

Contrôle d’une main robotique

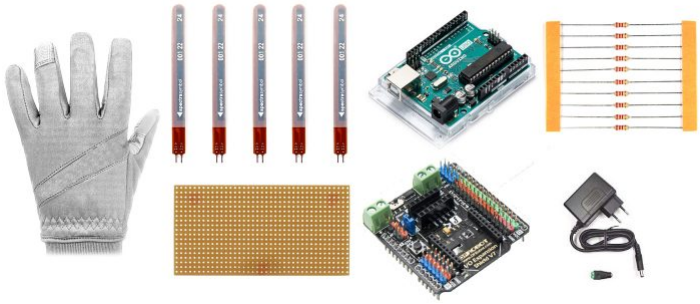
Cet article vous propose de contrôler les servomoteurs d’une main robotique grâce à plusieurs capteurs de flexion installés sur un gant et raccordés à une carte Arduino Uno.

Matériel nécessaire :

- Une main droite robotique (version gauche également disponible, voir ROB0142)
- Une carte Arduino UNO
- Un shield E/S Gravity V7
- Une alimentation 5 Vcc/3 A
- Un adaptateur bornier -> fiche alim 5,5 x 2,1 mm
- Un gant (voir par ex. gants antistatiques)
- 5 capteurs de flexion
- 5 résistances 10 kΩ
- Une plaque d’essais
- Un connecteur M/M 36 broches coudées MH190
- Un kit de soudure SOLD3

Pour la réalisation du câblage, nous recommandons l’utilisation de connecteurs à s

- 3 mètres de fil 0,2 mm² de différentes couleurs (bleu, rouge ,noir)
- Une pince à sertir HT225D
- 25 contacts NDR-T
- 25 boîtiers NSR-03



Ajout des composants au gant:

Dans un premier temps, nous allons prendre le gant et placer les 5 capteurs de flexion sur chaque doigt.

Etape 1: Positionnez la plaque d’essais sur le gant.

Mesurez la longueur de fil nécessaire vers chaque capteur en ajoutant 2 cm de marge pour l’articulation.

Etape 2: Soudez maintenant les capteurs de flexion au câble précédemment mesuré.

Soudez l’autre extrémité des câbles à la plaque d’essais.

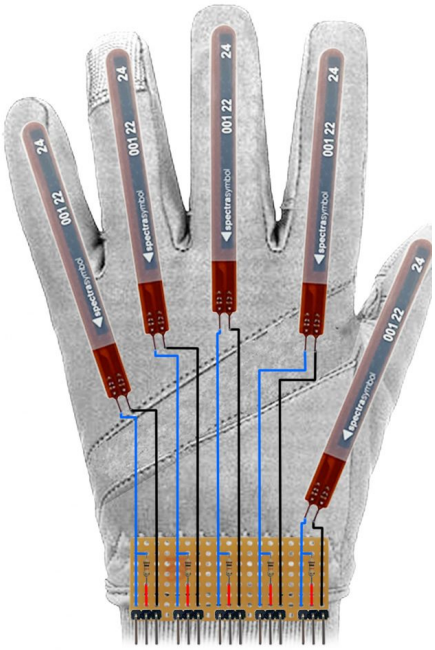
Enfin, soudez les résistances pour chaque capteur en laissant un emplacement libre pour le connecteur.

Les soudures terminées, fixez la plaque d’essais et les capteurs au gant.

La solution la plus efficace est de coudre les capteurs de flexion et la plaque d’essais au gant avec du simple fil de couture.

Remarques:

- Retrouvez nos fers à souder sur notre site gotronic.fr.
- Un tutoriel sur la soudure est disponible sur notre blog.



Raccordement des différents éléments

Pour faciliter la connexion entre le microcontrôleur et le gant, nous recommandons l'utilisation d'un ensemble de cosses à sertir et de boitiers compatibles:

- Cosses à sertir NDR-T.
- Boîtiers 3 contacts NSR-03.
- Fil de câblage en 0,20 mm² FC25.



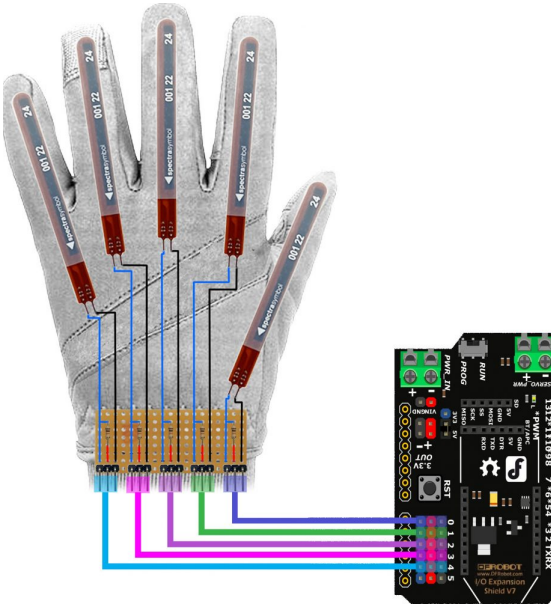
Le sertissage de ces connecteurs est simple à réaliser et nécessite seulement l'utilisation d'une pince à sertir de type HT225D.

1ère étape: connexion des capteurs de flexion au shield.

Cette étape vous permet de visualiser l'emplacement de raccordement des 5 capteurs de flexion.

Ceux-ci se connectent simplement sur les connecteurs analogiques 3 broches GVS (Ground, Vcc, Signal) du shield E/S Gravity V7.

Ce shield facilite grandement le raccordement de ce type d'éléments en proposant une connectique adaptée à l'alimentation.



Brochage:

- Pouce à connecter sur A0.
- Index à connecter sur A1.
- Majeur à connecter sur A2.
- Annulaire à connecter sur A3.
- Auriculaire à connecter sur A4.

2ème étape: connexion des servomoteurs au shield.

Cette étape indique le brochage à respecter pour la connexion des 5 servomoteurs animant la main robotique.

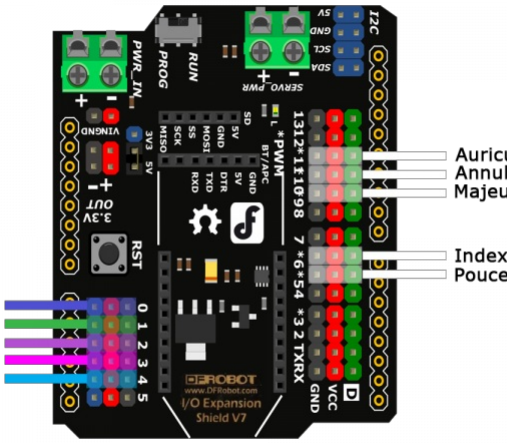
Ces connecteurs sont similaires à ceux de l'étape précédente mais délivrent un signal numérique actionnant vos servomoteurs.

Brochage:

-

Le servo correspondant au **pouce** se connecte sur la broche **D5**.

- Le servo correspondant à l'**index** se connecte sur la broche **D6**.
- Le servo correspondant au **majeur** se connecte sur la broche **D9**.
- Le servo correspondant à l'**annulaire** se connecte sur la broche **D10**.
- Le servo correspondant à l'**auriculaire** se connecte sur la broche **D11**.

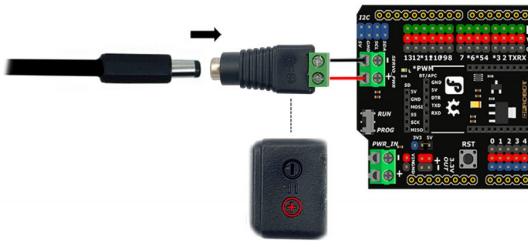


3ème étape: connexion de l'alimentation.

Raccordez maintenant l'adaptateur 5,5 x 2,1 mm au bornier à vis du shield avec du fil de câblage.

Remarque: attention aux polarités ! Cet adaptateur indique les pôles positif et négatif.

Ensuite branchez l'alimentation 5 Vcc dans l'adaptateur à bornier.



Calibrage et programmation

La première étape pour le calibrage des capteurs est de connecter votre carte Arduino Uno ou compatible en USB à votre ordinateur puis de lancer le logiciel de programme arduino.

Le premier programme que nous allons utiliser est un programme déjà existant que nous avons modifié. En ouvrant le moniteur série, ce programme va nous permettre de récupérer les données analogiques mini et maxi des capteurs appartenant à chacun de nos doigts.

Notez bien les données que va vous renvoyer le moniteur série du programme une fois ce dernier téléversé.
Exemple pour la broche A0 correspondant au capteur pour votre pouce:

- Valeur analogique maxi mesurée: 868
- Valeur analogique mini mesurée: 757

Remarque : Par défaut, le programme nous permet de récupérer les données collectées de la broche analogique**A0** qui est reliée au capteur du pouce dans cette ligne :

const int analogInPin = **A0**

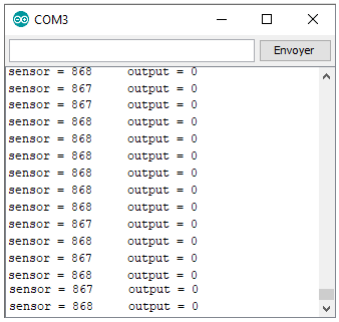
donc si l'on veut récupérer les données des différents doigts il suffit de modifier cette partie de cette façon :

l'index : const int analogInPin = **A1**
majeur : const int analogInPin = **A2**
annulaire : const int analogInPin = **A3**
auriculaire : const int analogInPin = **A4**

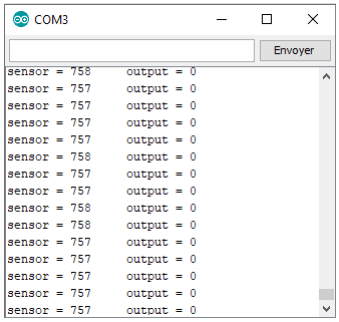
Attention il ne faut pas oublier de retéléverser le programme à chaque modification.

```
/*
  Analog input, analog output, serial output

  Reads an analog input pin, maps the result to a range from 0 to 255 and uses
  the result to set the pulse width modulation (PWM) of an output pin.
  Also prints the results to the Serial Monitor.

  The circuit:
- potentiometer connected to analog pin 0.
   Center pin of the potentiometer goes to the analog pin.
   side pins of the potentiometer go to +5V and ground
- LED connected from digital pin 9 to ground

  created 29 Dec. 2008
  modified 9 Apr 2012
  by Tom Igoe

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/AnalogInOutSerial
*/

// These constants won't change. They're used to give names to the pins used:
const int analogInPin = A0
;  // Analog input pin that the potentiometer is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
#include <Servo.h>
Servo index;

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
   index.attach(5);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // print the results to the Serial Monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop for the analog-to-digital
  // converter to settle after the last reading:
  delay(20);
}
```

Exemple de valeurs maxi données par le programme via le moniteur série:



Exemple de valeurs mini données par le programme via le moniteur série:



Nos 5 valeurs mini et 5 valeurs maxi relevées pour chacun de nos capteurs de flexion, nous pouvons passer à l'utilisation d'un programme actionnant les servomoteurs en fonction des mouvements de vos doigts.

Les valeurs mini et maxi (en bleu) sont à intégrer manuellement au programme :

```
sensorValuePouce = map(sensorValuePouce, 757,868, 80,110);//pouce 82° au min et 110° au max
sensorValueIndex = map(sensorValueIndex, 786,900, 60,120);//index 85° au min et 140° au max
sensorValueMajeur = map(sensorValueMajeur, 870,935, 60,120);//majeur 40° au min et 115° au max
sensorValueAnnulaire = map(sensorValueAnnulaire, 704,840, 60,120);//annulaire 40° au min et 115° au max
sensorValueAuriculaire = map(sensorValueAuriculaire, 750,822, 60,120);//auriculaire 95° au min et 145° au max
```

Remarque : selon la façon dont la main a été assemblée il peut exister des décalages, il faudra donc modifier les valeurs des servomoteurs (en vert).

Programme principal :

Le servomoteur a un potentiomètre intégré qui permet de contrôler le moteur tournant de 0° à 180°.

Pour le programme final nous allons utiliser la fonction **.write();** qui permet de modifier la valeur d'angle du servomoteur allant de **0° à 180°**.

Exemple pour le pouce:

pouce.write(sensorValuePouce); la valeur de sensorValuePouce convertit les valeurs du capteur de flexion correspondant aux valeurs d'angle du servomoteur grâce à la fonction **map();** vue ci-dessus.

```
// Définition des variables d'entrée analogiques :
const int analogInPinPouce = A0;
const int analogInPinIndex = A1;
const int analogInPinMajeur = A2;
const int analogInPinAnnulaire = A3;
const int analogInPinAuriculaire = A4;

// Définition des variables pour lire les valeurs analogiques des capteurs de flexions :
int sensorValuePouce;
int sensorValueIndex;
int sensorValueMajeur;
int sensorValueAnnulaire;
int sensorValueAuriculaire;

// Utilisation de la fonction Servo à partir de la bibliothèque :
#include <Servo.h>
Servo pouce;
Servo index;
Servo majeur;
Servo annulaire;
Servo auriculaire;

void setup() {
  // Initialisation de la communication série à 9600 bps :
  Serial.begin(9600);

   // Attribution de chaque servomoteur à une broche digitale :
   pouce.attach(5);
   index.attach(6);
   majeur.attach(9);
   annulaire.attach(10);
   auriculaire.attach(11);
}

void loop() {
  // Lecture de la valeur analogique du capteur de flexion :
  sensorValuePouce = analogRead(analogInPinPouce);
  sensorValueIndex = analogRead(analogInPinIdex);
  sensorValueMajeur = analogRead(analogInPinMajeur);
  sensorValueAnnulaire = analogRead(analogInPinAnnulaire);
  sensorValueAuriculaire = analogRead(analogInPinAuriculaire);


  // Utilisation de la fonction map qui convertit la valeur min et max du capteur en valeur min et max pour le servomoteur :
  sensorValuePouce = map(sensorValuePouce, 760,880, 82,110);//pouce 82° au min et 110° au max
  sensorValueIndex = map(sensorValueIndex, 786,900, 85,140);//index 85° au min et 140° au max
  sensorValueMajeur = map(sensorValueMajeur, 870,935, 40,115);//majeur 40° au min et 115° au max
  sensorValueAnnulaire = map(sensorValueAnnulaire, 750,822, 95,145);//annulaire 95° au min et 145° au max
  sensorValueAuriculaire = map(sensorValueAuriculaire, 704,840, 40,115);//auriculaire 40° au min et 115° au max


doigt();   // Fonction utilisée pour que chaque doigt ne dépasse pas les valeurs minimales et maximales imposées :

  delay(20);
}

void doigt()
  {

  if(sensorValuePouce<=82) // Si la valeur du pouce est inférieure à 82 alors la valeur sera =82
  { sensorValuePouce=82;  pouce.write(sensorValuePouce);}
  else{pouce.write(sensorValuePouce);}

  if(sensorValuePouce>=110) // Si la valeur du pouce est supérieure à 110 alors la valeur sera =110
  {sensorValuePouce=110;  pouce.write(sensorValuePouce);}
  else{pouce.write(sensorValuePouce);}

    if(sensorValueIndex<=85) // Si la valeur de l'index est inférieure à 85 alors la valeur sera =85
    {sensorValueIndex=85; index.write(sensorValueIndex);}
    else{index.write(sensorValueIndex);}

    if(sensorValueIndex>=140) // Si la valeur de l'index est supérieure à 140 alors la valeur sera =140
    {sensorValueIndex=140;  index.write(sensorValueIndex);}
    else{index.write(sensorValueIndex);}

      if(sensorValueMajeur<=40) // Si la valeur le majeur est inférieure à 40 alors la valeur sera =40
      {sensorValueMajeur=40;  majeur.write(sensorValueMajeur);}
      else{majeur.write(sensorValueMajeur);}

      if(sensorValueMajeur>=115) // Si la valeur le majeur est supérieure à 115 alors la valeur sera =115
      {sensorValueMajeur=115;majeur.write(sensorValueMajeur);}
      else{majeur.write(sensorValueMajeur);}

        if(sensorValueAnnulaire<=40) // Si la valeur de l'auriculaire est inférieure à 40 alors la valeur sera =40
        {sensorValueAnnulaire=40;annulaire.write(sensorValueAnnulaire);}
        else{annulaire.write(sensorValueAnnulaire);}

        if(sensorValueAnnulaire>=115) // Si la valeur de l'auriculaire est supérieure à 115 alors la valeur sera =115
        {sensorValueAnnulaire=115;annulaire.write(sensorValueAnnulaire);}
        else{annulaire.write(sensorValue);}

          if(sensorValueAuriculaire<=95) // Si la valeur de l'annulaire est inférieure à 95 alors la valeur sera =95
          {sensorValueAuriculaire=95;auriculaire.write(sensorValueAuriculaire);}
          else{auriculaire.write(sensorValueAuriculaire);}

          if(sensorValueAuriculaire>=145) // Si la valeur de l'annulaire est supérieure à 145 alors la valeur sera =145
          {sensorValueAuriculaire=145;auriculaire.write(sensorValueAuriculaire);}
          else{auriculaire.write(sensorValueAuriculaire);}
    }
```

Démonstration

Voir la vidéo sur YouTube