



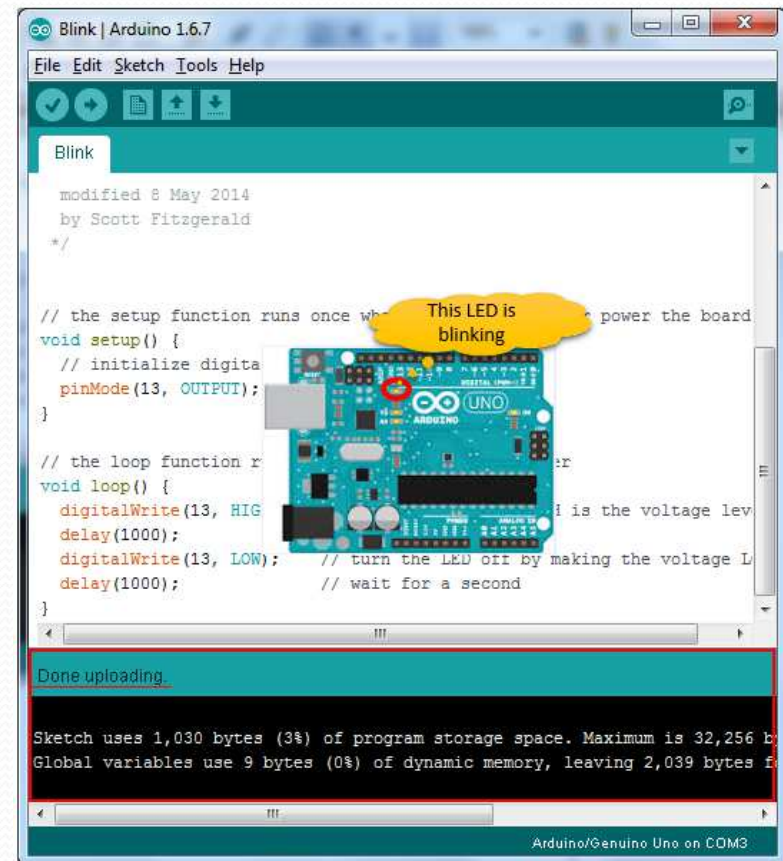
**FEDERACIÓ  
CATALANA  
D'AMICS DEL  
FERROCARRIL**

# ARDUINO I DCC

QUE ES POT FER AMB ARDUINO PER LA MAQUETA DIGITAL DCC

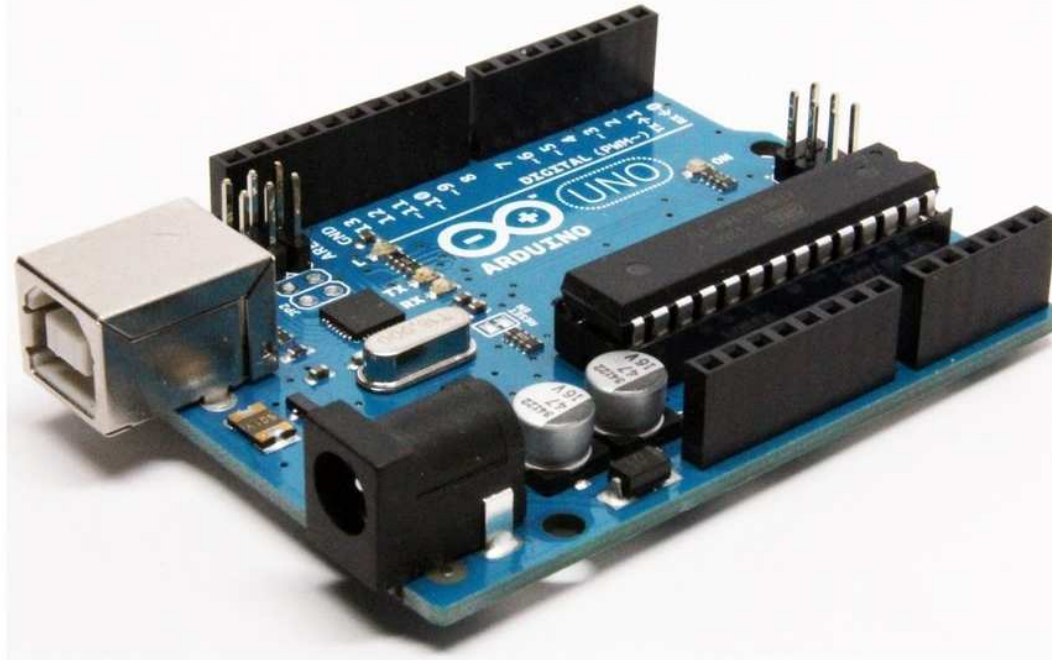
# QUÈ ÉS ARDUINO?

- Arduino és una plataforma electrònica de codi obert basada en hardware i programari de fàcil ús. Està pensat per a qualsevol persona que faci projectes interactius.
- El programari (Arduino IDE) fa que sigui fàcil d'escriure codi (llenguatge C) i pujar-lo a la placa Arduino.
- N'hi ha diferents tipus de plaques, bàsicament són un microprocessador amb diferents tipus d'entrades i sortides, amb connexió USB i d'alimentació.



# LA PLACA ARDUINO UNO

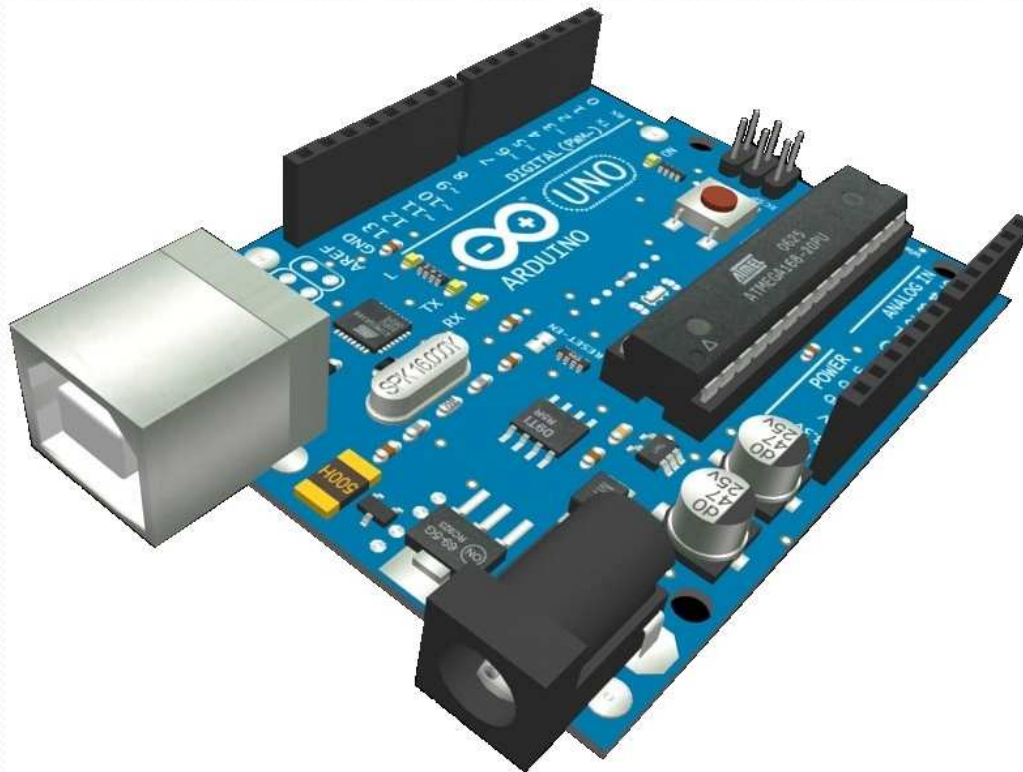
- La placa Arduino UNO és la més popular i utilitzada de tota la família Arduino





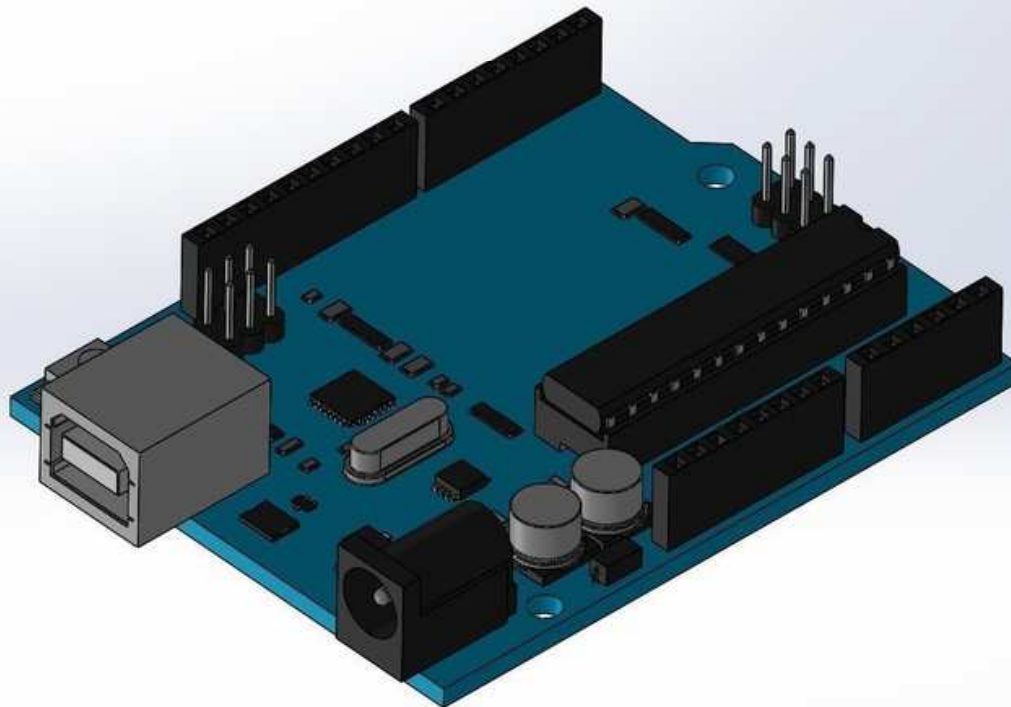
# LA PLACA ARDUINO UNO

- N'hi ha diferent versions de Arduino UNO. Amb diferent tamany però basicament igual n'hi la Arduino Nano.



# LA PLACA ARDUINO UNO

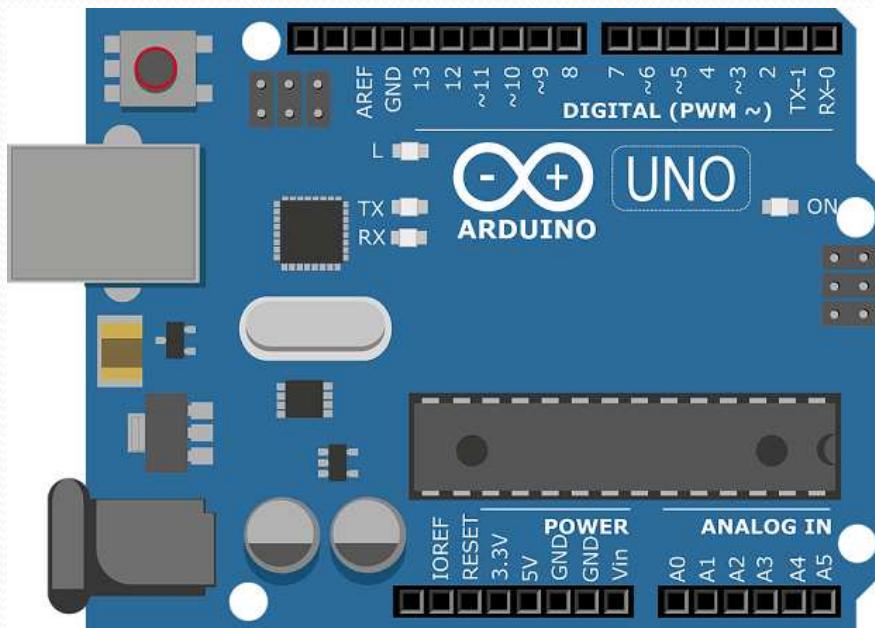
- Amb diferents microprocessador i nombre d'entrades i sortides n'hi ha altres plaques Arduino (MEGA, DUE, Leonardo,...) però totes es programen igual des de el programari Arduino IDE.





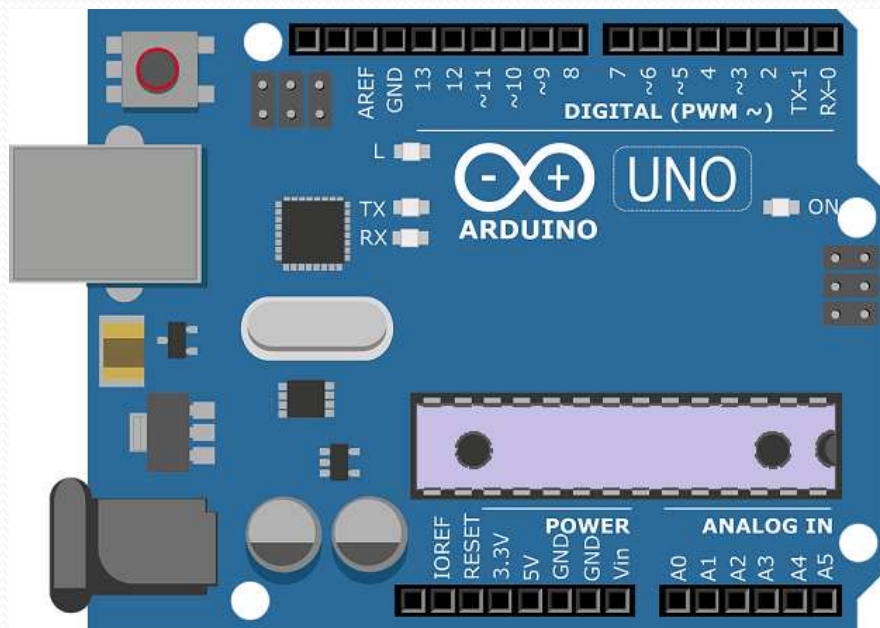
# LA PLACA ARDUINO UNO

- Podeu dir al vostre Arduino del que cal fer escrivint codi al llenguatge de programació Arduino o copiant el codi ja fet per altres i descarregant-lo a la placa desde el Arduino IDE



# CONEIXENT LA PLACA ARDUINO UNO

- Arduino Uno és una placa amb un microcontrolador basat en ATmega328P

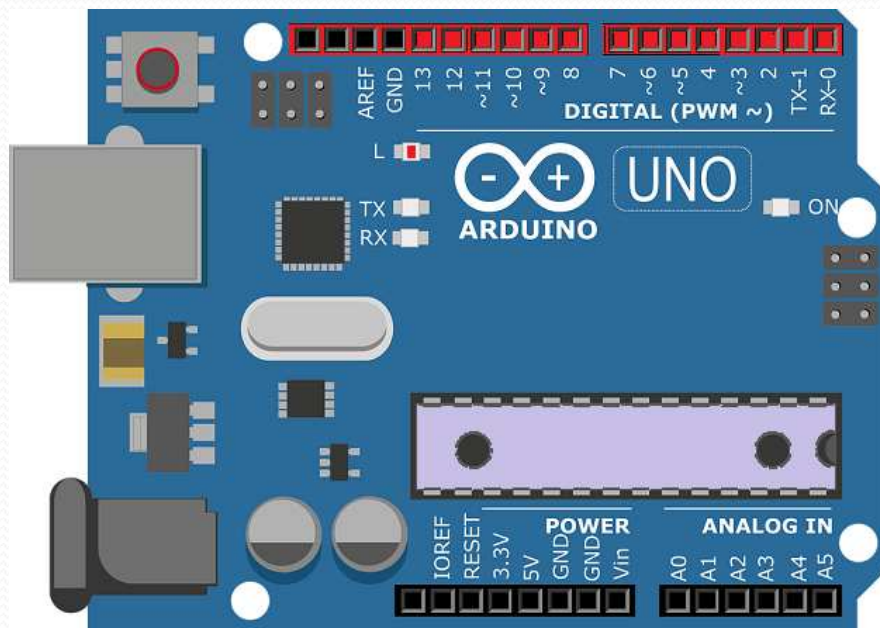


**Microprocessador**



# CONEIXENT LA PLACA ARDUINO UNO

- Té 14 pins digitals d'entrada / sortida (dels quals 6 es poden utilitzar com a sortides PWM)



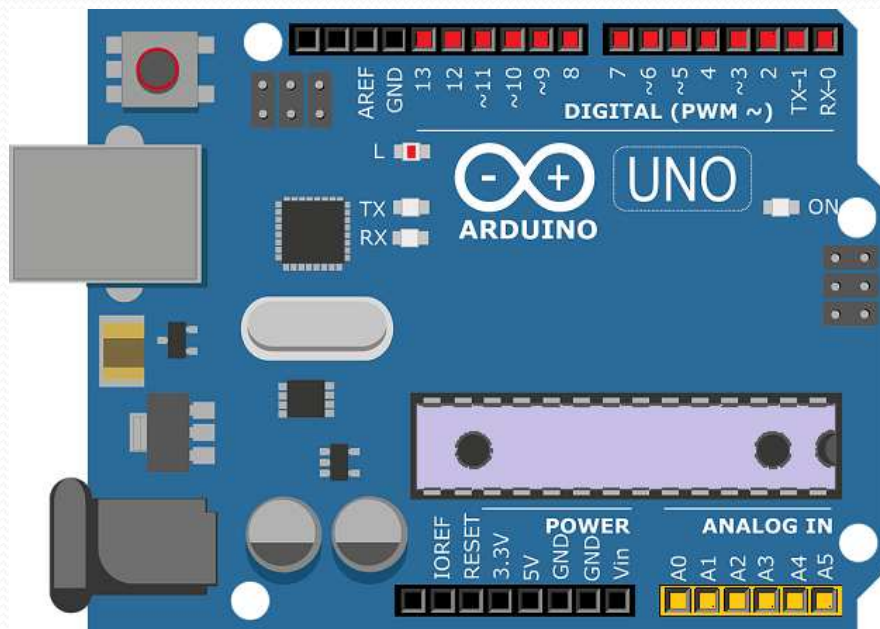
**Entrades/Sortides  
Digitals**

**Microprocessador**



# CONEIXENT LA PLACA ARDUINO UNO

- Fins a 6 entrades analògiques. Poden ser usats com pins digitals extra.



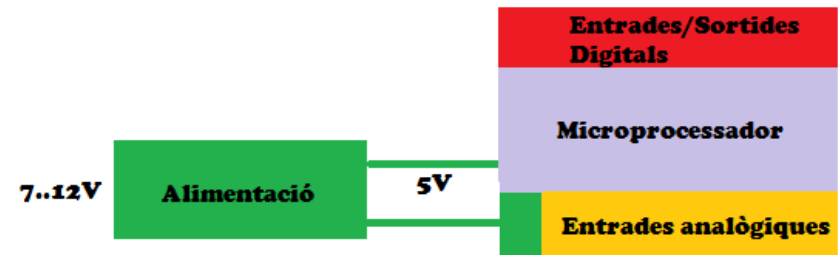
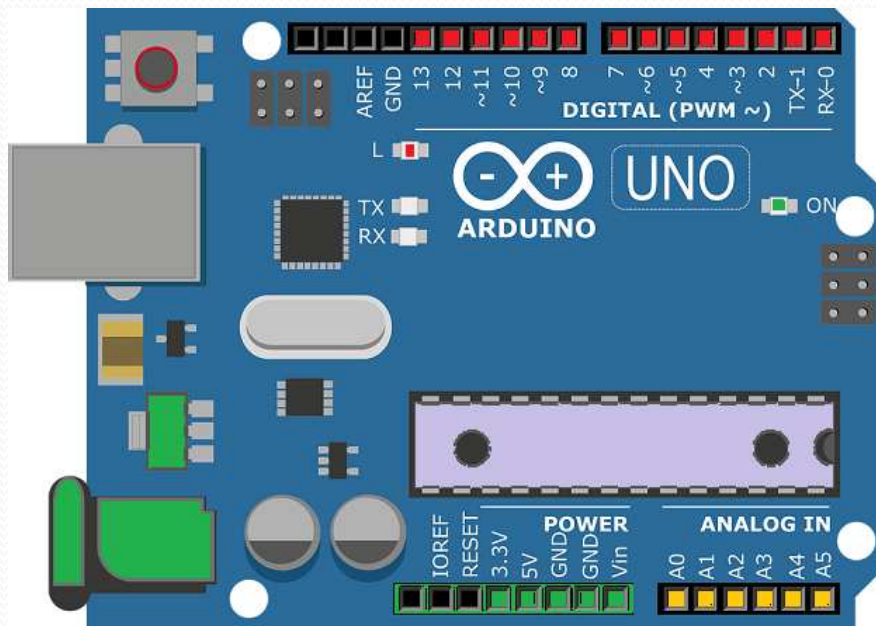
**Entrades/Sortides  
Digitals**

**Microprocessador**

**Entrades analògiques**

# CONEIXENT LA PLACA ARDUINO UNO

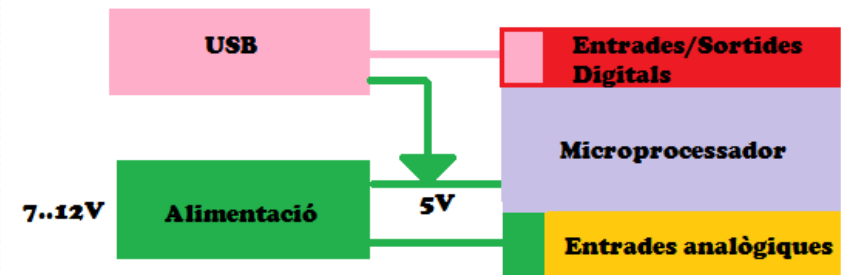
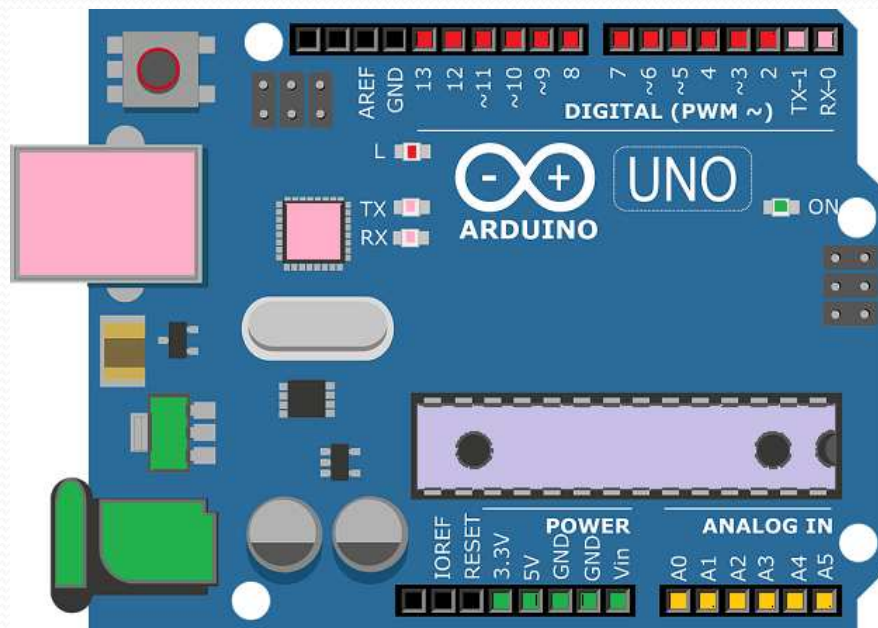
- Es pot alimentar externament amb tensió contínua entre 7V y 12V. El microprocessador treballa a 5V i al connector hi ha disponibles les tensions d'alimentació, els 5V i 3.3V, aquesta ultima només dona 50mA





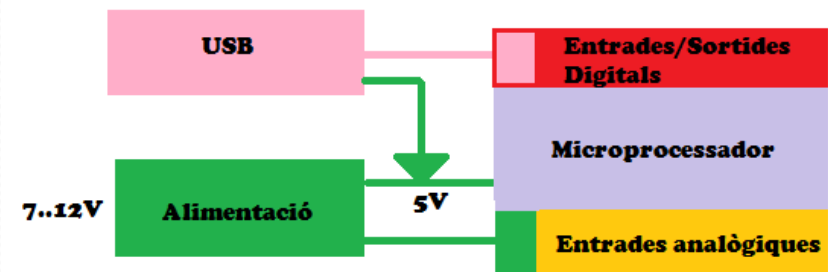
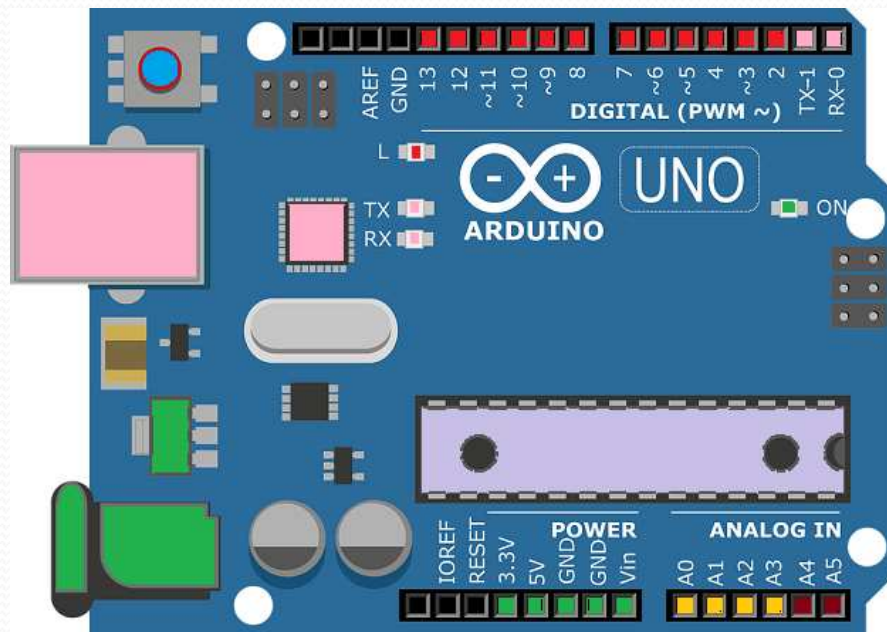
# CONEIXENT LA PLACA ARDUINO UNO

- Té una connexió USB per poder carregar el programari des de l'Arduino IDE i comunicar-se amb ell amb un port sèrie, fa servir dos pins digitals. També pot ser alimentat només des de l'USB si el consum es petit.



# CONEIXENT LA PLACA ARDUINO UNO

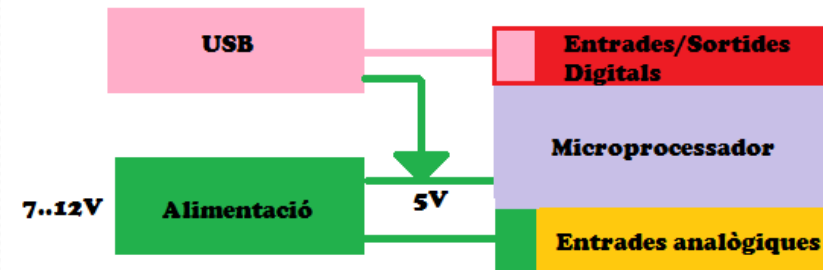
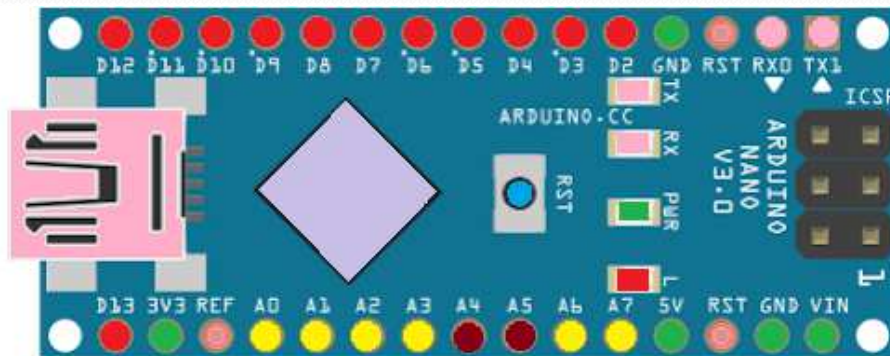
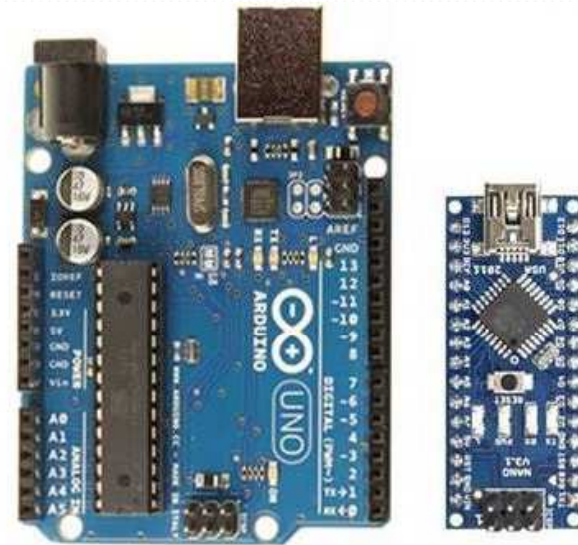
- Té una sèrie de LEDs indicadors: Alimentació, port sèrie i pin13.
- Botó de Reset
- Dos dels pins (A4, A5) poden funcionar com a un bus I2C





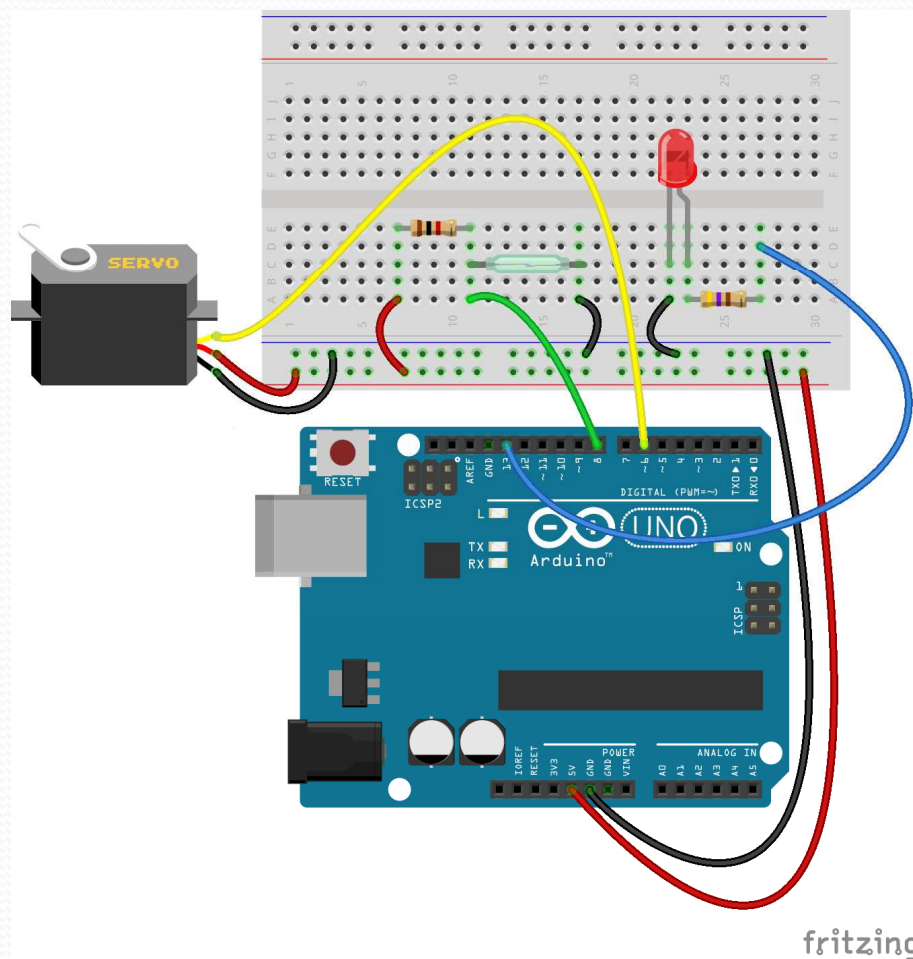
# CONEIXENT LA PLACA ARDUINO NANO

- El Arduino Nano, de mida molt més petita que el Arduino UNO, te les mateixes prestacions ja que fa servir el mateix processador. Han reduït connectors i mida de la placa.

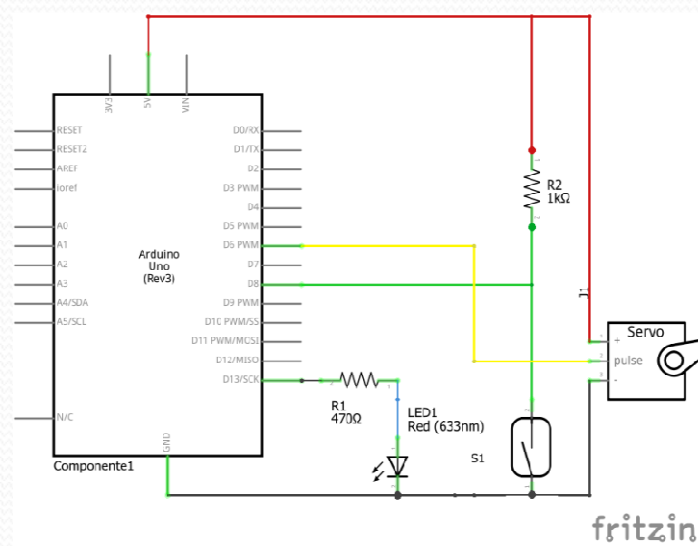


# COM CONECTAR ELS ELEMENTS

Per probes es pot fer servir un taulell on posar els elements que no necessita soldadura per fer les connexions, només cables.



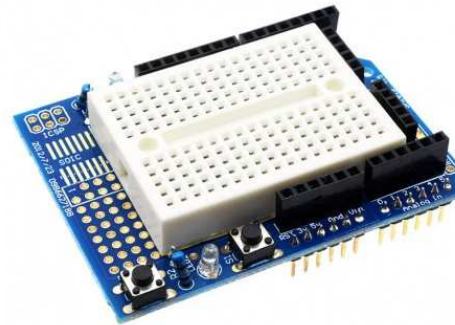
Els LED necessiten la seva resistència, el pin dona 5V i 20mA màxim  
Els REED necessiten una resistència de pull-up  
Els SERVOS normalment es connecten als pin PWM





# LES PLAQUES D'EXTENSIÓ O 'SHIELD'

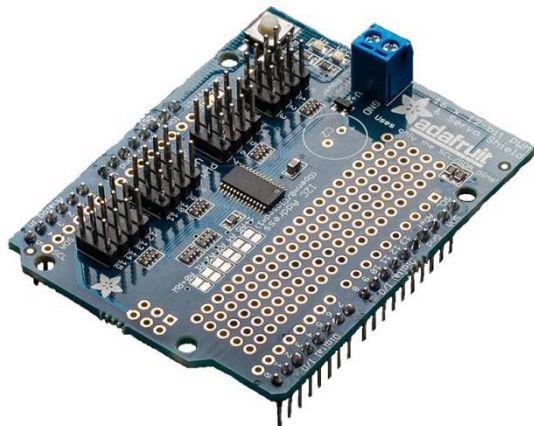
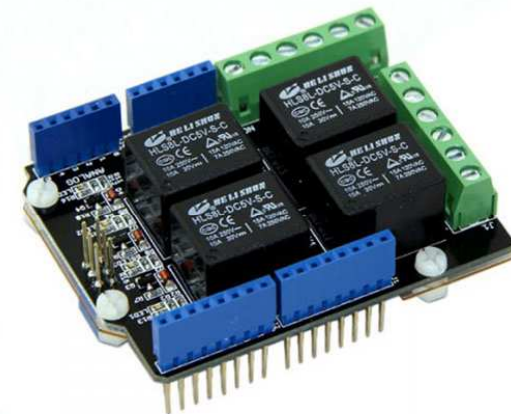
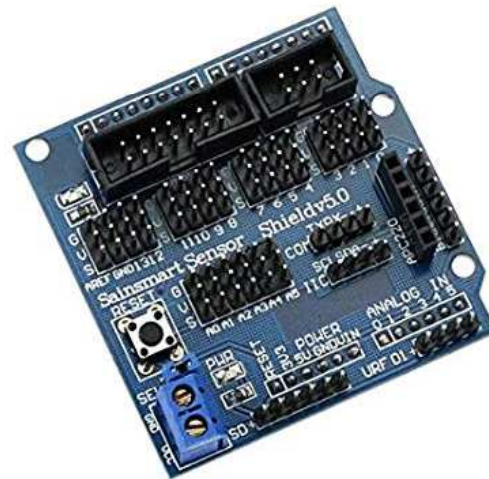
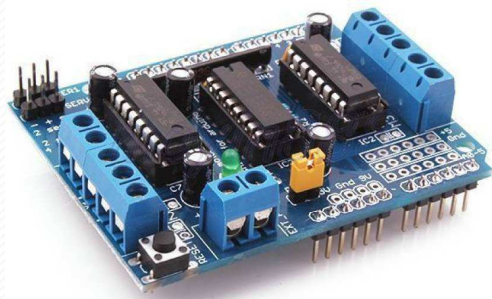
Un 'shield' en Arduino és una placa que s'apila sobre l'Arduino o sobre un altre 'shield', de manera que ens permet ampliar el hardware o per donar funcionalitat extra a un Arduino.



N'hi ha plaques 'shield' per prototips amb soldadura o amb taulell

# LES PLAQUES D'EXTENSIÓ O 'SHIELD'

Altres 'shield' interessants per a les maquetes inclouen control de motors de corrent continua, motors pas a pas, relés, sensors, servos,...

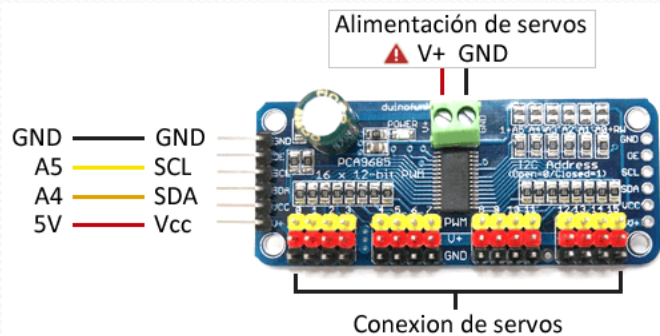
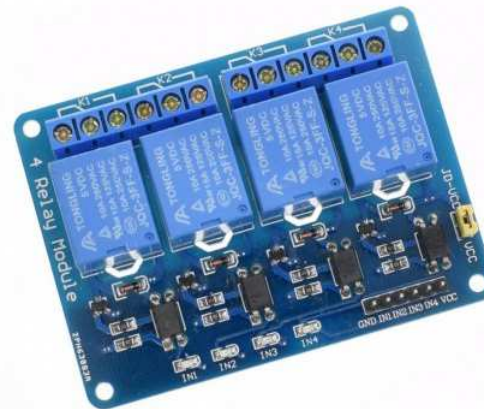
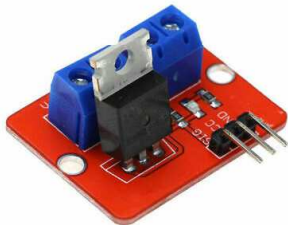




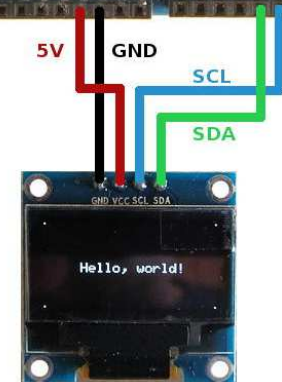
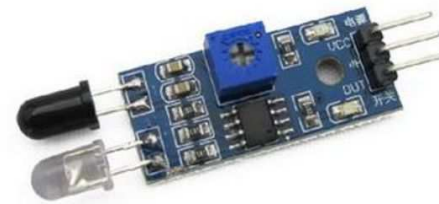
# ALTRES TIPUS DE PLAQUES D'EXTENSIÓ

Altres plaques d'extensió no es punxen a l'Arduino però també poden ser controlats des de els pins analògics i digitals o pel bus I2C.

N'hi ha de relés, servos, barreres de infraroig, pantalles OLED, transistors de potencia, ...



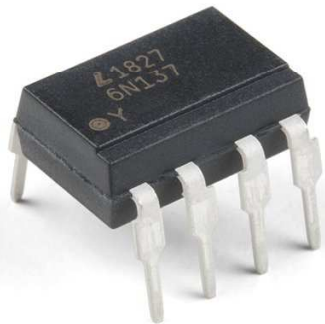
⚠ V+ entre 5 y 6V. Al usar alimentación externa SIEMPRE poner con GND común.



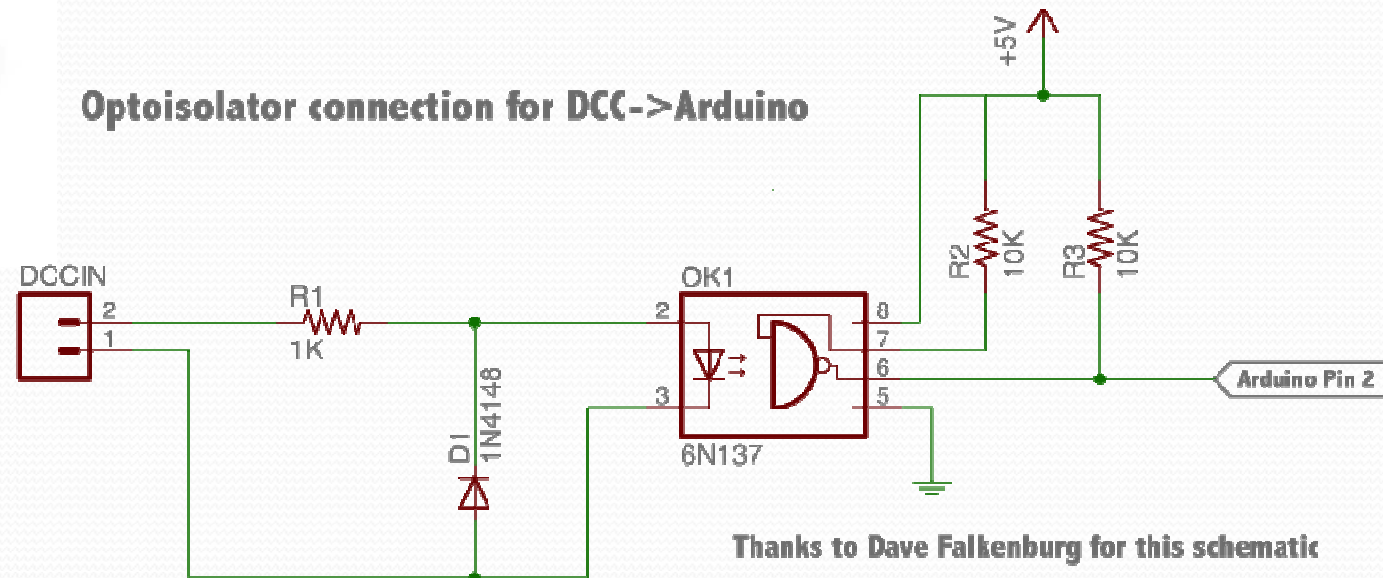
# CONEXIÓ ARDUINO A DCC

Per fer un descodificador d'accessoris amb Arduino necessitem d'un circuit per poder llegir la senyal DCC i que a la vegada ens aïlli de les tensions altes que tenim a la via perjudicials per l'Arduino, recordem que treballa a 5V.

Això s'aconsegueix mitjançant un optoacoblador, el 6N137 que es lo suficient ràpid per llegir el senyal DCC i donar-li a l'Arduino pel pin 2



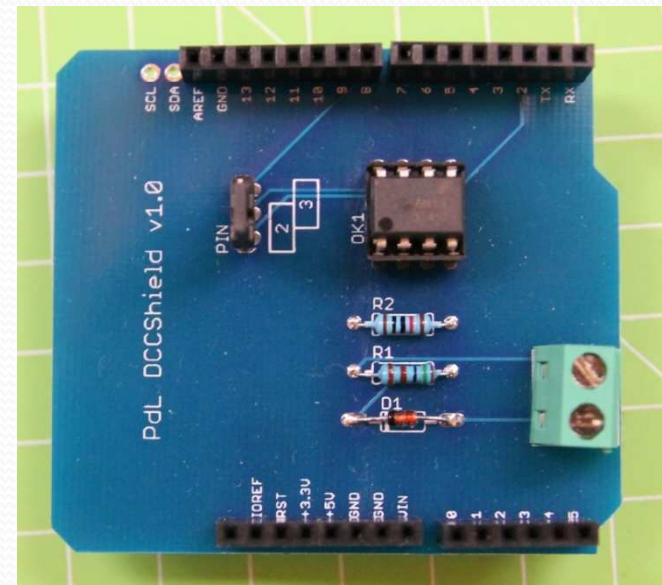
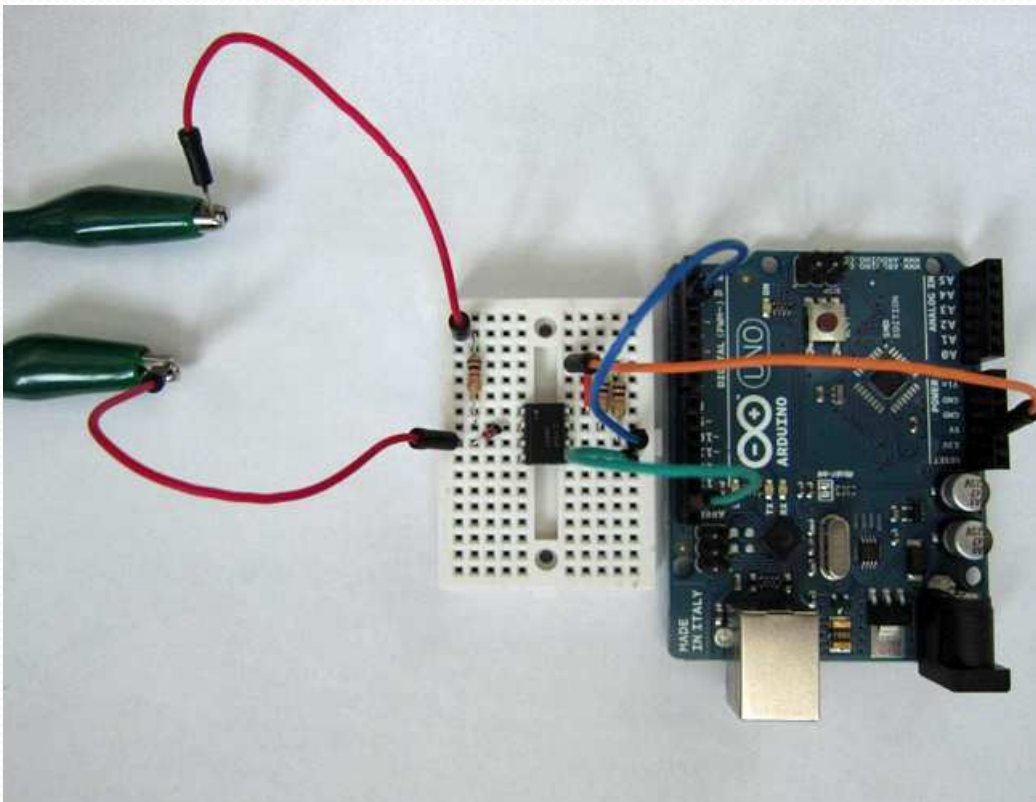
**Optoisolator connection for DCC->Arduino**





# CONEXIÓ ARDUINO A DCC

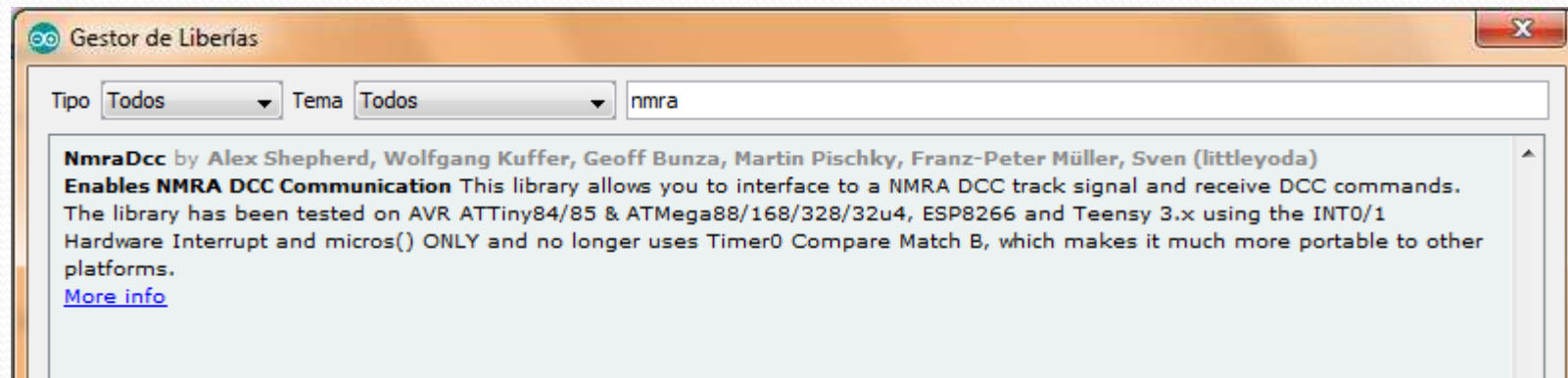
Malauradament no hi ha una 'shield' comercial així que l'haurem de muntar nosaltres amb una placa prototipus o fer-la fer des de un disseny d'una web



<https://github.com/lucadentella/arduino-dccshield>

# CONEXIÓ ARDUINO A DCC

La senyal DCC arribarà al pin 2 de l'Arduino però per fer-la servir al nostre programa caldrà descodificar-la. Per sort la comunitat Arduino ha posat a disposició una llibreria per l'Arduino IDE per poder rebre i entendre el senyal DCC .



Des de el mateix Arduino IDE es pot instal·lar. Amb la llibreria, la feina de fer per exemple, un descodificador d'accessoris DCC es bastant simple.



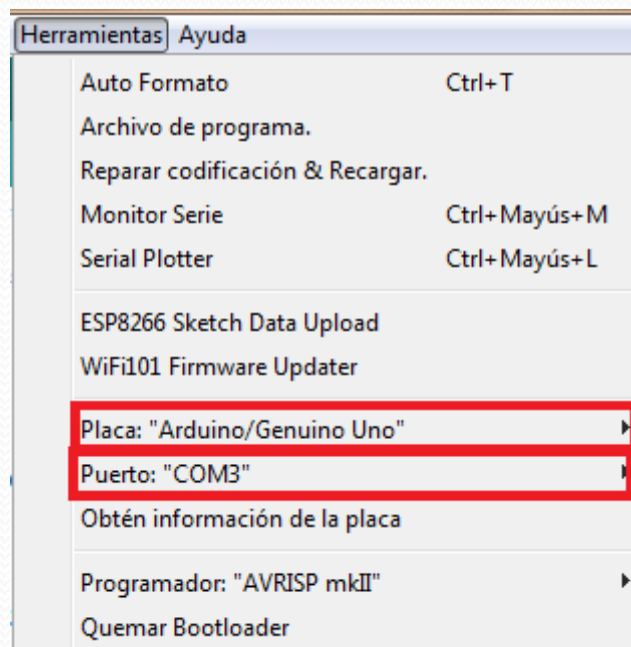
# CONEXIÓ ARDUINO A DCC

```
Archivo Editar Programa Herramientas Ayuda
AccDecDemo
1 // Demostració de la llibreria NmraDcc com descodificador d'accessoris -- Paco Cañada 2019
2
3 #include <NmraDcc.h>           // Llibreria DCC
4
5 NmraDcc Dcc ;                 // Crea el objecte DCC
6                               // Definim unes constants que pot canviar l'usuari
7 const byte DCC_PIN = 2;       // DCC pin
8 const byte LED_PIN = 13;      // LED pin
9 const int DCC_ADDRESS = 6;    // Adreça accessori
10
11 void setup() {                // Inicialització
12     pinMode (LED_PIN, OUTPUT ); // LED apagat per defecte
13     digitalWrite(LED_PIN, LOW);
14                               // Configurem pin i tipus de descodificador
15     Dcc.pin(digitalPinToInterrupt(DCC_PIN), DCC_PIN, 1);
16     Dcc.initAccessoryDecoder( MAN_ID_DIY, 1, FLAGS_OUTPUT_ADDRESS_MODE, 0 );
17 }
18
19 void loop() {                  // Bucle principal
20     Dcc.process();             // Processa la descodificació de la senyal DCC
21 }
22
23 // Quan arriba un paquet DCC per accessoris s'executa aquesta rutina
24 void notifyDccAccTurnoutOutput( uint16_t Addr, uint8_t Direction, uint8_t OutputPower ) {
25     if(( Addr == DCC_ADDRESS ) && OutputPower ) { // Si es la nostra adreça i esta activa
26         if (Direction == 0)                       // Segons sigui recte/desviat
27             digitalWrite(LED_PIN, LOW);           // apaguem LED
28         else                                       // o
29             digitalWrite(LED_PIN, HIGH);          // encenem LED
30     }
31 }
32
```

Aquest petit programa encén i apaga el LED de la placa Arduino (connectat al pin 13) quan rep la ordre corresponent per a l'activació de l'accessori 6 en posició recte/desviat.

Si volem que ho faci per un altre accessori només cal canviar a la línia 9 la adreça 6 pel accessori que escollim i tornar a pujar el programa a l'Arduino.

# CONEXIÓ ARDUINO A DCC

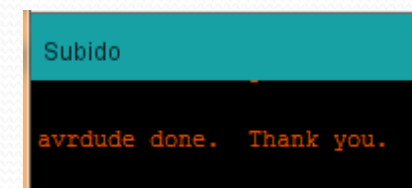


A l'Arduino IDE comprovem que hem seleccionat la placa Arduino que tenim i el port al que esta conectada.

Per carregar el programa premerem a sobre el boto de Pujar programa.



Si tot va bé ens informarà de que ha esta pujat



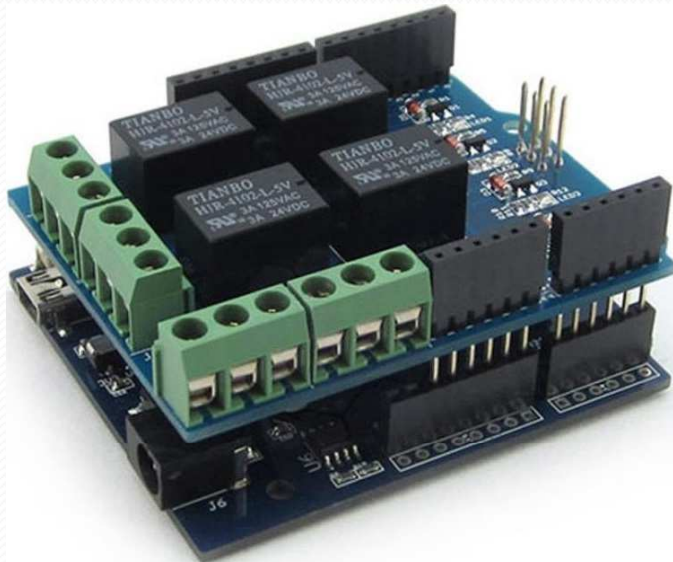
La llibreria permet fer tant descodificadors d'accessoris com descodificadors embarcats. També té funcions per poder gestionar les CV.

N'hi ha d'altres llibreries per descodificar la senyal DCC:

MynaBay: [https://github.com/MynaBay/DCC\\_Decoder](https://github.com/MynaBay/DCC_Decoder)

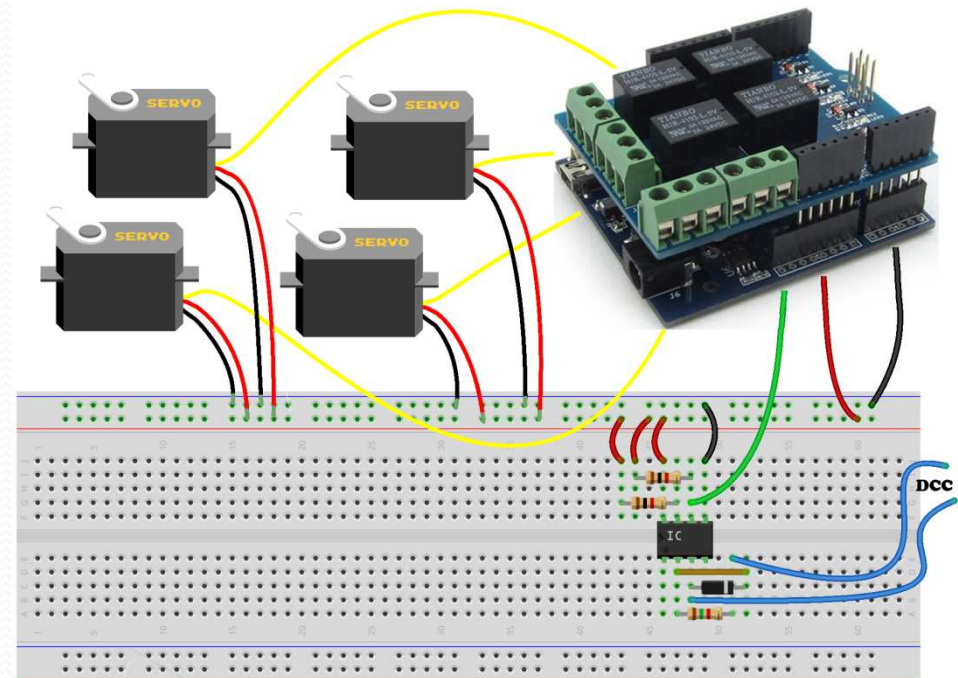


# CONEXIÓ ARDUINO A DCC



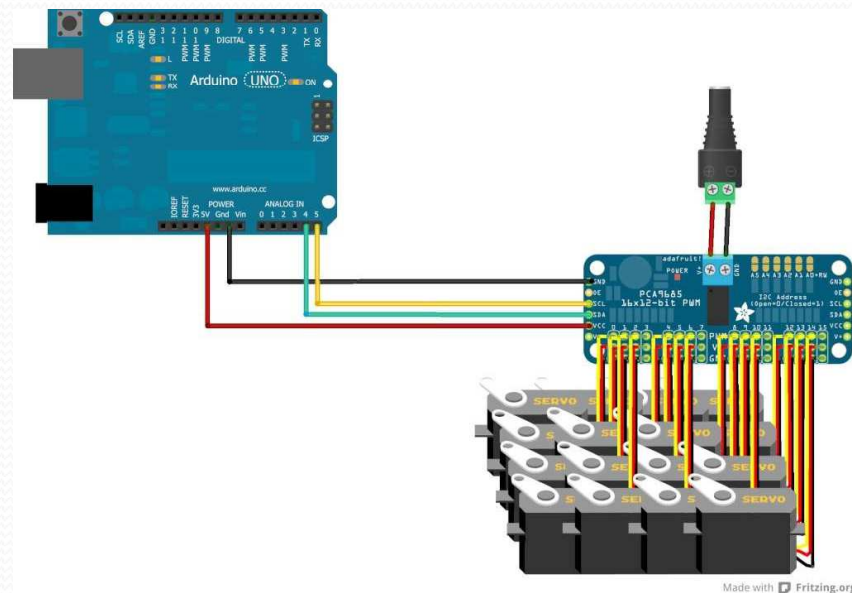
Punxant una 'shield' de relés i ampliant una mica el programa podem tindre un descodificador de accessoris amb 4 relés similar als comercials i encara podríem mirar d'aprofitar alguna de les sortides per encendre uns LEDs, per exemple.

O bé, afegint una altra de les llibreries disponibles que ens permet controlar un servo, i modificant convenientment el programa podem tindre un descodificador per agulles accionats per servo amb polarització dels cors

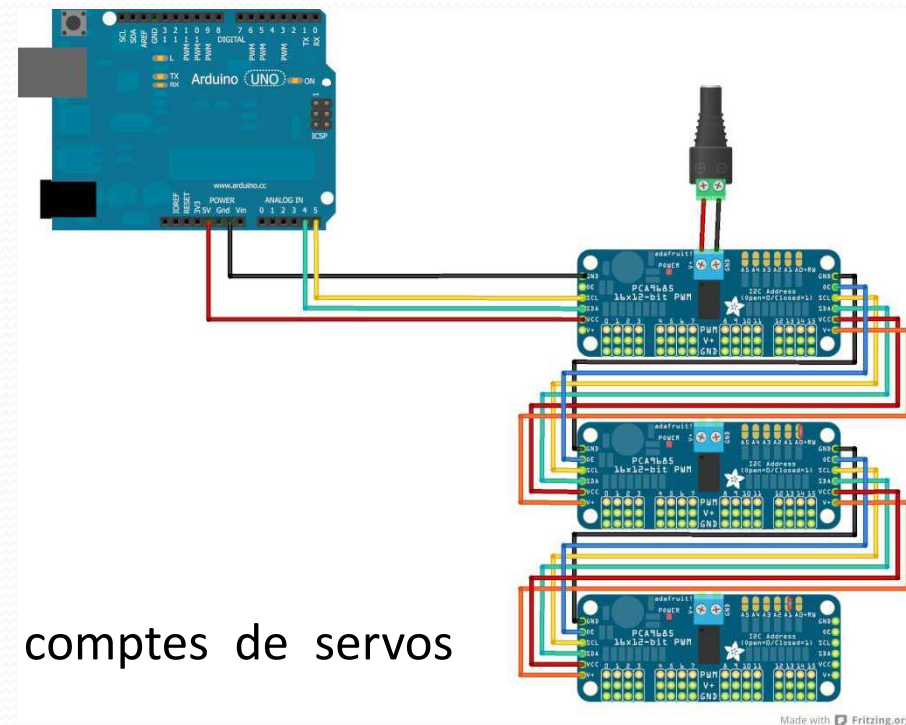


# CONEXIÓ ARDUINO A DCC

Amb la llibreria de servos només podríem controlar fins 12 servos a l'Arduino Uno, però amb plaques addicionals i fent ús del bus I2C podríem controlar de 16 a 1000. Necessitarem instal·lar les llibreries adients per treballar amb el bus I2C i el xip controlador de la placa de servos.



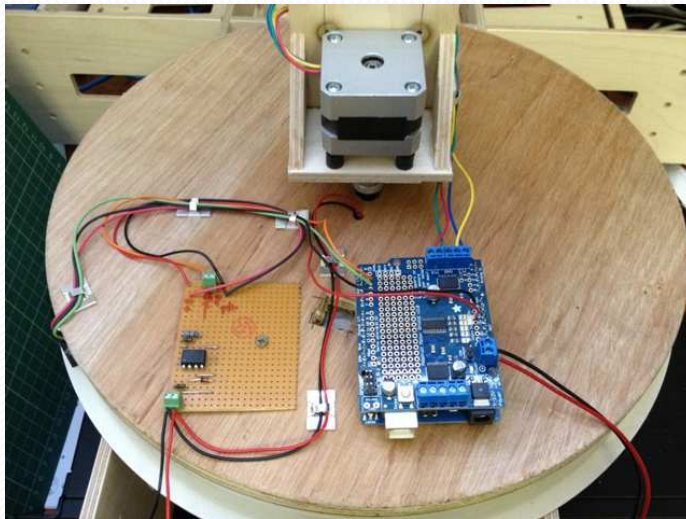
Les sortides son PWM per lo que en comptes de servos podríem controlar la intensitat de LEDs





# CONEXIÓ ARDUINO A DCC

Amb la llibreria I2C i una altra per a pantalles OLED, i una mica de feina de programació podem tindre cartells anunciadors a les estacions.

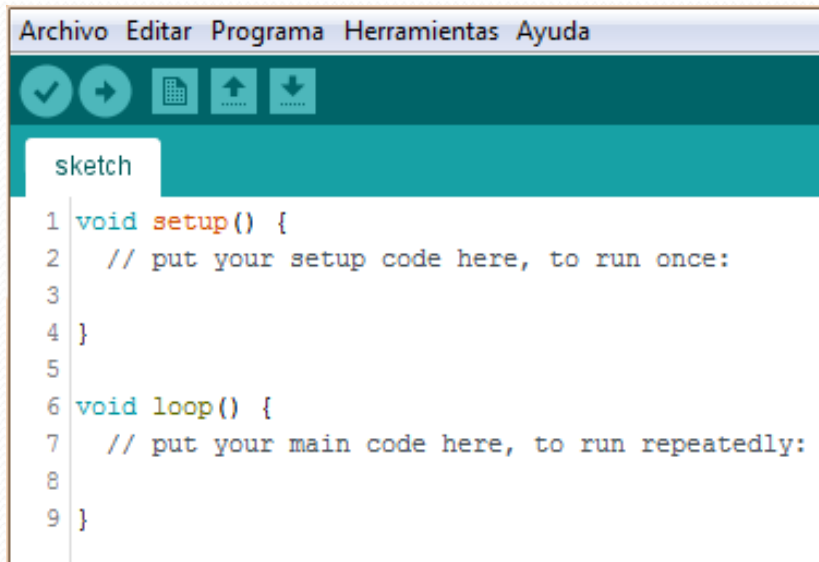


Amb una 'shield' de control de motors podrem controlar un motor pas a pas per moure una plataforma giratòria.

Hi ha una llibreria per a aquest tipus de motors.

# APENDRE A PROGRAMAR L'ARDUINO

S'anomena 'sketch' a un programa fet per a la plataforma Arduino.



```
Archivo Editar Programa Herramientas Ayuda
✓ → [Grid] [Up] [Down]
sketch
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Te dos rutines principals:

**setup()** – Conté el codi de configuració, s'executa una sola vegada al donar tensió o prémer el boto Reset.

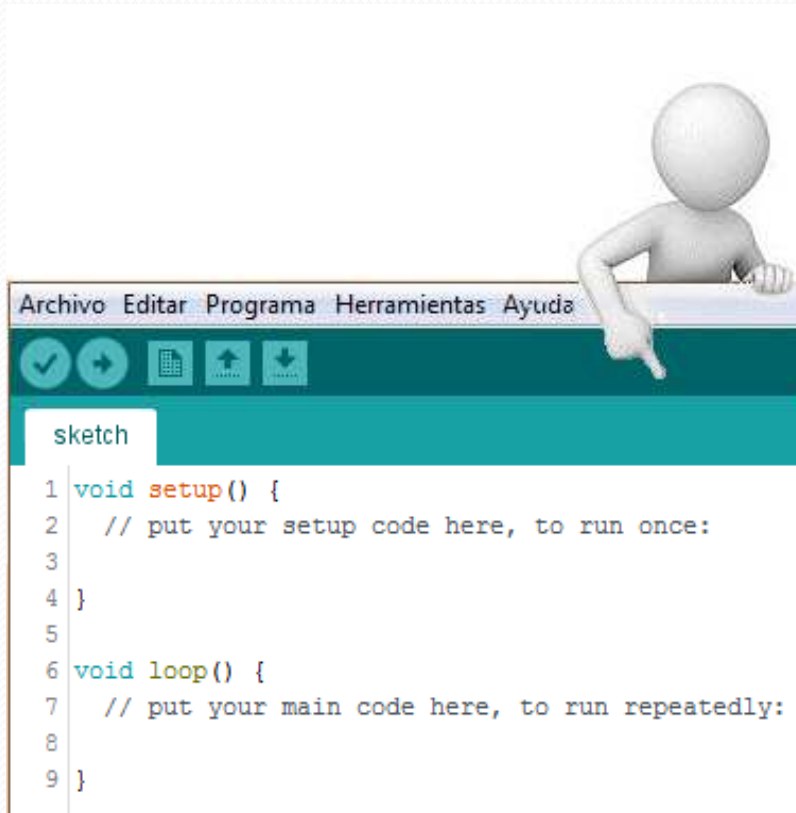
**loop ()** – Es el programa principal, s'executa una vegada i un altra sense parar.

Hi ha gran quantitat de llibreries que podem fer servir i que fan molt fàcil interactuar amb el hardware, per utilitzar-les les haurem d'instal·lar i s'han de incloure al codi fent servir la directiva **#include <llibreria.h>**

Haurem d'estudiar la llibreria o mirar els exemples que normalment porten per saber quines funcions afegeix i com les hem de fer servir.



# APENDRE A PROGRAMAR L'ARDUINO



## Ep! Un moment!

He d'aprendre tot això per fer servir l'Arduino a la maqueta?

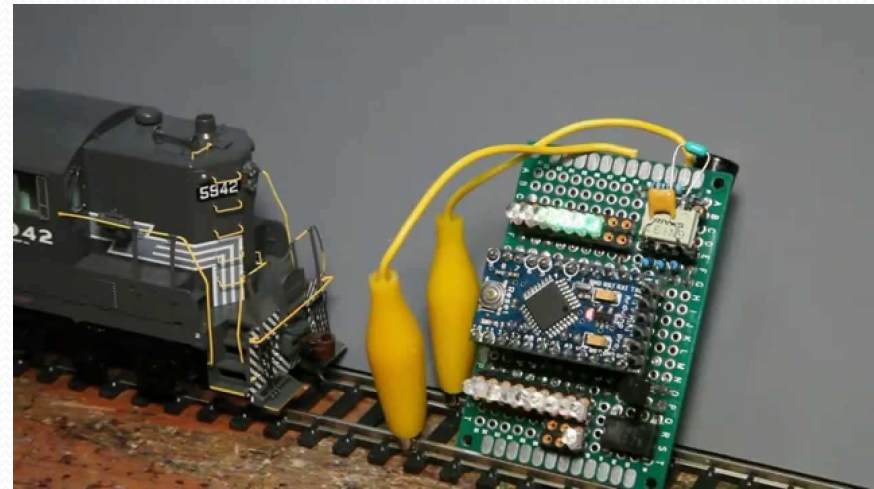


# APENDRE A PROGRAMAR L'ARDUINO



Vull un disseny especial ja que no he trobat res que faci el que necessito.

Em val amb el disseny que he vist a Internet en que la feina ja esta feta.



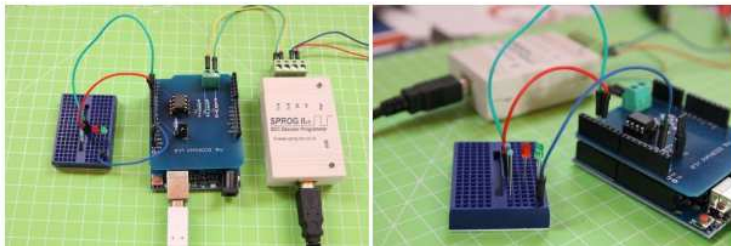


# NO APENDRE A PROGRAMAR L'ARDUINO

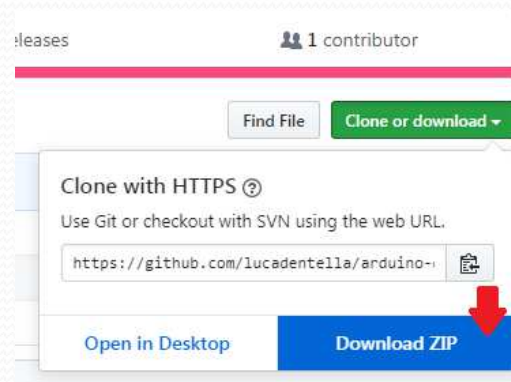
## DCC, LED ACCESSORY DECODER

LUCA 25/11/2017 3

After having designed a *shield* to interface Arduino with a DCC bus, today I'll show you how to realize a simple accessory decoder to control a couple of leds.



2 Descarreguem el software i el descomprimim si cal. L'obrirem a l'Arduino IDE.



1 He vist a Internet un descodificador que em serveix i a te el software llest per descarregar

Arduino projects for digital model railways in DCC

3 commits

1 branch

Branch: master

New pull request

lucadentella Decoder with CVs added

DccLedDecoderCV Decoder with CV

SimpleLedAccDecoder Simple led acces

README.md Initial commit

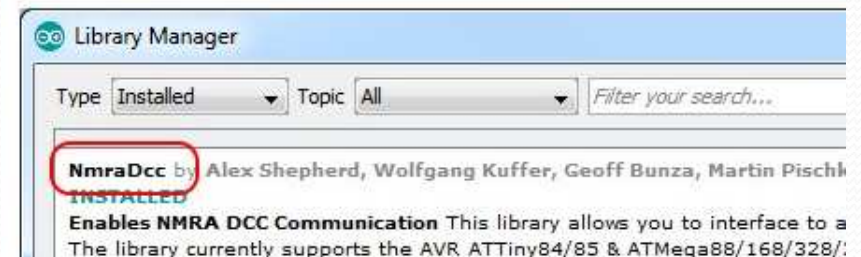
README.md

arduino-dcc

Arduino projects for digital model railways in DCC

3 Instal·lem les llibreries que ens indiquin o mirem al codi si ens fa falta alguna llibreria

```
#include "NmraDcc.h"  
NmraDcc Dcc;
```



# NO APENDRE A PROGRAMAR L'ARDUINO

```
#define BOARD_ADDRESS 5  
#define PORT_ADDRESS 1
```

4 Comprovem si hem de canviar alguna cosa al codi per a que s'adapti a les meves necessitats

Placa: "Arduino/Genuino Uno" ▶

Puerto: "COM3" ▶

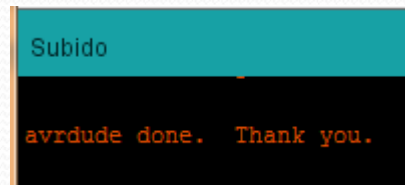
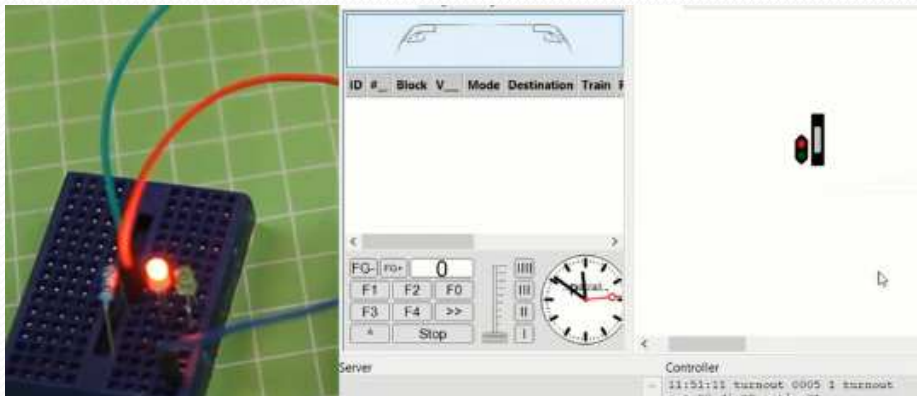
Obtén información de la placa

5 Escollim la nostra placa a l'Arduino IDE i el port on esta connectada



6 Li donem al boto de 'Pujar' el programa al nostre Arduino

7 Comprovem que hi hagi anat bé sense errors



8 Comprovem que funciona



# ARDUINO SENSE COMPLICACIONS

A Internet hi ha muntatges de descodificadors DCC en que la feina ja esta feta. Només cal abaixar-se el codi, instal·lar les llibreries que faci servir, i pujar el programa a l'Arduino.

Moltes vegades l'únic que s'ha de modificar del codi es la adreça a la que volem que respongui o algun valor que depengui de lo hem muntat si hi ha diferents opcions.

Si fan servir CV es com un descodificador normal comercial però hi ha que no cal, li pots dir des de el monitor sèrie:

```
Mardec on port COM3
MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 5.1

Mardec starting, please wait

Configuration mode of MARDEC #1

Settings of MARDEC #1

Default servo rotation speed: 25 ms/degree
Address offset: No
Servo's are detached at end of rotation
Startup mode: Last

Port 1: DCC 1, Servo , Angles 80/137, Inverted, Speed 25, Frog port no
Port 2: DCC 2, Servo , Angles 74/136, Not Inv., Speed 25, Frog port no
Port 3: DCC 3, Servo , Angles 65/135, Inverted, Speed 25, Frog port no
Port 4: DCC 4, Servo , Angles 65/140, Inverted, Speed 25, Frog port no
Port 5: DCC 9, Servo , Angles 69/125, Not Inv., Speed 25, Frog port no
Port 6: DCC 10, Servo , Angles 61/130, Not Inv., Speed 25, Frog port no
Port 7: DCC 11, Servo , Angles 75/148, Not Inv., Speed 25, Frog port no
Port 8: DCC 12, Servo , Angles 76/133, Inverted, Speed 25, Frog port no
Port 9: DCC 17, Servo , Angles 67/115, Not Inv., Speed 25, Frog port no
Port 10: DCC 18, Servo , Angles 48/118, Inverted, Speed 25, Frog port no
Port 11: DCC 19, Servo , Angles 65/111, Not Inv., Speed 25, Frog port no
Port 12: DCC 20, Servo , Angles 76/136, Not Inv., Speed 25, Frog port no
Port 13: DCC 5, Acc.type 1 (S. Steady), , Not Inv.
Port 14: not used
Port 15: not used
Port 16: DCC 5, Acc.type 1 (S. Steady), , Not Inv.

Servo on A1/P1 set to 80 degrees
Servo on A2/P2 set to 74 degrees
Servo on A3/P3 set to 65 degrees
Servo on A4/P4 set to 65 degrees
Servo on A9/P5 set to 125 degrees
Servo on A10/P6 set to 61 degrees
Servo on A11/P7 set to 75 degrees
Servo on A12/P8 set to 76 degrees
Servo on A17/P9 set to 115 degrees
Servo on A18/P10 set to 118 degrees
Servo on A19/P11 set to 65 degrees
Servo on A20/P12 set to 136 degrees
Accessory on A5/P13 is turned on
Accessory on A5/P16 is turned on

Mardec started

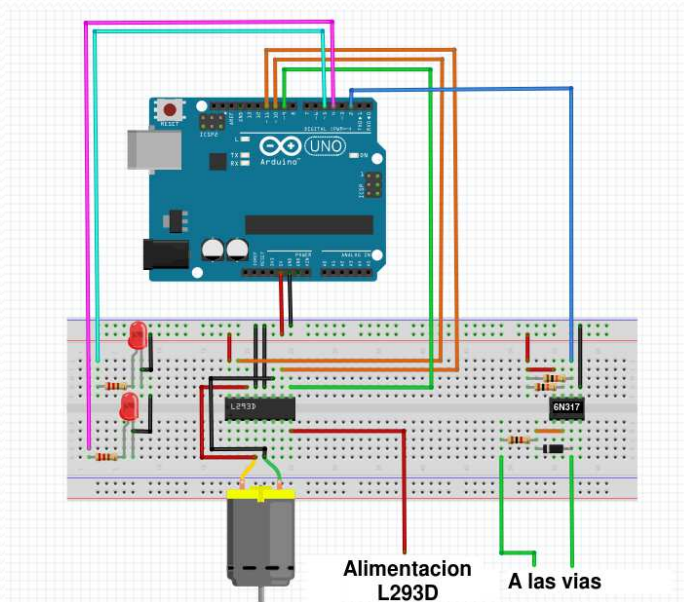
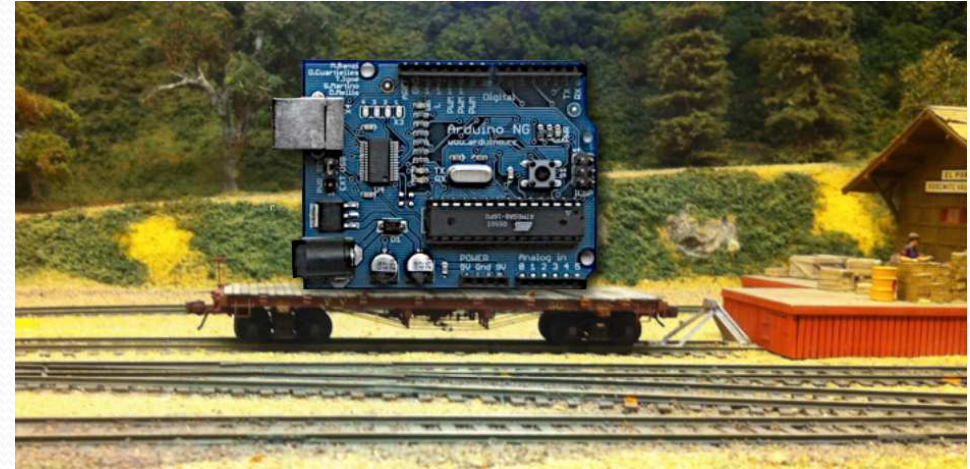
Specify action (P/A/R/T/D/E/I/?):
```



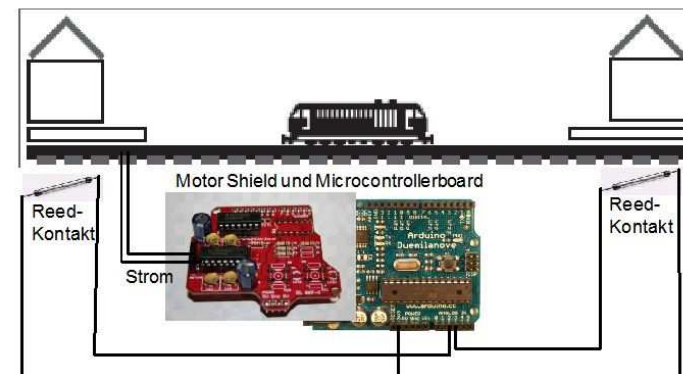
# DESCODIFICADORS PER LOCOMOTORES

També hi ha algun disseny de descodificadors embarcats per locomotores, però la mida es gran per a les escales petites.

Estan basats en fer servir el circuit de control de motors.



El més habitual es trobar circuits de control per anada i tornada de locomotores

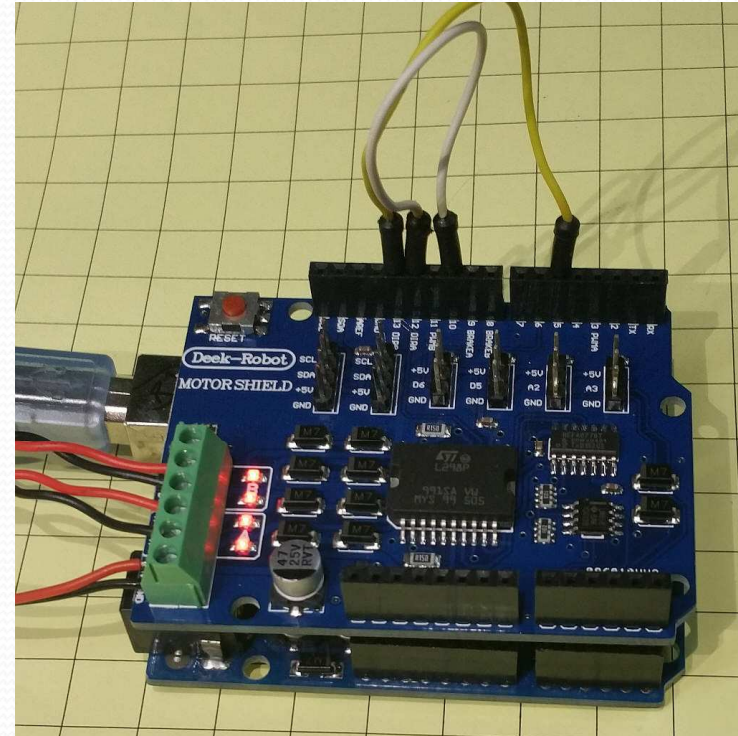
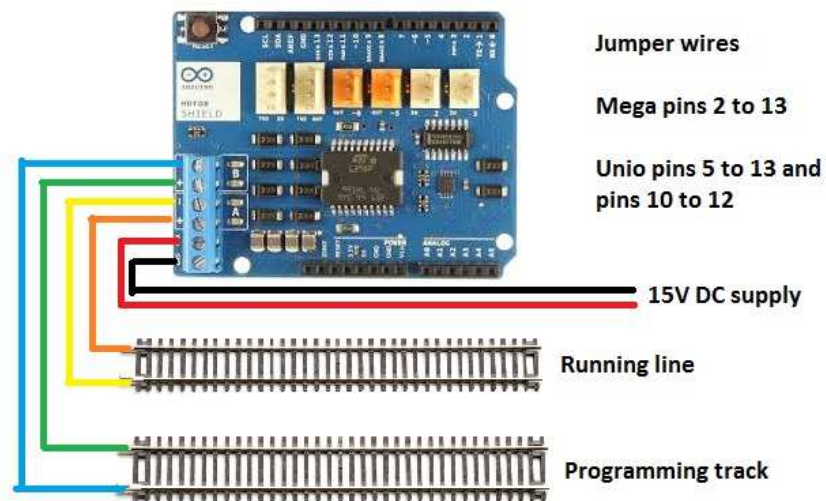




# CENTRAL DCC AMB ARDUINO

DCC++ es el disseny d'una central DCC feta amb l'Arduino i una 'shield' de control de motors.

Es controla des de l'USB amb el monitor sèrie mitjançant comandos de text.

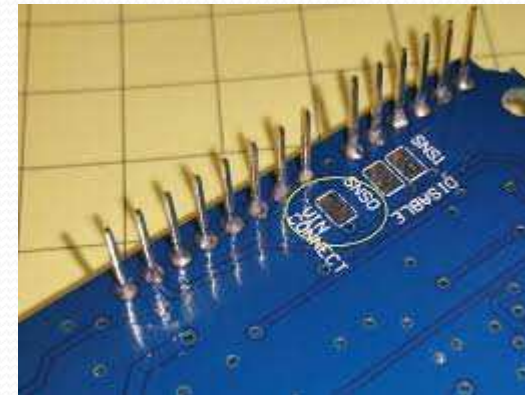


Té sortides per via principal i via de programació per llegir i escriure CV.

També es pot controlar des de JMRI i Rocrail.

# CENTRAL DCC AMB ARDUINO

Recordem que l'Arduino s'alimenta entre 7 i 12V, així que hem de aïllar la connexió  $V_{IN}$  de la 'shield' per a que la tensió a la que alimentem la DCC++ que es superior no es espatlli l'Arduino, aquest s'alimentarà des de l'USB.



A Internet, hi ha muntatges de la DCC++ amb 'shield' Ethernet o WiFi i amb una pantalla LCD per fer-la autònoma i no dependre de l'ordinador.

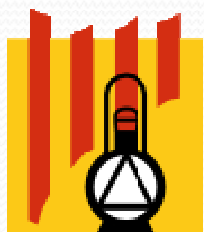
També s'han dissenyat comandaments per cable o inal·làmbrics per a la connexió sèrie.





# INTERNET

- Llibreries Arduino  
<https://www.arduino.cc/en/Reference/Libraries>
- Adafruit PCA9685 16-Channel Servo Driver  
<https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all>
- Arcomora  
<https://www.arcomora.com/>
- DCC++  
<https://sites.google.com/site/dccppsite/home>
- La maqueta de Infotronikblog  
<http://lomaquetade.infotronikblog.com/>
- Club N Caldes  
<http://www.clubncaldes.com/search/label/Arduino>
- Locoduino  
<http://www.locoduino.org/>



FEDERACIÓ  
CATALANA  
D'AMICS DEL  
FERROCARRIL

## GRÀCIES PER LA SEVA ATENCIÓ

Ponent: Paco Cañada

The **POWS**

[www.fut.es/~fmco](http://www.fut.es/~fmco)

