



Le Cerveau - Arduino capacitif

Une sculpture animée g ce   un Arduino en mode capacitif et un moteur pas   pas

 Difficult  Moyen

 Dur e 6 heure(s)

 Cat gories Art, D coration,  lectronique, Robotique

 Co t 150 EUR ( )

Sommaire

Introduction

Video d'introduction

 tape 1 - Fabrication du capteur capacitif

 tape 2 - Biblioth que Capacitive Sensor

 tape 3 - C blage et r glage du driver TB6600

 tape 4 - Quelques explications du code Arduino


Commentaires

Introduction

Dans le cadre d'une r sidence d'artiste, le Fablab de Saint Laurent du Maroni a  t  amen    collaborer avec un artiste camerounais Dieudonn  Fokou afin d'animer certaines de ses sculptures. Cette oeuvre baptis e "le Cerveau" utilise la technique de l'arduino capacitif et un moteur pas   pas. Nous d taillerons ici que le code et le c blage  lectronique, le reste relevant plus de la soudure   l'arc et de l'artiste.

Mat riaux

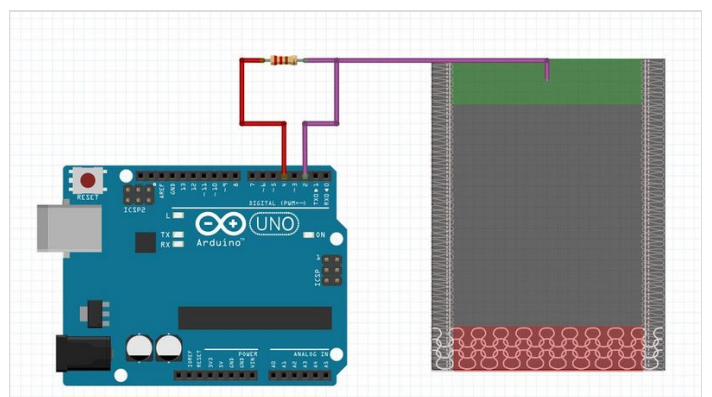
Outils

 Le_Cerveau_-_Arduino_capacitif_Thebrain_vf.ino

 tape 1 - Fabrication du capteur capacitif

Lorsqu'on place notre main sur ou pr s de la structure m tallique, une partie du cr ne va s'ouvrir r v lant son contenu. La sculpture fait office de capteur capacitif, elle envoie un signal   notre **Arduino** d s qu'on la touche par comparaison de temps entre l'impulsion qu'elle re oit sur une entr e r ceptrice et l'impulsion de sortie qu'elle envoie sur une autre.

Une **r sistance de 10 MOhm** (au minimum 1 MOhm) sera n cessaire pour obtenir cet effet et sera plac e entre 2 pins de l'Arduino. Un **moteur pas   pas** de type **Nema 17** activera l'ouverture et la fermeture du cr ne et sera pilot  par un tr s bon **driver TB6600**.



Étape 2 - Bibliothèque Capacitive Sensor

Pour transformer l'Arduino en capteur capacitif, il faut tout d'abord installer la bibliothèque de détection capacitive `CapacitiveSensor` depuis le logiciel Arduino. Extraire les fichiers dans `Documents/Arduino/libraries` puis redémarrer l'Arduino.

Pour le contrôle du moteur pas à pas, il est également possible de télécharger une librairie mais on économise de la mémoire en restant simple.

Étape 3 - Câblage et réglage du driver TB6600

On insère une résistance de 10 MOhm entre la pin 4 et 2 de l'Arduino.

La pin 2 nous sert de réception à laquelle est connectée également la structure métallique et la pin 4 de sortie du signal.

Selon le seuil atteint par la mesure, on va activer notre moteur avec le pin 9 (ENA+) et lui donner une direction sur le pin 7 (DIR+).

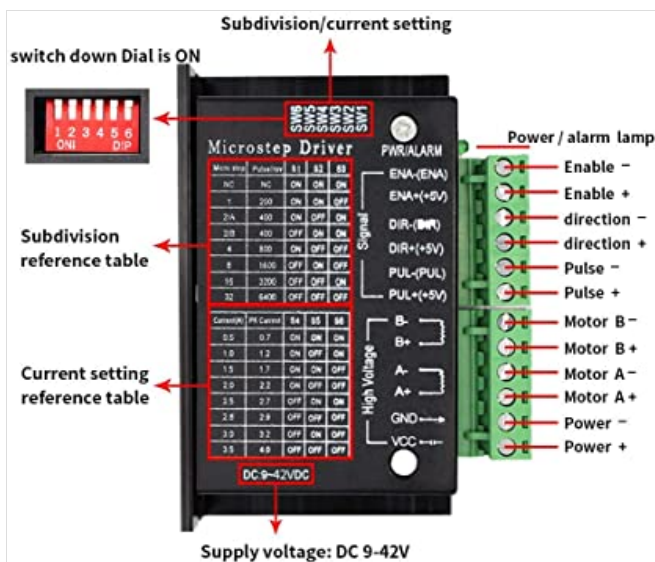
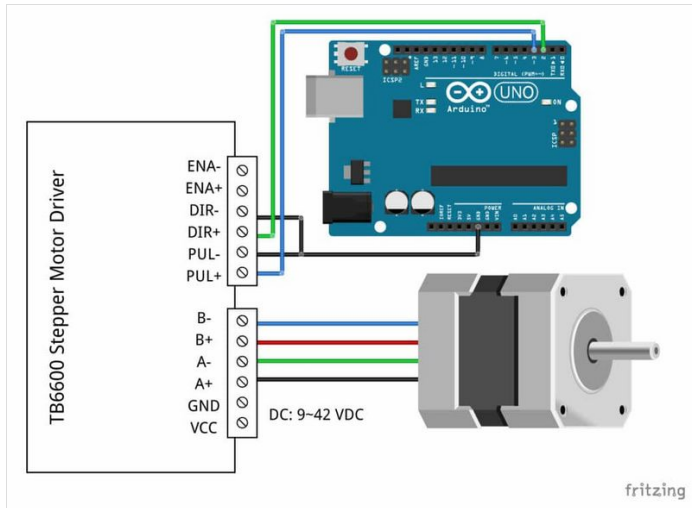
Le pin 8 (PUL+) nous sert à lui indiquer le nombre de pas.

Sur le TB6600, il y a 6 micro switches notés de SW1 à SW6. Les 3 premiers SW1 à SW3 ont été laissés à OFF, ce qui permet d'avoir un réglage à 32 micro steps, il faudra jouer avec la vitesse pour obtenir un mouvement fluide.

SW4 et SW5 ont été mis à ON mais pas SW6 pour limiter le courant à 1.5A (0.4A/phase soit $4 \times 0.4 = 1.6A$).

On relie également les 4 fils de notre moteur 17HS15-0404S à notre driver. Comme il y a 2 bobinages, les fils vont par paire (faire le test des fils croisés): noir(A+)/vert(A-) et rouge(B+)/bleu(B-).

Et on relie le (DIR-)/(PUL-)/(ENA-) au GND de l'Arduino afin de leur définir un état bas (il est généralement déconseillé de laisser des pattes en l'air).



Étape 4 - Quelques explications du code Arduino

Comment fonctionne le programme, on définit les pins entre lesquels on récupère l'information avec « `CapacitiveSensor cs_4_2 = CapacitiveSensor(4,2)` » : la pin 2 étant notre capteur relié à la structure.

Dans la partie `Setup()`, on définit juste nos sorties digitales avec output pour le DIR, le ENA et le STEP du moteur. Et on initialise la liaison série avec `Serial.begin()`.

Dans la boucle `Loop()` du programme, on active l'ouverture ou la fermeture du cerveau selon la valeur de trigger récupérée après l'exécution de `CSread()`.
