Relazione laboratorio di making

Davide Allevi
Filippo Soncini

Novembre 2019

1 Braccio Robotico

Il braccio robotico è di tipo articolato in plexiglass a 4 gradi di libertà, con base rotante (180^o), pinza per afferrare gli oggetti e 5 servocomandi gestibili da Arduino o altra elettronica. La pinza può essere montata sia in posizione orizzontale che verticale ed è in grado di ruotare di +/- 90^o al livello del polso.

Figure 1: Braccio robotico

1.1 Caratteristiche tecniche

In questa sezione sono elencate informazioni e caratteristiche tecniche del braccio.

Braccio:

- Lunghezza braccio e avambraccio (mm): 160
- Altezza massima braccio (mm): 270
- Altezza massima raggiunta dal polso (mm): 310
- Estensione massima (pinza inclusa) (mm): 400
- Rotazione braccio su base d'appoggio: 180^o
- Dimensioni base d'appoggio (mm): 145x145
- Pinze al vertice con ganasce ad arco
- Azionamento mediante 5 servocomandi
- Capacità di sollevamento al polso: 250 g
- Peso: 850 grammi

Figure 2: Avanbraccio

Pinza:

- Apertura massima ganasce (mm): 65
- Lunghezza ganasce (mm): 45

Figure 3: Pinza

Servo RC 13 kg*cm:

- Velocità di funzionamento: 0.17 s / 60 gradi (@ 4,8 V senza carico)
- Velocità di funzionamento: 0.13 s / 60 gradi (@ 6 V senza carico)
- Coppia di torsione: 9,4 kg·cm (@ 4,8 V)
- Coppia di stallo: 13 kg·cm (@ 6 V)
- Alimentazione: da 4,8 a 6 V
- Ingranaggi in metallo
- Dimensioni (mm): 40,7 x 19,7 x 42,9
- Peso: 55 grammi

Figure 4: Servo RC 13 kg*cm

Servo RC 1,2 kg*cm:

- Velocità di funzionamento: 0.12 s / 60 gradi (@ 4,8 V senza carico)
- Velocità di funzionamento: 0.11 s / 60 gradi (@ 6 V senza carico)
- Coppia di torsione: 1,2 kg·cm (@ 4,8 V)
- Coppia di torsione: 1,5 kg·cm (@ 6 V)
- Alimentazione: da 4,8 a 6 V
- Ingranaggi in materiale plastico
- Dimensioni (mm): 22x12x29
- Peso: 9 grammi

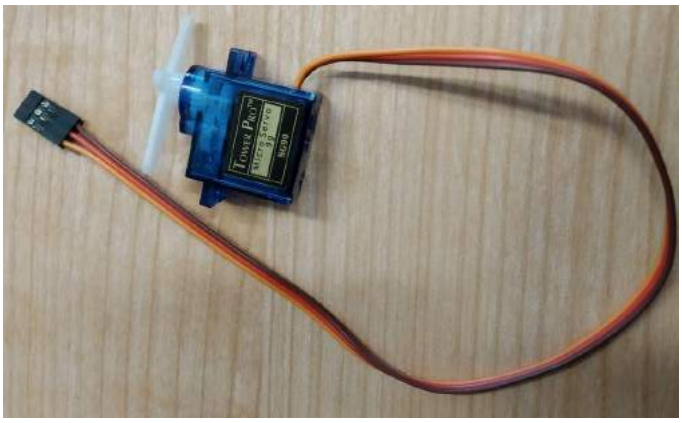


Figure 5: Servo RC 1,2 kg*cm

2 Scheda di controllo

La scheda è uno shield per Arduino/Fishino UNO che permette di controllare fino a sei servi RC tradizionali a 3 pin (+5V/–/PWM) e con feedback a 4 pin (+5V/–/PWM/Feedback). Dispone inoltre di un alimentatore da 5 volt 3A basato sul chip WP1584, due uscite di potenza a MOSFET con LED di stato, 1 ingresso digitale e due pulsanti per sviluppi futuri (I/O D7 e D8 di Arduino). L'alimentatore a bordo dello shield fornisce tutta la corrente che i servocomandi da 13 kg/cm richiedono per funzionare.
Lo shield **non prende quindi alimentazione da Arduino**, ma da un jack DC separato a bordo, che nello schema elettrico è siglato PWR; l'alimentazione d'ingresso passata attraverso il diodo di protezione dall'inversione di polarità viene filtrata e stabilizzata a 5 volt. Per ulteriori dettagli cosultare la documentazione tecnica qui.

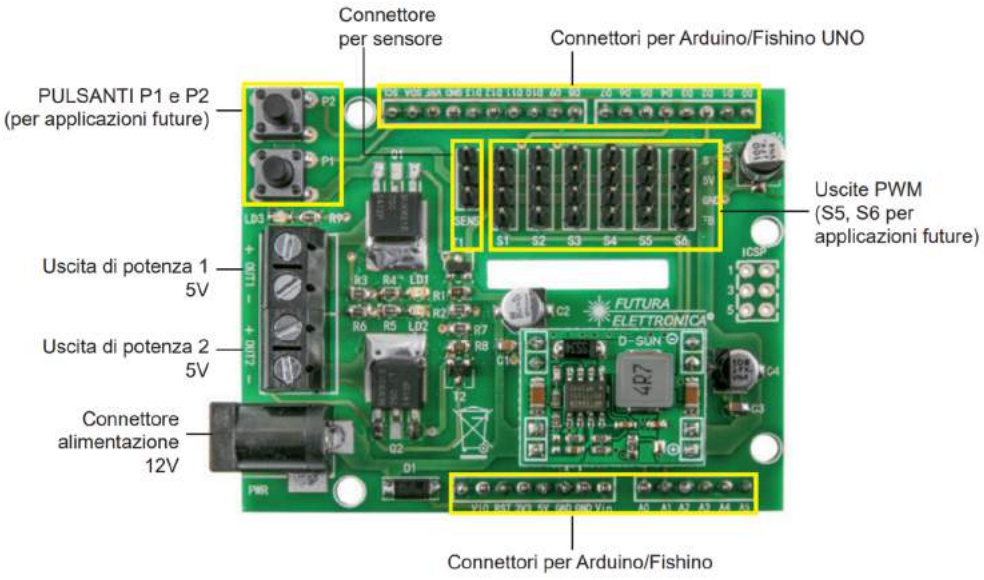


Figure 6: Scheda di controllo

3 Guida utente

3.1 Cablaggio

Connettere i vari servo allo shield come illustrato in figura 9

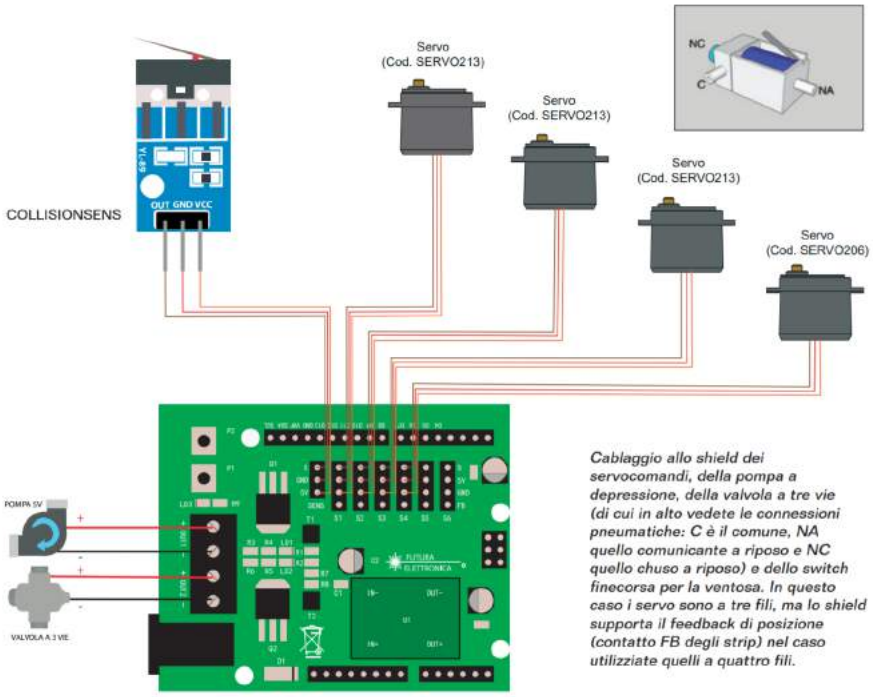


Figure 7: Esempio di cablaggio

Lo shield supporta fino ha 6 servo motori, ma il braccio ne possine solamente 5, non è presente un servo il controllo della rotazione del "polso" sull'asse Y o Z. Lo shield supporta anche gli accessori e i sensori per la pompa a depressione, ma anche quest'ultima non è per ora presente.

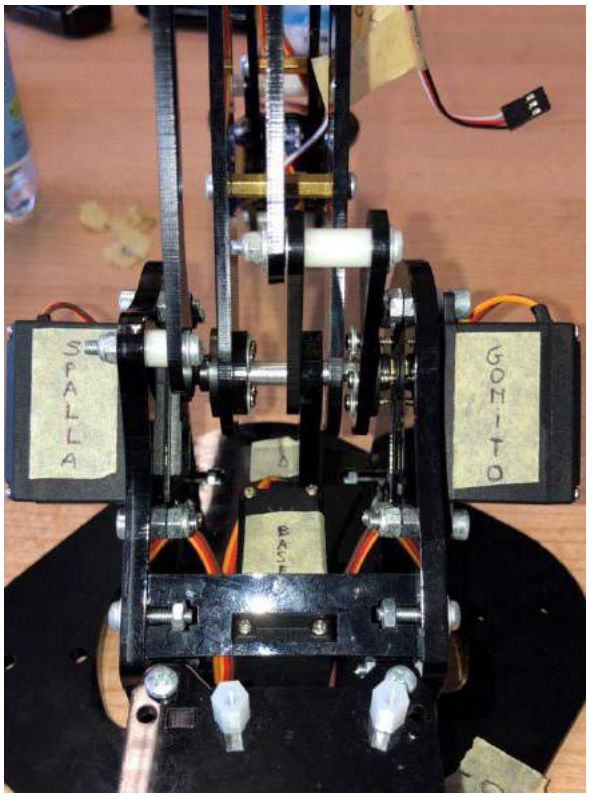


Figure 8: Servo Spalla, Gomito e Base

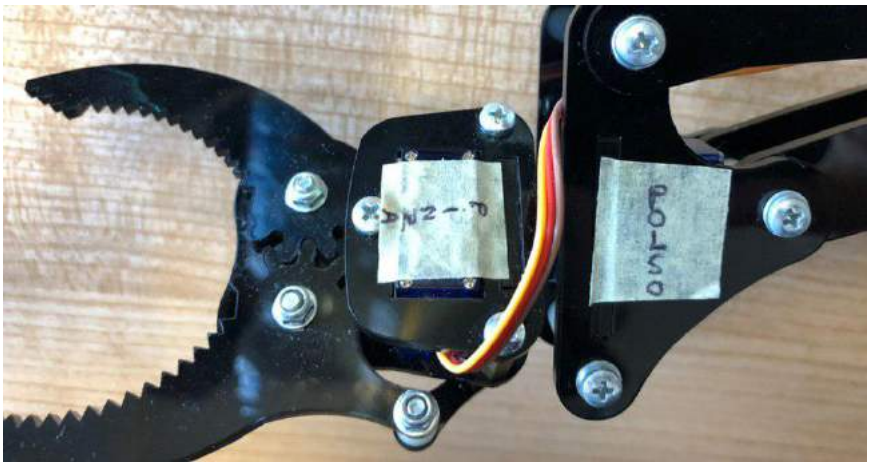


Figure 9: Servo Polso e Pinza

Importante tenere in considerazione la posizione dei singoli servo in corrispondenza dello slot di collegamento. Di norma i servo motori sono cablati in quest'ordine:

- Servo Base nello slot S1
- Servo Spalla nello slot S2
- Servo Gomito nello slot S3
- Servo Polso nello slot S4
- Servo Pinza nello slot S6
- lo slot S5 viene lasciato vuoto.

3.2 Accensione

Passi da eseguire per avviare il braccio:

1. Come prima cosa verificare che le viti d'assemblaggio nelle giunture del braccio non siano troppo strette, se così fosse è necessario allentare la tensione per permettere ai servo di funzionare correttamente.
2. Verificare che tutto sia cablato correttamente e nell'ordine corretto (sono stati fisicamente ettichettati sia i servo, sia i relativi cavi per facilitare questo passaggio).
3. Alimentare il dispositivo. Per alimentare il robot è necessario collegare un alimentatore con uscita a 12V / 1A al jack d'ingresso dello shield e collegare all'alimentazione anche Arduino/Fishino UNO.

   N.B: Attenzione come detto già in precedenza non collegare l'alimentatore nell'Arduino/Fishino UNO ma collegarlo allo shield!
4. A questo punto il braccio dovrebbe essere operativo e funzionante

3.3 Angoli consentiti

Il braccio non è in grado di muoversi liberamente per la presenza di diversi vincoli meccanici che impediscono certi moviment o superare certi angoli, o valori, dei servo. Gli angoli consentiti che ogni servo può sostenere sono elencati in tabella:

SERVO	MINIMO	MASSIMO
Base	0°	180°
Spalla	20°	170°
Gomito	50°	160°
Polso	30°	165°
Pinza	80°	140°

Avvertenze:

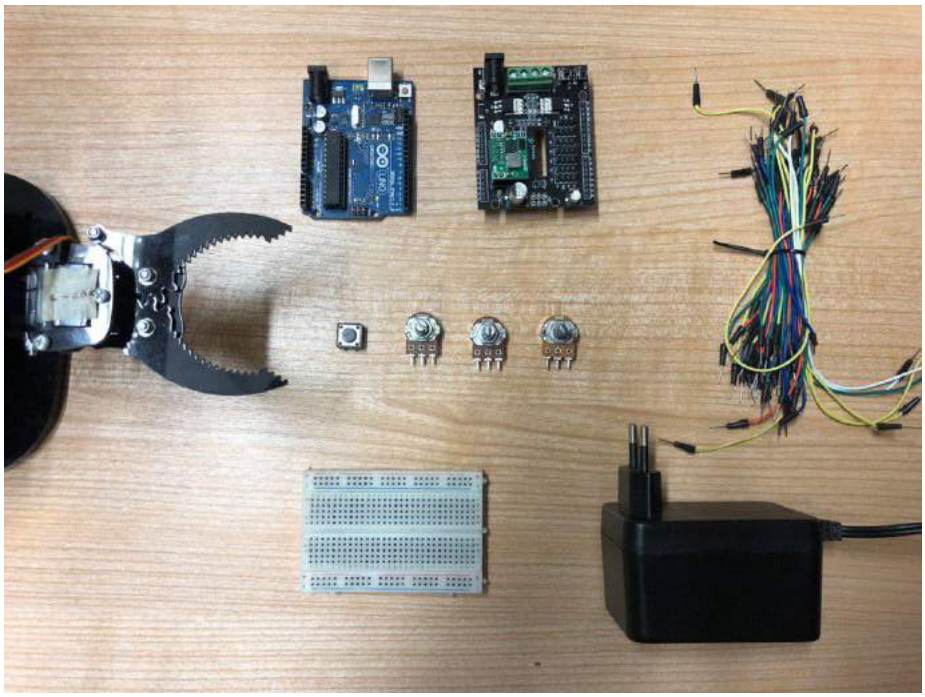
- E' estremamente importante non superare queste soglie, il superamento di esse potrebbe portare ad un eccessivo sforzo per i servo e alla conseguente rottura del prodotto.
- Prestare particolare attenzione ai servo di gomito e spalla. Tali servo sono connessi fra loro e ciò porta ad un legame fra i relativi angoli che finisco per vincolarsi a vicenda impedendo deteminate combinazioni di angoli. Per esempio se la spalla è ritratta al massimo, cioè con un angolo di 170°, il gomito non potrà estendersi al massimo, cioè ad un angolo di 50° ma sarà vincolato ad un angolo minimo di 70°

4 Demo

Il programma in esempio permette di controllare il braccio robotico utilizzando tre potenziometri, uno per ogni giuntura del braccio escluso il polso, e un button switch per l'apertura e chiusura della pinza. Il programma non ha nulla di particolarmente complesso ma è un ottimo esempio per iniziare a capire il funzionamento del braccio e come programmarlo. L'unico accorgimento particolare è stato l'utilizzo di una funzione per fare una media degli angoli tra 2 iterazioni successive, questo stabilizza i dati e permette al braccio di essere più preciso nei movimenti.

4.1 Componenti

In questo esempio sono stati utilizzati i seguenti componenti:



- Braccio,
- Shield di controllo,
- Arduino/Fishino,
- 1 Alimentatore da 12V 1A,
- 3 Potenziometri da 10Kohm,
- 1 Breadbord,
- 1 Push button switch,
- Qualche jumper e connettori.

4.2 Schema

Di seguito in figura 10 è illustrato lo schema completo.

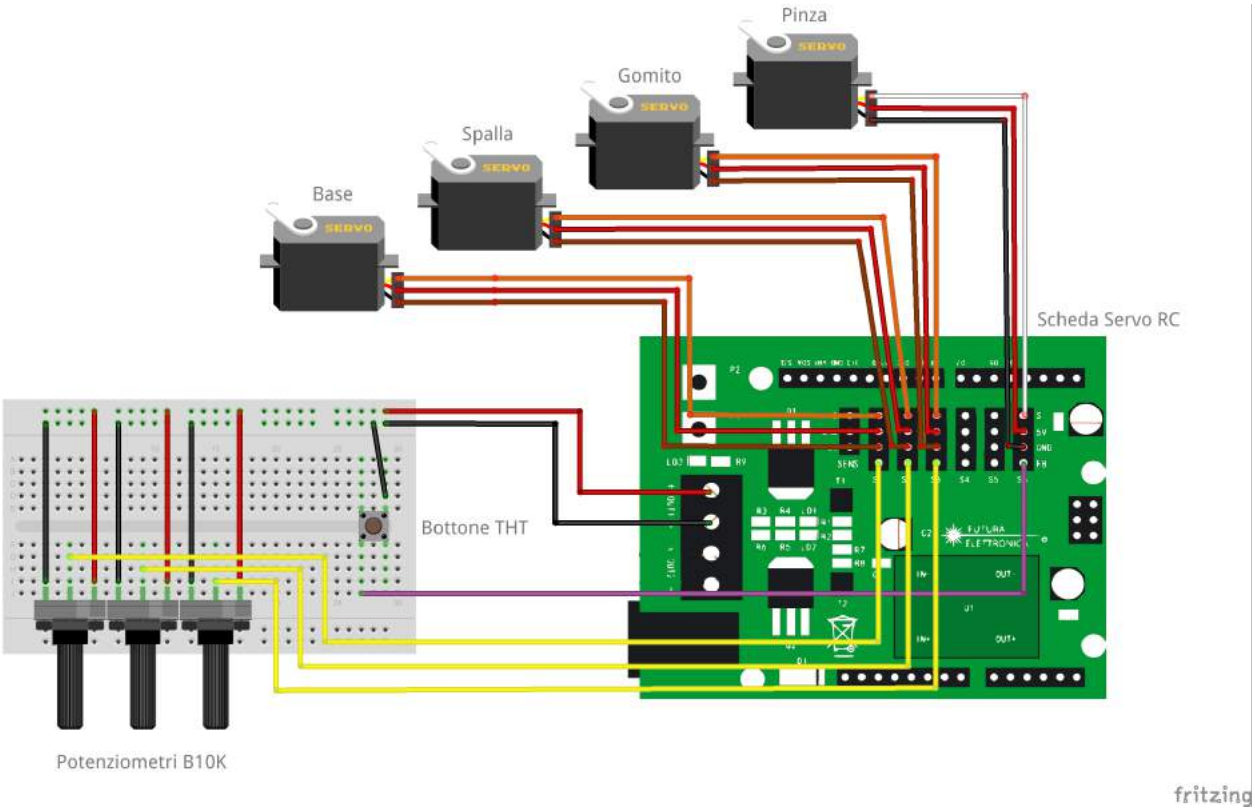


Figure 10: Schema

4.3 Codice

```
#include <Servo.h>

// Crea oggetti servo, uno per ogni servo del braccio.
Servo base;
Servo shoulder;
Servo elbow;
Servo gripper;

// Assegno i pin dell'arduino
int pin = A0;
int pinS = A1;
int pinE = A2;
```

```
int buttonPin = A5;

// Inizializzo le variabili angolo di ogni servo.
int firstangleB = 0;
int firstangleS = 0;
int firstangleE = 0;

// Funzione di inizializzazione dei servo.
void initservo(){
  base.attach(11);    // collega il servo sul pin 11 all'oggetto base,
  shoulder.attach(10); // collega il servo sul pin 10 all'oggetto
      shoulder,
  elbow.attach(9);    // collega il servo sul pin 9 all'oggetto elbow,
  gripper.attach(3);  // collega il servo sul pin 3 all'oggetto gripper.

  // imposto l'angoli iniziali per ogni servo.
  base.write(90);
  shoulder.write(90);
  elbow.write(110);
  gripper.write(100);
}

// Funzione che ottiene il valorede del potenziometro scalato.
/* Parametri:
 *  int: pin        Pin dell'arduino
 *
 *  Return:
 *  int: first      Angolo iniziale del servo collegato al pin
 */
int initangle(int pin){
  // Legge il valore del potenziometro (valore compreso tra 0 e 1023)
  int angle = analogRead (pin);
  // Ridimensiono il valore per poterlo utilizzare sul servo (valore tra
      0 e 180)
  int first = map(angle, 0, 1023, 0, 180);
  return first;
}

// Funzione che calcola la media del valore del potenziometro della
    precedente iterazione con il valore attuale del potenziometro.
/* Parametri:
 *  int: pin            Pin dell'arduino.
 *  int: firstangle     Valore del potenziometro dell'iterazione
     precedente.
 *
 *  Return:
 *  int: firstangle     Ritorna la media arrotondata.
 */
int anglewrite(int pin, int firstangle){
  // Legge il valore del potenziometro (valore compreso tra 0 e 1023)
```

```
  int angle = analogRead (pin);
  // Ridimensiono il valore per poterlo utilizzare sul servo (valore tra
      0 e 180)
  int mapping = map(angle, 0, 1023, 0, 180);
  firstangle = (mapping + firstangle) / 2;  // Calcomo la media
  return firstangle;
}

void setup() {
  pinMode(buttonPin, INPUT);    // inizializza il pin del pulsante come
      input
  digitalWrite(buttonPin, HIGH); // inizializza il valore del pulsante

  // Chiamo la funzione per inizializzare.
  initservo();
  firstangleB = initangle(pin);
  firstangleS = initangle(pinS);
  firstangleE = initangle(pinE);

}

void loop() {
  // Calcolo e aggiorno la media dei valori dei potenziometri,
  // e imposta la posizione del servo in base al valore,
  // Questo per ogni servo.
  firstangleB = anglewrite(pin, firstangleB);
  base.write(firstangleB);
  firstangleS = anglewrite(pinS, firstangleS);
  shoulder.write(firstangleS);
  firstangleE = anglewrite(pinE, firstangleE);
  elbow.write(firstangleE);

  // controlla se il pulsante viene premuto.
  if (digitalRead(buttonPin) == LOW) {
    // Se il pulsante viene premuto il servo si porta ad un angolo di
        140 (Chiusa)
    gripper.write(140);
  } else {
    // Altrimenti porta ad un angolo di 90 (Aperta)
    gripper.write(90);
  }
}
```