

Com construir un robot de sumo i no morir en l'intent

Taller de robòtica. Astrobanyoles

coordinador tècnic: Isaac de Palau

22 de juny de 2014

Índex

1	Introducció	3
1.1	Què és Arduino?	3
1.2	Anatomia d'una placa Arduino	3
1.3	La IDE d'Arduino	3
1.4	Material que cal portar	4
1.5	Instal·lar la IDE d'Arduino	4
1.5.1	Màquina virtual JAVA	4
1.5.2	Windows	4
1.5.3	Linux	5
1.6	Instal·lar les llibreries	5
1.6.1	Windows i Mac	6
1.6.2	GNU/Linux	6
2	Blink	7
2.1	Circuit blink	7
2.2	Codi blink	7
3	Blink amb pulsador	9
3.1	Circuit pulsador	9
3.2	Codi pulsador	9
4	Blink per Serial	11
4.1	Circuit blink per Serial	11
4.2	Codi blink per Serial	11
5	Sensor de distància SHARP	13
5.1	Circuit Sharp	13
5.2	Codi circuit Sharp	14
6	MUNTATGE DEL XASSÍS DE SUMO	16
6.1	Guia pas a pas	16
6.2	Pins del zumo motor shield	25
7	¡Levántate y... ¿rueda?	27
7.1	Circuit	27
7.2	Codi	27
8	"Yo... VIVO"! construcció d'un avoider	29
8.1	Circuit avoider	29
8.2	Codi avoider	29

9	Sensor de color	30
9.1	Circuit sensor de colors	30
9.2	Codi sensor de colors	30
10	Normes del minisumo	32
10.1	Els Robots	32
10.2	El Ring	32
10.3	El Combat	32
10.4	El Torneig	32
11	Consells a l'hora del torneig	33

1 Introducció

Si tens aquest dossier en pantalla o a les mans en format arbre mort suposaré que estàs interessat en entrar en l'apassionant, divertit i [introdueix el teu adjectiu favorit] món de la robòtica. També suposaré que no tens absolutament cap tipus de coneixement de robòtica, no has programat mai amb cap llenguatge i la paraula Shield et sona a pel·lícula de superherois. Aquest dossier et servirà com a base per construir, amb l'ajuda del teu equip, un robot de sumo durant el curs que tindrà lloc els dies 30, 1, 2, 3 i 4 de juny i juliol. El sumo amb robots és un joc en el que dues màquines autònomes s'enfronten en un ring, intentant tirar el contrari a fora. Aquí trobaràs les bases i els coneixements per entendre com funciona un robot d'aquest tipus. No obstant, de tu dependrà aplicar tot el que has après i combinar-ho per crear una màquina capaç de jugar a aquest joc. Per assolir el nostre objectiu farem un robot basat en la popular plataforma Arduino.

Els primers dies del curset, abans de començar a treballar amb el dossier, es farà una explicació sobre els conceptes bàsics de programació que cal tenir clars per tal d'entendre cada una de les pràctiques.

1.1 Què és Arduino?

Arduino és una placa pensada per realitzar projectes d'electrònica: des de complexos robots i drons voladors a sistemes domòtics o sintetitzadors musicals. La seva senzillesa fa que sigui molt fàcil connectar tot tipus de sensors i actuadors a aquesta placa, convertint-la en la plataforma més popular entre els aficionats a l'electrònica (i no tant aficionats!).

Arduino també disposa d'una **IDE**. La IDE és com un bloc de notes que ens permet crear un programa per la nostra placa Arduino i enviar-li. Un programa és com un seguit d'instruccions o d'ordres que l'Arduino anirà seguint. Podem dir-li que activi un motor que prèviament hem connectat a una sortida, o podem visualitzar la lectura que ens dóna un sensor de distància.

Hi ha molts models d'Arduino: *UNO*, *Nano*, *Mega*, *Yún*, *Tre*... Nosaltres farem servir **Arduino UNO r3**, que actualment és la versió estàndar.

1.2 Anatomia d'una placa Arduino

Arduino té pins. Té molts pins. Té infinitud de pins. Tot seguit s'especifiquen les funcions dels més importants junt amb algunes de les parts de la placa.

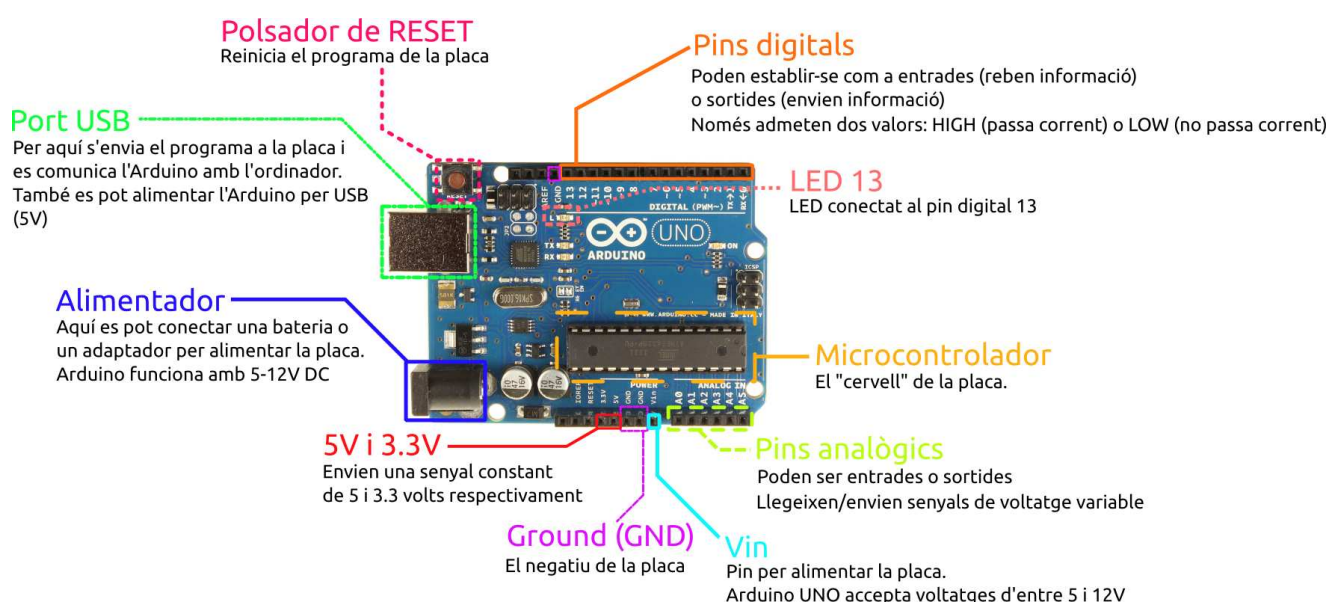


Figura 1: arduino

1.3 La IDE d'Arduino

La IDE d'Arduino és un programa que ens permet crear de forma senzilla el codi per la nostra Arduino. Un cop instal·lat i executat el programa, ens hauria de sortir una pantalla com aquesta:

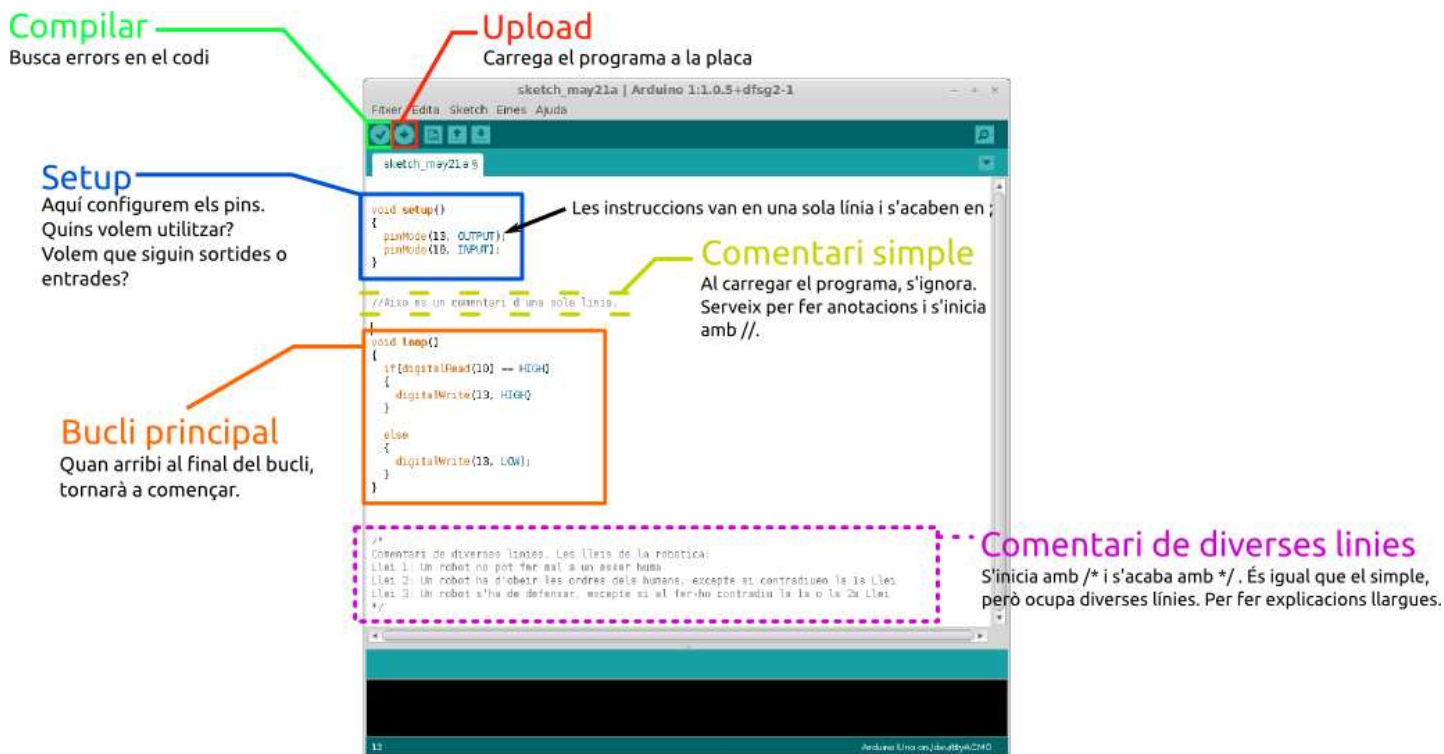


Figura 2: IDE

Val a dir que quan es fa un programa MAI s'han de fer servir accents, dièresis, "ç" ni caràcters estranys.

1.4 Material que cal portar

La gran majoria del material que necessitareu us el donem nosaltres (xassís, plaques, components, sensors...). No obstant hauríeu de portar:

- 4 piles AA recarregables (del mateix tipus)
- Portàtil i carregador (si en teniu). Hauria de tenir instal·lada la IDE d'Arduino (vegeu el capítol següent)
- Llapis, boli i quelcom per escriure si voleu anotar quelcom (no us farem prendre apunts!)
- Les ganes.

1.5 Instal·lar la IDE d'Arduino

Per tal de poder treballar i evitar perdre temps durant el curs set aniria bé que tothom portés el seu ordinador amb la IDE (Entorn per programar la placa) d'Arduino instal·lada. Tot seguit s'especifiquen els passos a seguir.

1.5.1 Màquina virtual JAVA

Primer de tot és necessari descarregar i instal·lar la màquina virtual Java si no la teniu en el vostre ordinador, per tal que la IDE d'Arduino funcioni. Escolliu i descarregueu la vostra versió depenent del sistema operatiu del següent enllaç: <https://www.java.com/en/download/manual.jsp>

1.5.2 Windows

- Descarregar l'arxiu de: [arduino.googlecode.com/files/arduino-1.0.5-r2-windows.zip](https://www.java.com/en/download/manual.jsp)
- Descomprimir l'arxiu i instal·lar-lo

- Instal·lar els drivers de windows del port **Serial Com**. Necessitarem fer-ho manualment perquè possiblement no detectarà la placa arduino a la primera quan es connecti al port USB. Anem a **Panel de Control** – > **Sistema i seguretat** – > **Sistema** – > **Administrador de dispositius** – > **Dispositivo desconocido** (si no l'ha trobat, si no posarà Arduino) – > Premem botó de la dreta de ratolí per a cercar el controlador mitjançant el menú d'arxius. Cerquem dins la carpeta que hem descarregat i descomprimit d'Arduino, la carpeta que porta per nom **Drivers**. Nota: Si en lloc de la versió zip hem descarregat i instal·lat la versió instalable, la carpeta d'Arduino estarà segurament a `c:\Archivos de programa\Arduino` o similar. Tot i que el concepte és el mateix, aquest procediment per instal·lar els drivers pot variar depenent de la versió de Windows que s'utilitzi.

1.5.3 Linux

- Opció 1: des del centre de control (Synaptic, Yast o similar, depenent de la vostra distribució) cercar arduino i instal·lar-lo
- Opció 2 (per a Debian, Ubuntu i derivats): Obrir la Terminal i escriure `sudo apt-get install arduino` Escriure el vostre password i s'instal·larà el programa
- En principi, les darreres versions d'ubuntu i derivats i altres GNU/Linux més recents ja porten el driver i detecta la placa Arduino a la primera. No hi hauria d'haver problema.

1.6 Instal·lar les llibreries

A més de la IDE d'Arduino també caldrà instal·lar unes llibreries. Una llibreria és un arxiu amb funcions i codi escrit per tercers que podrem “annexar” al programa que fem pel nostre robot i així simplificar la feina.

Les llibreries s'haurien de portar ja instal·lades. No obstant, el primer dia del curs et ajudarem a tots aquells que no hagin aconseguit instal·lar-les.

El primer que haurem de fer és baixar el fitxer .zip amb les llibreries.

Per això anem a <https://github.com/pololu/zumo-shield> i fem clic al botó “Download ZIP” (marcat amb vermell a la figura de baix). Recordem on guardem l'arxiu.

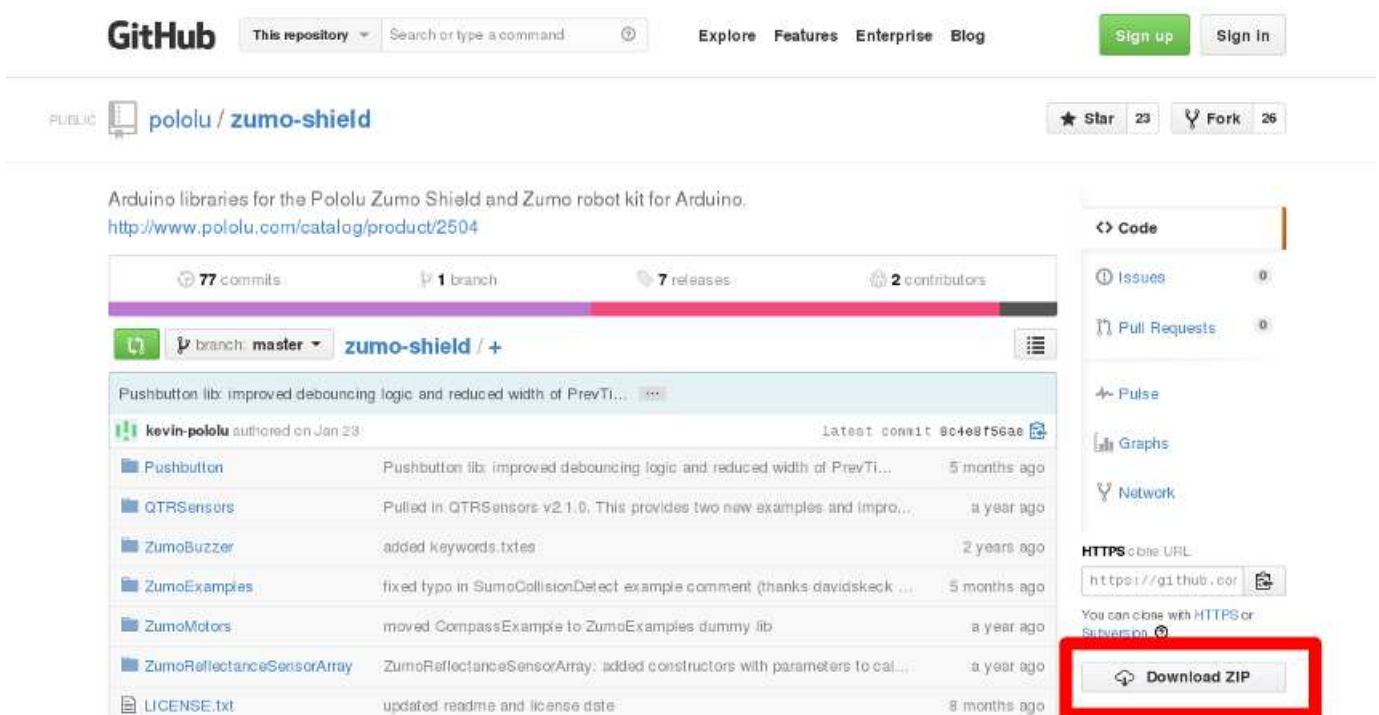


Figura 3: Descarreguem llibreries

Ara les haurem d'instal·lar. Segons el nostre sistema operatiu ho haurem de fer d'una forma o altra.

1.6.1 Windows i Mac

- Obrim la IDE d'Arduino que hem descarregat i instal·lat a l'apartat 1.5
- Fem clic a la pestanya Sketch – > Import Library – > Add Library (Sketch – > Importa Llibreria – > Afegeix Llibreria si el teniu en català)
- Seleccionem el fitxer .zip que hem descarregat.

1.6.2 GNU/Linux

La IDE de Linux no té actualment la opció d'importar llibreries automàticament, de manera que haurem de fer el següent:

- Descomprimim el fitxer .zip i recordem el nom del directori que es crea
- Obrim la Terminal i escrivim, en mode superusuari:

```
cp -rf /home/usuari/directori-on-he-baixat-la-llibreria/llibreria-que-hem-descomprimit /usr/share/arduino/libraries
```

Alerta! Canvieu els noms dels directoris a la instrucció anterior pel vostre nom d'usuari i carpeta.
- Un cop hem copiat el directori, li hem de donar permisos d'execució, lectura i escriptura (també en mode superusuari):

```
chmod +rwx -R /usr/share/arduino/libraries/llibreria-que-hem-copiat
```

2 Blink

L'exercici més bàsic que es proposa quan s'aprén a programar amb un llenguatge és el que es coneix com un "Hola Món". Un Holamón és un programa que simplement mostra per pantalla el text "Hola, món!" o algun derivat. Si bé ningú sap del cert l'origen del nom d'aquests programes, és un fet que tots els iniciats comencem per un programa així.

L'equivalent d'un Holamón en Arduino és el que es coneix com un **Blink!**, que consisteix en apagar i encendre un LED en intervals regulars de temps. El blink normalment es fa amb el LED que porta integrada la placa Arduino al pin núm 13. No obstant, nosaltres connectarem un LED exterior.

2.1 Circuit blink

Connectem el LED al pin 13 amb una resistència de 150 ohms, del contrari es podria fer malbé. Els LEDs tenen polaritat, de manera que hem de connectar la pota llarga (*ànode*) al positiu i la curta (*càtode*) al "GND"

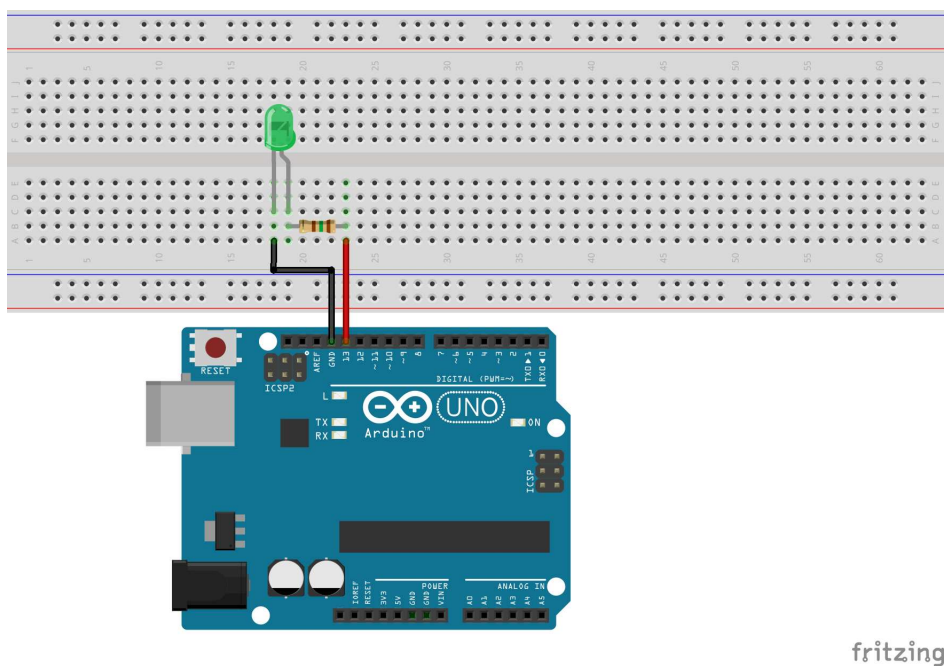


Figura 4: Circuit del Blink

2.2 Codi blink

```
#define LED 13 //Constant per guardar el pin del led

void setup()
{
  pinMode(LED, OUTPUT); //Establim el pin 13 com a sortida
}
void loop()
{
  digitalWrite(LED, HIGH); //Engeguem
  delay(500); //Esperem 0'5 segons (1000 = 1s)
  digitalWrite(LED, LOW); //Apaguem
  delay(500);
}
```

Una vegada hem escrit el programa al bloc de notes de la IDE, l'hem de compilar. Si ens dóna error, voldrà dir que hem escrit malament quelcom, de manera que haurem de revisar el codi. El missatge d'error ens pot donar pistes de què hem fet malament.

Quan compili correctament, connectem el cable USB a l'Arduino i carreguem el programa.

Si voleu, podeu intentar modificar el codi perquè pugui fer les següents tasques:

- Podeu provar a fer un semàfor afegint dos LEDs més: un de taronja i un de vermell. Heu de fer que s'encenguin i s'apaguin un rere l'altre.
- Guardant el temps dins una variable, podeu fer que a cada volta del loop el temps d'encesa del LED sigui el doble de l'anterior.

3 Blink amb pulsador

Tot doctor maligne i geni del mal ha desitjat alguna vegada tenir el seu propi panell de control per orquestrar la fi del món apretant la combinació correcta de botons. En aquesta pràctica no anirem tant lluny i no provocarem l'apocalipsi global, però aprendrem com fer servir pulsadors (botons) i com llegir informació a través dels pins digitals d'Arduino. També veurem, quan fem el codi, un cas pràctic de condicionals.

Quan apremem el botó, s'encendran i s'apagaran tres LEDs de forma escalonada.

3.1 Circuit pulsador

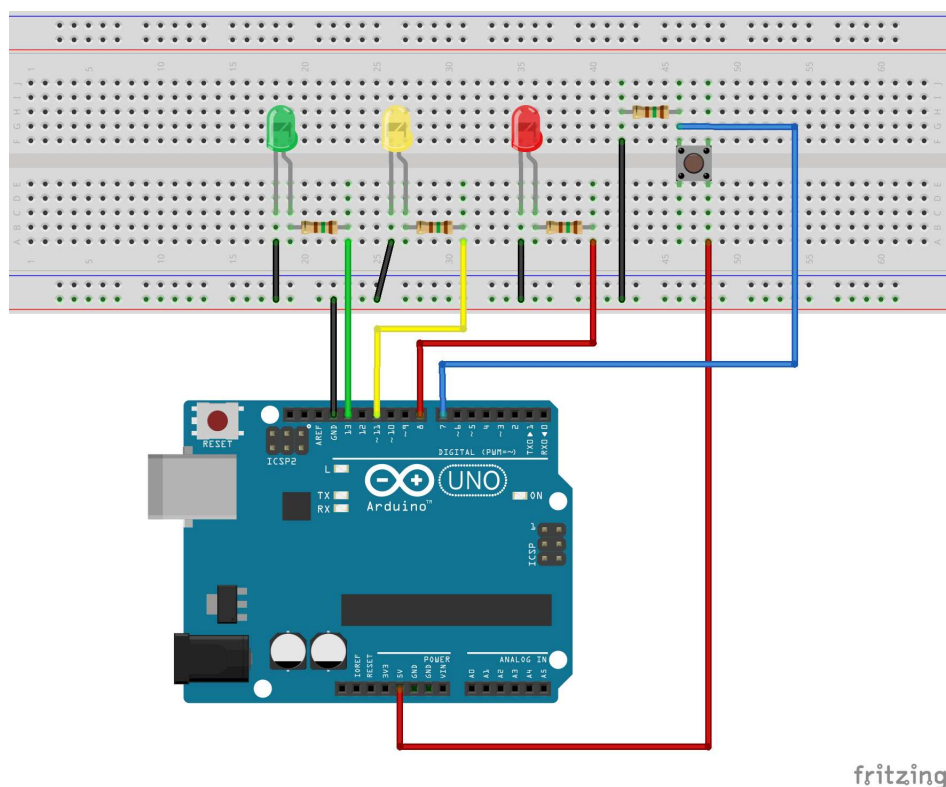


Figura 5: Fem un Blink amb pulsador

Ara ja sabem com es connecta un LED a l'Arduino, per tant serà qüestió de repetir el procés amb tres. El pulsador va connectat al pin 5V, que proporciona un voltatge constant (de cinc volts... sorpresa!). El connectem també al pin digital 7 de manera que quan es tanqui el circuit aquest pin rebrà voltatge i s'activarà. La resistència de 150 ohms a GND es necessària per evitar possibles interferències.

3.2 Codi pulsador

```
#define BOTO 7
#define VERD 13
#define GROC 11
#define VERMELL 8

int lectura; //Variable on guardarem el valor del boto (HIGH o LOW)

void setup()
{
  //Configurem els pins

  pinMode(BOTO, INPUT); //Establim el pin del boto com a entrada
  pinMode(VERD, OUTPUT); //Els pins dels LEDs seran sortides
  pinMode(GROC, OUTPUT);
```

```

    pinMode(VERMELL, OUTPUT);
}

void loop()
{
    lectura = digitalRead(BOTO); //Llegim el pin BOTO i guardem el seu valor

    if(lectura == HIGH) //Si el valor del pin BOTO es HIGH...
    {
        //... escala de color!!! :D
        digitalWrite(VERD, HIGH);
        delay(100);
        digitalWrite(VERD, LOW);
        digitalWrite(GROC, HIGH);
        delay(100);
        digitalWrite(GROC, LOW);
        digitalWrite(VERMELL, HIGH);
        delay(100);
        digitalWrite(VERMELL, LOW);
    }
}

```

Us heu quedat amb ganes de més? Podeu provar de modificar el codi perquè faci el següent:

- Afegiu un else, de tal manera que si no es prem el pulsador tots els LEDs estiguin encesos.
- Canvieu el codi, de tal manera que:
 - Si es prem una vegada el pulsador s’encengui el LED verd
 - Si es prem dues vegades s’encengui el LED groc
 - Si es prem tres vegades s’encengui el LED vermell
 - Si es prem quatre vegades s’apaguin tots
 - Si es prem cinc vegades torni a començar (podeu fer servir else if(condició))

EXTRA! Qui aconsegueixi fer el següent es guanya un minipunt:

- Modifiquen el codi original, de tal manera que faci exactament el mateix, però que si es deixa anar el pulsador quan està fent l’escala de color, es parin tots els LEDs

4 Blink per Serial

De vegades, necessitem rebre informació de l'Arduino i visualitzar-la en pantalla. Així mateix, també ens pot interessar enviar una instrucció des de l'ordinador a la nostra placa. Per això s'utilitza el protocol Serial, que bàsicament és una manera que té l'Arduino de comunicar-se amb dispositius externs. Ho farem a través del cable USB, però Arduino té alguns pins i connectors que es poden utilitzar per aquesta tasca.

En el següent exemple, l'Arduino encendrà un LED quan rebi per Serial un **1**, l'apagarà quan en rebi un **0**. Quan no rebi cap d'aquests dos números donarà error.

És possible que en funció de la versió de la IDE d'Arduino, aquesta pràctica no funcioni exactament igual per a tothom i que en algun cas sempre doni un missatge d'error. Això depèn de la versió de la IDE. Si es dona aquesta situació, demaneu ajuda als professors.

4.1 Circuit blink per Serial

El circuit és exactament el mateix del Blink. És important remarcar que si escrivim **'1'**, **val** no valdrà exactament 1. Arduino convertirà aquest 1 a un enter corresponent al caràcter ASCII del número 1, per tant la variable **val** valdrà 49. És per això que a les condicions posem **'1'** amb cometes simples, per mostrar que és un enter corresponent al caràcter ASCII '1'.

Per veure i enviar instruccions per Serial hem de fer servir la consola de la IDE.

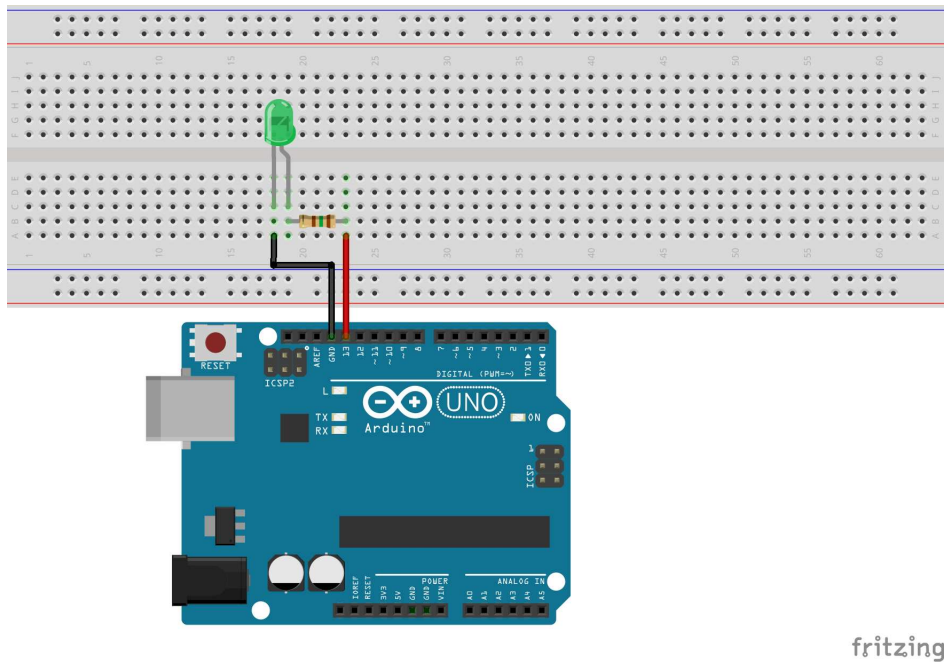


Figura 6: Deja Vu

4.2 Codi blink per Serial

```
#define LED 13

int val; //Variable per guardar l'ordre entrada

void setup()
{
  Serial.begin(9600); //Iniciem el Serial a 9600 bauds, velocitat estandar

  pinMode(LED, OUTPUT);
}

void loop()
```

```

{
  if(Serial.available()>0) //Si s'envia alguna cosa per Serial...
  {
    val = Serial.read(); //Guarda el que s'envia a val
    //Serial.println(val, DEC); //Pots descomentar la línia per veure l'entrada per Serial
    if(val == '1')
    {
      digitalWrite(LED, HIGH);
    }
    else if(val == '0')
    {
      digitalWrite(LED, LOW);
    }
    else
    {
      Serial.println("ERROR! :(");
    }
  }
}
}

```

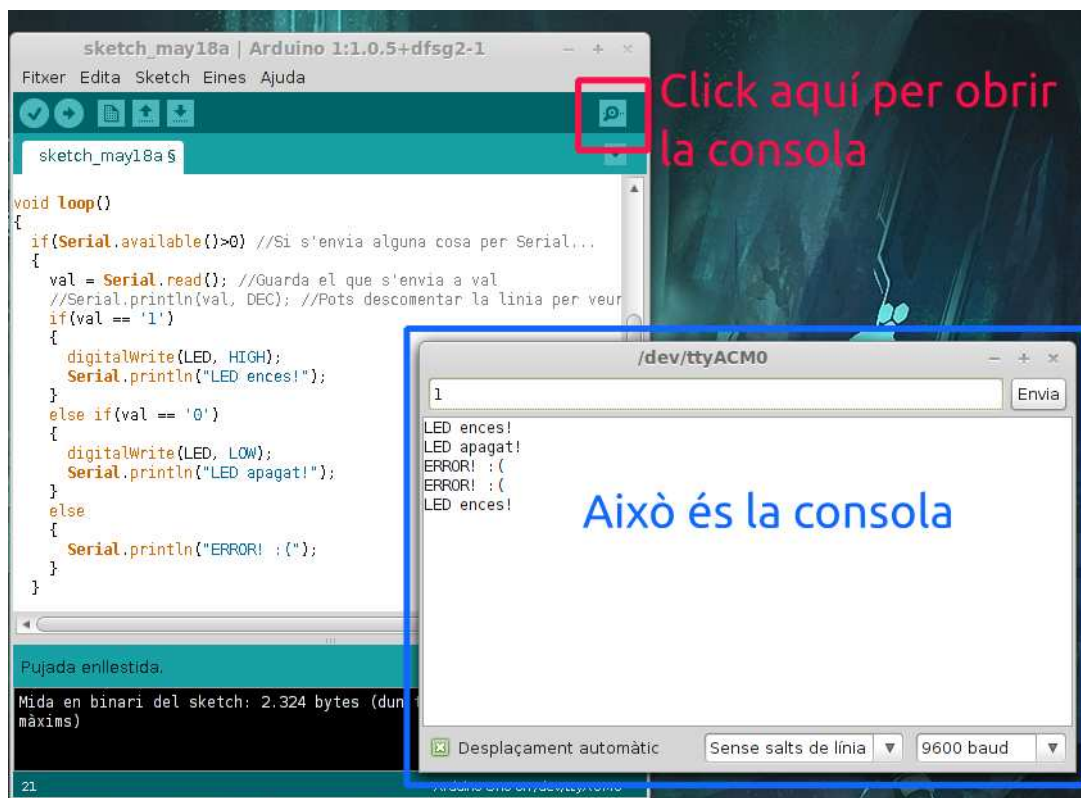


Figura 7: consola de la IDE

Si us sentiu amb ganes de més, podeu provar a fer algunes modificacions al codi:

- Poseu tres LEDs i, en funció de les entrades per Serial, n'enceneu un o altre.
- Modifiqueu el codi de tal manera que si entres un número, t'escrigui per Serial el mateix número que has entrat.

5 Sensor de distància SHARP

Davant d'una invasió extraterrestre com la de la pel·lícula Alien, és ben sabut que els llocs foscos solen estar ple de criatures horripilants amb l'únic desig de fer un bon caldo amb els ossos dels personatges secundaris. Per tal de no ser el primer plat d'alguna bestiola alienígena, res millor que tenir un sensor de distància per detectar els perills amb prou temps per poder sortir corrent.

El sensor de distància que farem servir envia rajos d'infrarrojos que, al col·lisionar amb un objecte, reboten. Segons l'angle amb que retornen al sensor, aquest calcula la distància a que es troba l'obstacle.

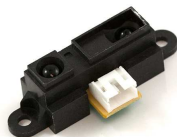


Figura 8: sensor Sharp

El sensor SHARP té un rang mínim de 4 centímetres, de manera que un objecte a menys d'aquesta distància pot donar lectures errònies tal i com es mostra a la següent imatge:

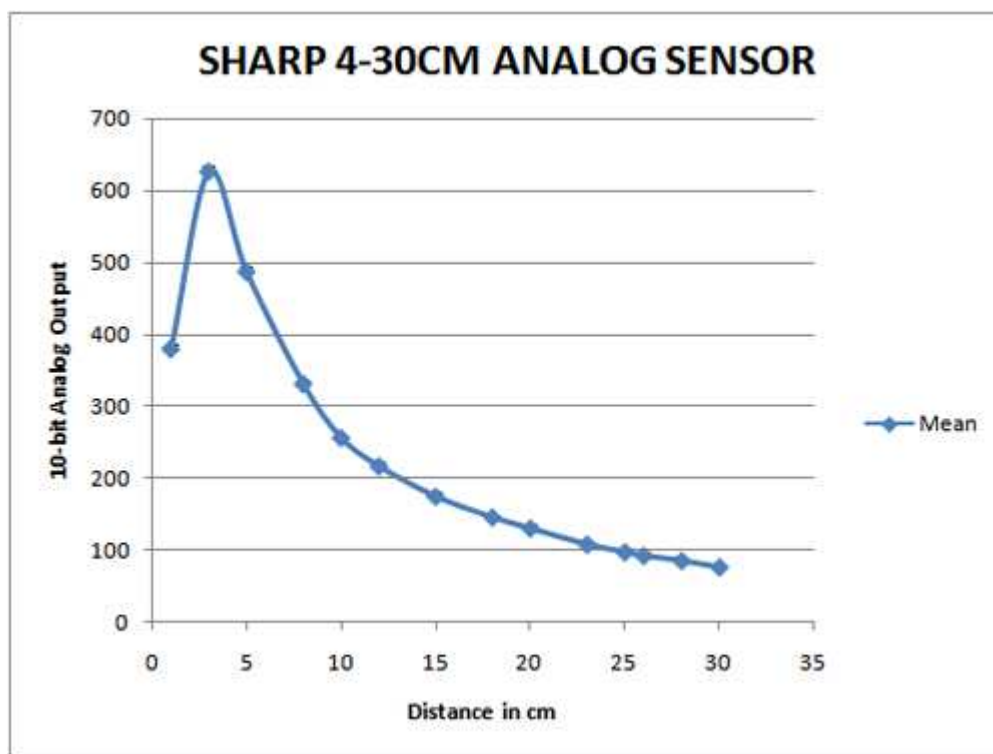


Figura 9: X = distància Y = voltatge analògic

En aquest capítol encendrem un LED quan el sensor detecti un objecte a menys de 15 centímetres aproximadament, i mostrarem la distància per la consola. Ens pot anar bé soldar un connector femella per cada cable del sensor per tal de connectar-lo més fàcilment quan construïm el nostre robot de sumo.

5.1 Circuit Sharp

El sensor sharp és un sensor analògic, de manera que el seu cable de senyal (el groc/blanc) haurà d'anar connectat a un pin analògic. Els altres dos cables vermell i negre van connectats a 5V i GND respectivament.

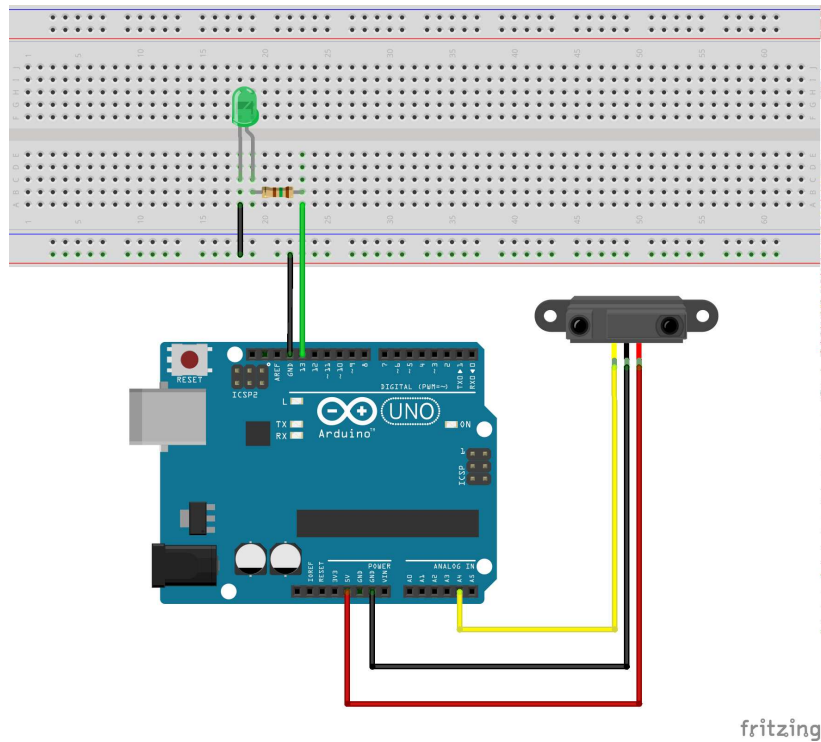


Figura 10: circuit amb el sensor SHARP

5.2 Codi circuit Sharp

Com més lluny estigui un objecte, més petit serà el valor que ens donarà el sensor. Com més a prop, més gran.

```
#define LED 13

int dist = 0; //variable per guardar distancia

void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  dist = analogRead(4); //Llegim A4 i guardem el seu valor
  Serial.println(dist); //Enviem dist per Serial

  //Fem servir condicionals d'una sola linia
  if(dist > 350) digitalWrite(LED, HIGH); //Si la distancia es mes gran de 350
  else digitalWrite(LED, LOW); //Si es inferior (l'objecte esta lluny)

  delay(100);
}
```

Les lectures del sensor poden variar en funció de la il·luminació de la sala, la forma de l'objecte i fins i tot la seva temperatura, així que la lectura sempre serà aproximada.

Si voleu experimentar amb el sensor, podeu provar a fer el següent:

- Segons la distància del sensor, s'ha de fer el següent:
 - (300-400): s'encén un LED verd i s'envia per serial un missatge a l'estil "OBJECTE ALLU-NYAT!"

- (400-500): s'encén un LED groc i s'escriu per serial "OBJECTE PROPER!" (distancia > 500)
s'encén un LED vermell i s'escriu per serial "OBJECTE MOLT PROPER!"
- A l'hora de fer el robot de sumo, ens serà molt útil tenir una funció per detectar objectes. Escriviu una funció booleana que rebi com a argument un número enter i retorni **cert** si el sensor d'infrarroig connectat al pin analògic del número entrat detecta un objecte.

6 MUNTATGE DEL XASSÍS DE SUMO

Per muntar el nostre robot de minisumo farem servir com a base un xassís prefabricat, el kit de zumo de Pololu, que ens estalviarà haver de construir una placa de motors, una base i suports per les bateries.

Aquest apartat del dossier només us explicarà com muntar el kit de sumo i us donarà tot un seguit de funcions bàsiques per fer funcionar els sensors i motors. També apareix al final un codi per construir un avoider (un robot que evita obstacles). A partir d'aquest exemple, dependrà de vosaltres decidir quin tipus de components i sensors voleu fer servir, com voleu col·locar-los i quin programa voleu que segueixi el vostre robot per convertir-vos en els vencedors de l'Arena. Hah, sona èpic i tot...

Hi ha moltes estratègies que un robot pot adoptar, i en realitat no n'hi ha una millor que altra: podeu moure el bot de manera aleatòria per la pista fins que troba un contrincant i atacar-lo frontalment, posant els motors a màxima potència. O potser decidiu fer moviments circulars i atacar al contrincant pel flanc quan aquest estigui desprevingut...

En tot cas cal tenir clares unes normes bàsiques a l'hora de construir el participant. Si voleu veure les normes complertes del minisumo, estan al final del dossier, però les més importants són:

- El robot no pot pesar més de 600 grams (piles incloses)
- El robot no pot passar dels 10 x 10 centímetres de mida. No hi ha limit a l'altura.
- El robot ha de ser capaç de tirar un dummy (una capsula de cartró) fora del ring en menys de 2 minuts.
- Al col·locar el robot al ring, aquest no ha de moure's fins al cap de tres segons de donar-li una senyal (podeu fer servir un pulsador i un delay)
- No podeu danyar intencionadament el robot contrincant. Així que res de motoserres, radials, llançafomes ni cap altre dels artefactes amb què estàveu pensant (no considerem dany si a l'apretar el contrincant aquest es desmunta)

6.1 Guia pas a pas

Per muntar el xassís del nostre robot de sumo farem servir el kit zumo de Pololu, que ens porta una placa de motors, rodes, motors i sensors de color la qual cosa ens estalvia bastanta feina a l'hora de dissenyar el hardware. A més, disposem de funcions especials i llibreries per programar amb més facilitat el codi del nostre robot. Quan haguem finalitzat el muntatge tindrem el següent:



Figura 11: Robot basic

És adorable, eh? El kit conté dos motors, un brunzidor (un "pito" bastant angoixant...), un array de sensors de color (a sota, per detectar els límits del ring), un acceleròmetre (que no farem servir) i una placa de motors per connectar amb Arduino i controlar tots els components abans esmentats. Tot seguit s'especifica cada part de la placa de motors.

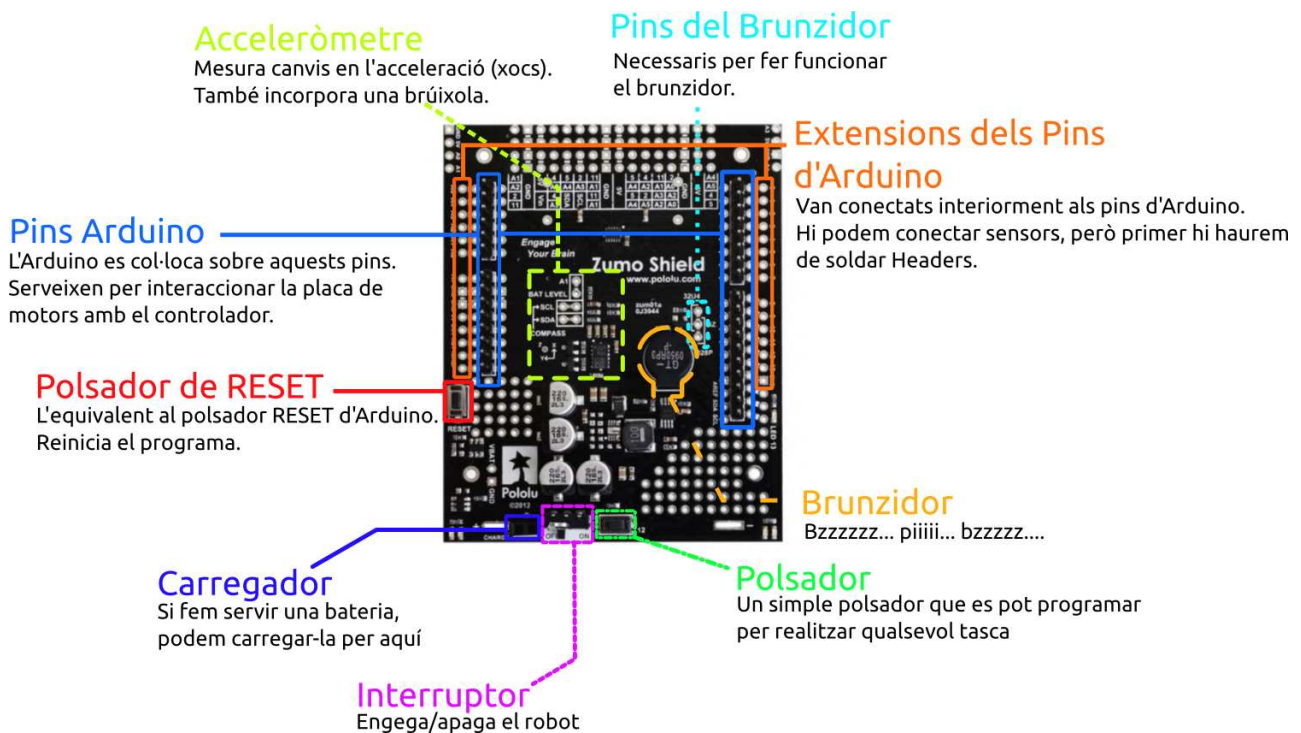


Figura 12: Esquema de la placa de motors

A l'hora de muntar el xassís és molt important seguir les instruccions al peu de la lletra i punt per punt. També hi haurà algunes parts que s'hauran de soldar, i això requereix força precisió. Si ho necessiteu, no dubteu en demanar ajuda.

- Abans de començar a muntar haurem de submergir en aigua les plaques del chasis que tinguin adhesiu (és una làmina marró que sembla de paper) durant uns quants minuts. Així serà fàcil de retirar-lo.
- Primer de tot soldem els headers a la placa de motors, que seràn els que connectaran amb la placa Arduino. Per fer-ho, podem connectar els headers a la placa Arduino i després col·locar la placa de motors al cim per soldar millor.

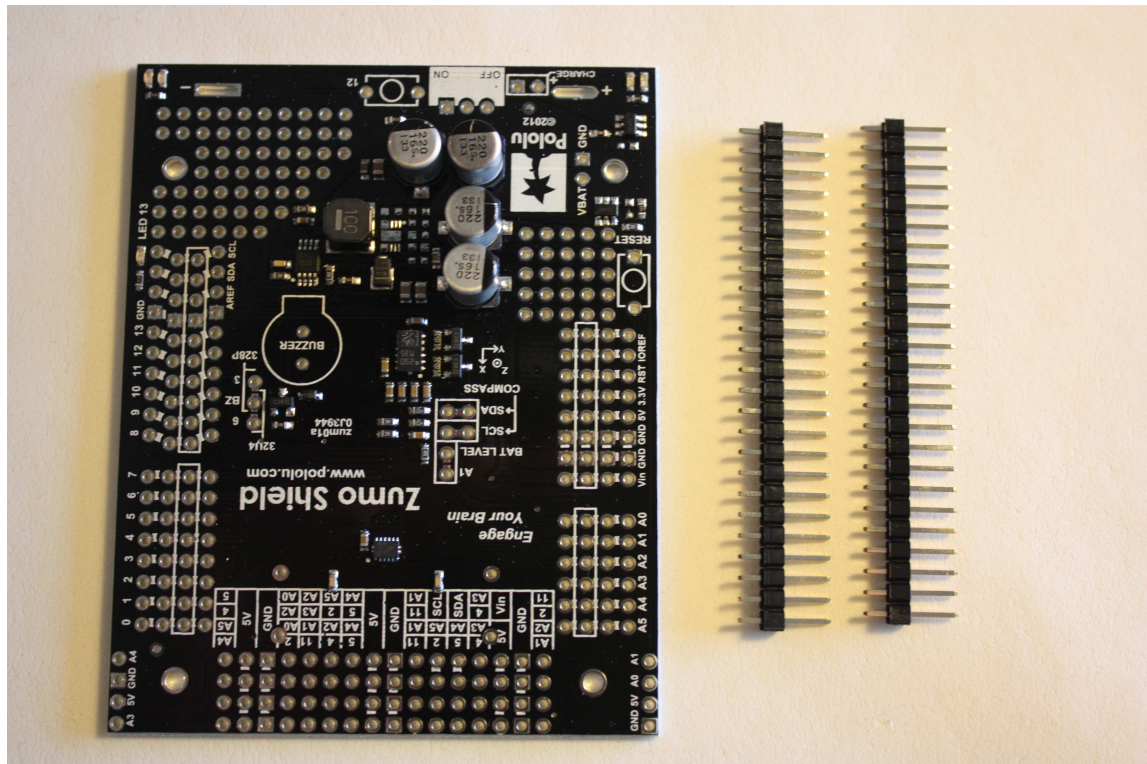


Figura 13: Placa de motors i headers

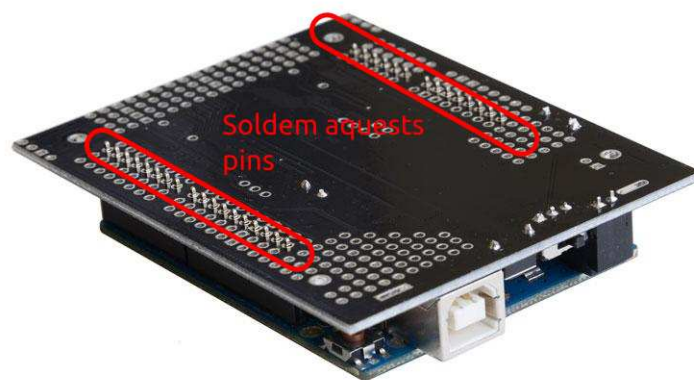


Figura 14: Col·locar la shield al damunt de l'Arduino ens ajudarà a l'hora de soldar

- Un cop acabat retirem la placa Arduino i soldem el brunzidor, també soldem els headers per poder connectar els sensors amb l'Arduino. Col·loquem un dels ponts blaus en els pins del Brunzidor marcats com a BZ i 3.
- Finalment soldem el pulsador de Reset i el botó. També soldem el carregador de bateries (opcional).

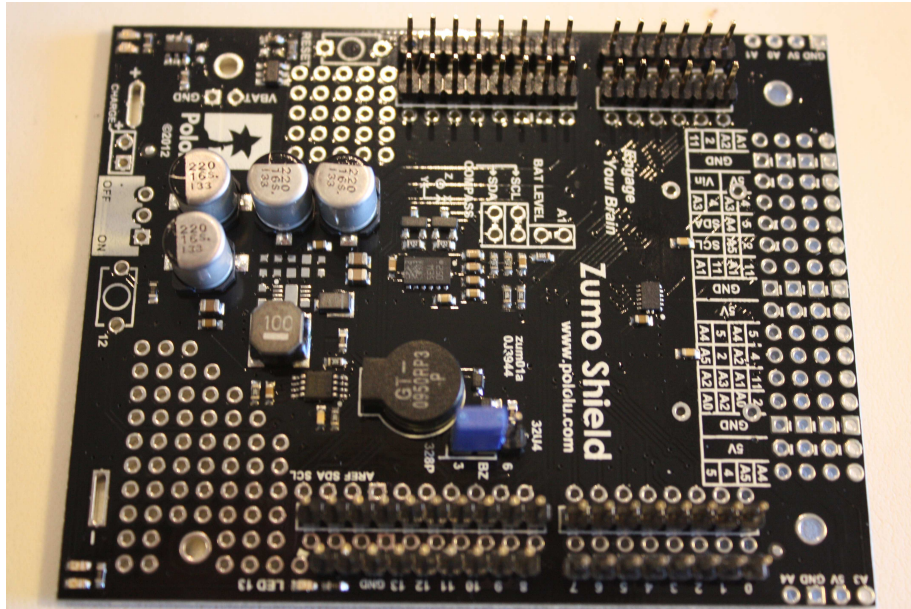


Figura 15: Placa soldada (aquí falten l'interruptor i els dos polsadors)

- Ara agafem els motors i soldem mig fil conductor al pol del motor marcat amb el signe "+". Després feu el mateix amb els pols negatius i col·loqueu els motors a la base del xassís de manera que quedin així:

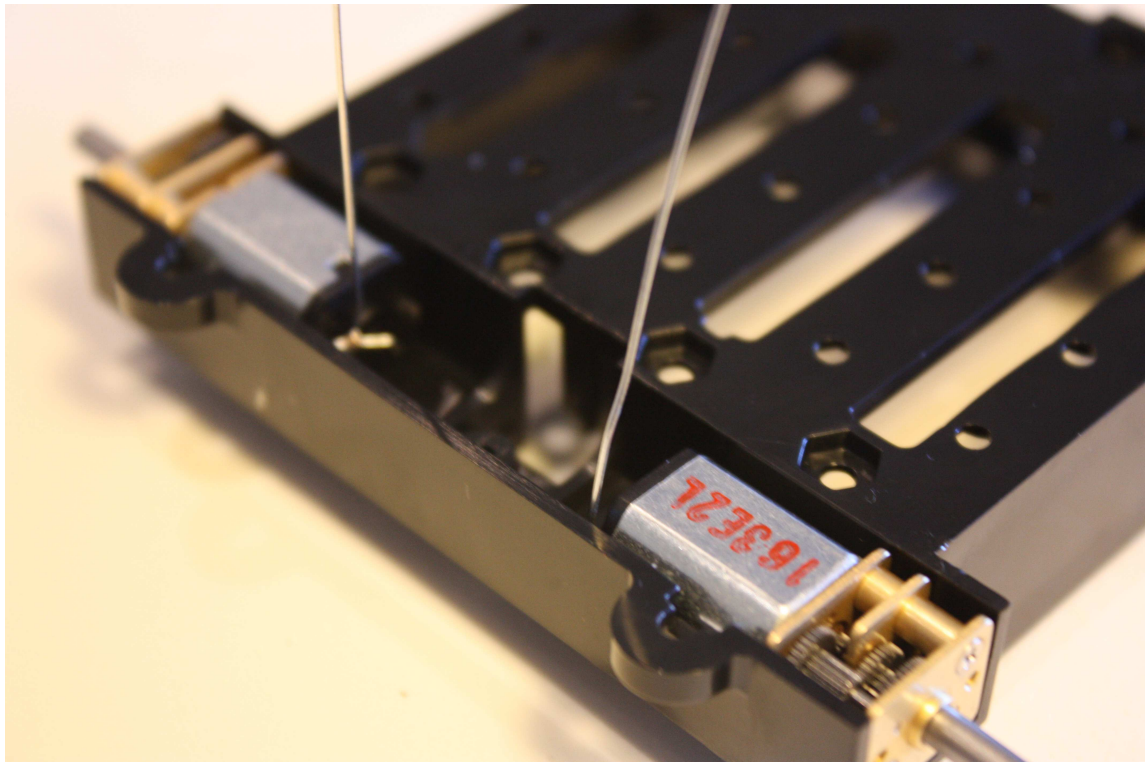


Figura 16: Cal vigilar la posició dels motors

- Agafem les dues plaques del xassís i les col·loquem al damunt dels motors, exactament com es mostra a la figura. COMPTE AMB LA COL·LOCACIÓ! Només hi ha una forma correcta de colocar-les i que encaixin bé amb la placa de motors.



Figura 17: Les plaques només tenen una orientació correcta

- Col·loquem la placa de motors al cim i collem els dos cargols de darrere. La femella s'ha de col·locar per dins, a la zona de les bateries. Per això ens pot anar bé aguantar-les amb cinta adhesiva.

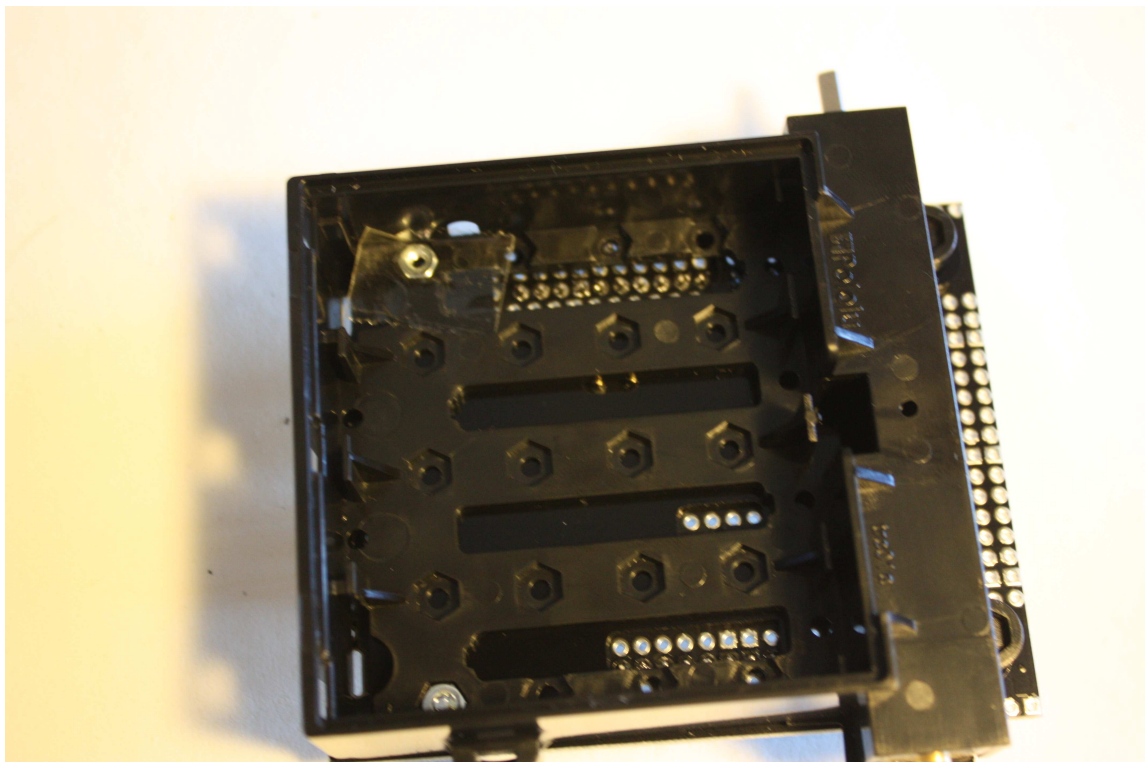


Figura 18: Posem cel·lo per aguantar les femelles

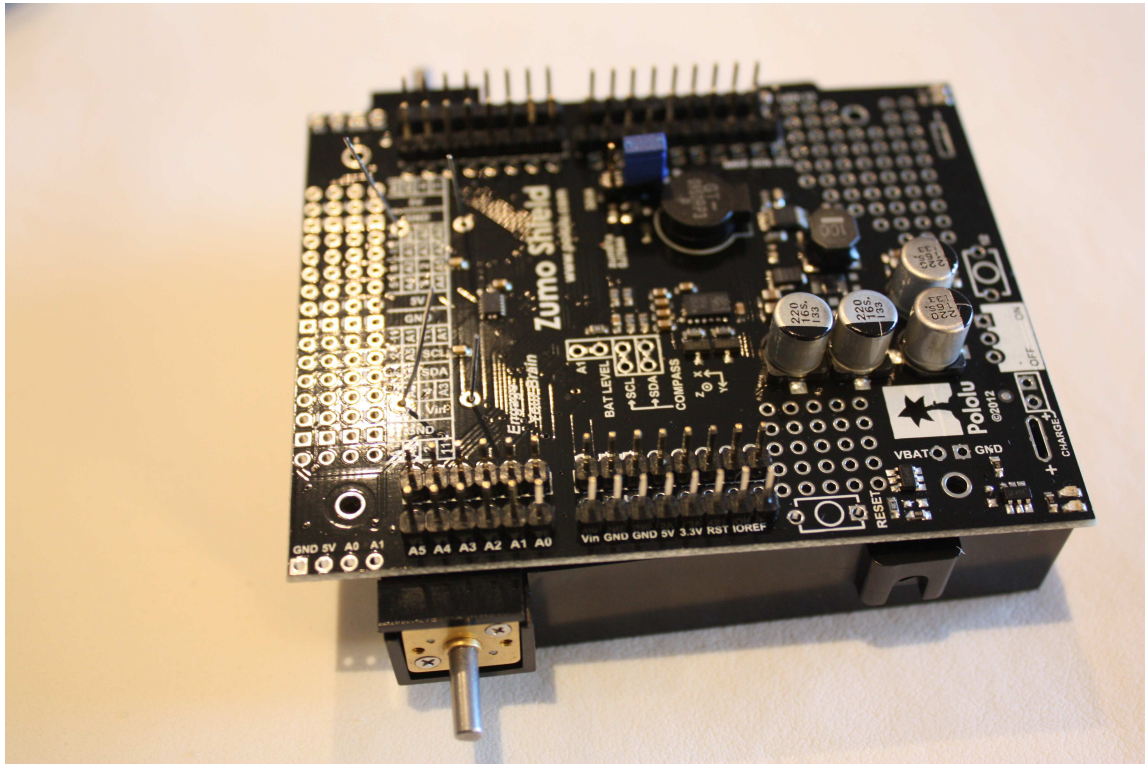


Figura 19: Placa de motors finalitzada

- Tallem els fils dels motors que sobresurten i els soldem.

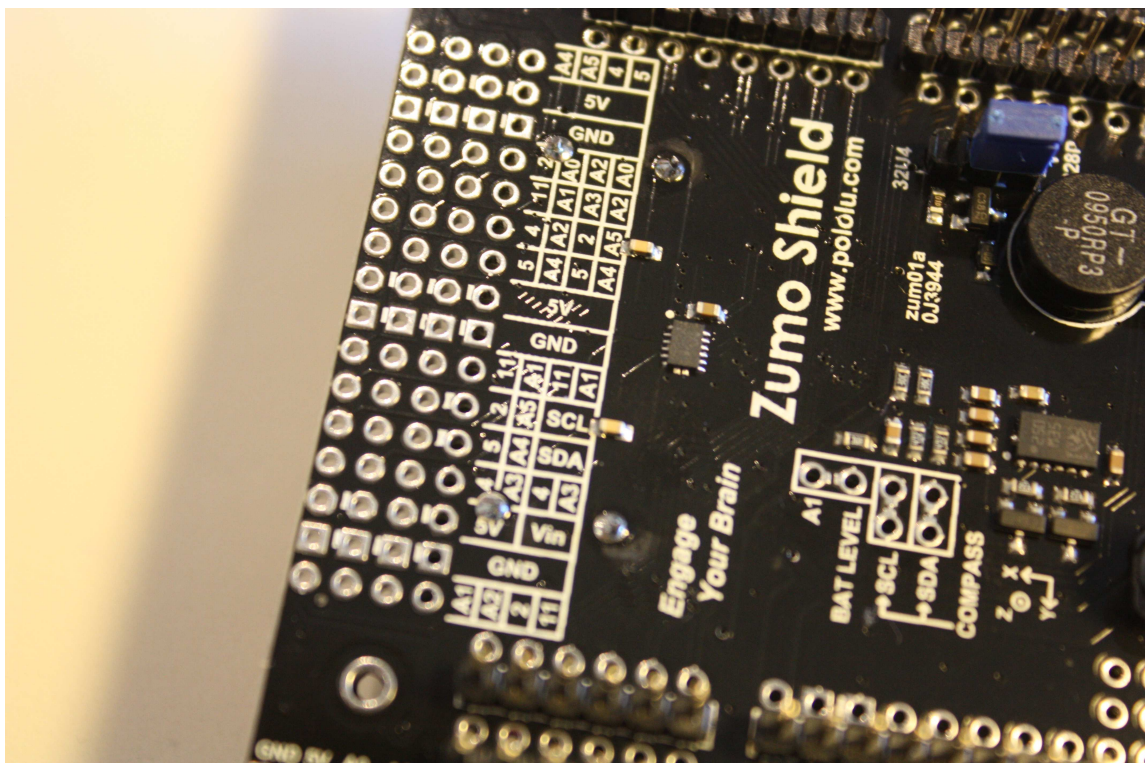


Figura 20: Tallem i soldem els fils dels motors

- Donem la volta al xassís i col·loquem els suports per les piles. Els dos que estan marcats a la figura sobresurten per sobre la placa de motors i s'hauran de soldar.

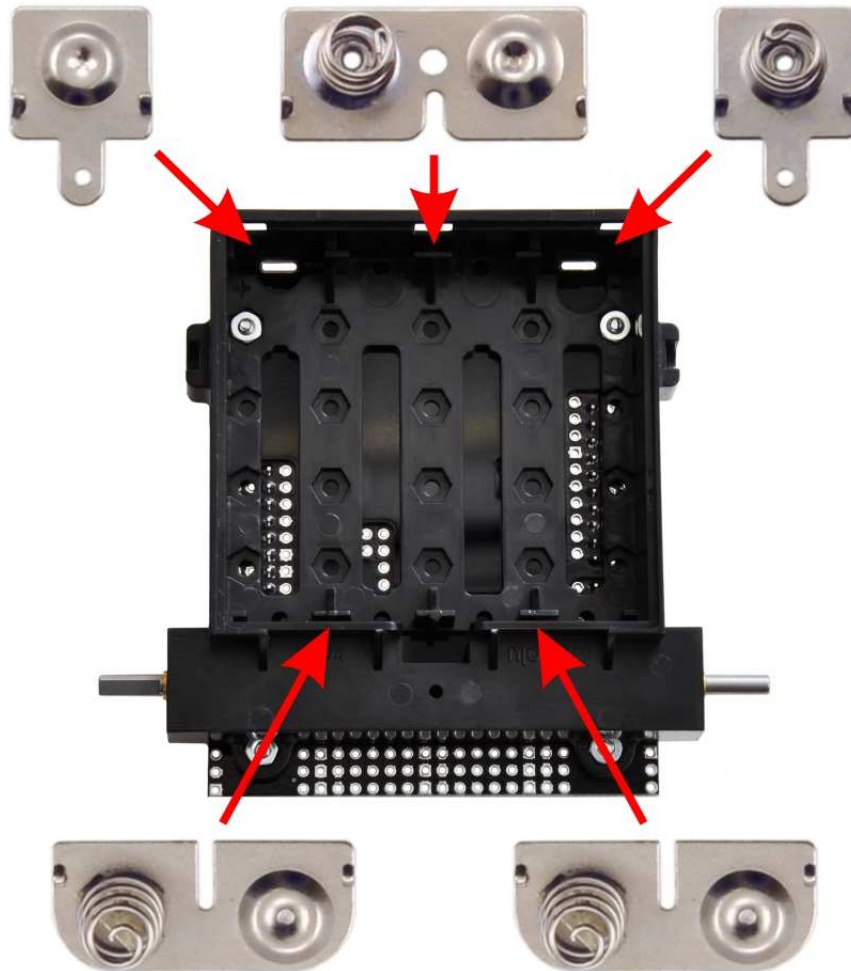


Figura 21: Col·locació dels suports

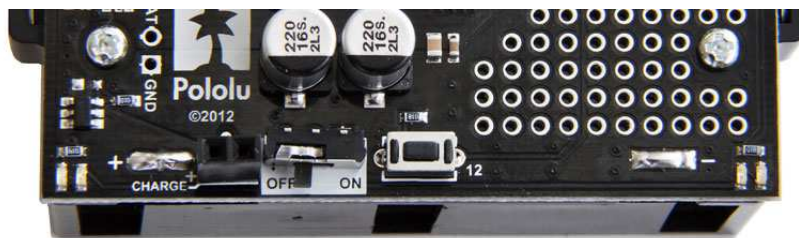


Figura 22: Cal soldar els extrems que sobresurten

- Ara hem de soldar el sensor de reflexió, que ens ajudarà a detectar els límits del ring. Comencem per soldar els pins mascle al sensor (ens podem servir de la protoboard per aguantar el sensor)

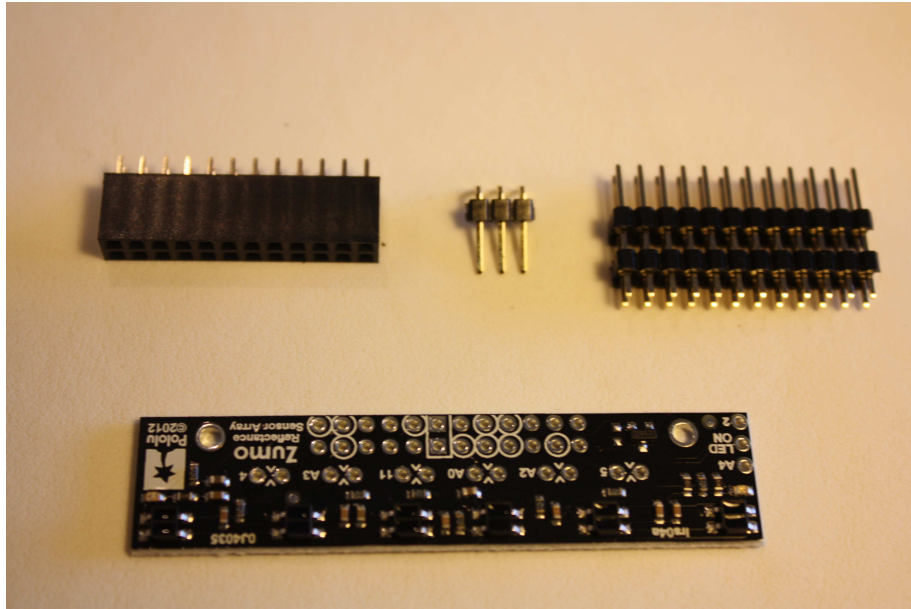


Figura 23: Material pel sensor

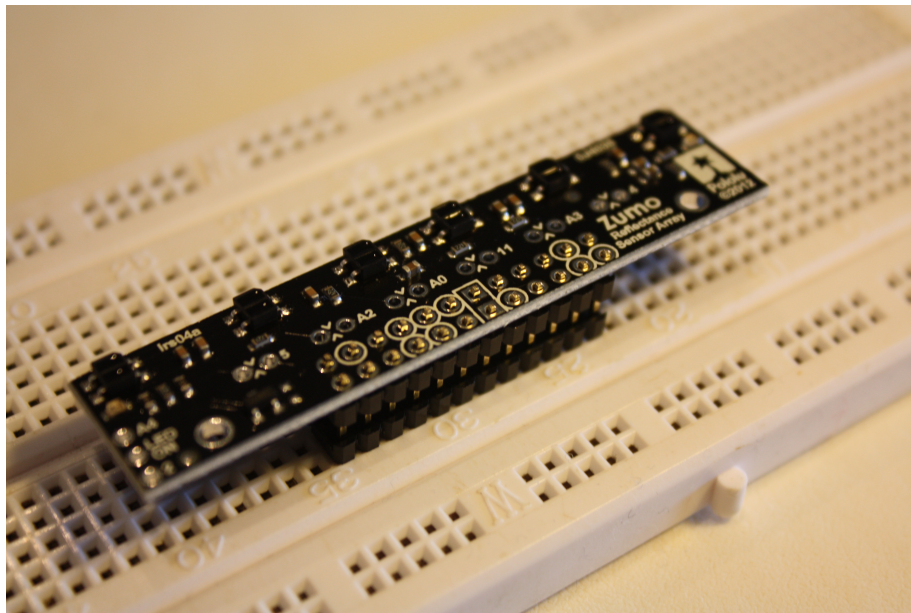


Figura 24: La protoboard ens ajudarà a soldar

- Soldem els pins femella per sota la part frontal de la placa de motors. **ATENCIÓ!** Fixeu-vos en quins pins ho soldeu, és molt fàcil equivocar-se.

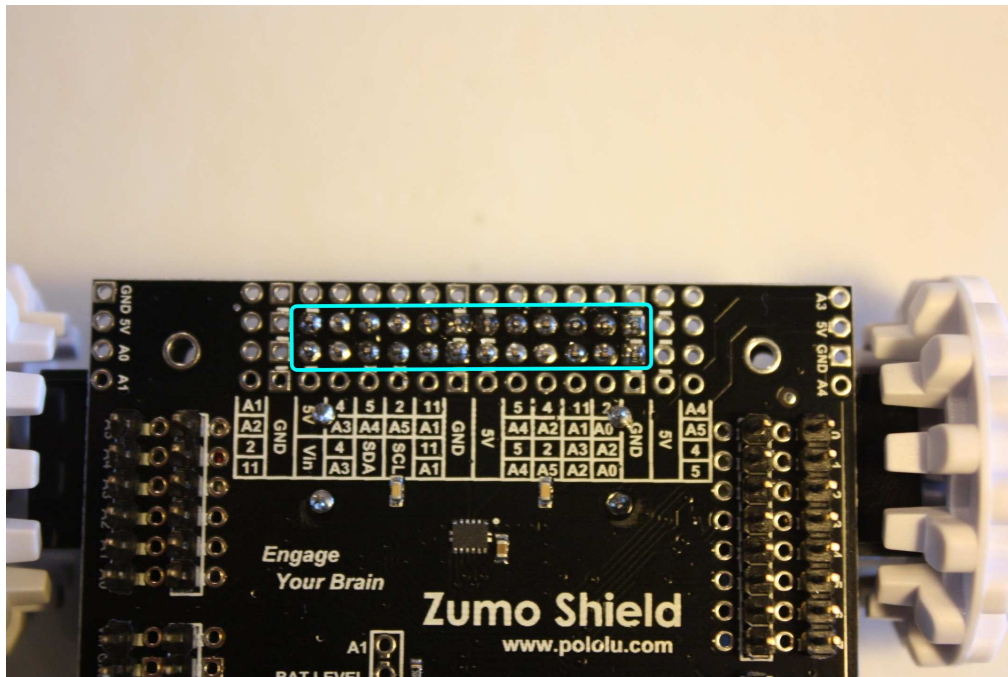


Figura 25: Compte amb els pins!

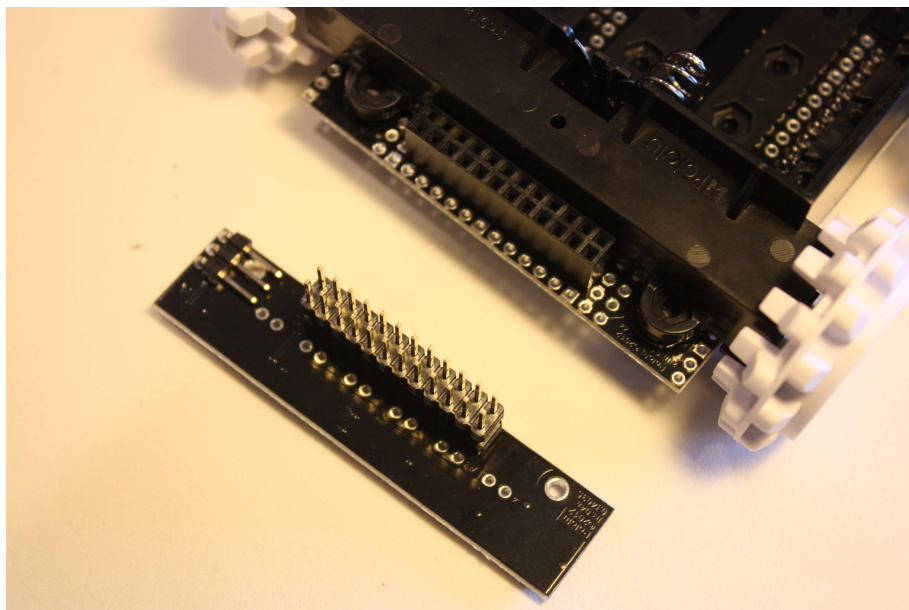


Figura 26: Col·loquem el sensor

- Col·loquem els suports blancs per les rodes (compte amb la forma del forat!) junt amb les orugues negres. Finalment cargolem la pala a la part frontal del xassís (les femelles van a la part inferior).

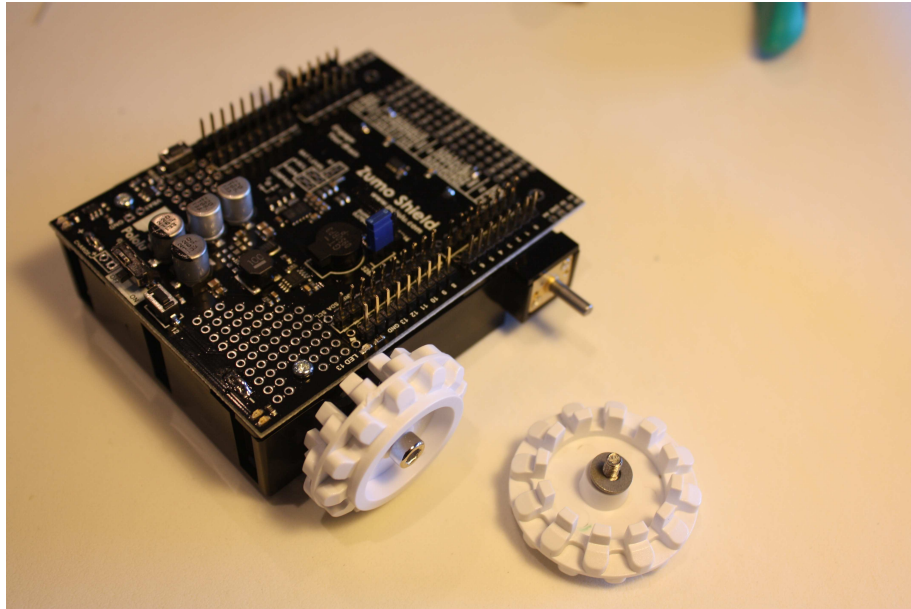


Figura 27: Cargolem els suports per les rodes

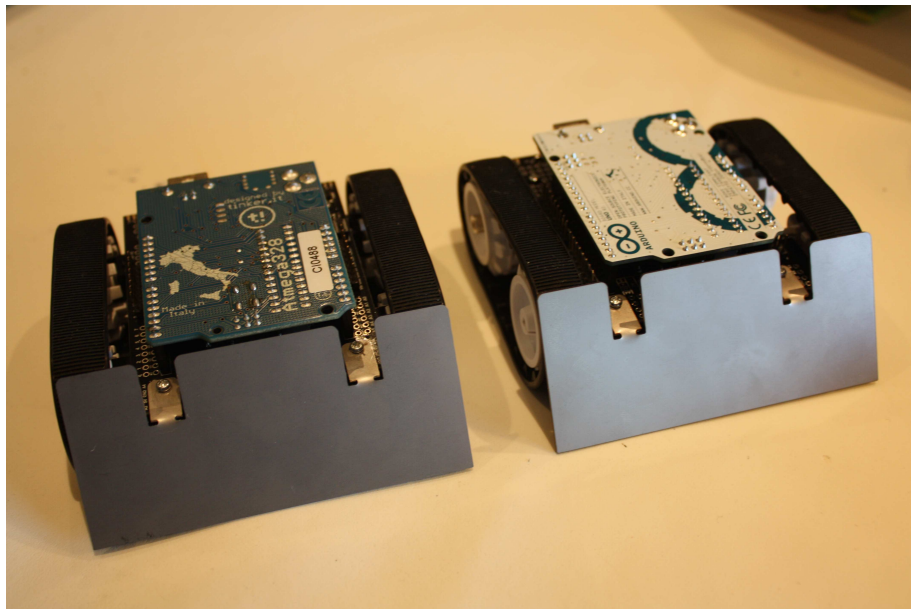


Figura 28: Finished! :3

6.2 Pins del zumo motor shield

Quan muntem el nostre robot no podrem fer servir tots els pins de l'Arduino ja que la majoria estan ocupats per la placa de motors i els sensors. Tot seguit s'especifica la tasca de cada pin i quins queden lliures per poder utilitzar.

Pin	Funció
Digital 0	RX (Comunicació Serial)
Digital 1	TX (Comunicació Serial)
Digital 2	SDA (Comunicació per I^2C)
Digital 3	Brunzidor
Digital 4	
Digital 5	
Digital 6	
Digital 7	Motors (Dreta)
Digital 8	Motors (Esquerra)
Digital 9	Motors (Generador de pulsos pel motor dret)
Digital 10	Motors (Generador de pulsos pel motor esquerra)
Digital 11	
Digital 12	Polsador
Digital 13	LED groc (però es pot fer servir per altres coses)
A0	
A1	
A2	
A3	
A4	SDA (Comunicació per I^2C)
A5	SCL (Comunicació per I^2C)

El sensor de color utilitza per defecte la majoria de pins que queden lliures pels seus sensors. No obstant, podrem desactivar-ne alguns per tal d'utilitzar els seus pins per altres funcions.

7 ¡Levántate y... ¡rueda?

Tot robot de sumo amb alguna aspiració d'èxit necessita moure's pel ring d'alguna forma per tal d'apretar els contrincants i esquivar possibles atacs. En aquest apartat es proposa un codi senzill per entendre totes les funcions per fer funcionar motors. El robot anirà endavant, endarrere, girarà a la dreta i a l'esquerra. Farem servir les llibreries `zumo` per la Shield, que s'han d'instal·lar al sistema. Us les passarem amb un pendrive, però les podeu baixar de: <https://github.com/pololu/zumo-shield> tal i com s'explica a l'apartat 1.6

7.1 Circuit

Simplement col·loquem la nostra Arduino al cim de la placa de motors. Recordeu que heu de tenir almenys quatre piles de tipus AA carregades

7.2 Codi

```
#include <ZumoMotors.h> //Importem la llibreria ZumoMotors

ZumoMotors motors; //Creem un objecte motors

void setup()
{
    //No hem d'inicialitzar res
}

void loop()
{
    motors.setSpeeds(200,200); //Posa els dos motors a la velocitat que li diem, en aquest cas 200 (esta
    delay(500);
    motors.setSpeeds(-200,-200); //La velocitat negativa inverteix el gir dels motors
    delay(500);
    motors.setSpeeds(0,0); //Podem parar els dos motors
    delay(100);
    motors.setLeftSpeed(200); //Posem el motor esquerra a 200
    delay(1000);
    motors.setRightSpeed(200); //Posem el motor dret a 200
    delay(1000);
}
```

En aquest codi s'introdueixen conceptes nous. Alguns són característiques avançades del món de la programació, així que de moment no ens cal comprendre com funcionen, sinó saber com s'han de fer servir en aquest cas pràctic.

```
#include <ZumoMotors.h>
```

és una llibreria. Una llibreria és com un catàleg de funcions que "anexionem" al codi que ja tenim, així podem fer-les servir en qualsevol moment.

`ZumoMotors`

serveix per crear un objecte `ZumoMotors` anomenat `motors`. En aquest curset no parlarem d'objectes i classes, simplement hem de saber que per fer funcionar el codi és necessari posar això al principi. Però si teniu curiositat, us posaré un exemple conceptual de què és una classe. Heu jugat mai a videojocs de rol? Una classe de programació és similar a una classe d'un joc. Creem la classe `Mag` que té tot un seguit de funcions: `tirarBolaDeFoc()`, `noPotsPassar()`, etc. També pot tenir variables per guardar la vida i l'armadura. Ara podem crear un objecte (una persona, en aquest cas) que es dirà `Gandalf` i serà de classe `Mag`, de manera que tindrà les funcions que abans hem dit. Podem crear infinitud de Mags amb noms diferents, i cada un tindrà les seves funcions i el seu valor de vida i armadura.

Per tant, el que fem amb això és crear un objecte de la classe `ZumoMotors` que anomenarem `motors`, i després dins el `loop` activem les funcions que té dins seu la classe `motors`. Però si això us sembla confús no us preocupeu, només us cal saber que heu d'escriure `ZumoMotors motors;` al principi de tot.

Ara col·loquem les piles a la part de baix del robot, tanquem bé la tapa (ha de fer un *çlec!*”), posem el robot a terra i engeguem l'interruptor. El robot hauria d'anar endavant, endarrere i girar a dreta i esquerra.

8 "Yo... VIVO"! construcció d'un avoider

Amb tot el que hem après, ja estem preparats per construir un robot mínimament seriós. Farem un avoider, és a dir, un robot que es mourà en línia recta i quan trobi un obstacle girarà. També farem servir la llibreria Pushbutton.h per tal de fer servir el pulsador i activar el robot passat un cert temps.

8.1 Circuit avoider

Necessitem col·locar la placa Arduino al cim de la Shield de motors. Després enganxem un sensor infraroig al cim del nostre robot (podem fer servir cinta adhesiva de doble cara, o brides), connectem el cable de senyal al pin analògic 3 i el vermell i negre a 5V i GND respectivament.

8.2 Codi avoider

```
#include <ZumoMotors.h>
#include <Pushbutton.h>

ZumoMotors motors;
Pushbutton button(ZUMO_BUTTON); //A l'igual que amb els motors, creem un objecte button

int val = 0; //Aquí guardarem la lectura del sensor
int vel = 200; //Variable per establir la velocitat. 200 es una velocitat estandar.

void setup()
{
    while(not button.isPressed()); //Mentre no es premi el boto, no es continuara el programa
    delay(3000); //Esperem tres segons
}

void loop()
{
    val = analogRead(3); //Llegim el pin analògic 3
    if(val > 450) //Si la lectura es superior a 450 (
    {
        motors.setSpeeds(-vel, -vel);
        delay(500);
        motors.setSpeeds(-vel, vel);
        delay(200);
    }
    else
    {
        motors.setSpeeds(vel, vel);
    }
}
```

Poseu el robot al terra, prepareu-li uns quants obstacles, premeu el botó i... bé, anava a dir que gaudiu de l'espectacle, però millor aparteu-vos.

Si voleu, podeu fer unes quantes variacions:

- Afegiu el segon sensor al pin A2. Col·loqueu un dels sensors que apunti a davant del robot, inclinat cap a la dreta i l'altre cap a l'esquerra. Modifiqueu el codi de manera que si el sensor de la dreta detecta un obstacle, giri a l'esquerra, i si és el de l'esquerra que detecta un obstacle, que giri a la dreta.
- Modifiqueu el codi perquè el robot vagi donant voltes sobre si mateix i, si detecta un objecte, avanci en línia recta cap a ell.

9 Sensor de color

Finalment, només ens queda aprendre a configurar el sensor de color. El sensor de color consta en realitat de diversos sensors infrarojos que mesuren la reflectància del terra. Els colors més foscos, com el negre, tenen menys reflectància que els clars. Per defecte, els sensors de reflectància ocupen tots els pins que ens queden lliures de la placa de motors:



Figura 29: pins sensor de color

No obstant, per detectar línies n'hi ha prou amb els dos sensors de les puntes, així que podem desactivar els del mig:



Figura 30: pins sensor de color remap

Per exemplificar-ho, crearem un robot que es mourà si el terra és negre i girarà si és blanc. Si el ring té els colors al revés, haureu de canviar la funció `linia()`.

9.1 Circuit sensor de colors

Simplement col·loqueu l'Arduino al cim de la shield i connecteu l'array de sensors infrarojos.

9.2 Codi sensor de colors

```
#include <ZumoMotors.h>
#include <Pushbutton.h>
#include <QTRSensors.h>
#include <ZumoReflectanceSensorArray.h>

byte pins[] = {4, A3, A2, 5};
ZumoReflectanceSensorArray sensors(QTR_NO_EMITTER_PIN);
```

```

unsigned int sensor_values[4];

ZumoMotors motors;

int vel = 200;

void setup()
{
  sensors.init(pins, 4);
  Serial.begin(9600);
}

void loop()
{
  if(detectaLinia())
  {
    motors.setSpeeds(-vel, -vel);
    delay(300);
    motors.setSpeeds(vel, -vel);
    delay(400);
  }
  else
  {
    motors.setSpeeds(vel, vel);
  }
}

bool detectaLinia()
{
  sensors.read(sensor_values); //Llegim els sensors

  bool hiHaLinia = false; //Variable booleana

  int i = 0;

  //Llegim els quatre sensors i, si almenys un marca linia,
  //posem la variable hiHaLinia a true
  while(i < 4)
  {
    //A l'igual que els SHARP, aquests sensors de color
    //donen una intensitat
    if(sensor_values[i] > 500)
    {
      hiHaLinia = true;
      break;
    }
    i++;
  }
  return hiHaLinia;
}

```

Afegint només una petita variació en el codi, ja tindreu el robot de sumo més simple que es pot fer:

- Connecteu el sensor de distància de tal forma que el robot sempre eviti la línia exterior fins que no detecti un objecte.

10 Normes del minisumo

Aquí es recullen les normes completes del nostre combat de minisumo.

10.1 Els Robots

- Els robots han de pesar, com a molt, 600 grams amb les piles/bateries incloses. No es permet augmentar el seu pes durant el combat.
- Els robots han de medir, a l'inici del combat, 10x10 centímetres com a molt. No hi ha límit a l'altura. Poden augmentar el seu tamany durant el combat.
- Els robots no poden danyar ni malmetre intencionadament els contrincants. No es considera dany apretar el contrincant.
- Els robots no poden deixar anar líquids, fum, foc ni cap substància.
- Per tal de considerar-se aptes, els robots han de poder tirar un dummy (una capsa) fora del ring en menys de 2 minuts.

10.2 El Ring

- El ring serà de color negre amb una vora de 5 centímetres blanca.
- Al centre del ring hi haurà dues línies grises.
- El ring tindrà una altura d'entre 2 i 3 centímetres.

10.3 El Combat

- Un combat consta de 3 rondes. El vencedor del combat és el que guanyi 2 de les 3 rondes.
- L'objectiu dels robots és tirar el contrari fora del ring. Un robot perd quan toca el terra fora del ring.
- A l'inici de la ronda s'han de col·locar els dos robots darrere de les ratlles grises, en qualsevol orientació.
- Els robots s'han de començar a moure tres segons després de que l'àrbitre doni la senyal. Si un robot es mou abans d'hora, es tornarà a donar la sortida.
- A la primera ronda es decideix a sorts qui comença col·locant el robot. A les següents serà el guanyador de la ronda anterior.
- En cas que un combat s'allargui més de 4 minuts, es considerarà empat.

10.4 El Torneig

- El torneig final serà una eliminatòria.
- Els equips combatran en rondes 1 vs 1. Els robots que perdin quedaran eliminats, i els guanyadors passaran a la següent ronda.
- El robot que guanyi l'última ronda, la final, serà el vencedor del torneig

11 Consells a l'hora del torneig

- Tingueu les piles carregades, i diversos recanvis.
- Abans de començar, comproveu que el robot detecti bé els colors del ring.
- Netegeu les orugues després de cada combat. El millor és fer-ho amb un cotó i alcohol, però també funciona amb aigua amb cura per no mullar els components elèctrics.
- Tingueu ben subjectats tots els sensors i cables. Poseu una mica de cinta aïllant a la tapa de les piles.
- Comproveu en tot moment el pes del robot.