



Arduino

Tutorial #16 - Mocho - Robot pulitore.

Materiale utilizzato – Arduino UNO - Motor Shield (2A Motor Shield) - motoriduttori - 2 ruote - sensore Sharp di distanza - struttura robot - porta batterie - interruttore - mocho.

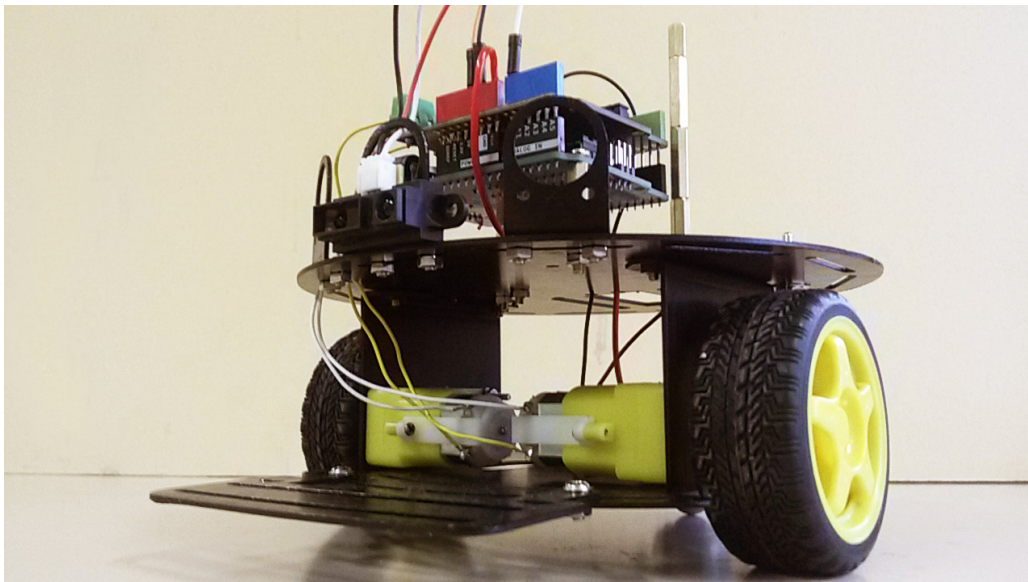
Per vedere tutti i tutorial: www.marcopucci.it/arduino

N.B.: [Clicca qui](#) per scaricare il file .zip con tutti gli sketch utilizzati per realizzare Mocho.

Questo tutorial è stato realizzato in collaborazione con www.robotics-3d.com che mi ha fornito i componenti per costruire il robot Mocho. In fondo al tutorial il link per trovare i componenti che ho utilizzato.

Mocho è un robottino in grado di girovagare autonomamente per casa, evitando ostacoli e cambiando direzione all'evenienza. L'idea si ispira ovviamente ai robot pulitori ormai comuni anche se la sua funzione (con uno straccio mocho poggiato sulla sua testa) è puramente ironica. Grazie ad un sensore di distanza è in grado di evitare gli ostacoli e girare liberamente.

La struttura

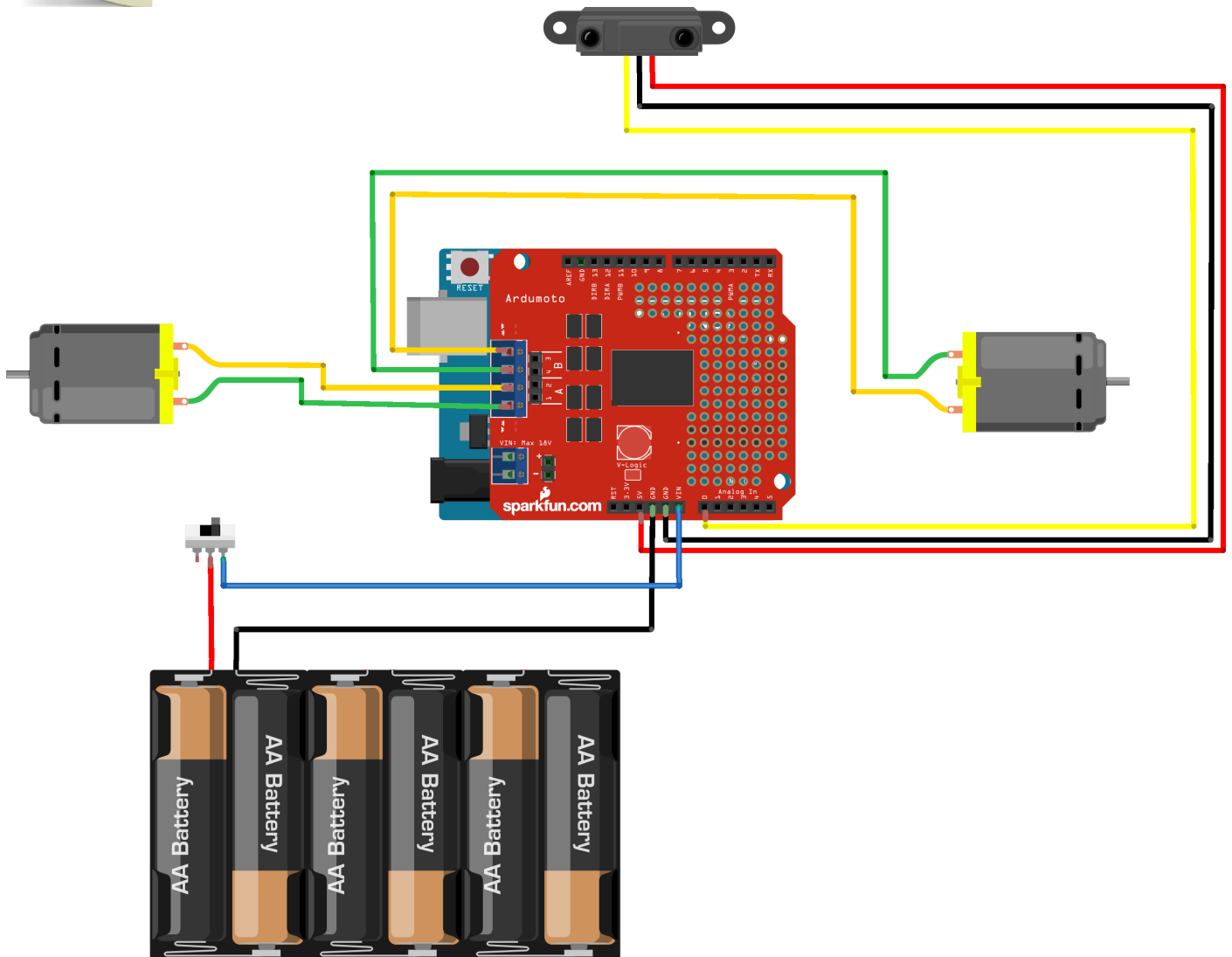


Potete utilizzare qualsiasi tipo di struttura come base per il vostro Mocho. In questo tutorial ho utilizzato la piattaforma robotica 2WD acquistabile sul sito www.robotics-3d.com. Questa base è un'ottima soluzione per realizzare piccoli robot se non siamo in grado di costruircela da soli. Inoltre al suo interno sono già presenti due motorini, due ruote, un pacco batteria e un interruttore.



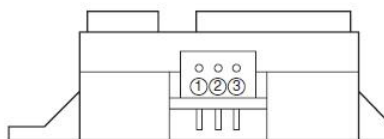
Schema elettrico

Qui sotto lo schema elettrico del circuito che dobbiamo realizzare.

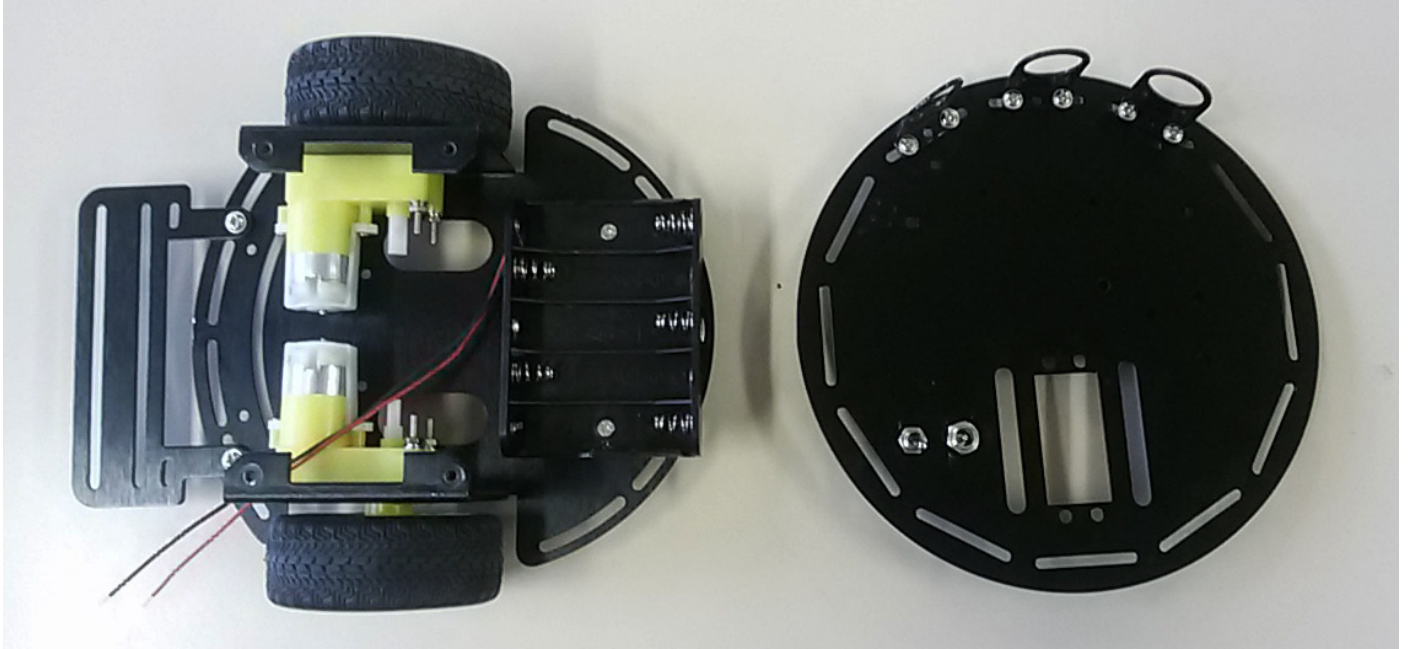


fritzing

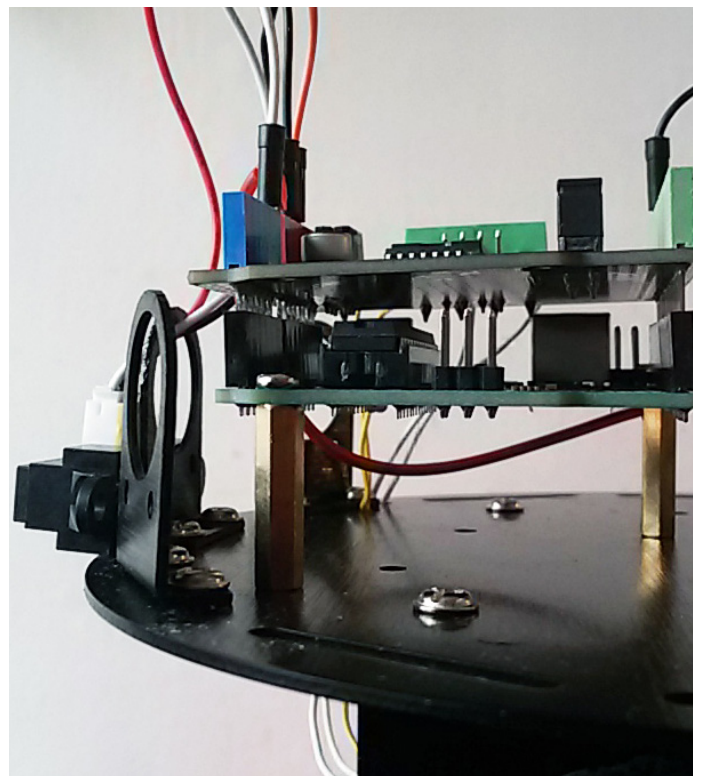
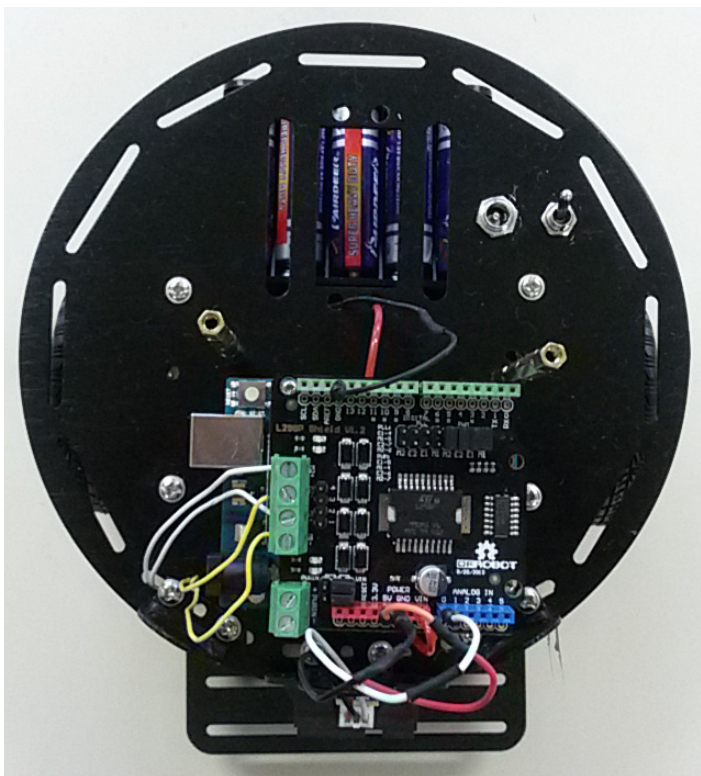
- 1** - Inseriamo la Motor Shield sopra il nostro Arduino (più avanti la spiegazione di questo componente).
- 2** - Colleghiamo il porta batterie all'interruttore. Il cavo rosso va nel pin centrale dell'interruttore (off), mentre quello di sinistra va collegato con il pin VIN di Arduino (il pin che alimenta Arduino). Il cavo nero va collegato al pin GND di Arduino. In questo modo abbiamo chiuso il circuito. Quando l'interruttore verrà spostato a sinistra la corrente delle batterie alimenterà il robot.
- 3** - Collegate i due motorini alla Motor Shield seguendo lo schema elettrico.
- 4** - Il cavo rosso del sensore Sharp va collegato al pin 5V. Il cavo nero al pin GND. Il cavo giallo al pin 0 di Arduino (se nel vostro sensore non dovessero essere presenti i tre cavi colorati fate riferimento allo schema qui sotto).



PIN	SIGNAL NAME
①	V _O
②	GND
③	V _{CC}



Alcuni dettagli della struttura del robot.
In basso a destra la scheda Arduino con sopra la Motor Shield.





Motor Shield

La Motor Shield è un componente utilizzato per gestire due motorini. Il suo funzionamento è molto elementare: ciascun motore è collegato con due cavi a questa. Il primo cavo gestisce la direzione (avanti o dietro), il secondo la sua velocità.

Apriamo il file **_1_motorini.ino** (se non avete scaricato la cartella con tutti gli sketch [cliccate qui](#)).

Uplodate lo sketch e guardate come si muovono i due motori.
Qui sotto l'analisi del codice.

```
int E1 = 5;
int M1 = 4;
int E2 = 6;
int M2 = 7;
```

Dichiaro 4 pin: E1 e E2 sono la velocità dei due motorini.
M1 e M2 gestiscono la loro direzione

```
void setup()
{
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
}
```

```
void loop()
{
  avanti();
  delay(5000);
  dietro();
  delay(2000);
  sinistra();
  delay(2000);
  destra();
  delay(2000);
}
```

Richiamo la funzione avanti() (guardare sotto)
Dopo cinque secondi la funzione dietro()...

```
void avanti(){
digitalWrite(M1,HIGH);
digitalWrite(M2, HIGH);
analogWrite(E1, 255);
analogWrite(E2, 255);
}
```

Funzione avanti:
I pin delle direzioni vengono impostati su HIGH così la direzione è avanti. La velocità dei due motorini è al massimo (per variare la velocità possiamo inserire un valore compreso tra 0 e 255).

```
void dietro(){
digitalWrite(M1,LOW);
digitalWrite(M2, LOW);
analogWrite(E1, 100);
analogWrite(E2, 100);
}
```

Funzione dietro:
Le due direzioni sono impostati su LOW.

La velocità è leggermente minore.

```
void sinistra(){
digitalWrite(M1,LOW);
digitalWrite(M2, HIGH);
analogWrite(E1, 200);
analogWrite(E2, 200);
}
void destra(){
digitalWrite(M1,HIGH);
digitalWrite(M2, LOW);
analogWrite(E1, 200);
analogWrite(E2, 200);
}
```

Funzione sinistra:
il motorino di sinistra va indietro mentre quello di destra va avanti (così facciamo ruotare il robot a sinistra)



Sensore di distanza

Vediamo ora come funziona il sensore Sharp di distanza. Per una spiegazione approfondita guardate il tutorial del Sensore Sharp - [clicca qui](#).

Apriamo lo sketch **_2_sharp_sensor.ino** e uplodiamolo sulla nostra scheda Arduino. Apriamo il Serial Monitor per vedere i valori del sensore.

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

```
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(1);  
}
```

Stampiamo sul Serial Monitor i valori del sensori.

Se non c'è nessun ostacolo davanti al sensore, il numero che appare è uguale a 0. Se posizioniamo una mano davanti al sensore questo valore cresce mano a mano che ci avviciniamo. A una distanza di circa 10 cm il suo valore è uguale a 100. Utilizzeremo questo dato per segnalare al robot che sta per andare incontro a un ostacolo.

Aprirete il file **_3_motorini_e_sharp.ino**. Questo sketch è l'unione dei due precedenti.

Ecco lo schema del suo funzionamento:

1 - il robot va avanti finchè non individua un ostacolo (valore del sensore maggiore di 100).

2 - A questo punto si blocca e va indietro per un secondo.

3 - Attraverso il comando **random()** scegliamo se il robot deve andare a destra o a sinistra.

Random(2) vuol dire che quando questo comando è attivato sceglie casualmente un valore che può essere 0 o 1. Se il valore è 0 allora il robot va a sinistra, se è uguale a 1 va a destra.



```
int E1 = 5;
int M1 = 4;
int E2 = 6;
int M2 = 7;
int direzione;
void setup()
{
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(100);

  if (sensorValue>100) {
    dietro();
    delay(1000);
    direzione=random(2);

    if (direzione==0) {
      sinistra();
      delay(2000);
      fermo();
      delay(10000);
    }
    if (direzione==1) {
      destra();
      delay(2000);
      fermo();
      delay(10000);
    }
  }
  avanti();
}

void avanti() {
  digitalWrite(M1, HIGH);
  digitalWrite(M2, HIGH);
  analogWrite(E1, 100);
  analogWrite(E2, 100);
}

void dietro() {
  digitalWrite(M1, LOW);
  digitalWrite(M2, LOW);
  analogWrite(E1, 100);
  analogWrite(E2, 100);
}

void sinistra() {
  digitalWrite(M1, LOW);
  digitalWrite(M2, HIGH);
  analogWrite(E1, 100);
  analogWrite(E2, 100);
}
void destra() {
  ...
```

Se il valore del sensore è maggiore di 100
va indietro per un secondo

funzione random - sceglie a caso un numero tra 0 e 1

Se è uguale a zero gira a sinistra per due secondi

poi si ferma per dieci secondi.

Se è uguale a uno gira a destra per due secondi

poi si ferma per dieci secondi.



Mocho

La programmazione del robot Mocho è finita.

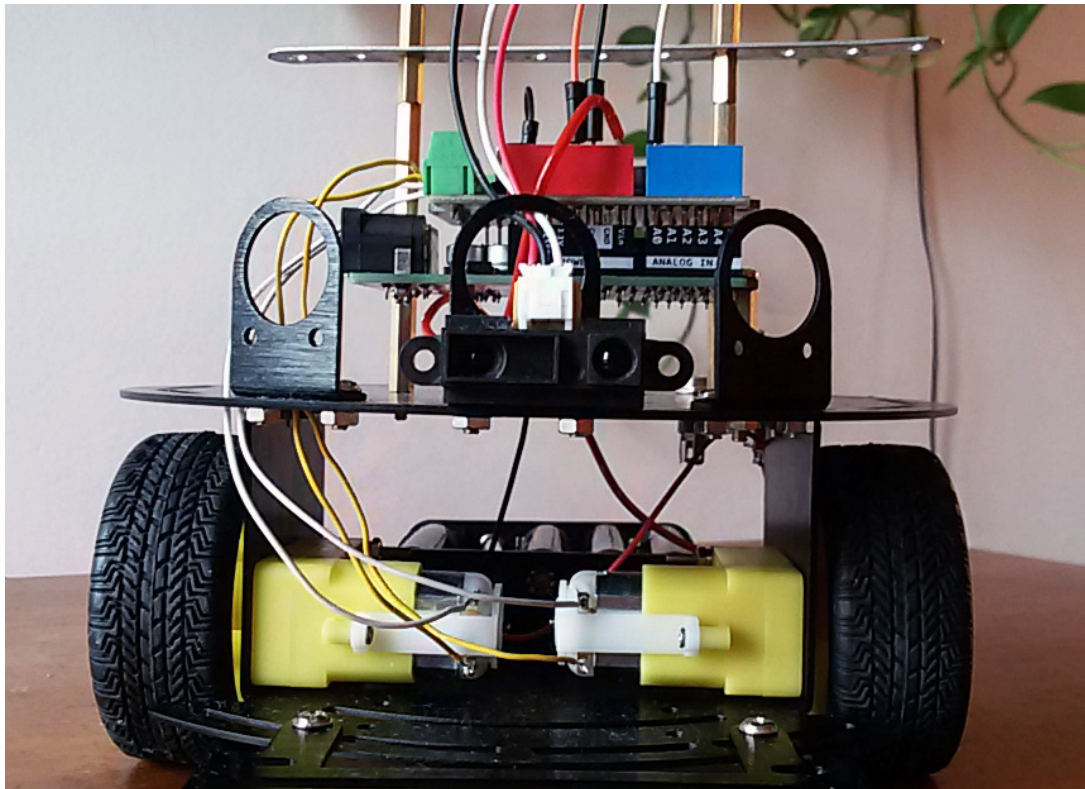
Chiaramente è possibile aggiungere altri sensori Sharp per renderlo più abile a evitare ostacoli.

Non ci resta che inserire un mocho per le pulizie sopra il robot.

Costruiamo una struttura per sollevare il mocho appena sopra Arduino (nel mio caso ho utilizzato dei distanziatori e una barra di metallo (vedere foto qui sotto).

Accorciate con delle forbici il panno del mocho per evitare che vada sotto le ruote.

N.B: nel codice ho inserito un tempo di fermata di dieci secondi (quando trova l'ostacolo) per evitare di far surriscaldare troppo i due motorini.



MATERIALE UTILIZZATO

Potete trovare tutto il materiale per realizzare il Robot Mocho acquistando il box completo presso il sito www.robotics-3d.com