Arduino
Tutorial #13 - Robotic face

Materiale utilizzato – Arduino UNO - 2 servo motori - accelerometro - magnetometro - maschera di carnevale.

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 la faccia robotica.

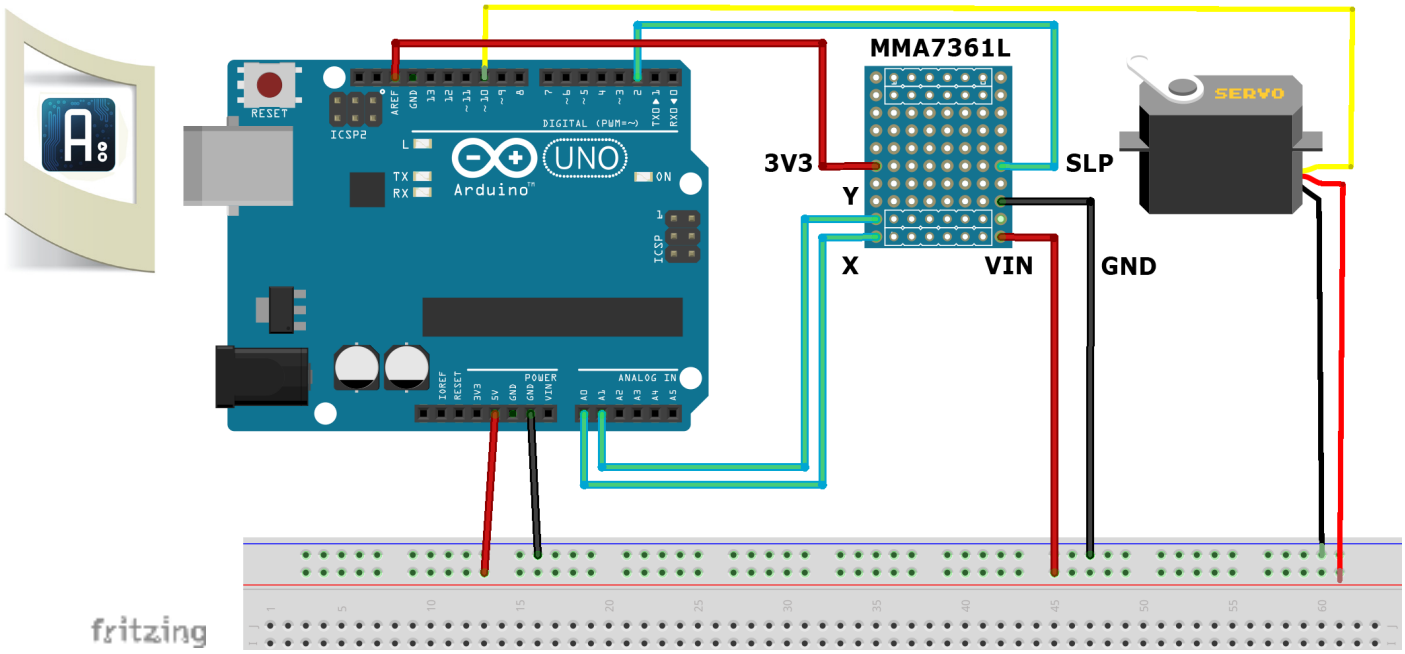
In questo tutorial realizziamo una testa robotica (low cost) in grado di replicare i nostri movimenti.
Applichiamo a un cerchietto per capelli un accelerometro e un magnetometro per individuare i movimenti della nostra testa. Questi dati, una volta analizzati, faranno muovere due motorini servo collegati a una maschera attaccata al cerchietto.
Se non avete familiarità con i motorini servo vi consiglio di leggere i precedenti tutorial - clicca qui.

Esistono numerosi accelerometri e magnetometri per Arduino. Non verrà spiegato la parte di codice che riguarda i calcoli matematici che i due componenti compiono per evitare mal di testa o svenimenti da codice. La cosa importante è capire dove mettere mano a pezzi di codice trovati su internet. In questo modo sarete in grado di gestire tutti i tipi di sensori, anche quelli più complessi.
Divido il tutorial in due parti che poi uniremo alla fine: la prima è quella relativa al controllo della testa robotica in senso verticale, ovvero "su" e "giù". La seconda quella del movimento orizzontale, destra, centro e sinistra.
Perché utilizziamo un accelerometro e un magnetometro? Non è possibile utilizzare uno solo dei due componenti? Questa è la domanda che mi sono posto mentre realizzavo la testa robotica. La risposta è no, dobbiamo utilizzare due sensori per controllare tutti i movimenti della nostra testa. Con l'accelerometro possiamo individuare l'inclinazione della testa (su, centro, giù) mentre con il magnetometro abbiamo la posizione della testa se è rivolta verso sinistra, al centro o a destra.

Schema elettrico

Iniziamo a costruire la parte elettronica del progetto. Per realizzare la faccia robotica ho utilizzato due motorini servo, un accelerometro MMA7361L e un magnetometro MAG3110.
Nella prossima pagina utilizzeremo del codice trovato su internet per far funzionare l'accelerometro MMA7361L. Quando acquistate un qualsiasi sensore per Arduino verificate se qualcuno lo ha già utilizzato per altri progetti e se sono disponibili le librerie da installare.

Come primo passo colleghiamo il motorino servo e l'accelerometro ad Arduino seguendo questo schema elettrico.



Per prima cosa quando dobbiamo acquistare un nuovo sensore cerchiamo su internet il suo schema elettrico (datasheet) per evitare di collegarlo male e bruciarlo. Sempre nella stessa fase di ricerca controlliamo se il sensore è stato già utilizzato per qualche altro progetto. In questo modo potete essere sicuri che il vostro acquisto vada a buon fine. In questo caso cercando tra i vari siti che vendono online questo tipo di sensori ho trovato questo link, http://www.robotstore.it/product/288/Accelerometro-3-assi-MMA7361L-con-regolatore.html che, oltre a vendere il componente a buon prezzo, ha in fondo alla pagina lo sketch di Arduino. Cliccate sul link "3 axis Accelerometer with Voltage Regulator test sketch" e si aprirà una nuova finestra con il codice da copiare e incollare realizzato da **Mirko Prosseda** (06-2013) * email: mirko.prosseda@gmail.com. Altrimenti potete aprire il file precedentemente scaricato **_2_accelerometro**.

Lo schema elettrico per collegarlo ad Arduino è molto semplice da trovare. Basta scrivere su Google "Arduino + MMA7361L" ed effettuare una ricerca per immagini.

Analizziamo ora il codice scaricato.
All'inizio del codice, nei commenti, sono presenti le info dell'autore e come collegare il sensore ad Arduino.

Connections:
 * BOARD -> ARDUINO
 * VIN   -> 5V
 * GND   -> GND
 * SLP   -> PIN 2
 * X   -> PIN A0
 * Y   -> PIN A1
 * Z   -> PIN A2
 * 3V3   -> AREF

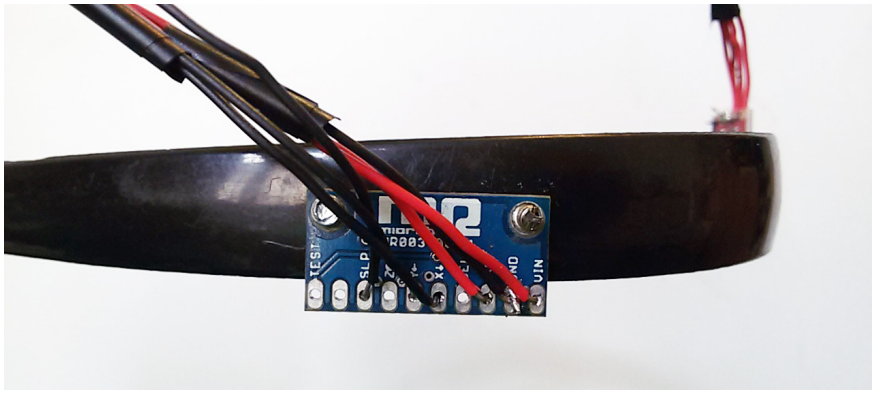
Dando uno sguardo molto veloce al codice possiamo notare che all'interno della funzione loop() sono presenti tre comandi, Serial.print(valueX); , Serial.print(valueY); e Serial.print(valueZ);

```
void loop() {
  valueX = (analogRead(analogInX) * (3.3 / 1023.0));
  valueY = (analogRead(analogInY) * (3.3 / 1023.0));
  valueZ = (analogRead(analogInZ) * (3.3 / 1023.0));
  valueX=valueX*10;
  valueY=valueY*10;
  valueZ=valueZ*10;
  Serial.print("X = ");
  Serial.print(valueX);
  Serial.print("v  Y = ");
  Serial.print(valueY);
  Serial.print("v  Z = ");
  Serial.print(valueZ);
  Serial.println("v");
  delay(2);
}
```

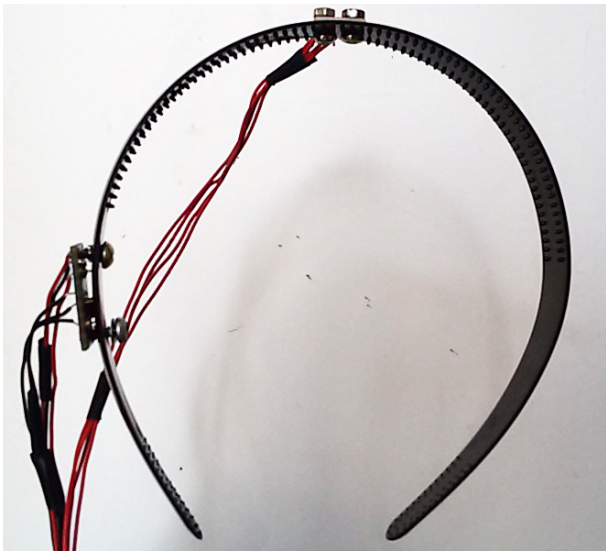
Lettura dei tre dati x,y e z che vengono elaborati per restituire tre valori.

La variabile "valueX" contiene il valore dell'asse X dell'accelerometro.
La variabile "**valueY**" contiene il valore dell'asse Y dell'accelerometro che è quello che serve a noi per individuare lo spostamento verticale della testa.

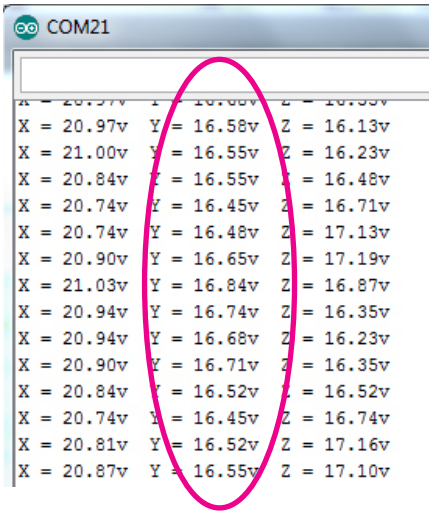
Se uplodiamo il codice sulla scheda Arduino e apriamo il SerialMonitor vediamo scorrere i tre valori X,Y e Z in base a come muoviamo il nostro accelerometro. Fissiamo l'accelerometro ad un cerchietto per capelli (in una delle parti laterali) e controlliamo i valori.

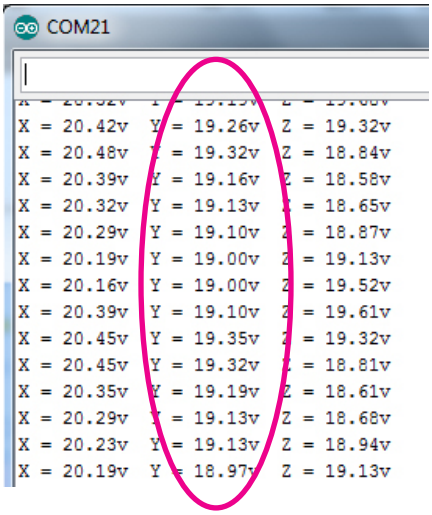


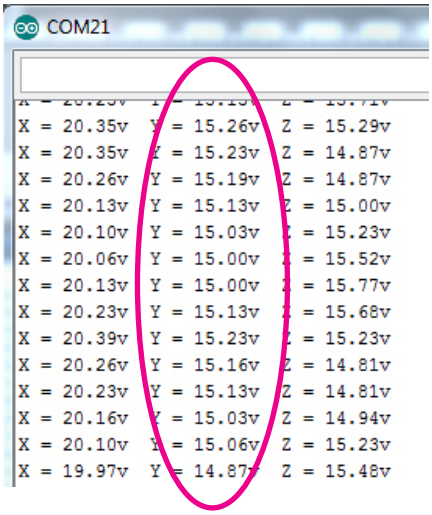
L'accelerometro deve essere fissato con due viti al lato del cerchietto.



In alto il magnetometro fissato nella parte alta del cerchietto (questa operazione verrà eseguita più avanti), in basso a sinistra l'accelerometro.







Il valore della Y è quello che ci serve. Se l'accelerometro è posizionato frontalmente il suo valore è uguale a circa 16. Se l'accelerometro viene inclinato verso l'alto il suo valore è uguale a circa 19, se verso il basso il valore è di circa 15. In questo modo possiamo determinare l'esatta posizione della nostra testa e muovere un servo motore.

Aprite il file **_3_accelerometro_servo_if**
Il codice precedente è stato leggermente modificato: in base al valore dell'asse Y muoviamo un servo motore. Se non avete seguito i tutorial precedenti scaricate la libreria **SERVO** da internet e posizionatela nella cartella **libraries** di Arduino.

```
#include <Servo.h>
Servo myservo;
const int analogInY = A1;
const int sleep = 2;
float valueY = 0;
void setup()
{
  myservo.attach(10);
  Serial.begin(9600);
  pinMode(sleep, OUTPUT);
  digitalWrite(sleep, HIGH);
  analogReference(EXTERNAL);
}

void loop() {
  valueY = (analogRead(analogInY) * (3.3 / 1023.0));
  valueY=valueY*10;
  Serial.print("Y = ");
  Serial.println(valueY);
  delay(15);

if (valueY<15.8){
   myservo.write(30);
}
if ((valueY>15.7)&(valueY<18.6)){
   myservo.write(60);
}
if ((valueY>18.7)){
   myservo.write(100);
}
}
```

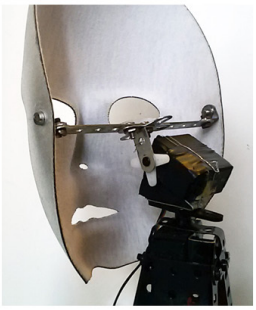
Importiamo la libreria Servo

Utilizziamo solamente la variabile analogInY

Il servo motore è collegato al pin 10

Stampiamo il valore di Y

Se il valore della Y è più piccolo di 15.8 allora il motorino deve posizionarsi all'angolo 30 (testa in giù)

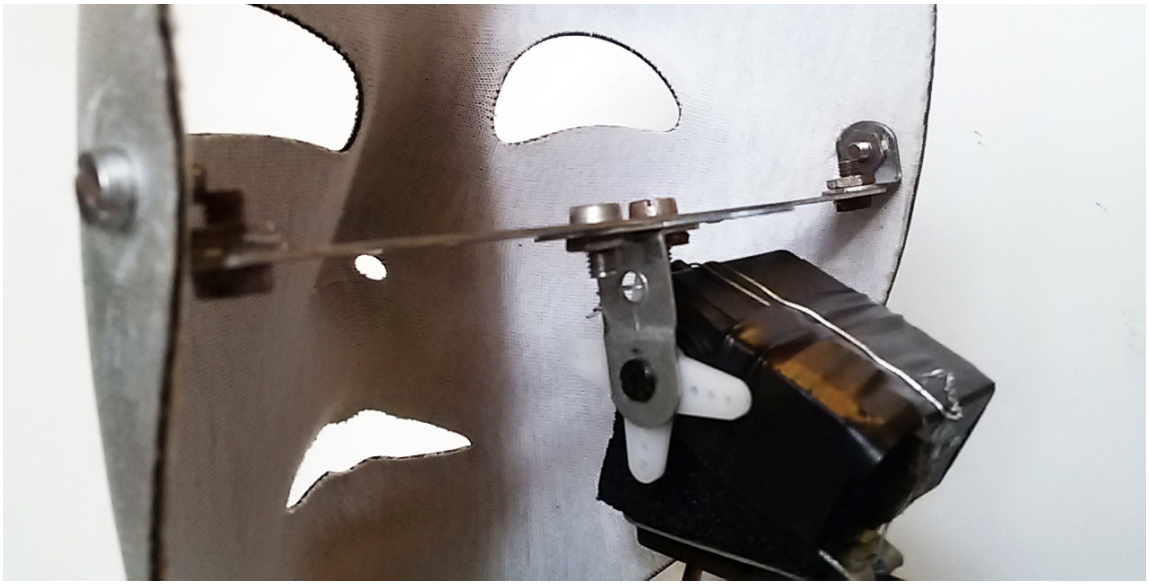


se valueY<15.8
il motorino va a 30°

se valueY>18.7
il motorino va a 100°

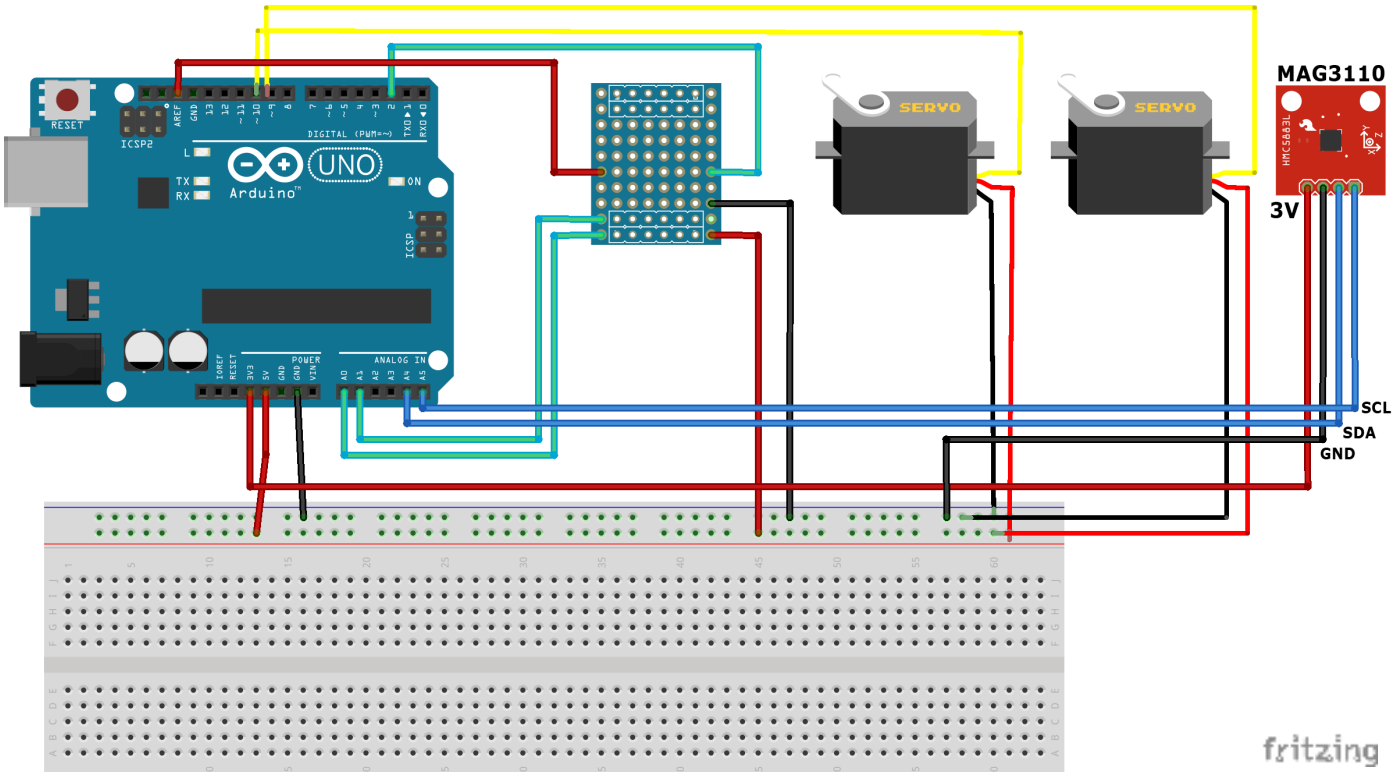
se valueY>15.7 e <18.6
il motorino va a 60°

Possiamo ora costruire la testa robotica.
Prendiamo una comune maschera di carnevale. Facciamo due buchi ai lati e fissiamo un pezzo di metallo alle due estremità. Questa barra deve essere collegata al primo servo motore. Con questa struttura elementare creiamo il movimento verticale della testa.



Indossate il cerchietto per capelli e verificate se la maschera segue i vostri movimenti verticali.
Se non dovesse andare bene basta verificare che i dati della variabile Y siano esatti oppure tarare la testa con i vostri valori.
Un altro problema potrebbe essere aver montato in diverso modo il motorino. In questo caso basterà modificare i gradi di spostamento della testa robotica nella riga di codice: **`myservo.write();`**

Passiamo ora alla realizzazione della parte orizzontale, ovvero i movimenti destra, sinistra, centro.
In questo caso utilizzeremo il magnetometro MAG3110.
Qui sotto lo schema elettrico. Potete utilizzare qualsiasi altro magnetometro, ma prima di acquistarlo verificate se in rete trovate il suo datasheet e il codice per farlo funzionare.



Effettuiamo ora una nuova ricerca su Google: “Arduino + magnetometro MAG3110”. Tra i primi risultati della ricerca troverete il sito sparkfun, un rivenditore di componenti elettronici. Anche in questo caso sono presenti i codici per far funzionare il sensore con Arduino.

images are CC BY-NC-SA 3.0

Triple Axis Magnetometer Breakout - MAG3110

SEN-10619 RoHS

Description: Freescale's MAG3110 is a small, low-power, digital 3-axis magnetometer. The device can be used in conjunction with a 3-axis accelerometer to produce orientation independent accurate compass heading information. It features a standard I2C serial interface output and smart embedded functions. It's also a tiny QFN package which isn't very easy to play with so here is our easy to use breakout board. This board breaks out all of the pins for the MAG3110FCR1 to a standard 0.1" header and also supplies the necessary filtering capacitors so that you can easily use it in your next navigation project.

Features:

- 1.95V to 3.6V Supply Voltage
- 7-bit I2C address = 0x0E
- Full Scale Range ±1000 µT
- Sensitivity of 0.10 µT

Dimensions: 13.3 x 14.5 mm

Documents:

- Schematic
- Eagle Files
- Datasheet (MAG3110FCR1)
- Example Code

Cliccando sul link in basso **Example Code** si apre una pagina con il codice di Arduino. Copiamolo e incolliamolo in un nuovo sketch di Arduino (oppure aprite il file precedentemente scaricato **_4_Mag3010**).

Analizziamo il codice e cerchiamo di individuare la parte che ci interessa saltando le funzioni matematiche per il calcolo dei valori del sensore.

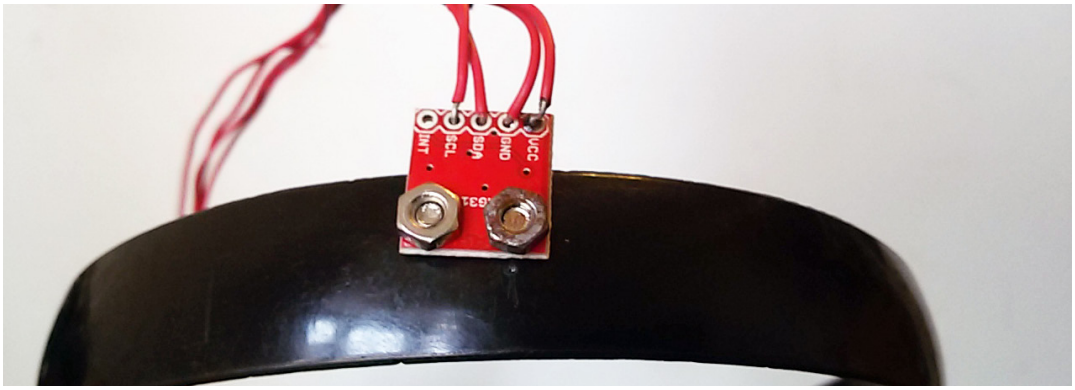
```
void loop()
{
  print_values();
  delay(5);

}
```

All’interno della funzione loop() appare solamente il comando print_values(); che serve a richiamare la funzione. Spostiamoci giù per vedere cosa succede all’interno di print_values();

```
void print_values(void)
{
  Serial.print("x=");
  Serial.print(readx());
  Serial.print(",");
  Serial.print("y=");
  Serial.print(ready());
  Serial.print(",");
  Serial.print("z=");
  Serial.println(readz());
}
```

Vengono stampati i tre valori del magnetometro x,y e z. La parte di codice che calcola i tre singoli dati sono scritte all’interno delle funzioni **int readx(void)**, **int ready(void)** e **int readz(void)**.

Fissiamo il magnetometro nella parte alta del cerchietto. Uplodiamo il codice e apriamo il Serial Monitor. Come potete notare indossando il cerchietto e muovendo la testa verso destra, sinistra e centro il valore della X cambia. Utilizzeremo questo dato per gestire il secondo motore.



Fissiamo il secondo motore sotto il primo.
Per permettere al motore in alto di ruotare liberamente utilizziamo una piccola staffetta di metallo piegata a V (come nella figura qui a lato). In questo modo con i due motorini incollati possiamo ricreare tutte le posizioni orizzontali e verticali della nostra testa.

Aprite il file **_5_Mag3010_servo**.

```
void loop()
{
  print_values();
  delay(5);


  int valoreX=readx();
  Serial.println(valoreX);




  valoreX = map(valoreX, -100, -550, 30, 130);
  myservo.write(valoreX);
}
```

A parte le dichiarazioni della libreria servo e il nome del servo utilizzato (pin 9) la parte di codice modificata all'interno della funzione loop() è piccolissima.

Utilizziamo una variabile di appoggio per copiare i valori della X e stamparli.
A questo punto segniamo su un foglietto i valori della X quando la testa è girata a sinistra e quando è girata a destra. Nel mio caso i due valori sono di circa -100 e -550.

Come nel tutorial precedente utilizziamo la funzione map per creare un rapporto tra i valori della X e il movimento del servo motore.

A seconda del magnetometro utilizzato o di come è stato montato il motorino, questi valori potrebbero essere diversi dai miei.

Indossate il cerchietto e muovete la testa. La testa robotica seguirà i vostri movimenti.
Non resta che unire i due sketch.
Visto che il programma del magnetometro è molto più complesso utilizzeremo questo come base del nuovo sketch. Salviamo con nome per fare una copia di sicurezza.
Riga per riga incolliamo le parti di codice dell'accelerometro dentro il nuovo sketch, facendo attenzione alla loro posizione.

```
#include <Wire.h>
#include <Servo.h>
Servo myservo_dx_sx;
Servo myservo_su_giu;
const int analogInY = A1;
const int sleep = 2;
float valueY = 0;
int risultato=0;
#define MAG_ADDR  0x0E

void setup()
{
  myservo_dx_sx.attach(9);
  myservo_su_giu.attach(10);
  pinMode(sleep, OUTPUT);
  digitalWrite(sleep, HIGH);
  analogReference(EXTERNAL);
  Wire.begin();
  Serial.begin(9600);
  config();
}

void loop()
{
  print_values();
  delay(5);
  int valoreX=readx();
  Serial.println(valoreX);
  valoreX = map(valoreX, -107, -550, 30, 130);
  myservo_dx_sx.write(valoreX);
  valueY = (analogRead(analogInY) * (3.3 / 1023.0));
  valueY=valueY*10;
  Serial.print("Y = ");
  Serial.println(valueY);
  delay(15);

if (valueY<15.8){
   myservo_su_giu.write(30);
}

if ((valueY>15.9)&(valueY<18.6)){
   myservo_su_giu.write(60);
}

if ((valueY>18.7)){
   myservo_su_giu.write(100);
}
}
```

In grassetto le parti incollate nel nuovo sketch.
Realizzate una base per sollevare la testa robotica e testate il cerchietto con i due sensori. Di seguito una piccola galleria fotografica per vedere come ho realizzato la mia RobotFace.
Una volta realizzata la vostra RobotFace, potete inviarmi le immagini o il video alla mail: puccimarco76@yahoo.it. Il materiale verrà pubblicato con i vostri contatti sul mio sito web e sulla pagina Facebook "Tutorial Arduino".