

Campus Félix Leclerc

**Étude de faisabilité :**

**Graveur de circuit imprimé (PCB) sur machine CNC**

PRÉPARATION AU PROJET ET AU STAGE

*(247-5E1-HU)*

Session Automne 2016

Benoît Houle

Présenté aux professeurs

du programme de

Technologies du génie électrique

7 février 2020

Table des matières

[Matériel utilisé 2](#_Toc31947769)

[Explications des composantes 3](#_Toc31947770)

[Diagrammes schématiques 4](#_Toc31947771)

[Alimentation & contrôleur 4](#_Toc31947772)

[Contrôle des moteurs 5](#_Toc31947774)

[Diagrammes fonctionnels 6](#_Toc31947775)

[Matériel 6](#_Toc31947776)

[Serveur 7](#_Toc31947777)

[Contrôleur 8](#_Toc31947778)

[Code 9](#_Toc31947779)

[UML 9](#_Toc31947780)

[Serveur 9](#_Toc31947781)

[Moteur 10](#_Toc31947782)

[En-tête personnel 11](#_Toc31947783)

[Contrôle des moteurs 11](#_Toc31947784)

[Serveur 13](#_Toc31947785)

[En-tête d’autres librairies 17](#_Toc31947786)

[Contrôle des moteurs 17](#_Toc31947787)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Quantité** | **numéro de pièce (SKU)** | **Fabricant** | **Fournisseur** | **Description** | **Prix** | **Page web (URL)** | **Fiche technique** |
| 4 | POLOLU DRV8834 | Polulu | Creaton Inc. | DRV8834 Low Voltage Stepper Motor Driver - 2A | 13,83$ | <https://www.pololu.com/product/2134> | <https://www.pololu.com/file/0J617/drv8834.pdf> |
| 1 | 2070-OM-O2-ND | Onion Corporation | Digi-key electronics | Onion Omega2 Module | 15,97$ | <https://octopart.com/om-o2-onion-99318893> | <https://docs.onion.io/omega2-docs/omega2p.html> |
| 1 | 2070-OM-D-ARD2-ND | Onion Corporation | Digi-key electronics | Arduino Dock 2 | 26,60$ | <https://octopart.com/om-d-ard2-onion-99318876> | <https://docs.onion.io/omega2-docs/arduino-dock-2.html> |
| 4 | UA7805CKCS | Texas Instruments | Digi-key electronics | 1.5-A, 40-V fixed positive linear regulator 3-TO-220 0 to 125 | 0,87$ | <https://octopart.com/ua7805ckcs-texas+instruments-525328> | <https://octopart.com/click/track?ai=1165&ct=datasheets&sig=029cb9c&at=physicalpart&sid=370&ppid=525328&hlid=25213747> |
| 1 | 750-KBPC1004W-G | Comchip Technology | Mouser | Bridge Rect 1P 400V 10A Kbpc-w | 2,87$ | <https://octopart.com/kbpc1004w-g-comchip-22746349> | (aucune fiche technique) |
| 1 | TCT50-03E07K | Triad Magnetics | Digi-key electronics | Power Transformer 1500VAC 50VA 4 Terminal Quick Connect Flange Mount | 24,54$ | <https://octopart.com/tct50-03e07k-triad+magnetics-28565153> | <http://datasheet.octopart.com/TCT50-03E07K-Triad-Magnetics-datasheet-43795339.pdf> |

# Matériel utilisé

## Explications des composantes

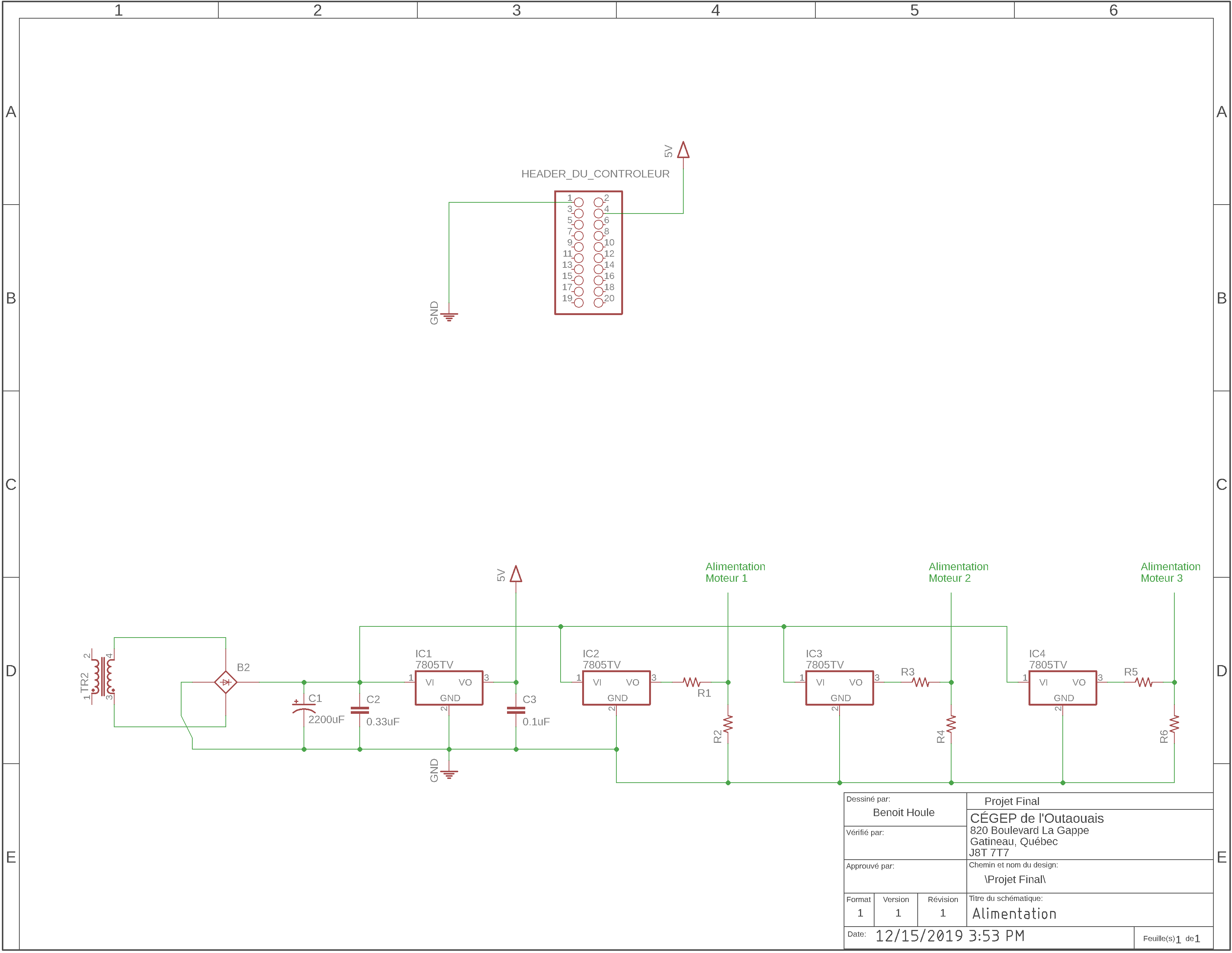
Pour mon contrôleur, j’ai choisi le Omega2 parce qu’il a tous ce dont j’ai besoin et plusieurs choses sont déjà fait. Il a un module Ethernet qui fonctionne dès que je le branche et je peux aussi le configurer comme dans la majorité ses systèmes linux. C’est aussi un appareil fait pour du IoT, alors il y a beaucoup d’outils pour faire des serveurs web facilement.

J’ai choisi un extension Arduino parce que je vais avoir besoin des pins GPIOs et aussi parce qu’il y a déjà des librairies pour contrôler des moteurs stepper. Une de ces librairies est celle qui va me permettre de contrôler l’accélération des moteurs.

Je vais utiliser un DRV8834 pour contrôler mes moteurs puisque avec cette composante, j’ai besoin de seulement une pin pour le PWM qui fait bouger un moteur, une pin pour la direction du moteur et optionnellement une ou deux autres pins pour contrôler les microsteps. La composante fait tout le reste du travail. Elle a en plus un limiteur de courant intégré configurable avec un petit potentiomètre.

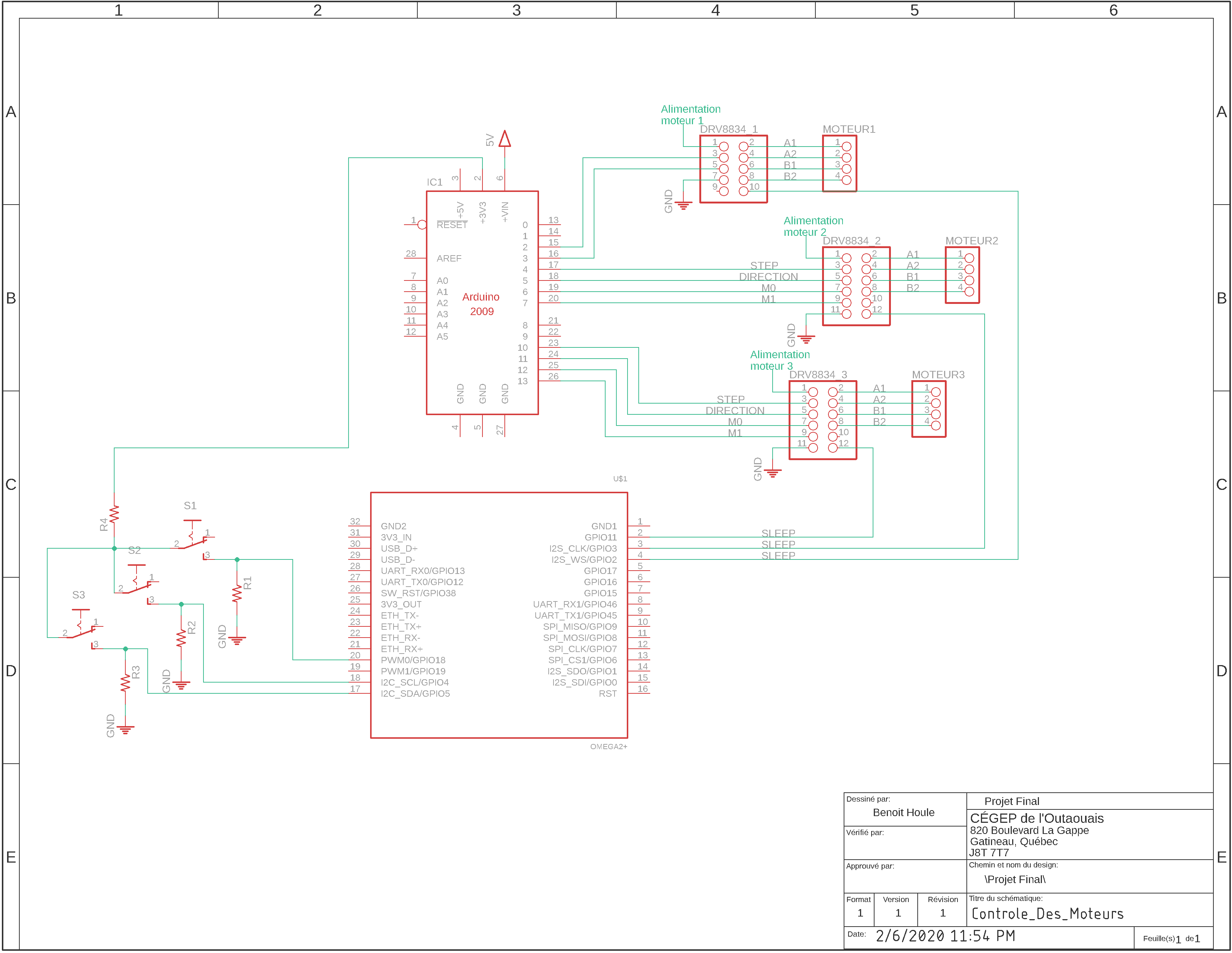
Pour l’alimentation, je transforme et stabilise ma tension pour avoir du 5V, que mon contrôleur a besoin, et ensuite j’utilise un diviseur de tension pour baisser jusqu’à 4.1V que les moteurs utilisent.

# Diagrammes schématiques



## Alimentation & contrôleur

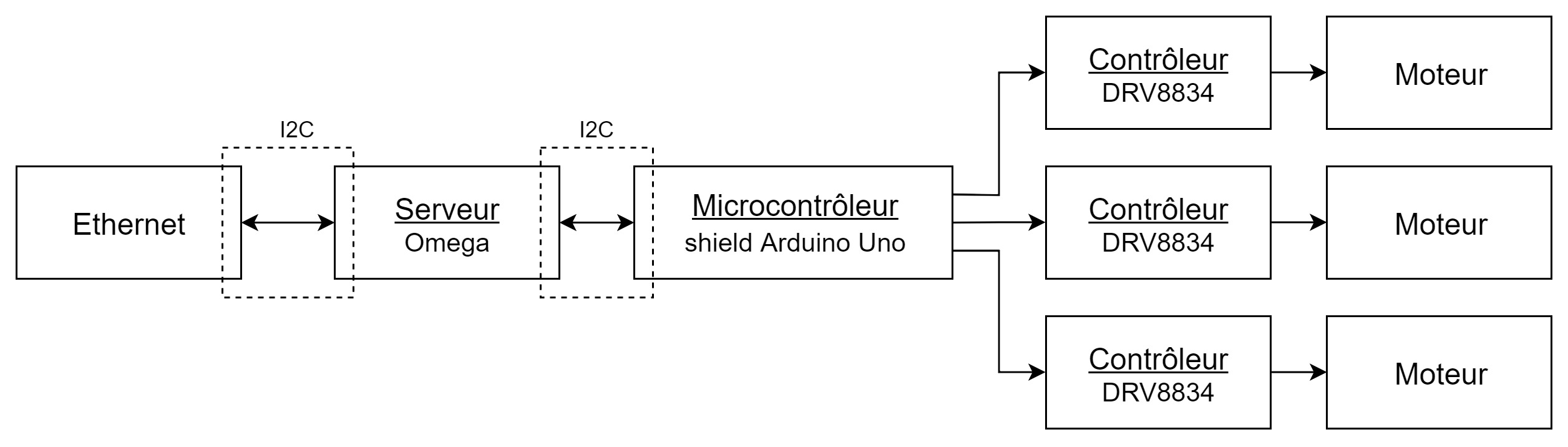
## 



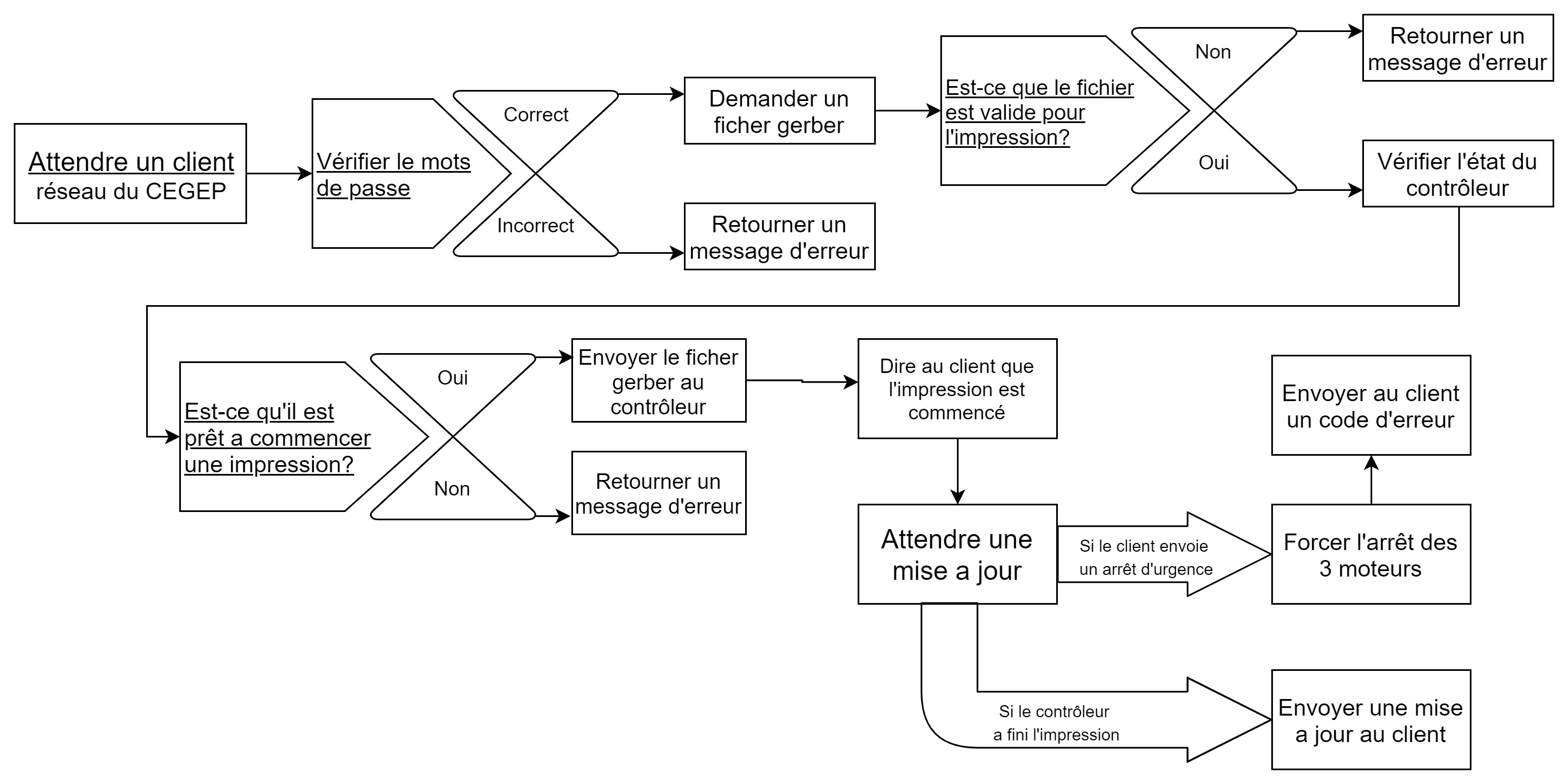
## Contrôle des moteurs

# Diagrammes fonctionnels

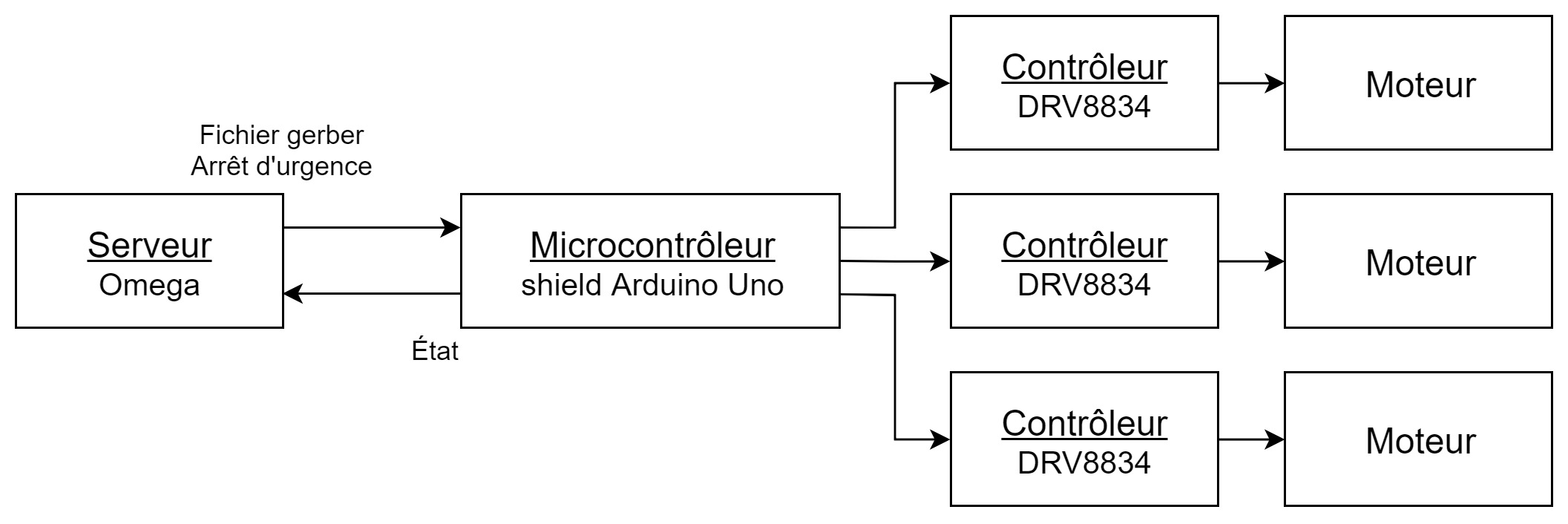
## Matériel



## Serveur



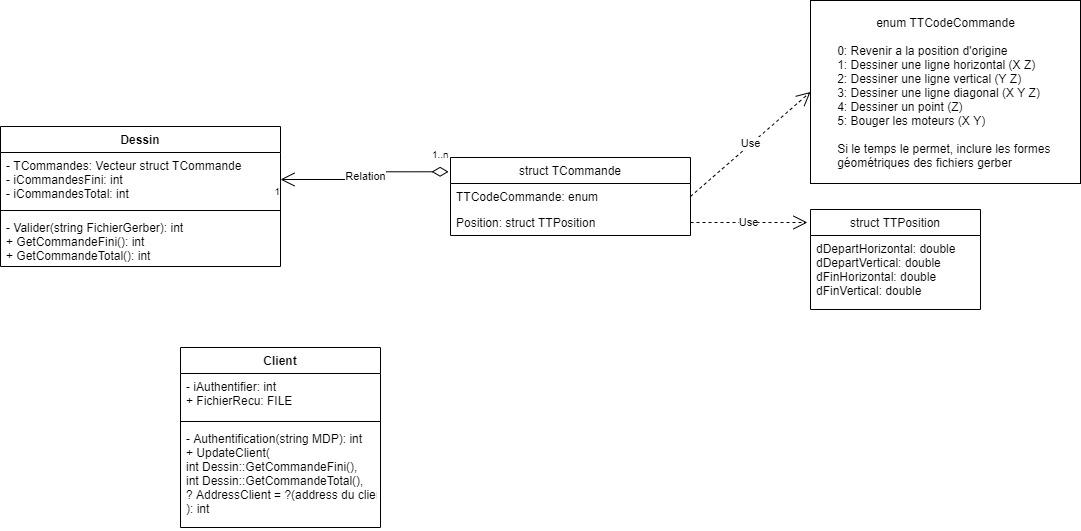
## Contrôleur



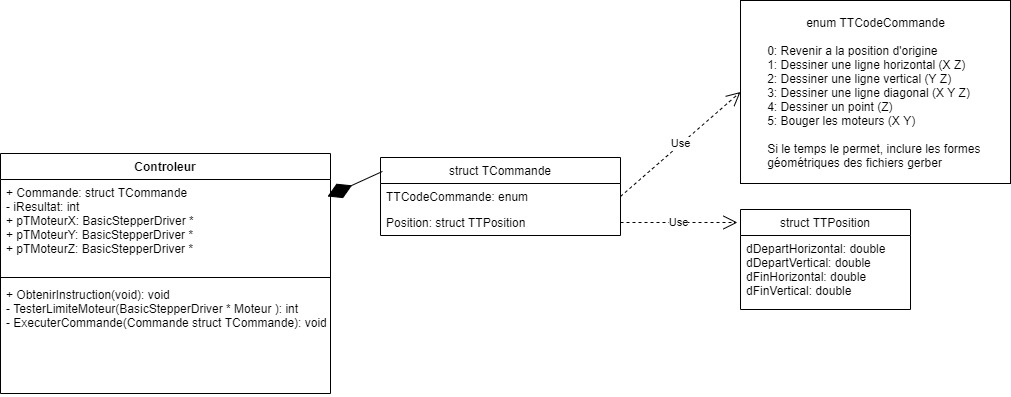
# Code

## UML

### Serveur



### Moteur



## En-tête personnel

### Contrôle des moteurs

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : ExecuterCommande

Paramètre

-Entrée : La commande à effectuer

-Sortie : -

Description : Effectue la commande désirer

Note : En gros, regarde le code de la commande pour savoir quel moteur faire bouger.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Si on veut dessiner une ligne horizontal

S’assurer que le moteur X est à la position de départ

Descendre le moteur Z

Bouger le moteur X vers la destination

Monter le moteur Z

Sinon, si on veut dessiner une ligne vertical

S’assurer que le moteur Y est à la position de départ

Descendre le moteur Z

Bouger le moteur Y vers la destination

Monter le moteur Z

Sinon, si on veut dessiner une ligne diagonal

S’assurer que les moteurs X et Y sont à leur position de départ

Descendre le moteur Z

Bouger les moteurs X et Y vers leur destination

Monter le moteur Z

Sinon, si on veut dessiner un point

S’assurer que les moteurs X et Y sont à leur bonne position

Descendre et monter le moteur Z

Sinon, si on veut reset tous les moteurs

S’assurer que le moteur Z est levé

Exécuter la fonction pour bouger les moteurs X et Y vers la position (0,0)

Sinon

Retourner un code d’erreur

FinDuSi

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : TesterLimiteMoteur

Paramètre

-Entrée : Le moteur

-Sortie : Code d’erreur

Description : Bouge un moteur à faible vitesse pour tester sa limite

Note : Démarre un moteur lentement et attend que le serveur lui dise que la limite a été atteinte. Fait cette opération deux fois pour être certain de la précision

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Mettre tous les moteurs à leur position de départ

Dire au serveur qu’il commence à bouger le moteur demandé

Tant qu’aucun message est reçu du serveur

Avancé le moteur d’un pas

Voir si le serveur nous envoie un message

Fin de la boucle pour faire bouger le moteur

Mettre le moteur à sa position de départ

Dire au serveur qu’il commence à bouger le moteur demandé

Tant qu’aucun message est reçu du serveur

Avancé le moteur d’un pas

Voir si le serveur nous envoie un message

Fin de la boucle pour faire bouger le moteur

Mettre le moteur à sa position de départ

### Serveur

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : Initialisation

Paramètre

-Entrée : -

-Sortie : Code d’erreur

Description : Initialisation à effectuer au démarrage

Note :

* Établie une connexion avec le contrôleur
* Teste la limite des moteurs
* Démarre le serveur

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Établir une connexion avec le contrôleur

Si la connexion a été établie

Démarrer le serveur normalement

Si la connexion n’a pas été établie

Démarrer le serveur en mode << sans contrôleur >>

Tant que la connexion n’a pas été établie avec le contrôleur

Attendre 1 seconde

Tenté de se connecter

FinDuTantQue

Dire au serveur que le contrôleur s’est connecter

FinDuSi

Tester la limite des moteurs X, Y et Z

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : VerifierAutorisation

Paramètre

-Entrée : Mot de passe envoyé par le client

-Sortie :

* Bon mot de passe (0)
* Mauvais mot de passe (1)
* Erreur (2)

Description : S’assurer que le client a l’autorisation d’utiliser ce matériel

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Décrypter le mot de passe reçu

Comparer le mot de passe avec

Si c’est le même

Traiter les données

Sinon

Renvoyer un code d’erreur

FinDuSi

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : ArretUrgence

Paramètre

-Entrée : -

-Sortie : Code d’erreur

Description : Force les 3 moteurs à arrêter

Note : Vu que c’est un arrêt d’urgence, c’est le serveur qui s’en charge. Comme ça, il n’y a aucun délai supplémentaire et moins de source d’erreur.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Activer le sleep mode sur les 3 moteurs.

Dire au contrôleur d’arrêter les moteurs

Envoyer une confirmation au client que l’arrête c’est bien fait.

Attendre un temps raisonnable pour que le mouvement des moteurs ait normalement le temps de se finir.

Réactiver les moteurs.

Envoyer au contrôleur la commande pour reset la position des 3 moteurs à leur position original.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fonction : ValiderFichier

Paramètre

-Entrée : Le fichier à valider

-Sortie :

* Fichier valide (0)
* Fichier invalide (Code d’erreur 1-\*)

Description : Détermine si le fichier reçu du client est valide

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Ouvrir le fichier Gerber en mode R/W

S’assurer qu’il y a au moins une commande pour définir la taille du crayon même si on ne l’utilise pas (Même si dans notre cas la taille du crayon ne change rien, il doit en avoir un pour un fichier valide. Ex : ADD10C,0.010\*)

S’assurer que la dernière commande est celle de fin de fichier (M02\*)

Si la taille du crayon n’est pas définie ou que la commande de fin est absente

Dire au client que le dessin peut être fait, mais qu’il peut y avoir des problèmes

Retourner un code d’erreur signifiant seulement un problème

FinDuSi

S’assurer qu’il y a au moins une commande pour dessiner (fini par D01\*)

S’il n’y a aucune commande pour dessiner

Dire au client qu’aucune ligne a été détecter dans le fichier envoyé

Retourner un code d’erreur pour une erreur fatal

FinDuSi

## En-tête d’autres librairies

### Contrôle des moteurs

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Librairie : DRV8834.h

Fonction : stepper

Paramètre

-Entrée :

* Nombre de pas pour faire un tour complet
* Pin pour la direction du moteur
* Pin pour les pas du moteur
* Pin pour activer ou désactiver le moteur (optionnel)
* Une des deux pins pour contrôler le microstepping (optionnel)
* L’autre des deux pins pour contrôler le microstepping (optionnel)

-Sortie : Objet pour contrôler un moteur

Description : Permet de contrôler un moteur à travers un DRV8834

Note : A utiliser avec la librairie MultiDriver

Source : <https://github.com/laurb9/StepperDriver/tree/master/src>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Librairie : MultiDriver.h

Fonction : MultiDriver

Paramètre

-Entrée :

* Moteur 1
* Moteur 2

-Sortie : Objet pour contrôler deux moteurs

Description : Permet de contrôler deux moteurs en même temps

Note : A utiliser avec la librairie SyncDriver

Source : <https://github.com/laurb9/StepperDriver/tree/master/src>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Librairie : SyncDriver.h

Fonction : startMove

Paramètre

-Entrée :

* Nombre de pas du moteur 1
* Nombre de pas du moteur 2

-Sortie : -

Description : Bouge les deux moteurs pour qu’ils finissent en même temps

Note : A utiliser pour faire des lignes diagonales

Source : <https://github.com/laurb9/StepperDriver/tree/master/src>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/