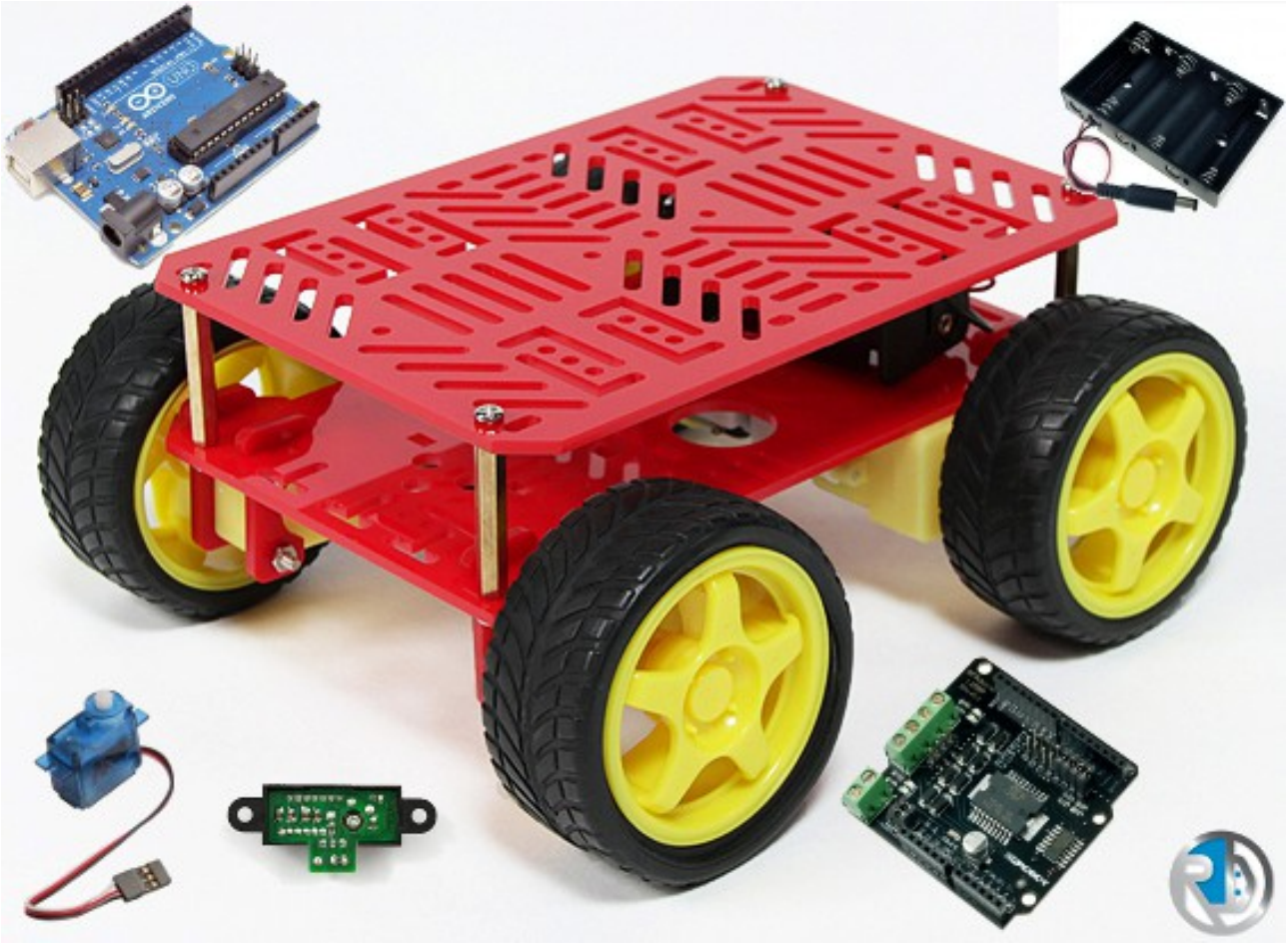
Robot Beginner Kit 4WD



- Lista dei componenti

Controlla che la lista dei componenti corrisponda al kit:

- Arduino Uno Rev3
- Dagu 4WD Magician Chassis
- 2A Motor Shield
- Infrared Proximity Sensor – Sharp GP2Y0A21YK
- Sub Micro Servo
- JST Jumper 3 Wire Assembly
- 10 sets M3 * 6 nylon screws
- 10 jumper M/F
- 10 jumper M/M

- Lo chassis

Come primo step, andremo ad assemblare lo chassis.

Questa parte non verrà trattate poichè è presente nella busta una dettagliata descrizione per l'assemblaggio

(la parte superiore dello chassis andrà avvitata per ultima altrimenti non avremo modo di posizionare Arduino e gli altri accessori).

Aggiungete i distanziatori in nylon, 2 quantità per ogni lato.

- I motori

In questo step procederemo con la saldatura dei motori. È da considerare la parte più "complessa" della guida dato che si andrà a determinare il movimento totale del nostro robot.

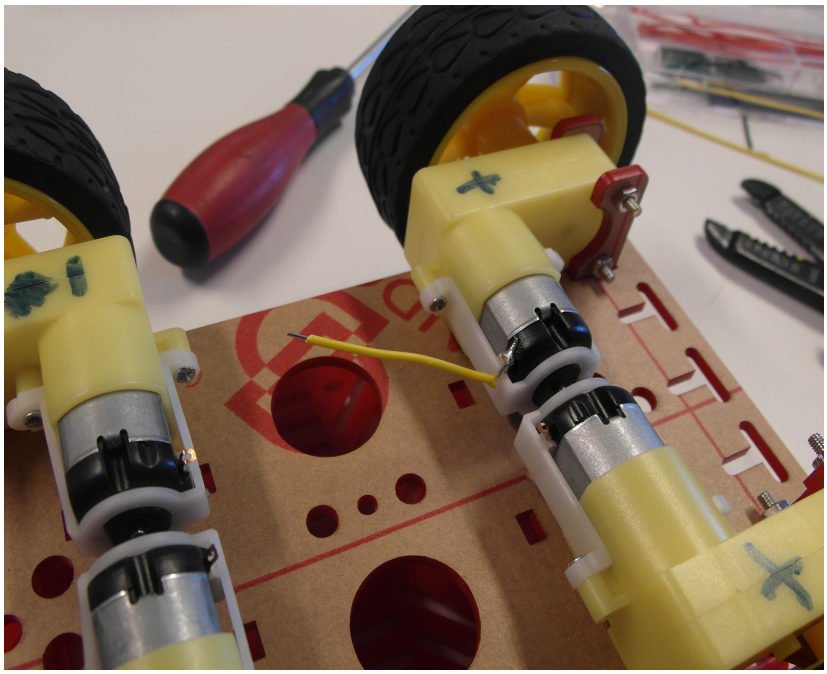
Innanzitutto bisogna identificare il senso di rotazione dei motori (ciò andrà a determinare la parte anteriore e posteriore dello chassis). Questo servirà a noi per avere maggiore chiarezza sulla saldatura che andremo ad effettuare fra poco. Se non avete un'alimentatore regolabile in casa, potete utilizzare l'uscita regolata di Arduino (5V). Quindi collegate 2 ponticelli a 5V e GND di Arduino e appoggiarli sui poli dei motori tenendo d'occhio la rotazione di quest'ultimo.

NB: (questo procedimento va fatto per tutti e 4 i motori).

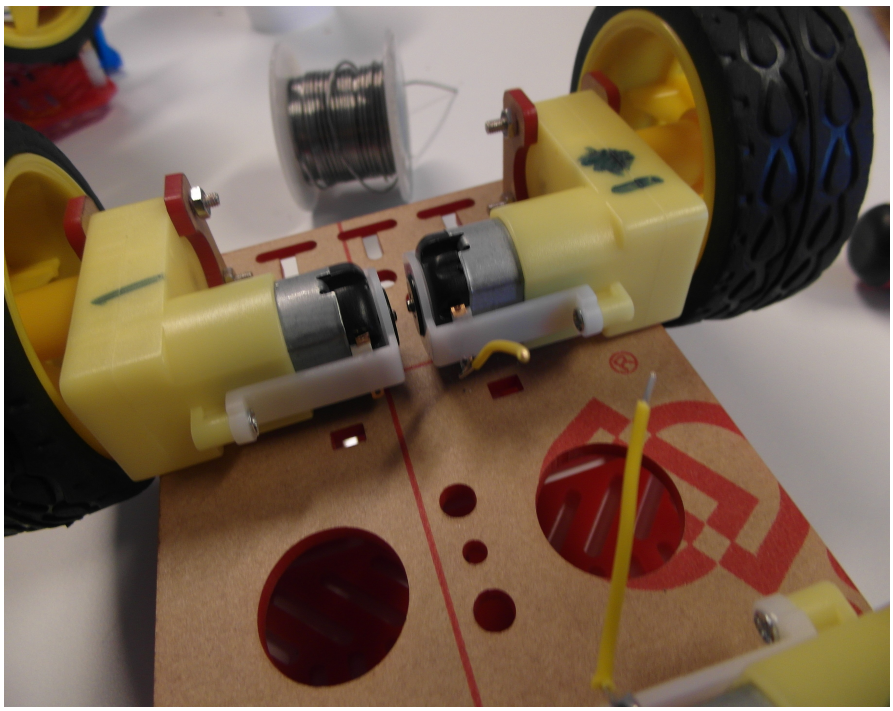
Li contrassegneremo con "+" o "-".

ecco le varie fasi della saldatura dei motori:

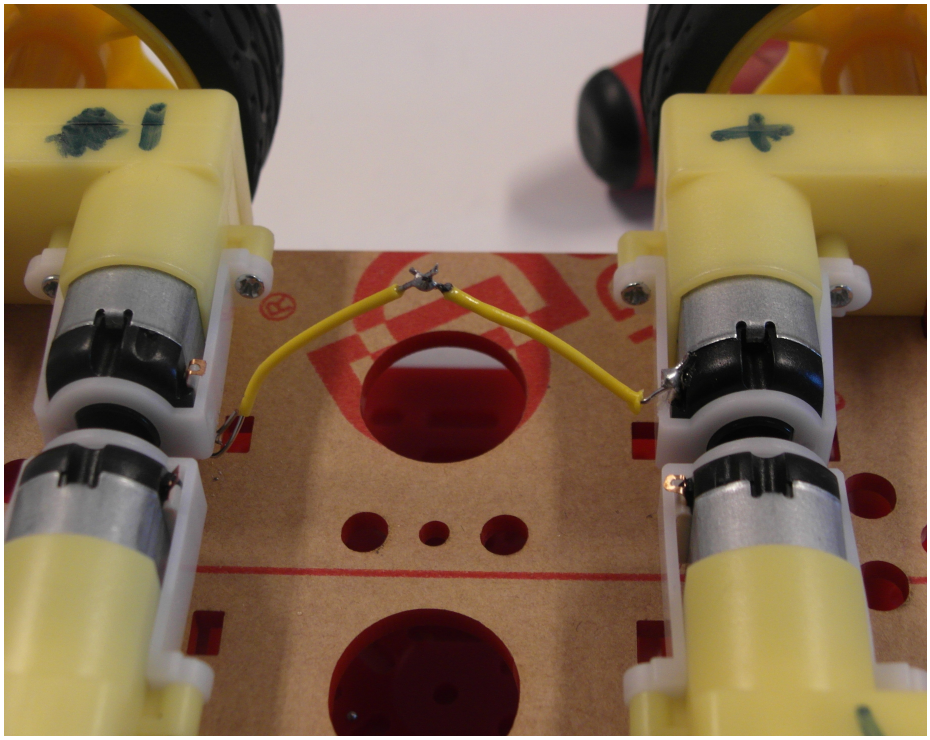
1.



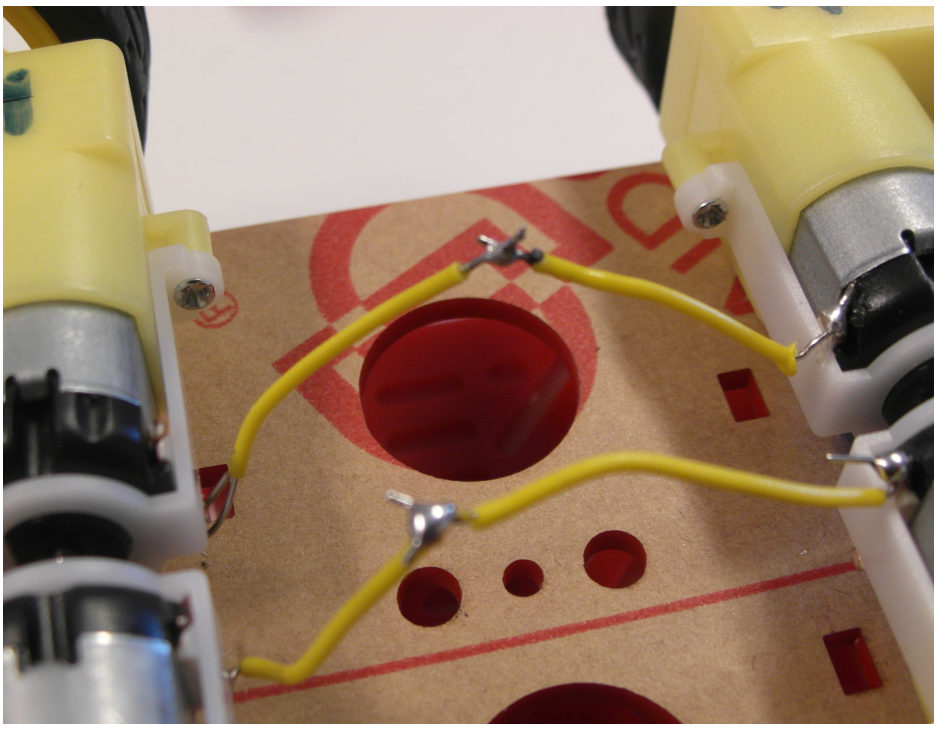
2.



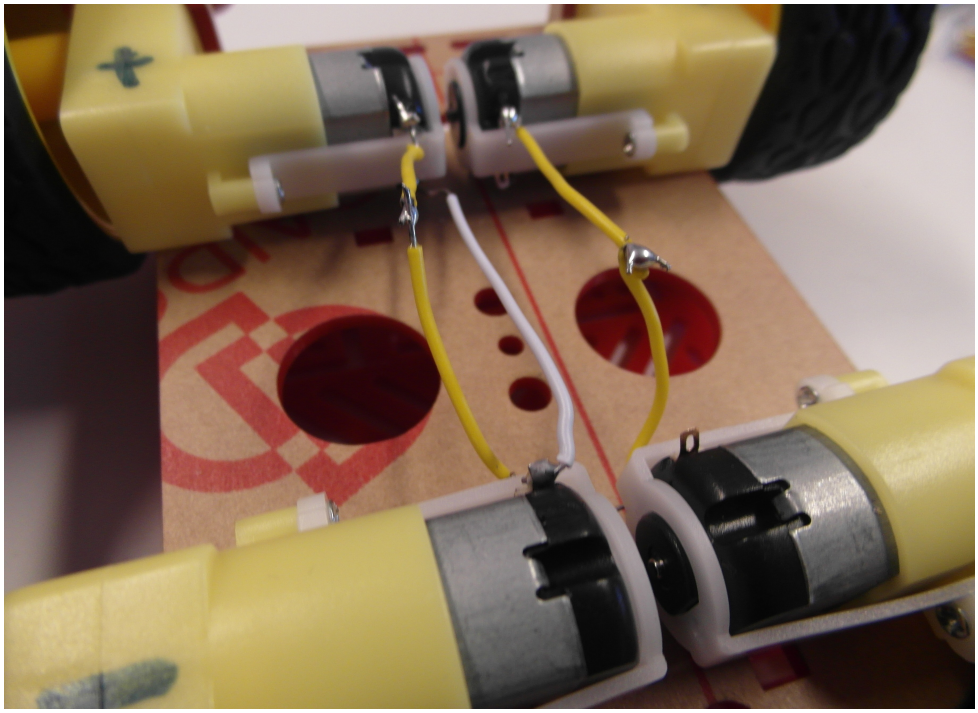
3.



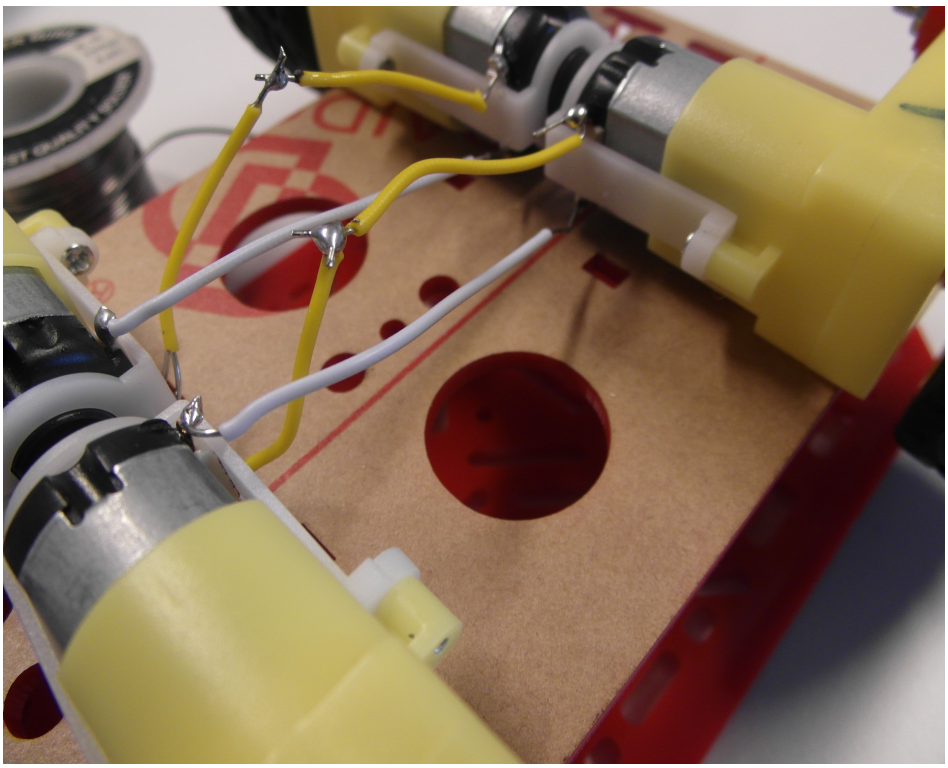
4.



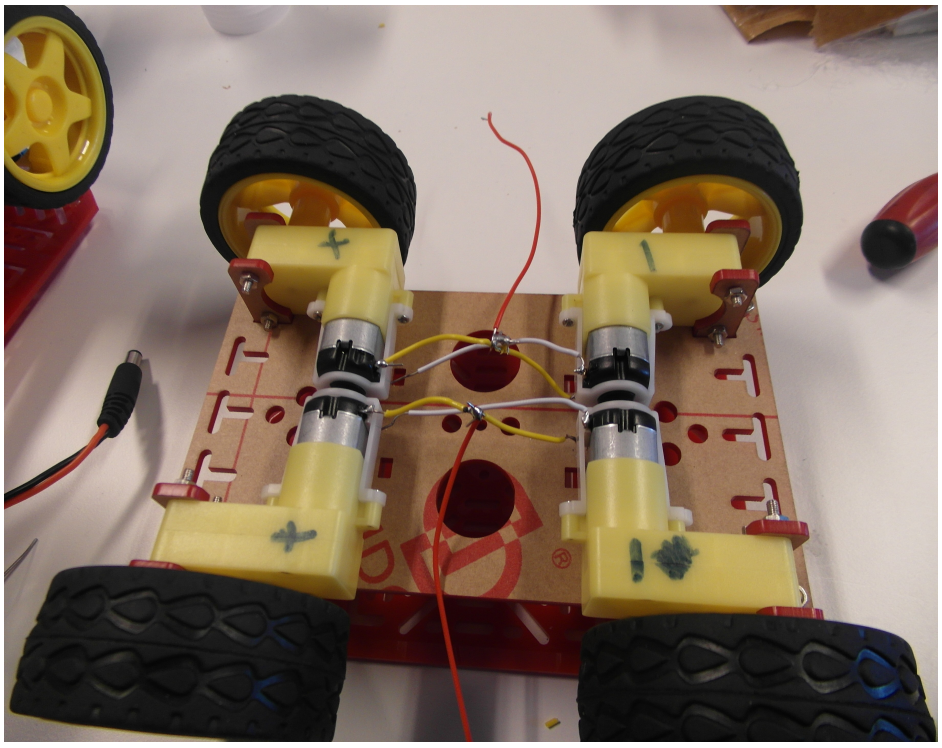
5.



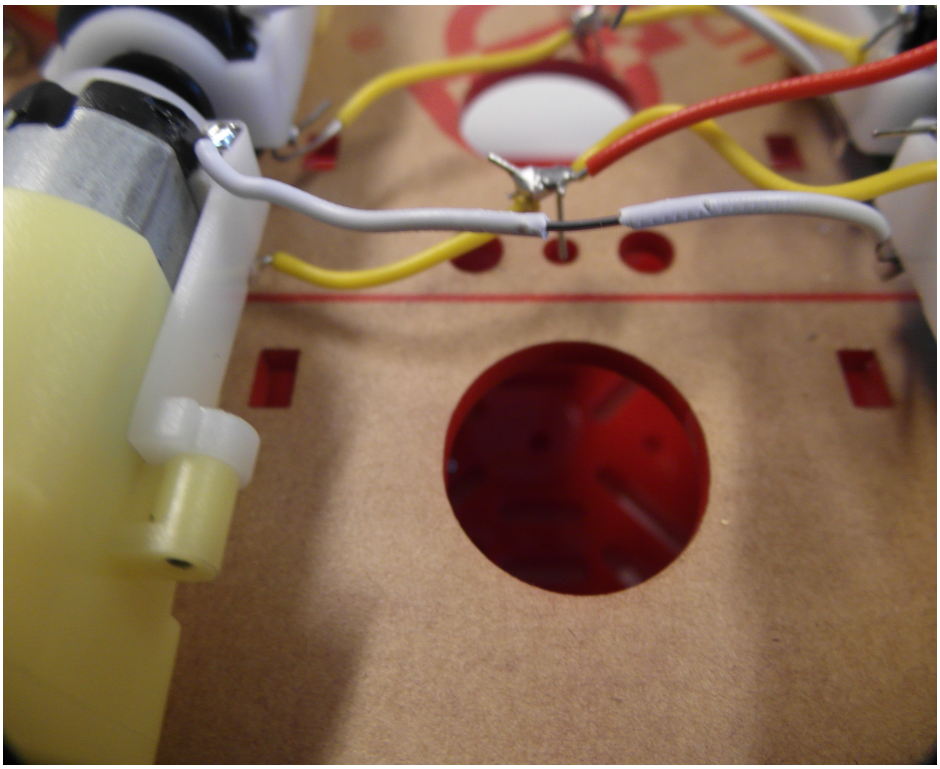
6.



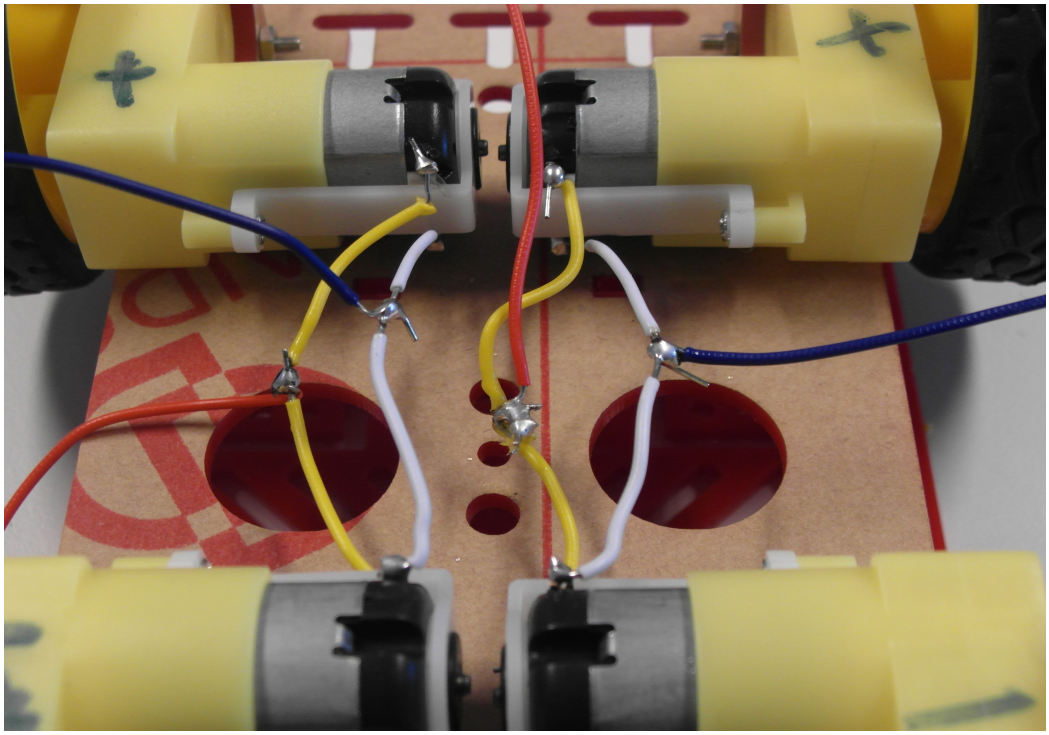
7.

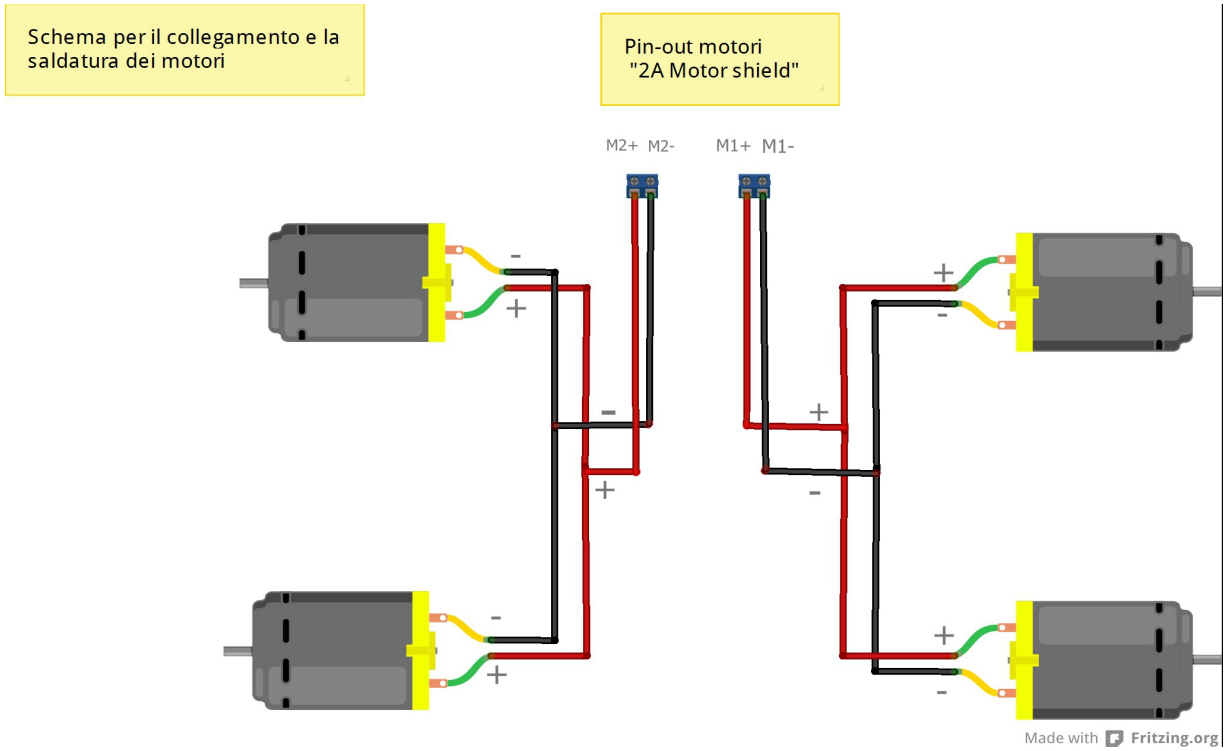


8.



9.

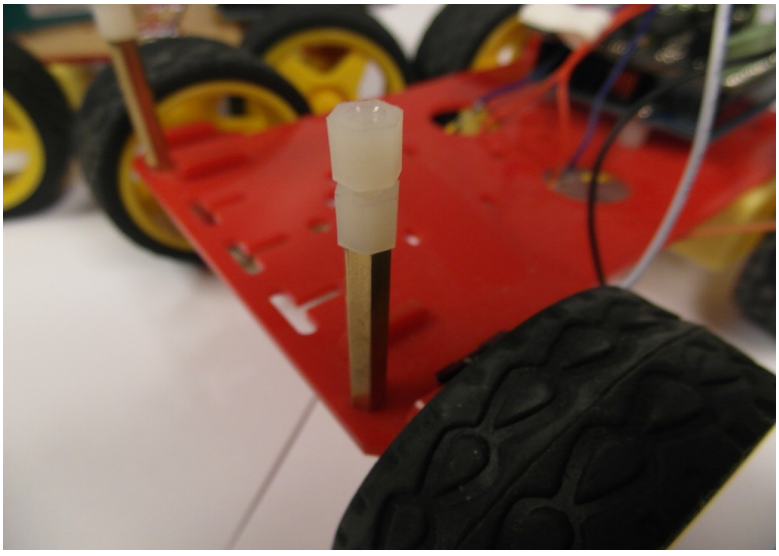




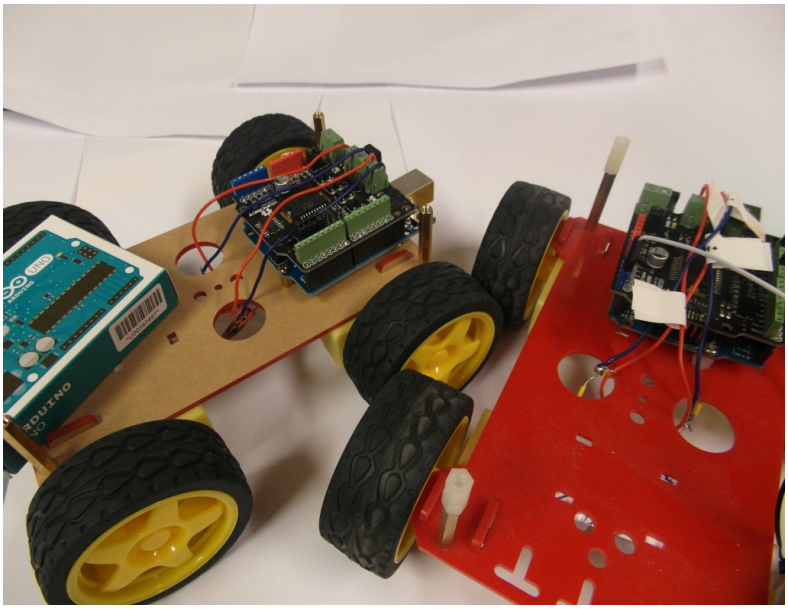
Schema in Fritzing

- Arduino sullo chassis

Completata la parte più delicata,andiamo a posizionare Arduino e Motor shield, fissandoli con viti e distanziatori sullo chassis. Cercate di avvitarli per bene in modo che, in caso di urti, non cadano dalla piattaforma.



Il risultato sarà il seguente:



- Collegamenti

Una volta estesi i jumper, li colleghiamo ai morsetti della motor shield, (vedi step "I motori").

Anteriore

M2+ M2- | M1+ M1-

|

red black | red black

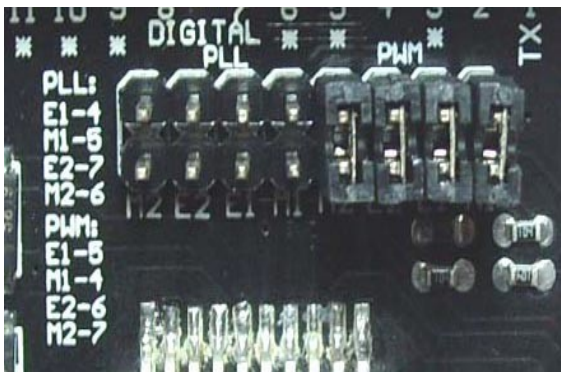
|

Posteriore

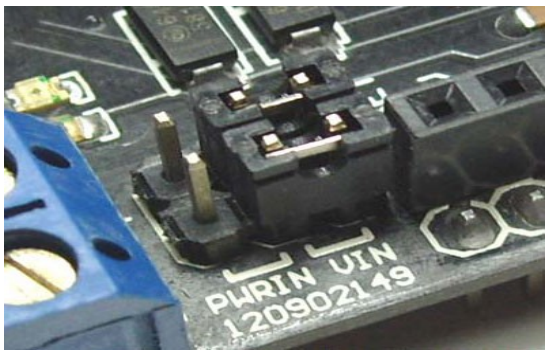
NB: utilizzando sempre l'uscita regolata di Arduino (5V), provate a fare il test precedente per controllare se le saldature sono ok.

Nota:

La motor shield dovrà essere configurata in questo modo:



I pin dovranno essere in modalità PWM



La motor shield riceverà l'alimentazione da Arduino

ora ci occupiamo a collegare tutto il resto!

- Posiziona il servo

I servi standard consentono all'albero di ruotare solitamente da 0° a 180°.

Il servo ha tre fili: alimentazione, massa e segnale. Il cavo di alimentazione è tipicamente rosso, e deve essere collegato al pin 5V sulla scheda Arduino. Il cavo di terra è in genere nero o marrone e deve essere collegato a al pin GND sulla scheda Arduino. Il pin segnale è tipicamente giallo, arancione o bianco e deve essere collegato ad un pin digitale sulla scheda Arduino.

Per poter controllare il servo utilizzeremo una libreria standard di Arduino (Servo.h)

La libreria Servo utilizza i seguenti metodi:

- **attach**()
- **write**()
- **writeMicroseconds**()
- **read**()
- **attached**()
- **detached**()

quelli che andremo ad utilizzare sono fondamentalmente 2: attach() e write()

>Attach()

Descrizione

Fissa la variabile Servo ad un pin digitale

Sintassi

servo.attach(pin)

Parametri

servo: una variabile di tipo Servo

pin: il pin dove è collegato il servo

>Write()

Descrizione

Scrive un valore al servo, controllandone l'albero. Su un servo standard, questo imposterà l'angolo dell'albero (in gradi), spostando l'albero a tale orientamento. Su un servo a rotazione continua, questo imposterà la velocità del servo (con 0 che piena velocità in una direzione, 180 essendo piena velocità nell'altro, ed un valore vicino a 90 essendo nessun movimento).

Sintassi

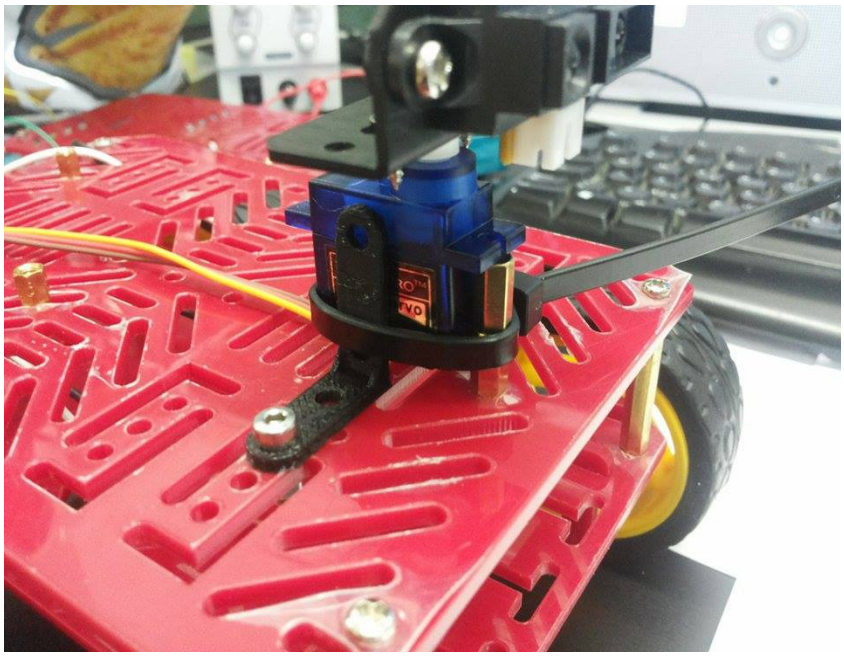
servo.write(angle)

Parametri

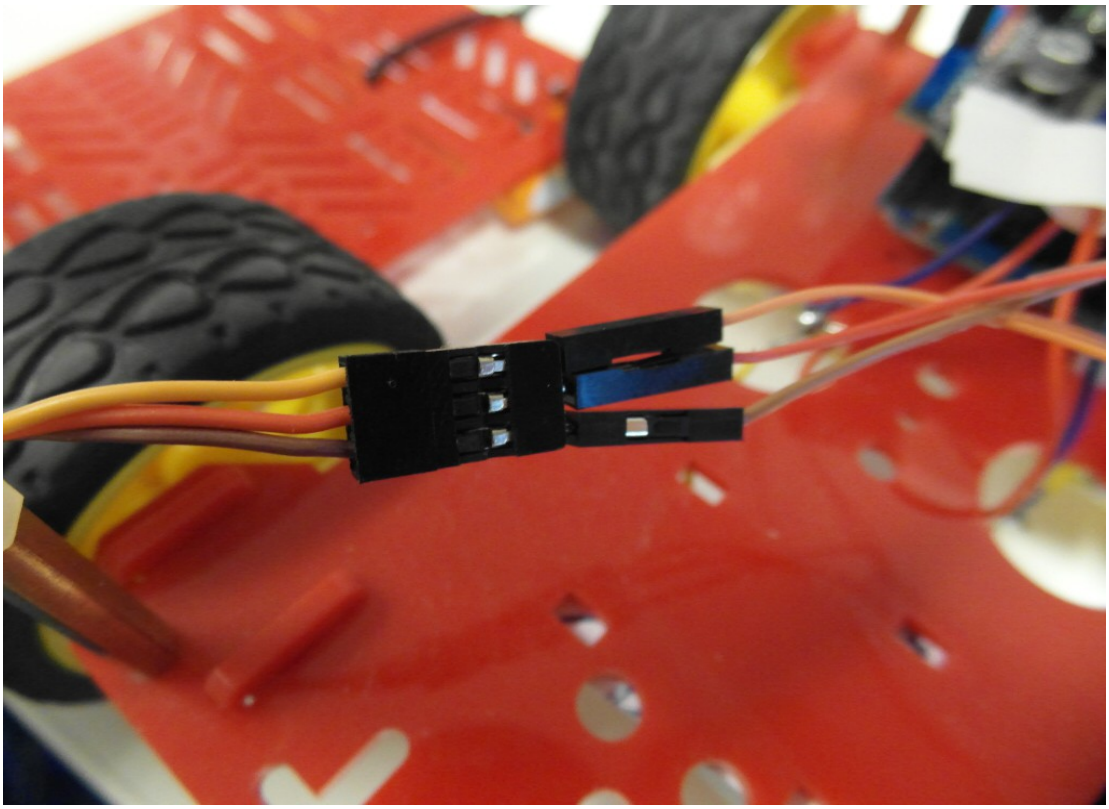
servo: una variabile di tipo Servo

angle: il valore dell'angolo del servo, da 0° a 180°

Posizioniamo il servo con le relative staffe sullo chassis in questo modo:



Collegate il servo ad Arduino con dei jumper:



- Posiziona il sensore

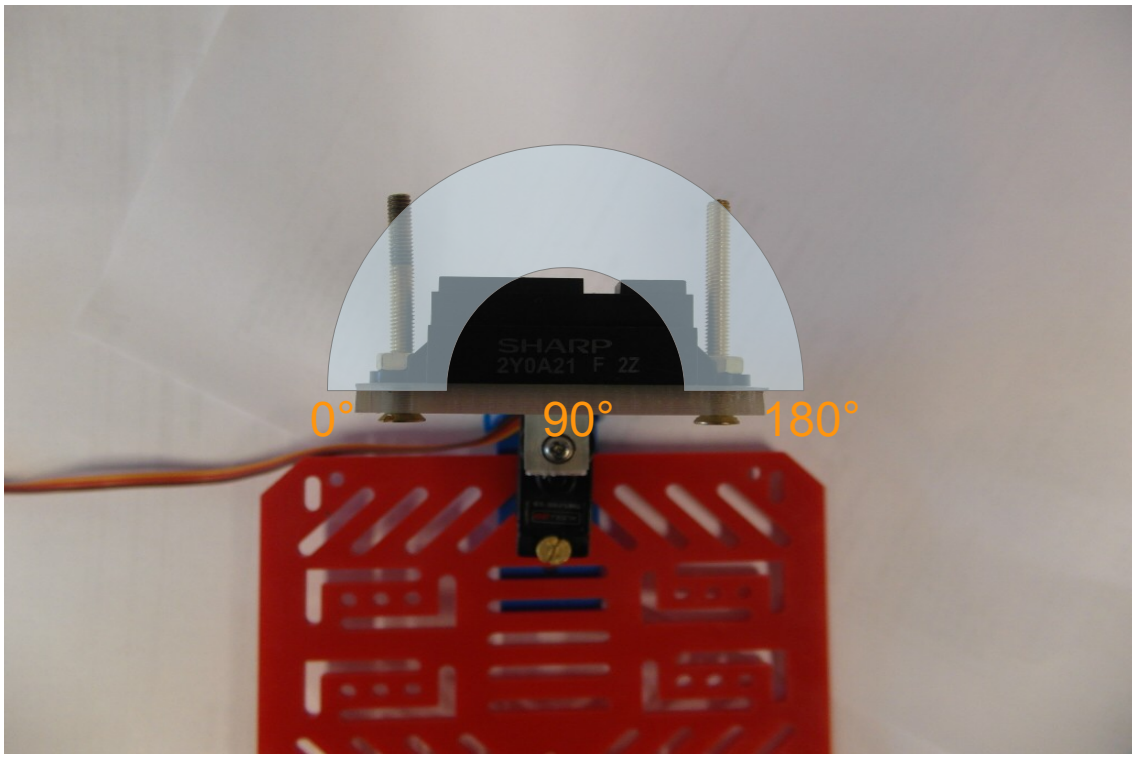
Lo SHARP GP2Y0A21YK è un sensore di prossimità a raggi infrarossi. Ha un'uscita analogica che varia da 3.1V a 0.4V a 10cm a 80 cm. Emana un raggio IR da un LED, e un fototransistor misura l'intensità di luce rimbalzata. Se guardi la parte frontale del sensore, è possibile vedere uno dei LED luminosi leggermente rosso.

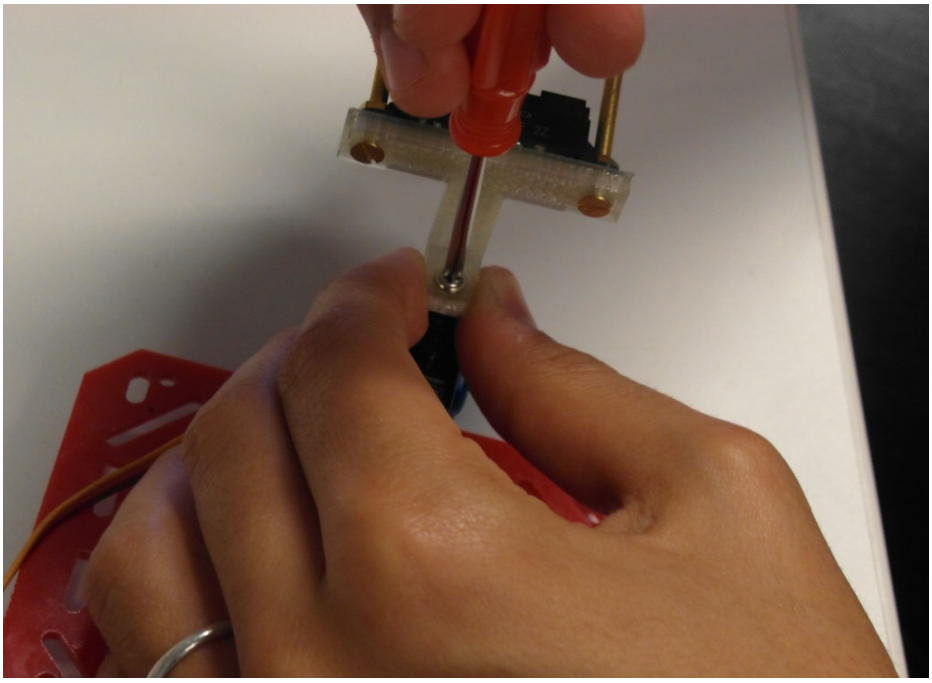
Codice d'esempio

```
int Pin_sensore = 5; //pin analogico 5

void setup(){
  Serial.begin(9600);
}

void loop(){
  int val = analogRead(Pin_sensore);
  Serial.println(val);


  delay(1000); // ritardo di 1s

}
```

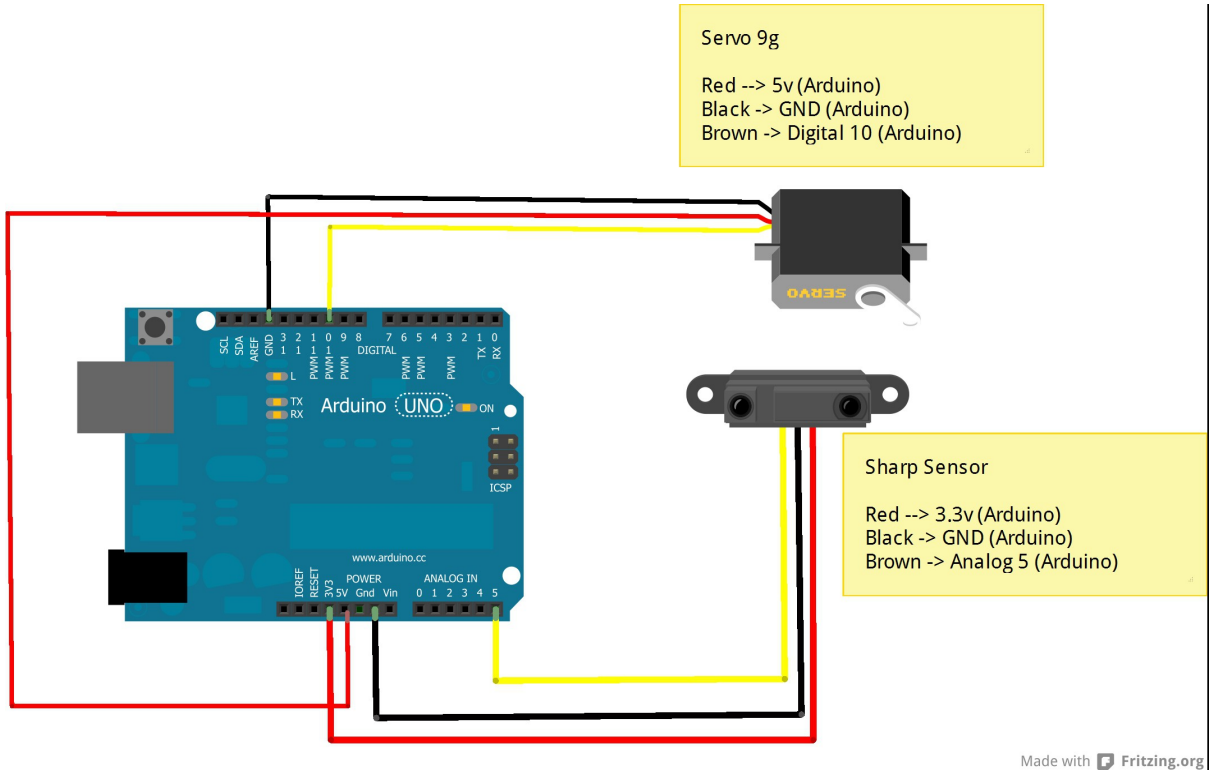
Calibrazione

dato che il servo potrà compiere un movimento di 180°, noi fisseremo la staffa (con il sensore) al punto medio (90°), in modo tale da facilitarne la programmazione.





- Collegamenti Fritzing



Schema fritzing collegamenti Arduino

- Programmazione

Sicuramente lo step più divertente la programmazione del microcontrollore.

Per semplificare il tutto, abbiamo già a disposizione una libreria per comandare i motori la quale ha a disposizione i seguenti metodi:

- **up**(**valore**) ---> muovi il robot in avanti
- **down**(**valore**) ---> muovi il robot indietro
- **right**(**valore**) ---> ruota il robot a destra
- **left**(**valore**) ---> ruota il robot a sinistra
- **shutdown**() ---> fai arrestare il robot

Sintassi

robot.up(valore)

Parametri

valore: valore del PWM che va da 0 a 255.

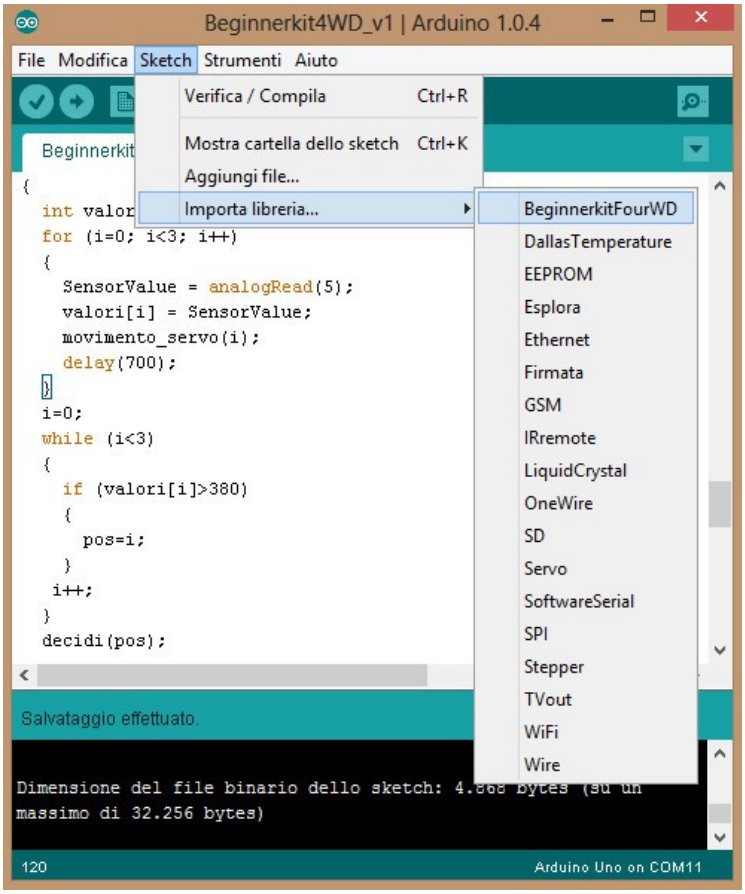
inserite il valore che volete tenendo conto di queste informazioni:

150 ---> lento

200 ---> medio

255 ---> alto

Abbiamo creato già un programma basilare che evita gli ostacoli. Basta solamente caricarlo sulla board.



- **Sketch**

- BeginnerKit4WD_v1.rar

Buon divertimento!