També és possible que cada un dels IDEs d'[arduino.cc](http://www.arduino.cc) i [arduino.org](http://www.arduino.org) tinguin versions diferents de les llibreries que tenen incloses, això ens pot portar problemes a l'usar un sketch en un IDE o en un altre i caldrà tenir-ho en compte.

3. Parts de l'Arduino UNO:



Alimentació:

- o connexió USB (proporciona 5 V)
- o jack d'alimentació (mitjançant pila de 9 V o font d'alimentació entre 7 - 12 V)

Pins d'alimentació són per a alimentar els circuits connectats a la placa:

- o 3.3 V proporciona un voltatge de 3,3 V i una intensitat màxima de 50 mA
- o 5 V proporciona un voltatge de 5 V i una intensitat màxima de 300 mA
- o GND Ground (massa) o nivell 0 V de referència
- o Vin proporciona el voltatge màxim d'alimentació d'Arduino

Valors d'entrada i sortida: hi ha pins que s'utilitzen com INPUTs o OUTPUTs:

- o Sortida i entrada digital: els valors de sortida (OUTPUT) poden ser 0 V (LOW) o 5 V (HIGH). Els pins digitals van del 0 al 13

LOW $\rightarrow$ 0 - 2V; HIGH $\rightarrow$ 3 - 5V

- o Sortida analògica: els valors de sortida van des de 0V a 5V en un rang de 0 a 255 (precisió de 8 bits). Coincideixen amb els pins digitals i estan indicats com PWM (Pulse Width Modulation) amb el símbol $\sim$.

- o Entrada analògica: els valors d'entrada van des de 0V a 5V en un rang de 0 a 1023 (precisió de 10 bits). Aquests pins són A0, A1,... A5

Altres pins:

- o REF Referència de voltatge
- o TX Transmissió d'escriptura en comunicació sèrie
- o RX Lectura en comunicació sèrie
- o RESET Reinicia el programa
- o A4 (SDA) Entrada analògica i comunicació sèrie I²C per l'enviament de dades
- o A5 (SCK) Entrada analògica i comunicació sèrie I²C pel senyal de rellotge
- o ICSP (In Circuit Serial Programming):
 - per a comunicació SPI (Serial Peripheral Interface), estàndard de comunicacions, utilitzat principalment per a la transferència d'informació entre circuits integrats en equips electrònics
 - per programar el BootLoader del Microcontrolador ATmega sense necessitar programadors externs

La intensitat màxima d'aquests pins és de 40mA

Per saber més sobre la placa d'Arduino:

<https://www.arduino.cc/en/Guide/BoardAnatomy>

4. Descàrrega i instal·lació de l'IDE d'Arduino

Descarreguem la versió més recent de l'IDE (Entorn Integrat de Desenvolupament) d'Arduino segons el sistema operatiu del nostre sistema i des de la pàgina:

<http://arduino.cc/en/main/software>

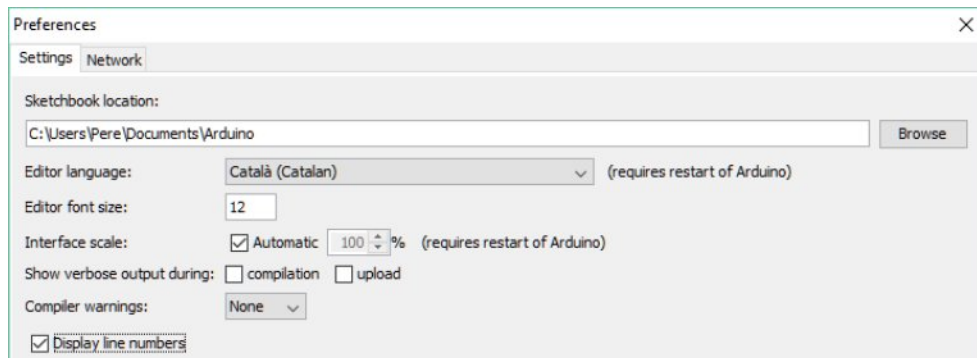
Un cop descarregat, executem el fitxer descarregat i instal·lem el software d'Arduino. Ens apareixerà la següent icona a l'escriptori.



Si volem canviar l'idioma de l'aplicació:

[Menú] / File / Preferences / Editor Language → Català

I per visualitzar el número de cada línia assenyalar [Display line numbers](#)

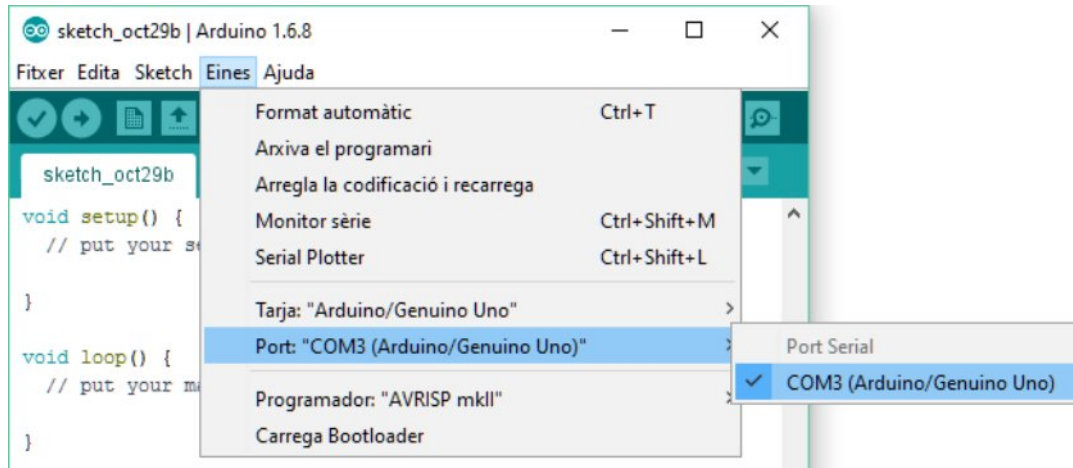


Per fer efectius els canvis s'haurà de reiniciar l'IDE

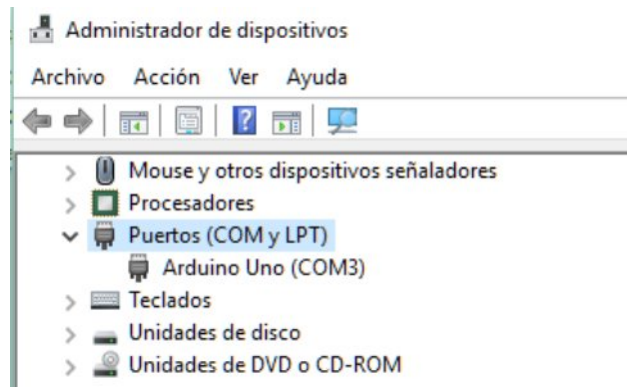
5. Comprovació de la instal·lació de l'IDE

Un cop instal·lat l'IDE d'Arduino, anem a comprovar que reconeix la nostra placa correctament i que podem programar-la. Per a això, connectem la placa Arduino a l'ordinador mitjançant el cable USB i comprovem que els llums de la placa s'il·luminen indicant que té alimentació.

Des del [Menú] / Eines / Tarja seleccionem Arduino Genuino UNO i a [Menú] / Eines / Port observem que hi ha un port sèrie COM marcat amb un tick



En el cas que apareguin més COMs a la llista i no el puguem identificar podem anar al Panel de control/Hardware/Administrador de dispositius, mirem la pestanya Ports i ens apareix el nostre Arduino i el COM al que està connectat.



Per comprovar que ja funciona correctament, executarem un programa d'exemple. Anem a:

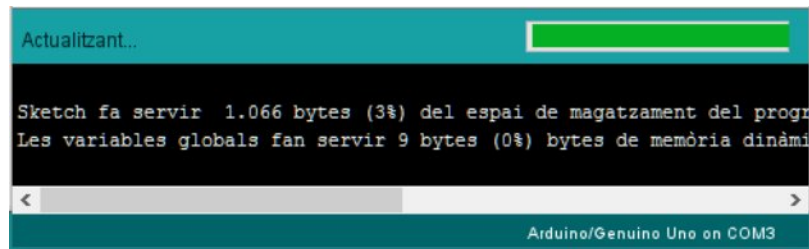
[\[Menú\] / Arxiu / Exemples / 01.Basics / Blink](#)



Clicar sobre la fletxa

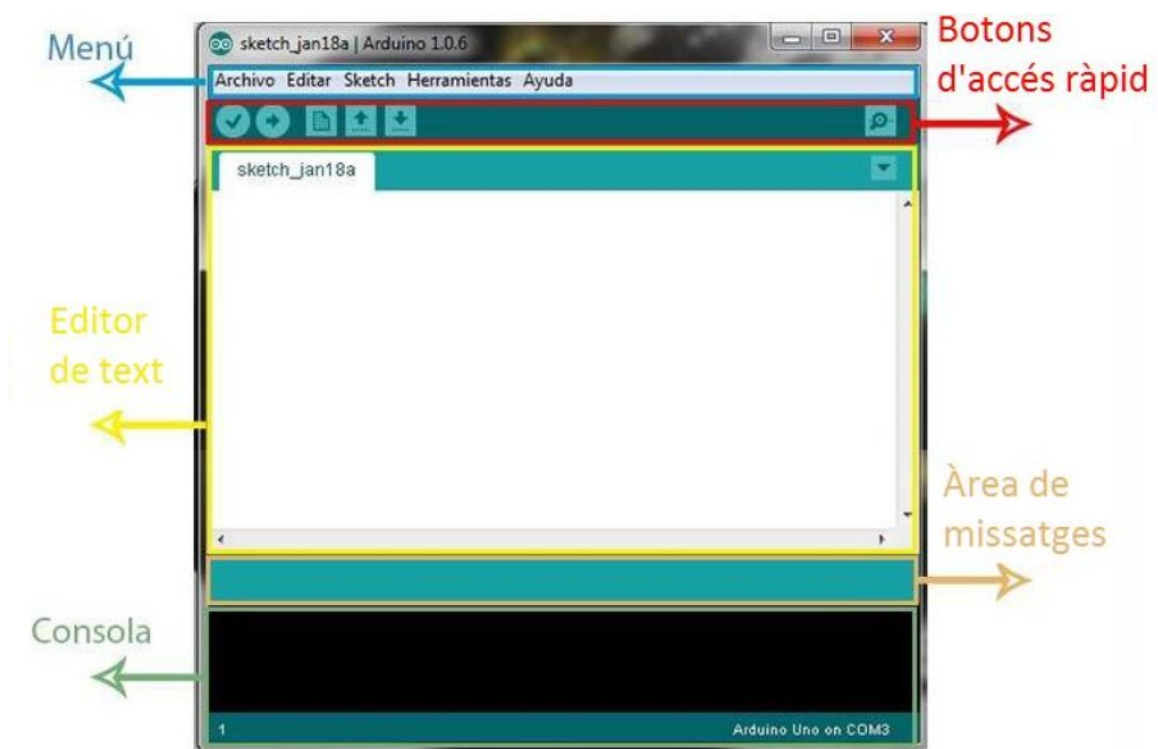


I comprovar que després de veure com s'omple una línia de color verd de compilació de programa apareix un missatge amb text de color blanc a la part inferior del programa, indica que la pujada està enllestida i parpelleja un LED de la placa d'Arduino.



Si ha anat tot així, significa que l'entorn ja està instal·lat i configurat correctament.

6. L'IDE d'Arduino



Menú

A la part de menú tenim una zona per accedir a funcions com càrrega d'arxius, edició del text del codi, càrrega de llibreries i exemples, configuració, eines ... etc.

Botons d'accés ràpid



Verifica si el programa està ben escrit i funciona



Carrega el programa a la placa d'Arduino després de compilar-lo



Crea un programa nou



Obre un programa



Guarda el programa en el disc dur



Monitor Serial. Obre una finestra de comunicació amb l'Arduino connectat amb el cable USB

Editor de text

En el quadre de l'editor de text escrivim el codi del programa que vulguem que Arduino executi.

Àrea de missatges i consola

L'àrea de missatges i la consola Arduino ens aniran donant informació sobre si la consola està compilant, carregant ... i sobre els errors que es produeixen tant en el codi com en el propi IDE.

Un programa d'Arduino s'anomena sketch o projecte i és un fitxer amb extensió `.ino`. Aquest sempre ha d'estar en una carpeta amb el mateix nom que el fitxer.

7. Estructura bàsica dels codis

Un codi Arduino és una sèrie d'ordres de programació que li diran al microcontrolador:

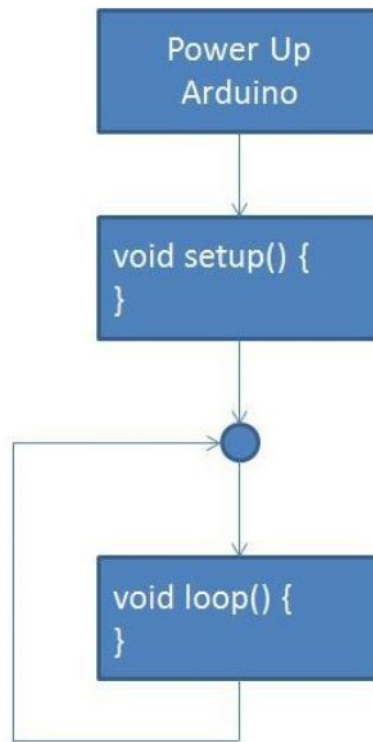
- com configurar l'inici i
- quines accions ha de realitzar mentre estigui en funcionament.

El bloc de codi ha de tenir 2 parts, o funcions principals, que sempre hem d'incloure:

- `void setup () { }`
 - Part encarregada de recollir la configuració.
 - Hi escrivim les sentències de configuració (mode de funcionament dels pins, comunicació sèrie, activació de sensors, ...) que necessitem perquè Arduino treballi correctament.
 - Aquestes s'executaran en iniciar l'Arduino i una única vegada.
- `void loop () { }`
 - Sempre després de `void setup ()`.

- Hi escrivim totes les sentències, bucles i crides a funcions que necessitem que el nostre Arduino repeteixi constantment.
- S'escriuran en ordre d'execució.

Existeixen altres parts del codi que no són obligatòries però que podem necessitar per organitzar-ho tot. Aquestes zones poden ser un espai per al títol, l'autor, llibreries, definicions, variables i funcions.



8. Sentències bàsiques i funcions

Les sentències i funcions són les eines que utilitzem per dir-li al nostre microcontrolador que és el que ha de fer en cada moment. I aquestes poden ser:

- Estructurals:

Donen la forma computacional del codi i les operacions lògiques a executar. Amb aquests indiquem el camí que ha de seguir el codi a executar-se i quines operacions ha de fer.

if, for, while, +, -, =, *, /, ==, >, < ...

- Variables:

Defineixen els tipus i característiques que tindran els valors que farem servir en el codi. Poden ser nombres, caràcters o estats.

int, float, HIGH, LOW, char, string, true, false ...

- Funcions:

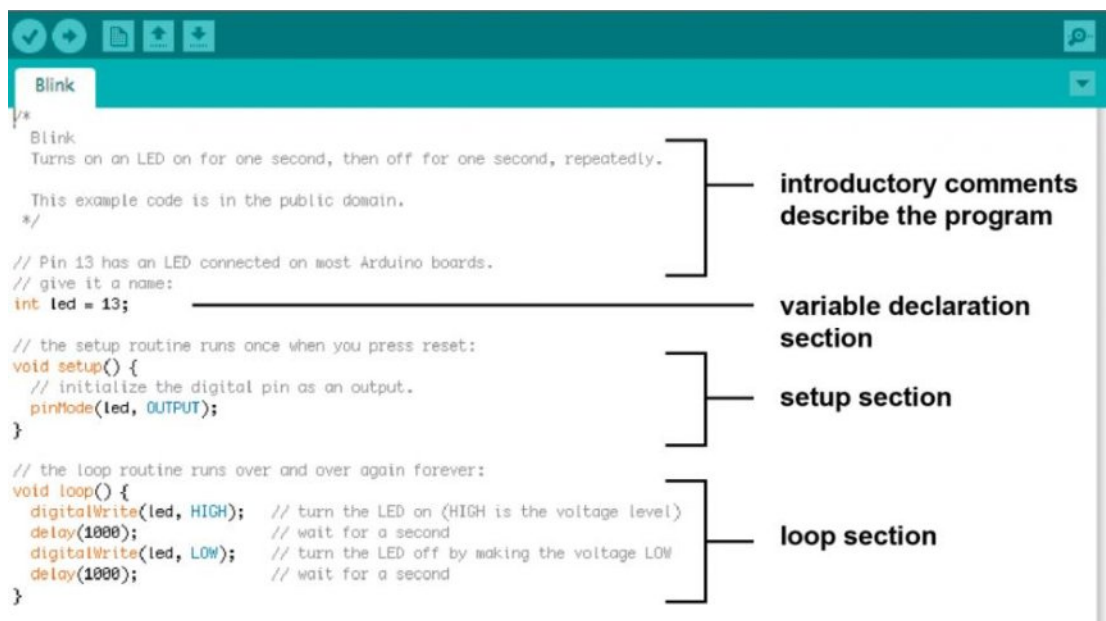
Realitzen una sèrie d'operacions concretes fora del codi principal.

Al següent enllaç de la web oficial d'Arduino hi ha un resum de totes les sentències estructurals, variables i funcions:

<https://www.arduino.cc/en/Reference/HomePage>

Dos detalls necessaris per poder programar el nostre codi:

- Cada sentència o funció que s'executi ha de dur al final de cada ordre el signe de puntuació punt i coma ;
- Utilitzarem la doble barra // per indicar que el que s'escriu a la dreta és un comentari. El codi no l'executarà. També es poden fer comentaris entre els símbols /* i */



9. Llibreries

Les llibreries són:

- trossos de codi fets per tercers que fem servir en el nostre sketch.
- col·leccions de codi que faciliten la interconnexió de sensors, pantalles, mòduls electrònics, etc.

- una sèrie de sentències i funcions específiques de cada element o shield que connectem a Arduino i que realitzen accions específiques de cada un facilitant-nos molt la programació.

Les llibreries inclouen arxius comprimits en un arxiu ZIP o dins d'un directori. Aquestes sempre contenen:

- un arxiu .cpp (codi de C ++),
- un arxiu .h capçalera de C, que conté les propietats i mètodes o funcions de la llibreria,
- un arxiu Keywords.txt , que conté les paraules clau que es ressalten en l'IDE (opcional),
- un arxiu readme amb informació addicional del que fa i amb instruccions de com usar-la (opcional),
- un directori denominat examples amb diversos esquets d'exemple que ens ajudarà a entendre com utilitzar la llibreria (opcional).

Per afegir una llibreria al nostre projecte simplement s'afegeix al nostre codi la paraula clau `#include` seguit del nom de la llibreria.

Com instal·lar llibreries:

<http://arduino.cc/en/Guide/Libraries>

El mateix IDE d'Arduino ja porta integrades diverses llibreries, però en podem descarregar altres i incorporar-les al nostre IDE per utilitzar-les en els nostres programes.

Les llibreries instal·lades per nosaltres es guarden en el directori indicat des de les preferències de l'IDE.

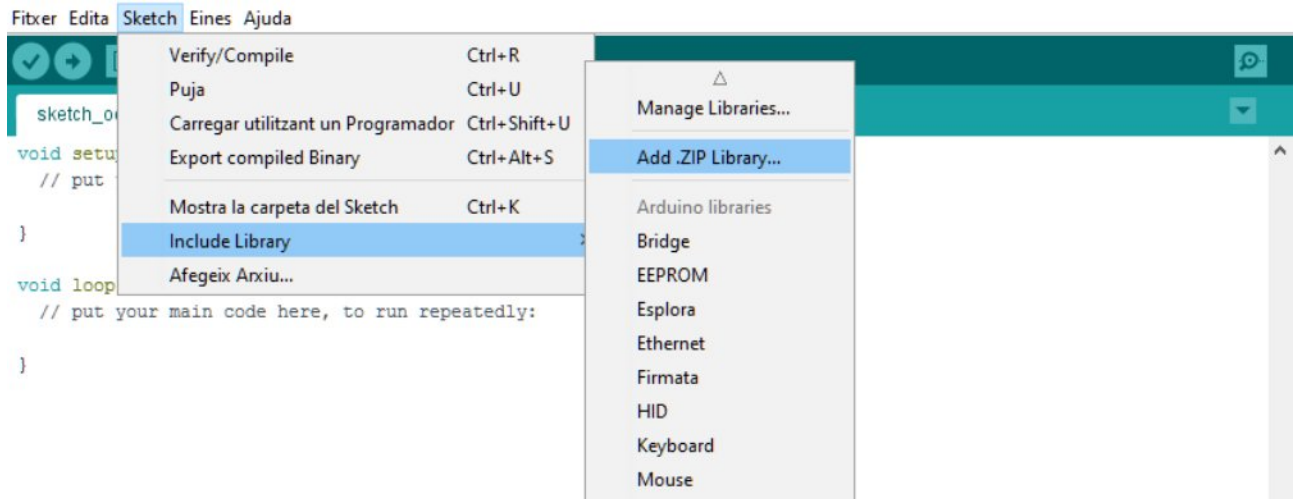
[\[Menú\]](#) / [Fitxer](#) / [Preferències](#)

Podem instal·lar llibreries:

- Mitjançant l'IDE d'Arduino de manera automàtica des de

[\[Menú\]](#) / [Sketch](#) / [Include Library/ Add .ZIP Library...](#)

Admet la instal·lació des d'un fitxer zip o des d'una carpeta ja descomprimida.



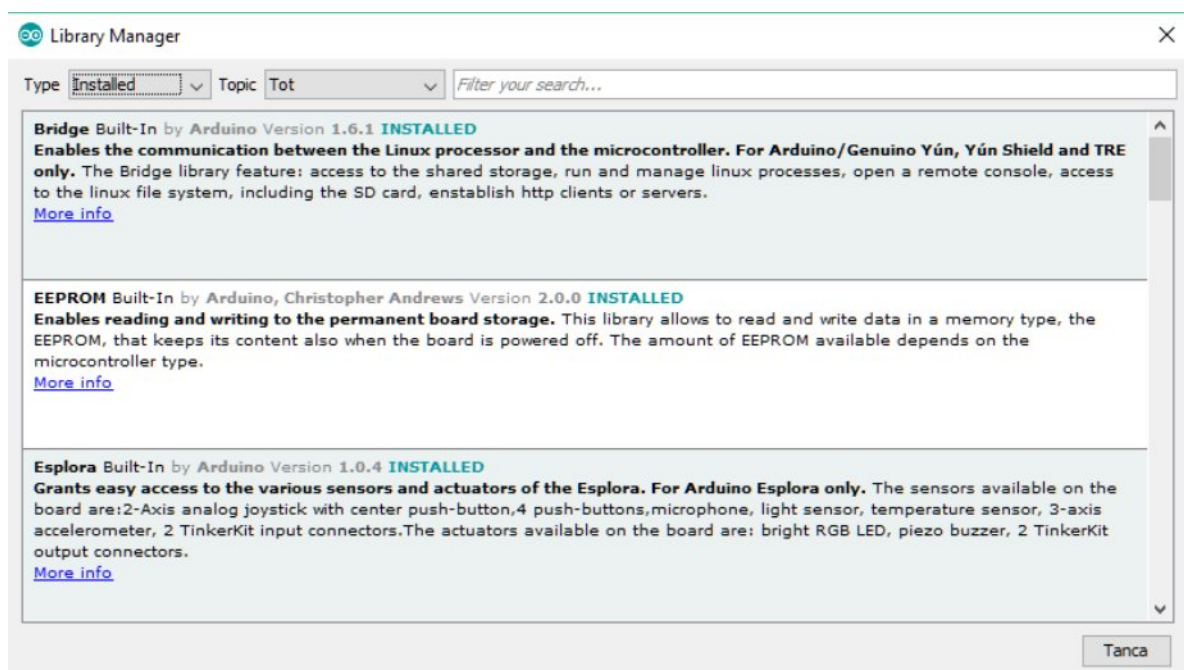
- Instal·lació Manual. Descomprimint en un directori la llibreria i copiant al directori llibries. Generalment a Els meus Documents

C:\Users\Nom_usuari\Documents\Arduino\libraries

- Des del gestor de llibries. Aquesta eina és accessible des de

[\[Menú\] / Sketch / Include Library / Manage Libraries...](#)

Des d'aquí podem veure les llibries instal·lades, buscar llibries disponibles, instal·lar llibries i actualitzar-les. Aquesta eina també ens permet gestionar les llibries instal·lades manualment.



Aquest enllaç explica com actualitzar una llibreria:

<https://github.com/firmata/arduino#updating-firmata-in-the-arduino-ide-arduino-164-and-higher>

NOTA: Abans d'instal·lar amb el gestor de llibreries, s'ha de desinstal·lar la llibreria si s'ha instal·lat anteriorment. Per desinstal·lar, esborrar el directori de la llibreria i reiniciar l'IDE d'Arduino.

10. Tipus de memòria en Arduino: Flash, SRAM i EEPROM

Hi ha tres tipus de memòria en els microcontroladors utilitzats per les plaques Arduino (ATmega168, ATmega328, ATmega1280, etc.):

- memòria Flash,
- memòria SRAM
- memòria EEPROM.

Memòria Flash (espai del programa)

La memòria Flash (espai del programa):

- És on Arduino emmagatzema l'sketch.
- La memòria és no volàtil, si Arduino deixa de ser alimentat elèctricament les dades que hi hagi en aquesta memòria romandran.

Memòria SRAM (Static Random Access Memory)

La memòria SRAM (Static Random Access Memory o memòria estàtica d'accés aleatori):

- és de tipus volàtil, és l'espai on els esquetsos emmagatzemen i manipulen variables a l'executar-se,
- la informació guardada en aquesta memòria serà eliminada quan Arduino perdi l'alimentació.
- és d'ús exclusiu per al programa en execució.
- Si la SRAM es queda sense espai, el programa d'Arduino fallarà de manera imprevista, tot i que es compili i es pugi a Arduino correctament.

Memòria EEPROM en Arduino

La memòria EEPROM:

- memòria no volàtil, pel que les dades guardades en ella romandran encara que Arduino perdi l'alimentació i pot ser utilitzat pels programadors per emmagatzemar informació a llarg termini,
- només pot llegir byte a byte i el seu ús pot ser una mica incòmode.

Per exemple, el microcontrolador ATmega328 (com el que incorpora Arduino UNO) utilitza aquestes quantitats de memòria:

- Flash = 32KB, dels quals el bootloader fa servir 0,5KB
- SRAM = 2KB (2048 bytes)
- EEPROM = de 1KB (1024 bytes). Arduino admet afegir mòduls de memòria EEPROM externa de major grandària.

ACONSELLABLE: desenvolupar els programes de forma molt optimitzada, fent servir els tipus de variables que menys memòria requereixin i optimitzar el codi font de l'aplicació per no repetir línies de codi.

11. Compilació i càrrega

Un cop escrit el nostre codi a la IDE Arduino procedirem a verificar que està correctament escrit. Per a això premem el botó:



Arduino compilarà el codi del llenguatge de programació a un llenguatge que entengui el microcontrolador directament.

En el cas de contenir un error ens apareixerà a la consola una explicació orientativa de l'error.

Si la compilació ha estat correcta, ja podem carregar el nostre programa a Arduino. Amb la placa connectada per USB, premem el botó de càrrega:



12. Com espatllar l'Arduino

Abans de començar les pràctiques i treballar amb l'Arduino UNO, vegem 10 formes de destruir un Arduino, per saber que NO hem de fer:

<http://www.trastejant.com/2013/07/03/10-formas-de-destruir-un-arduino/>

13. Programació Arduino

El llenguatge de programació d'Arduino és una adaptació de C++. Aquest és un llenguatge de mig nivell que tracta amb objectes com caràcters, números, bits, adreces de memòria, etc.

Tota la informació per programar Arduino es troba al Reference de la web d'Arduino:

<https://www.arduino.cc/en/Reference/HomePage>

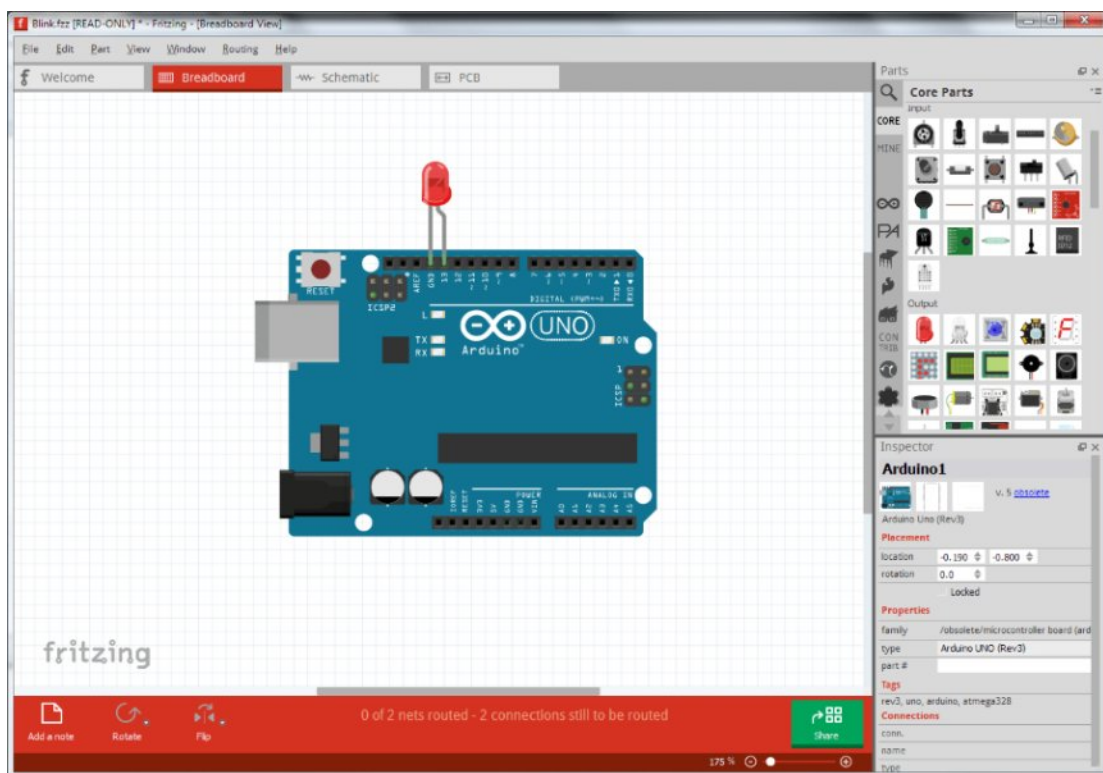
És possible utilitzar ordres estàndard de C++ en la programació d'Arduino sempre que estiguin inclosos en l'avr-libc (eina bàsica per programar els microcontroladors AVR d'Atmel).

SIMULADORS D'ARDUINO

1. Fritzing

Principals característiques:

- programari codi obert per realitzar esquemes elèctrics en projectes amb Arduino,
- disposa de biblioteques amb la majoria de components inclòs els propis Arduinos, plaques de connexions, LED, motors, displays, etc.
- permet fer esquemes elèctrics, dissenyar el nostre PCB final i una infinitat d'opcions que converteixen aquest programa en una eina molt útil.
- permet obtenir l'esquema elèctric, llistat de component usats i el diagrama per poder fabricar la PCB. Després podem demanar que ens facin la PCB amb el servei Fritzing Fab.
- Les noves versions de Fritzing permeten guardar el codi d'un disseny i fins i tot carregar-lo en Arduino.

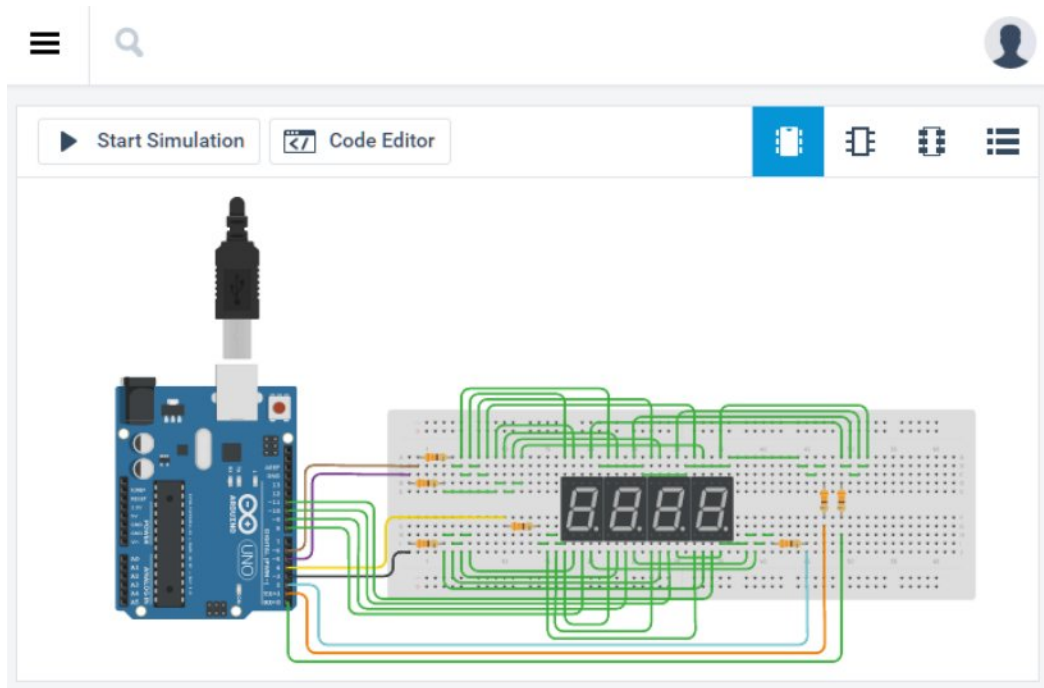


Enllaços:

- Web Principal: <http://fritzing.org/home/>
- Descàrrega: <http://fritzing.org/download/>
- Sol·licitar fabricar disseny realitzat amb Fritzing: <http://fab.fritzing.org/fritzing-fab>

2. 123D Circuit.io

<https://circuits.io/>



- Eina online gratuïta d'Autodesk que permet dibuixar esquemes de manera similar a Fritzing.
- Permet simulació de circuits mitjançant la "programació virtual" de les plaques Arduino i comprovar el funcionament.

PROGRAMACIÓ GRÀFICA D'ARDUINO

1. Scratch for Arduino (S4A)



http://s4a.cat/index_ca.html

Scratch for Arduino (S4A) és una versió modificada d'Scratch, preparada per interactuar amb plaques Arduino. Va ser desenvolupat per l'equip de Smalltalk del Citolab a l'any 2010 i des de llavors s'ha fet servir a tot el món per a una gran varietat de projectes.

Avantatges:

- forma senzilla per interactuar amb el món real, aprofitant la facilitat d'ús d'Scratch
- no cal és necessari definir variables o inicialitzar pins ja que l'script fa tot aquest treball per nosaltres
- S4A és una implementació molt completa d'Scratch per treballar amb Arduino

Inconvenient:

- exigeix tenir connectada la placa Arduino en tot moment a l'ordinador i per tant, no és possible que l'Arduino funcioni de manera autònoma.

2. Bitbloq



<https://bitbloq.bq.com/#/>

Bitbloq és un entorn de programació gràfica dissenyat per l'empresa espanyola BQ, està pensat per ser utilitzat amb el kit de robòtica que ells mateixos distribueixen però pot ser usat amb la placa Arduino.

Avantatges:

- Es programa online de manera que no necessita instal·lació. El seu únic requisit és tenir instal·lada una versió recent del navegador Chrome i connexió a internet.
- la interfície és molt intuïtiva,
- els projectes és poden guardar i obrir online
- Es pot visualitzar el codi generat pel programa amb la possibilitat de copiar-lo i enganxar-lo a l'IDE d'Arduino.

Inconvenients:

- Actualment no està massa depurat i presenta alguns errors
- els blocs no és corresponen 100% amb la nomenclatura Arduino

3. Altres entorns gràfics

- Ardublock: <http://blog.ardublock.com/>
- Visualino: <http://www.visualino.net/>
- mBlock: <http://www.mblock.cc/>