

Què és l'
Arduino?

És una plataforma
de codi obert per a
prototips electrònics.
I això què vol dir?

plataforma -

"Arquitectura de maquinari amb un sistema de programari
sobre la qual es pot executar un (altre) programa."

codi obert -

"Recursos que poden ser usats, redistribuïts o reescrits lliures de càrrec.
Sovint programari o maquinari."

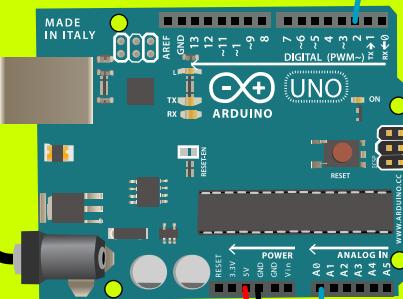
prototip -

"Model original que pot servir com a base per altres coses."

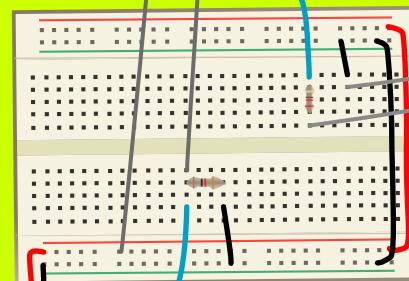
electrònica -

"Tecnologia que utilitza el corrent elèctric per
controlar quelcom automàticament."

microcontrolador



fotocel·la

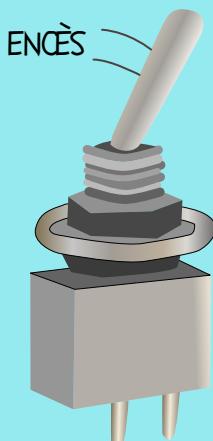


LED



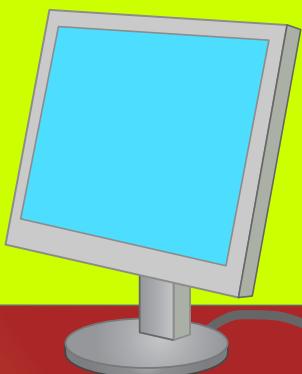
placa de connexions

Un Arduino conté un microcontrolador, que és un ordinador en miniatura que pots programar. Li pots connectar sensors per mesurar condicions (per exemple quanta llum hi ha a l'habitació). Pot controlar com altres objectes reaccionen a aquestes condicions (si l'habitació s'enfosqueix el LED s'engega).

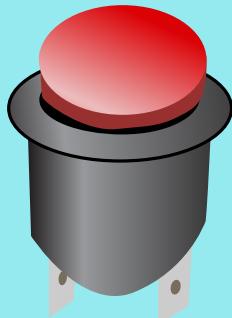


O pot respondre a quelcom tan senzill com accionar un interruptor.

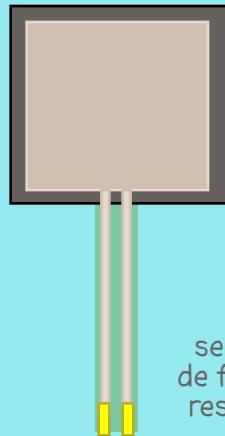
un ratolí és un dispositiu d'entrada d'un ordinador, una pantalla és un dispositiu de sortida.



Els microcontroladors utilitzen les entrades i sortides com qualsevol ordinador. Les entrades capten informació de l'usuari o de l'entorn, mentre que les sortides fan alguna acció amb la informació que ha estat captada.



pulsador

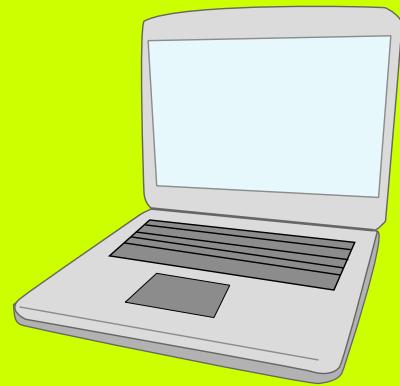


sensor
de forces
resistiu

Un interruptor o un sensor poden ser una entrada per l'Arduino.



motor de C.C.



Qualsevol objecte que volguem engegar o aturar i controlar pot ser una sortida. Pot ser un motor o fins i tot un ordinador.



Quina diferència hi ha entre entrades i sortides digitals i analògiques?

Les entrades i sortides poden ser digitals o analògiques. La informació digital és binària -és o bé cert o fals-. La informació analògica és contínua, pot abarcar un rang de valors.

La informació digital és discreta i finita. Tota la informació es descriu en dos estats, 1 o bé 0, on o bé off.

La informació analògica es caracteritza per la seva naturalesa contínua. Pot tenir un infinit nombre de possibles valors.



Un interruptor és una entrada digital, un sensor és una entrada analògica. El rang d'un sensor analògic està limitat per la seva conversió a dades digitals.



tensió?
intensitat?
resistència?
llei d'Ohm?

Abans de connectar l'Arduino revisarem uns conceptes i principis relacionats amb el comportament del corrent elèctric (i per tant de l'electrònica).

Tensió (U)
és la mesura
del potencial
elèctric.
Es mesura
en **volts (V)**.

Intensitat (I)
és la velocitat
del flux a través
d'un material
conductor.
Es mesura
en **ampères (A)**.

Resistència (R)
és l'oposició
del material
al flux del
corrent elèctric.
Es mesura
en **ohms (Ω)**.

El corrent elèctric és el flux d'energia a través d'un material conductor.



la pressió del raig
ve determinada per la tensió



la resistència fa decreixer o
creixer el raig



el cabal d'aigua equival
a la intensitat

L'analogia de l'aigua s'utilitza habitualment per explicar aquests conceptes. Aquí tenim un model.

Ilei d'OHM

intensitat = tensió / resistència
 $(I = U / R)$

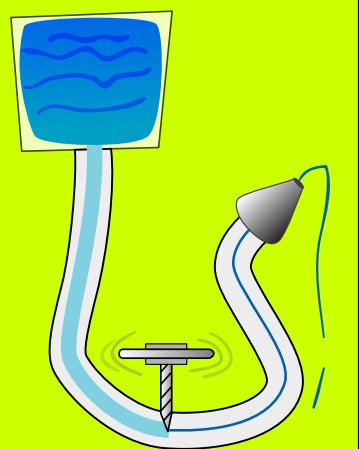
o bé

resistència = tensió / intensitat
 $(R = U / I)$

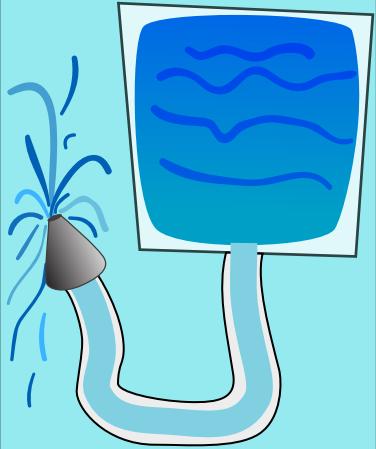
o bé

tensió = resistència * intensitat
 $(U = R * I)$

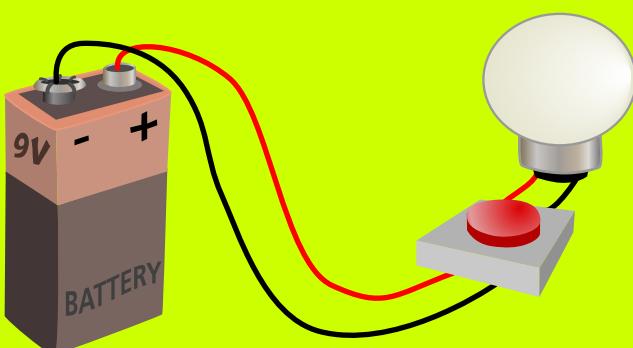
Hi ha una relació entre tensió, intensitat i resistència, descoberta per en Georg Ohm, un físic alemany.



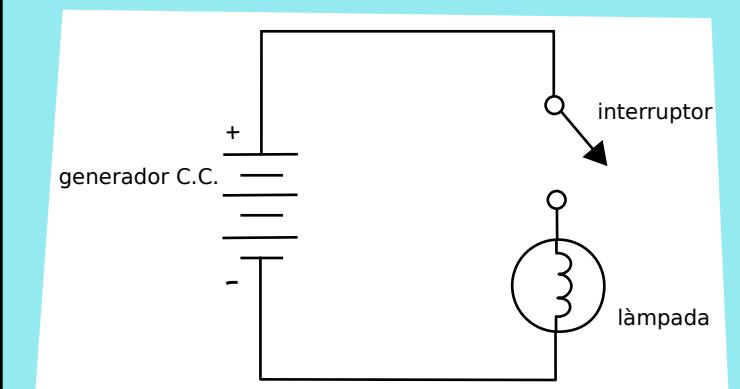
Per exemple, si s'incrementa la resistència es redueix la intensitat.



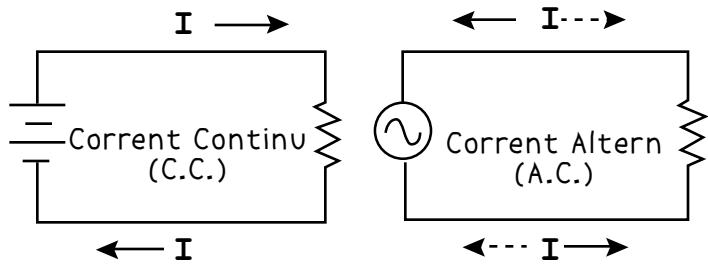
O si s'incrementa la tensió hi ha més corrent.



Anem a veure ara un circuit senzill. Cada circuit és un recorregut tancat que té una font d'energia (pila) i una càrrega (làmpada). La càrrega converteix l'energia elèctrica en una altra d'útil. Aquest circuit també té un interruptor



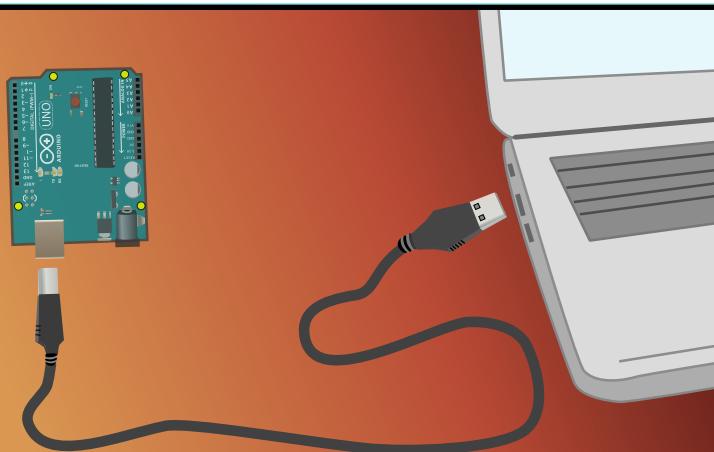
Aquest és l'esquema del mateix circuit (representa el circuit mitjançant símbols pels components electrònics). Quan l'interruptor és tancat, el corrent flueix des del generador i fa encendre la làmpada



Hi ha dos tipus habituals de circuits, els de Corrent Continu i els de Corrent Alternant. En un circuit de C.C. el corrent sempre flueix en el mateix sentit. En C.A. el corrent flueix en sentits opositos en cicles regulars. Aquí només parlarem de circuits de C.C.



L'Arduino necessita energia per funcionar. L'hem de connectar a un ordinador per programar-lo.



Al connectar l'Arduino a l'ordinador amb un cable USB, aquest proporcionarà els 5V que necessita i podrem programar-lo.



S'ha de descarregar i instal·lar programari per poder programar l'Arduino. Es troba disponible en l'URL de dalt gratuïtament. El programari Arduino és executable en plataformes Mac OS X, Windows i Linux.

Instruccions per instal·lar el programari

Arduino en un Mac:

<http://www.arduino.cc/en/Guide/MacOSX>

Instruccions per instal·lar el programari

Arduino en Windows:

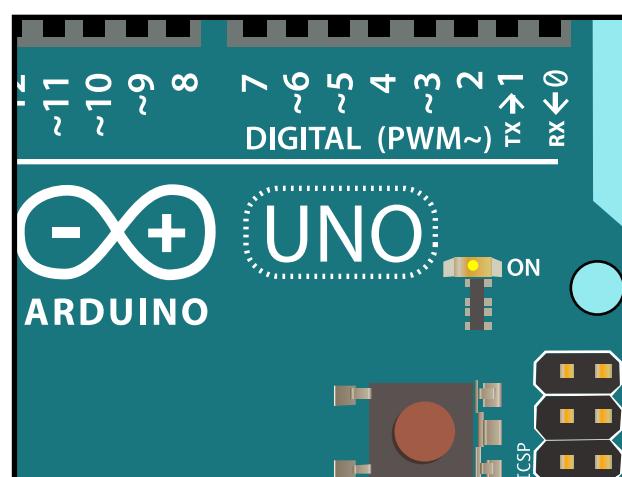
<http://www.arduino.cc/en/Guide/Windows>

Instruccions per instal·lar el programari

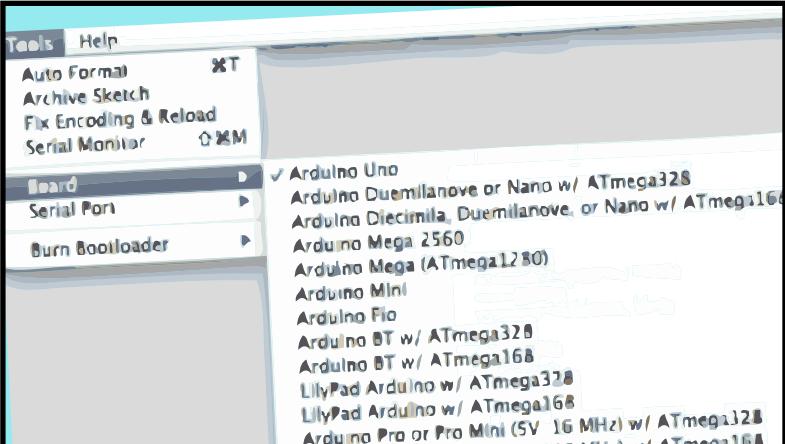
Arduino en Linux:

<http://www.arduino.cc/playground/Learning/Linux>

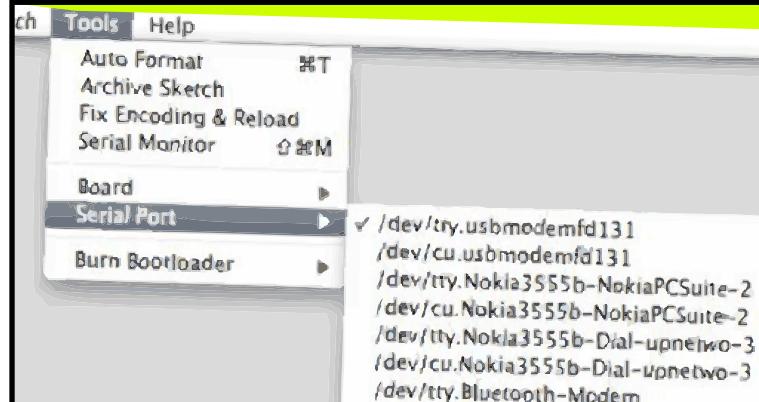
A les URL de dalt hi ha instruccions detallades per instal·lar el programari en aquestes plataformes.



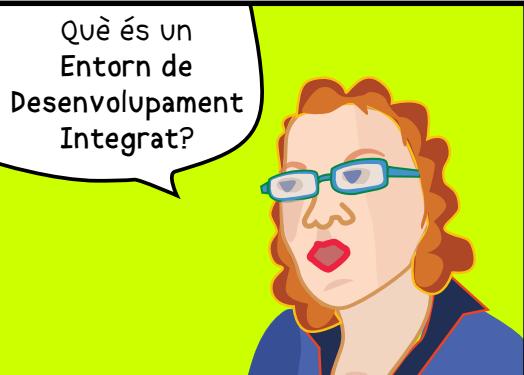
Després d'haver instal·lat el programari connecteu l'Arduino. Un LED marcat ON s'hauria d'encendre a la placa.



Executeu el programari Arduino. En el menú Tools escolliu la placa que estigueu utilitzant (Tools > Board). Per exemple, Arduino UNO.

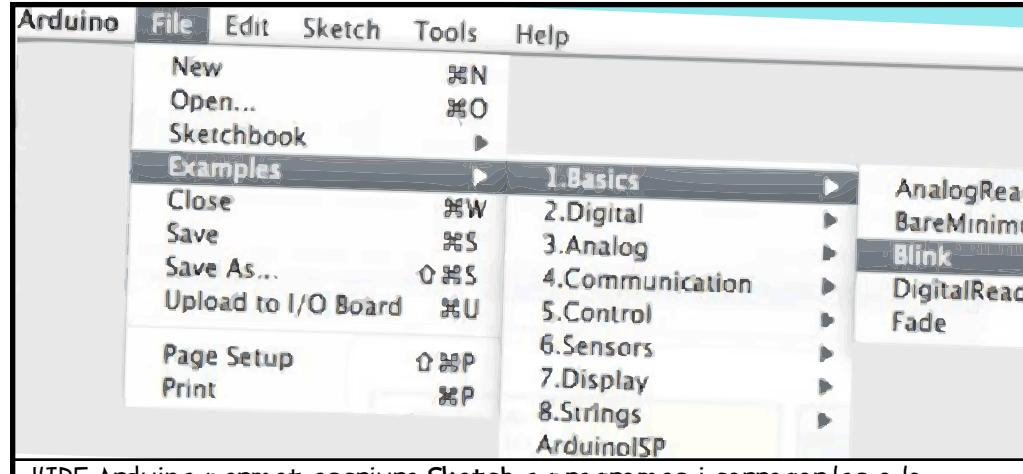


A continuació seleccioneu el port sèrie (Tools > Serial Port). En un Mac serà quelcom semblant a /dev/tty.usbmodem. En un ordinador amb Windows serà com3 o quelcom similar.



Quan heu descarregat el programari Arduino heu descarregat un **IDE**.

Combina un editor de text amb un compilador i altres funcions per ajudar als programadors a desenvolupar programari.



L'IDE Arduino permet escriure Sketch o programes i carregar-los a la placa Arduino. Obriu l'exemple Blink en el menú File:
File > Exemples > 1.Basics > Blink.

```

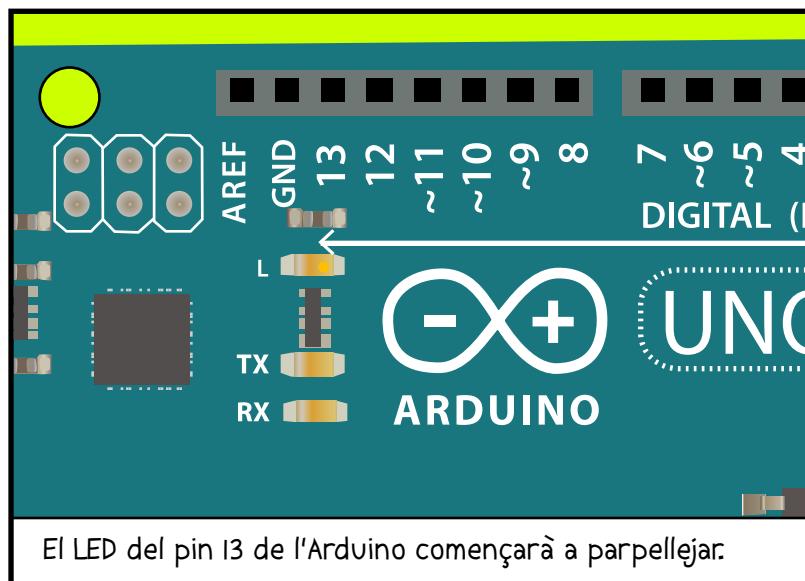
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
}

```

Per carregar l'sketch a la placa Arduino cliqueu el botó Upload en la tira de botons de dalt de la finestra. Apareixeran alguns missatges a la part de baix de la finestra i finalment Done Uploading.



```

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has 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
}

```

Un sketch, com qualsevol programa escrit en un altre llenguatge és un conjuts d'instruccions per l'ordinador. Si observem detingudament l'exemple Blink veurem que hi ha dues parts principals: **setup** i **loop**.

<http://arduino.cc/es/reference/homepage>



Al web d'Arduino podeu consultar la guia de referència i molts altres recursos per aprendre el llenguatge.

setup: s'executa una vegada quan el programa s'inicia

loop: es repeteix una i altra vegada

Aquests dos són blocs de codi anomenats **funcions** que tindran tots els sketch. Estan tancat per claudàtors { }.

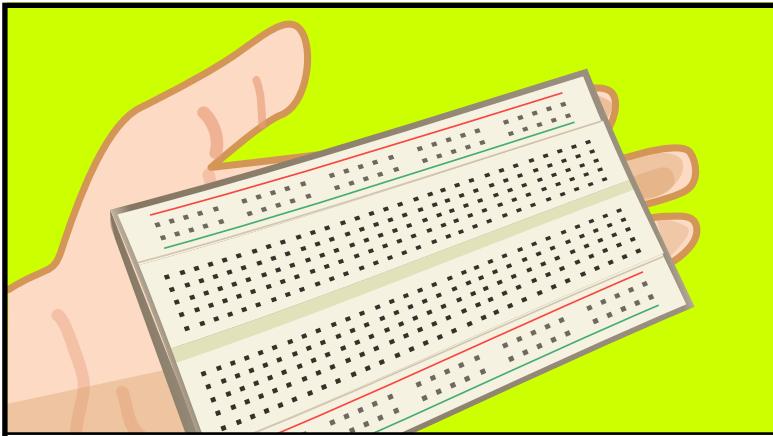
```

void setup() {           // declara un bloc de codi
  pinMode(13, OUTPUT); // posa el píu 13 com sortida
}                      // finalitza el bloc de codi

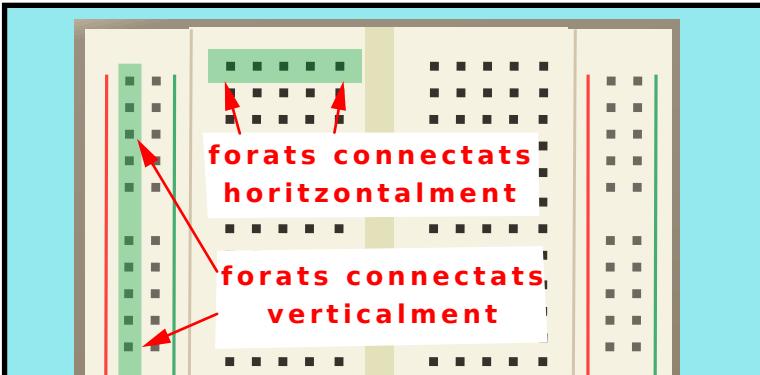
void loop() {            // declara un bloc de codi
  digitalWrite(13, HIGH); // posa píu 13 a nivell alt
  delay(1000);           // paua 1 segon
  digitalWrite(13, LOW);  //posa píu 13 a nivell baix
  delay(1000);           //paua 1 segon
}                      //finalitza el bloc de codi

```

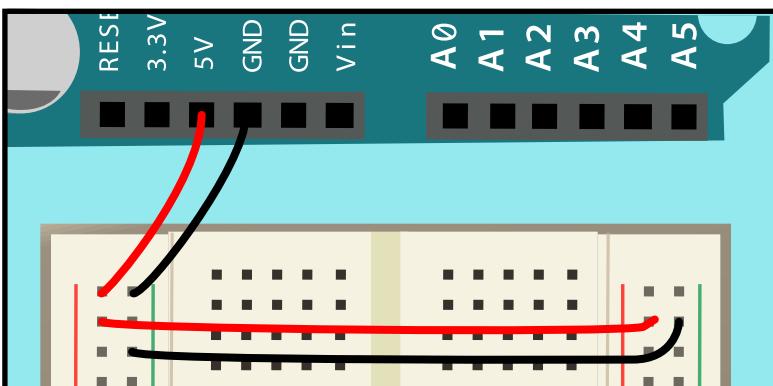
Ara mireu aquest script senzill línia a línia per veure que fa cadascuna.



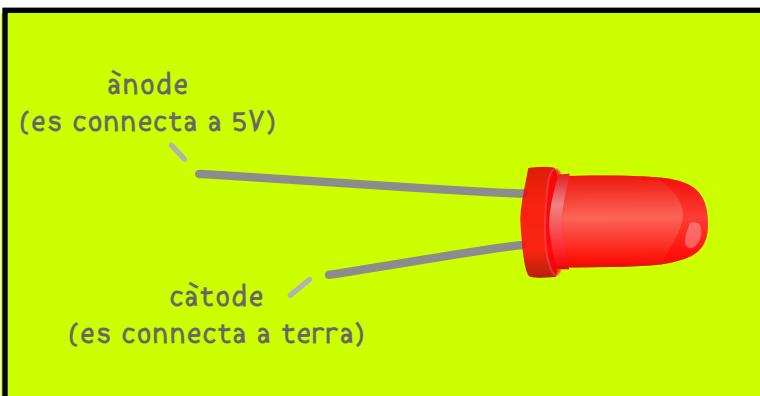
Com es controlen objectes que no estan en la placa Arduino? Connectarem l'Arduino a una **placa de prototips sense soldadura**. Això ens permetrà construir i comprovar circuits ràpidament.



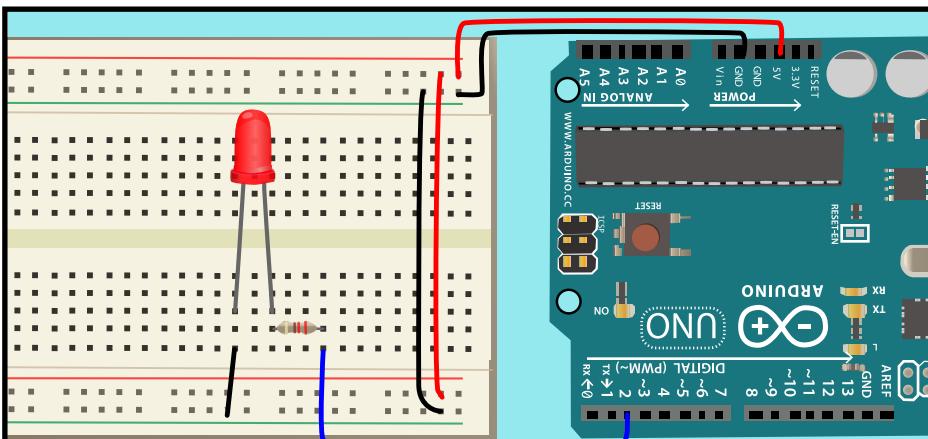
Aquesta placa de prototips té 2 files de forats verticals a esquerra i dreta, i files de 5 forats horitzontals als costats de la ranura central. Les tires dels costats estan connectades verticalment, i cada tira de 5 forats del mig és independent de les altres.



Connectarem els 5V i la terra de la placa Arduino a les tires connectades verticalment a esquerra i dreta amb cables primis. Els components es poden interconnectar amb els forats del mig i a 5V i terra si és necessari.



Quan circula corrent travessant un LED (Díode Emissor de Llum) en la direcció correcta aquest s'encén. Anem a connectar un LED a la placa de prototips i llavors a l'Arduino per tal de poder controlar-lo amb codi.

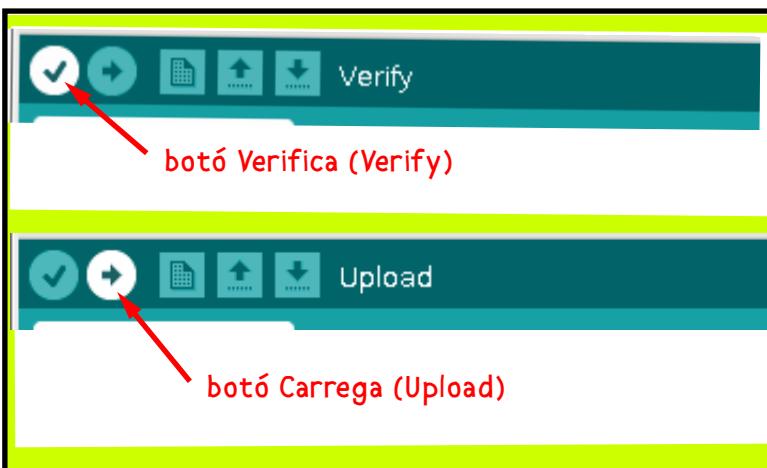


L'ànode està connectat al piu 2 de l'Arduino mitjançant un resistor de 220 ohms. El càtode és connectat a terra. Del piu 2 fins el I3 es poden configurar com entrades o sortides digitals. Clica el botó New per començar un sketch.

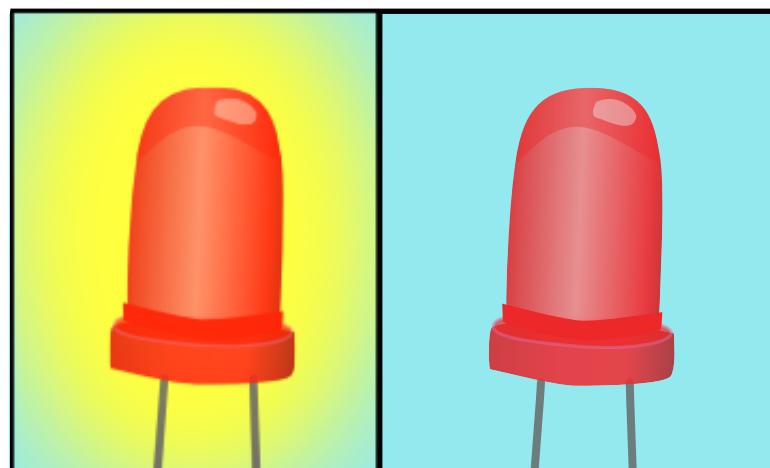
```
voi d setup() {
    pi nMode(2, OUTPUT);
}
```

```
voi d loop() {
    di git al Wite(2, HI GH);
    del ay(500);
    di git al Wite(2, LOW);
    del ay(500);
}
```

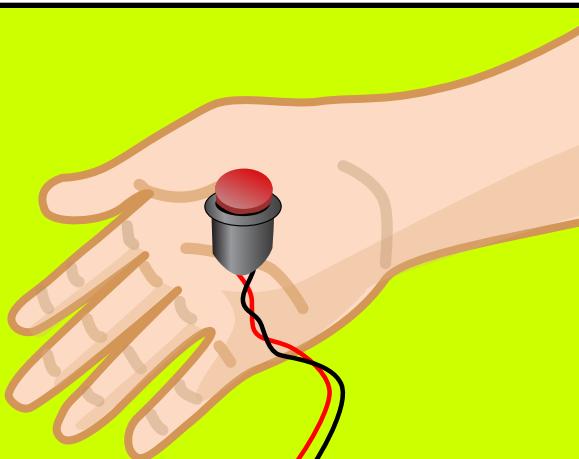
A setup hem configurat el piu 2 com a sortida. A loop, primer hem posat el piu 2 a nivell alt. Delay atura 500 milisegons -o mig segon-. Quan el piu 2 es posa a nivell baix el LED s'apaga, llavors hi ha una altra pausa de mig segon.



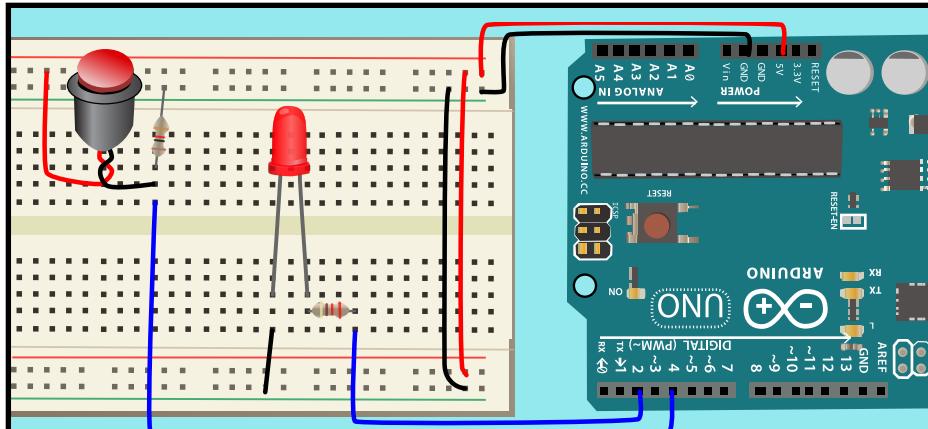
Clica Verifica al menú per comprovar el codi. Si no hi ha cap errada, clica Carrega per enviar el programa a l'Arduino.



El LED s'encén durant mig segon, després s'apaga durant un altre mig segon, i així una i altra vegada.



Ara afegirem un interruptor, una entrada digital, per poder encendre i apagar el LED.



Connecta un extrem del pulsador al piu 4 de l'Arduino, i també un extrem d'un resistor de 10k connectat per l'altre extrem a terra. Connecta l'altre extrem als 5V. Deixa el LED connectat al mateix piu que abans.

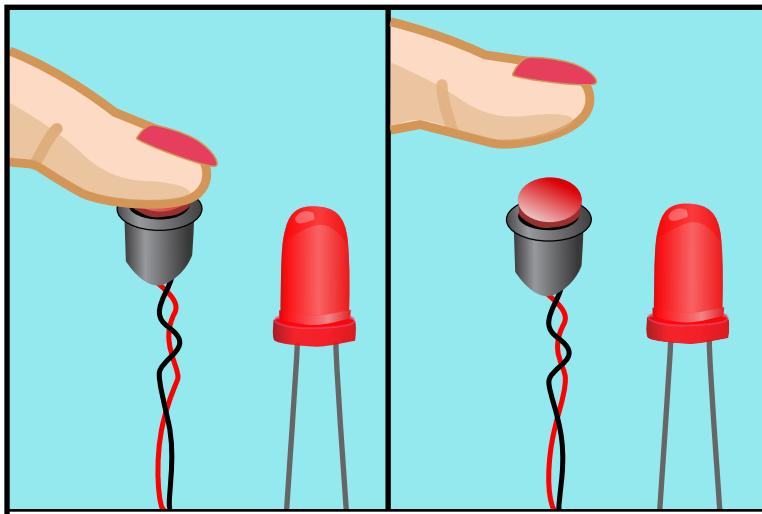
```

void setup() {
    pinMode(2, OUTPUT);
    pinMode(4, INPUT);
}

void loop() {
    if(digitalRead(4)) {
        digitalWrite(2, HIGH);
    } else {
        digitalWrite(2, LOW);
    }
}

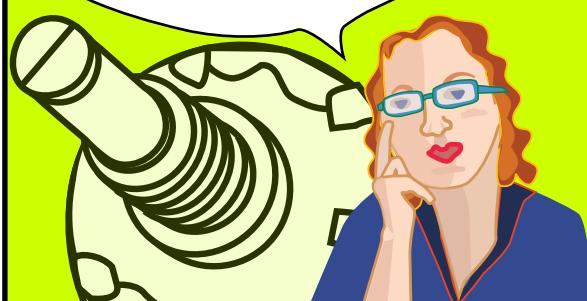
```

A continuació escriurem el codi. A setup, declarem el piu 2 com sortida i el piu 4 com entrada. A loop, utilitzarem una sentència if: si al comprovar el piu 4 està a nivell alt, posarem el piu 2 a nivell alt, en cas contrari posarem el piu a nivell baix tot apagant-lo.

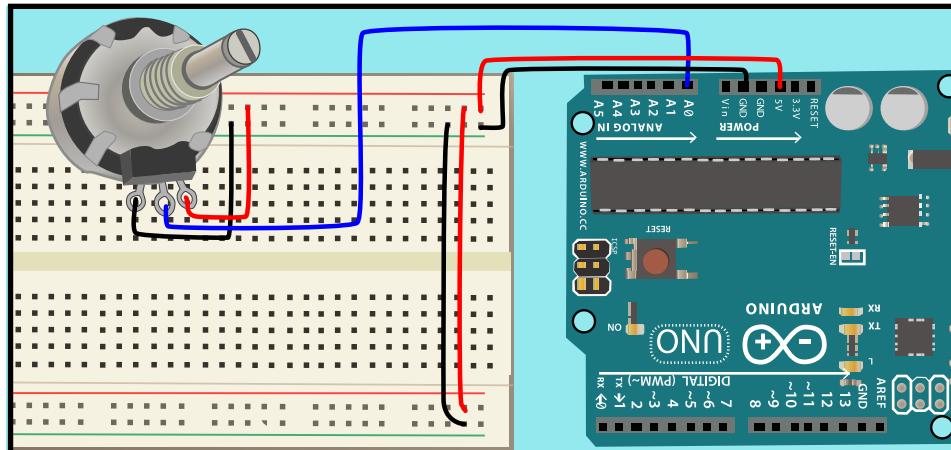


El LED s'encén quan el pulsador es manté premut.

un potenciòmetre és un resistor variable. La seva resistència canvia quan es fa girar, tot augmentant o disminuint segons el sentit en que es fa girar.



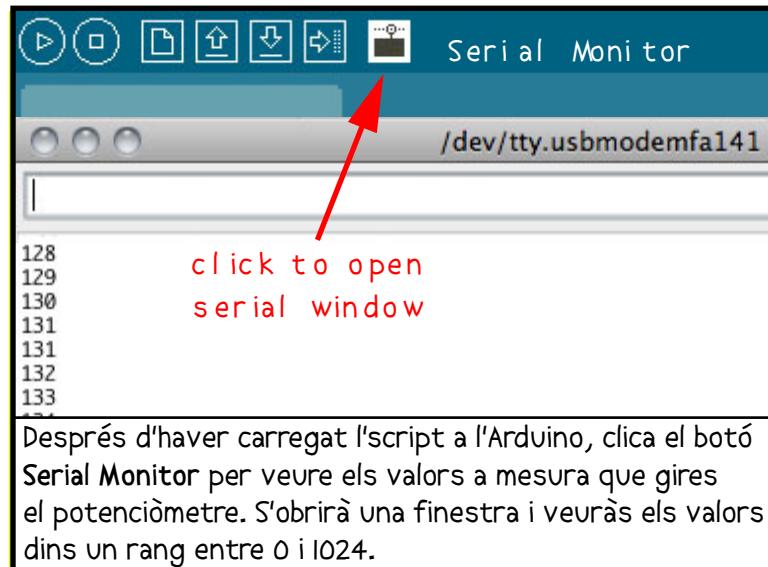
Ara establirem una entrada analògica. Utilitzarem un potenciòmetre.



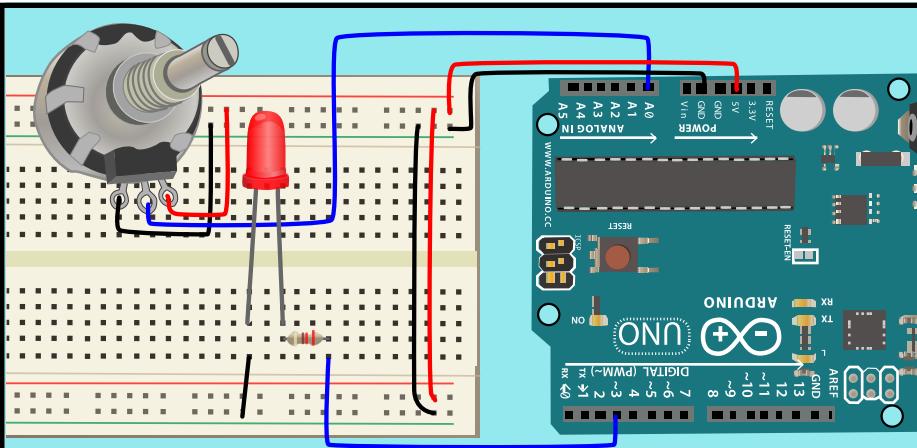
Connecta el terminal del mig del potenciòmetre al piu analògic A0. Connexта un extrem del potenciòmetre a 5V i l'altre a terra.

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println(analogRead(A0));  
}
```

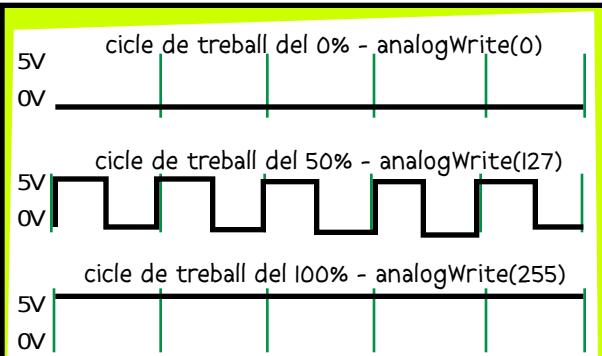
Primer mirarem el rang de valors que obtenim al girar el potenciòmetre utilitzant el **Serial Monitor**. En el codi inicialitzarem l'objecte serial al **setup**, configurant una velocitat de 9600 bauds. A **loop** llegirem el valor del piu analògic A0 i l'imprimirem a l'objecte serial amb la funció **println**.



Després d'haver carregat l'script a l'Arduino, clica el botó **Serial Monitor** per veure els valors a mesura que gires el potenciòmetre. S'obrirà una finestra i veuràs els valors dins un rang entre 0 i 1024.



Ara utilitzarem els valors variables que es reben del potenciómetre com atenuador per controlar un LED. Connecta l'ànode mitjançant un resistor al píu 3 de la placa i el càtode a terra.



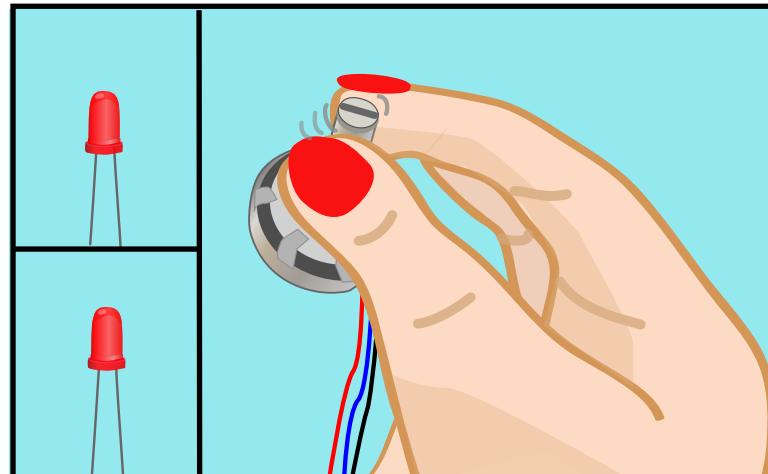
Utilitzarem la modulació per amplada de pulsos (PWM). És una manera de simular un valor analògic tot alterant la tensió entre el màxim i el mínim a diferent ritme o cicle de treball. Es pot utilitzar la PWM en els píus 3, 5, 6, 9, 10 i 11.

```
int sensorValue = 0;

void setup() {
  pinMode(3, OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

Primer crearem una variable per emmagatzemar el valor del potenciómetre. A `setup` definim el píu 3 com sortida. A `loop` emmagatzemem el valor que s'ha llegit del pin A0 a la variable. Llavors escrivim el valor al píu 3, el del LED. Hem de dividir la variable entre 4 per tal de tenir un rang de valors entre 0 i 255 que és un byte.



La lluentor del LED canvia, entre completament apagat i molt luent al girar el potenciómetre.



Guies

<http://arduino.cc/es/Tutorial/HomePage>

Libres

<http://arduino.cc/playground/Es/Manuales>

Projectes

<http://arduino.cc/playground/Es/Projects>

Enllaços

Programari

Descàrrega de programari

<http://arduino.cc/es/Main/Software>

Referència del llenguatge

<http://arduino.cc/es/Reference/HomePage>

On comprar

<http://arduino.cc/es/Main/Buy>

<http://store.arduino.cc/eu/index.php>

Text i dibuixos de **Jody Culkin**
vegeu-ne més a jodyculkin.com
Traduït al català, adaptat i remaquetat per Antoni Ubieto Xirgas

Agraïments especials a Tom Igoe, Marianne Petit, Calvin Reid, al professorat i personal del Interactive Telecommunications Program de la NYU, particularment Dan O'Sullivan, Danny Rozin i Red Burns. Gràcies a Cindy Karasek, Chris Stein, Sarah Teitler, Kathy Goncharov i Zannah Marsh.

Moltes, moltes gràcies al Arduino Team per oferir-nos aquesta plataforma de codi obert robusta i flexible.

I gràcies a la animada, activa i sempre creixent comunitat Arduino.

Introduction to Arduino by Jody Culkin
està subjecta a una llicència Creative Commons
Attribution-NonCommercial-ShareAlike 3.0
Unported License.



La traducció i adaptació d'Antoni Ubieto Xirgas
és per tant també subjecta a la mateixa llicència