

Tutoriel pour câbler les composants de la ZG2

Bonjour cher.e étudiant.e, bienvenue dans ce tutoriel qui permet de vérifier le fonctionnement de chaque composant et de savoir comment les câbler. Avant toute chose, tu auras besoin de la datasheet des composants et bien évidemment des composants. **Fais bien attention aux précautions lorsqu'il y en aura !**

Voici le matériel dont tu as besoin :

- 1 carte arduino R3 board et son câble usb
- 1 carte arduino motor shield
- 50:1 Metal Gearmotor 37Dx70L mm 12V with 64 CPR Encoder (Pololu) (moteur item 4753 + encodeur)
- Pololu Digital Distance Sensor 200cm (capteur)
- Câbles
- 1 oscilloscope
- 1 générateur de tension continue
- 1 multimètre ou ampèremètre

Objectifs :

- Alimentation du moteur avec le générateur
- Alimentation et visualisation des données de l'encodeur et du capteur sur l'oscilloscope
- Câblage des composants sur la carte arduino et son motor shield
- Visualisation des données des composants sur un port série

Alimentation du moteur :

Nous souhaitons commander le moteur à l'aide de la carte arduino et de la carte de puissance qui se monte sur la première. La carte de puissance nous permet d'utiliser une alimentation extérieure pour alimenter un ou deux moteurs. Avant de le faire, nous vérifierons que nous ne dépassons pas le courant max admissible par la carte.

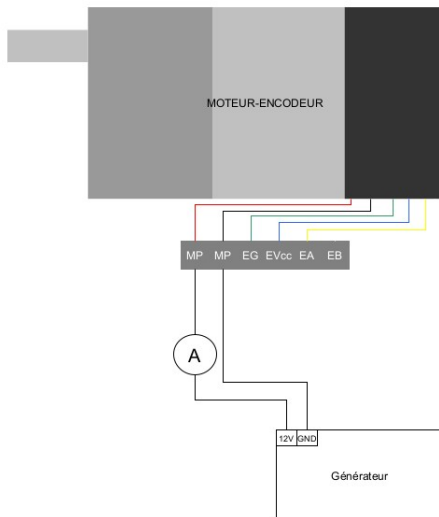
Selon la datasheet de la carte de puissance (motor shield arduino), nous ne pouvons pas dépasser 4A :

TECH SPECS

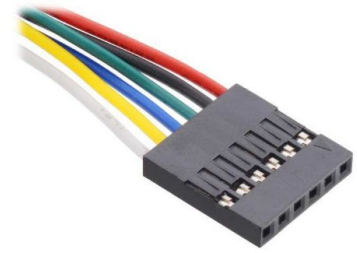
Operating Voltage	5V to 12V
Motor controller	L298P, Drives 2 DC motors or 1 stepper motor
Max current	2A per channel or 4A max (with external power supply)
Current sensing	1.65V/A
Free running stop and brake function	

La carte nous indique également que nous ne pouvons pas dépasser les 12V pour alimenter notre moteur.

Avant de câbler le moteur sur la carte, nous pouvons le tester en l'alimentant sur le générateur. Un moteur consomme beaucoup plus d'énergie au démarrage, alors nous allons contrôler ce pic d'intensité en veillant à brancher les câbles électriques aux bons connecteurs (bleus et noirs) :



Lead Color	Function
Red	Motor power
Black	Motor power
Green	Encoder ground
Blue	Encoder Vcc (3.5 V to 20 V)
Yellow	Encoder A output
White	Encoder B output



Règle le multimètre sur « peak » et allume le générateur.

Le pic d'intensité sans charge du moteur est d'environ $I_{\max}=2,13A$. La datasheet du moteur indique un pic d'intensité maximal de 5,5A (avec charge), dans notre cas nous pouvons utiliser la carte car $I_{\max} < 4A$.

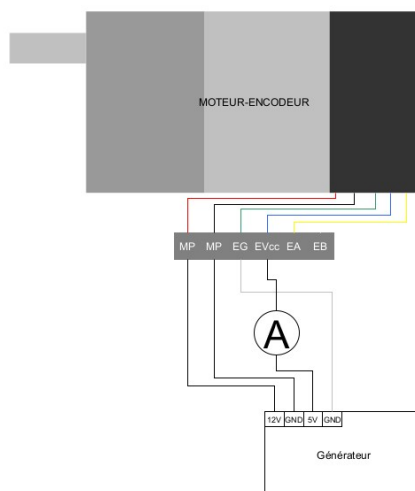
En inversant le + et le – du générateur, on change le sens de rotation du moteur.

Alimentation de l'encodeur :

Nous souhaitons alimenter l'encodeur directement avec la carte arduino, **pour le moment nous effectuerons les tests avec le générateur**. Comme la tension de l'encodeur doit être entre 3,5V et 12V, nous choisissons d'utiliser le pin 5V de la carte arduino. L'intensité maximale du pin 5V que peut fournir la carte n'est pas indiquée, par précaution $I_{\max}= 20mA$ (intensité maximale d'une entrée/sortie).

La datasheet de l'encodeur nous indique que le courant maximal pour l'alimenter est de 10mA :

The Hall sensors require an input voltage, V_{cc} , between 3.5 V and 20 V and draw a maximum of 10 mA



(Les couleurs des fils indiquent sur quel port du moteur-encodeur brancher)

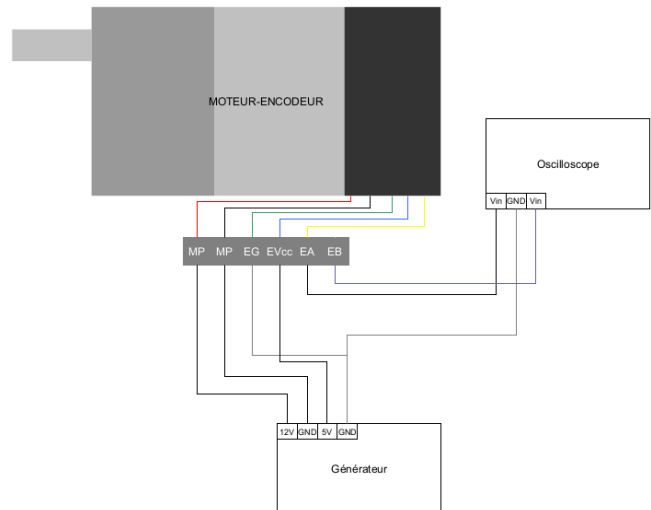
Toujours en utilisant la fonction pic, l'ampèremètre indique : $I_{\max}= 8,17mA$ et sans la

fonction peak : $I_{moy} = 8,15\text{mA}$. Nous pourrons donc alimenter le moteur sur la carte.

Visualisation des données de l'encodeur :

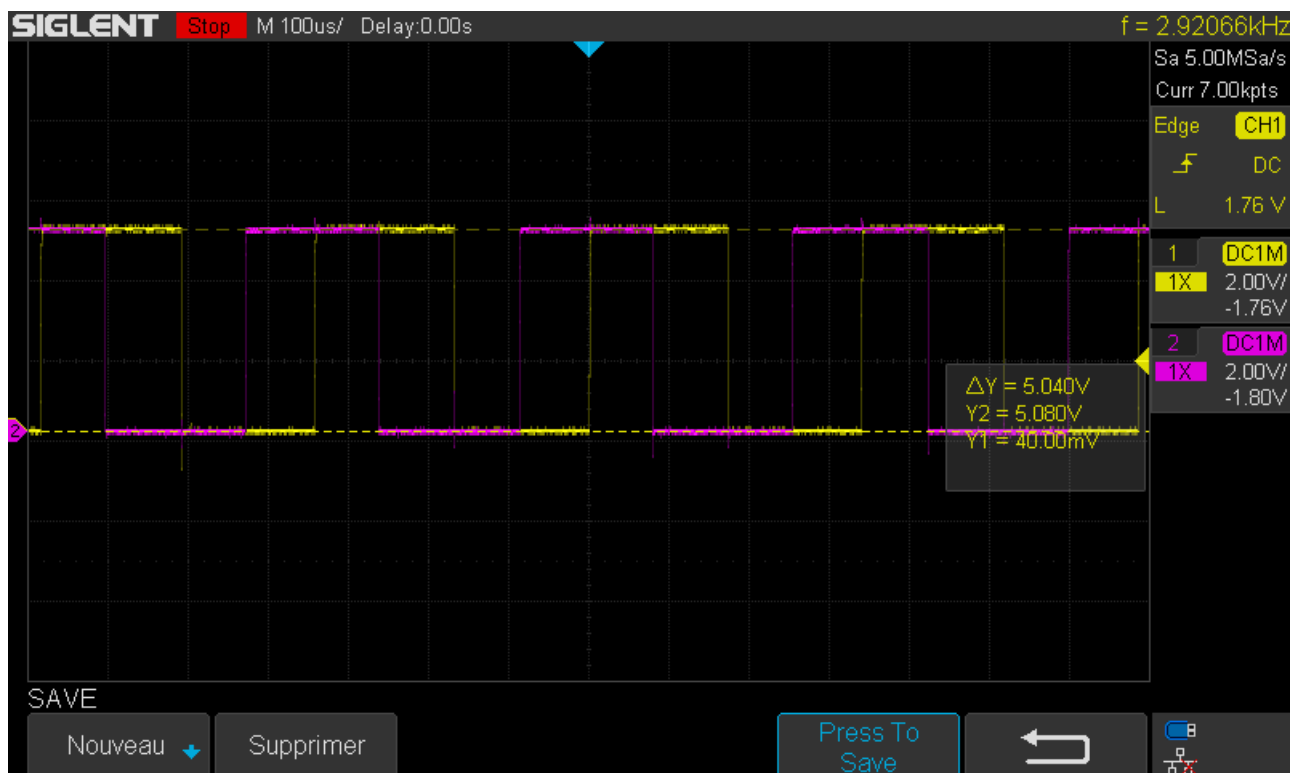
Pour visualiser les données de l'encodeur, nous aurons besoin de l'oscilloscope. L'encodeur nous permettra de connaître la vitesse de rotation et si besoin le sens de rotation du moteur.

Voici le schéma électrique :



Après avoir câblé l'ensemble, vous pouvez désormais régler l'oscilloscope et observer les sorties de l'encodeur. En inversant le + et le – du moteur on inverse son sens de rotation :

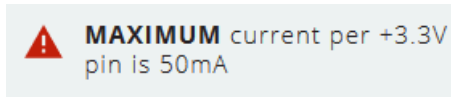
Cas du moteur tournant dans le sens inverse



(En jaune l'encodeur A et mauve l'encodeur B)

Alimentation du capteur :

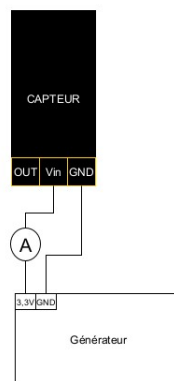
Nous souhaitons également alimenter directement le capteur sur la carte arduino, **pour le moment nous effectuerons les tests avec le générateur**. Comme la tension d'alimentation du capteur doit être entre 3V et 5,5V, nous choisirons d'utiliser le pin 3,3V de la carte. Selon la datasheet de la carte, le pin 3,3V peut fournir une intensité maximale de 50mA :



Quant à celle du capteur, il y est indiqué que le courant maximal est 30mA :

Maximum range:	200 cm
Sampling rate:	30 Hz ¹
Minimum operating voltage:	3.0 V
Maximum operating voltage:	5.5 V
Supply current:	30 mA ²
Output type:	digital ³

Voici le schéma électrique :

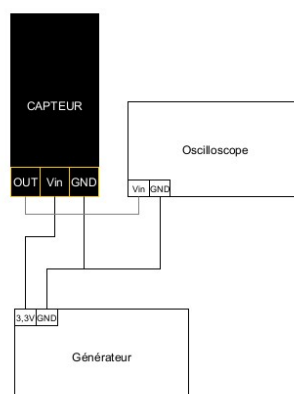


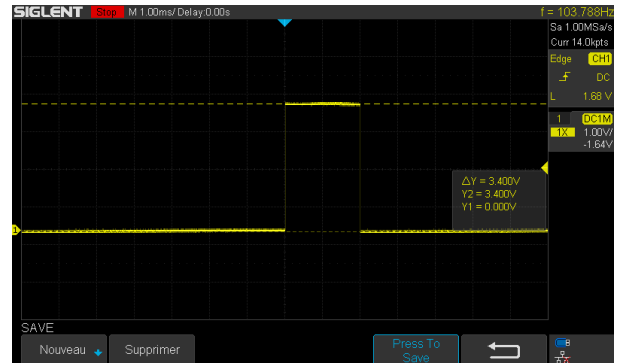
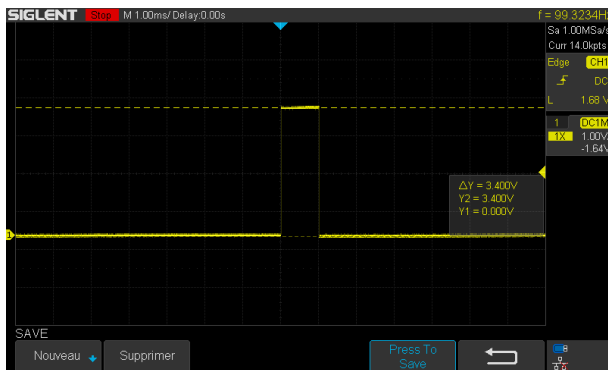
Toujours en utilisant la fonction pic, l'ampèremètre indique : $I_{\max} = 32\text{mA}$ et sans la fonction peak : $I_{\text{moy}} = 24\text{mA}$. Nous pourrions donc alimenter le moteur sur la carte.

Visualisation des données du capteur :

Le capteur que nous utilisons est un capteur à infrarouge. Le principe est le suivant : plus l'objet sera éloigné, plus le temps que met la lumière pour parcourir cette distance sera long. Donc plus l'objet sera loin plus la largeur de la porte du signal sera grande.

Voici le schéma électrique afin de visualiser le signal :





(À gauche à distance minimale et à droite distance maximale)

Nous savons désormais qu'on peut utiliser la carte pour alimenter les capteurs et comment ils fonctionnent. Nous pourrions récupérer leurs signaux afin d'afficher les informations dont nous avons besoin pour notre système qui sera asservi. Avant toute manipulation avec les cartes, il est important de prendre quelques précautions pour ne pas les endommager (les cartes ont un certain coût).

Précautions à prendre :

- Fais attention à ton câblage, vérifie bien sur quel pin se branche chaque fil.
- Laisse la carte arduino sur son support plastique. En prenant la carte dans tes mains lors de son utilisation, tu peux créer un court circuit sur la carte en reliant 2 pins (le corps est conducteur).
- Ne pas débrancher la liaison USB (carte arduino uno - PC)

Montage de la carte arduino-motorshield et câblage des composants sur la carte :

Avant de lier les 2 cartes nous devons choisir les pins que nous utiliserons pour piloter le moteur et récupérer les informations des composants. Pour le moteur, nous utiliserons le canal A. Voici les pins que tu utiliseras :

Moteur :

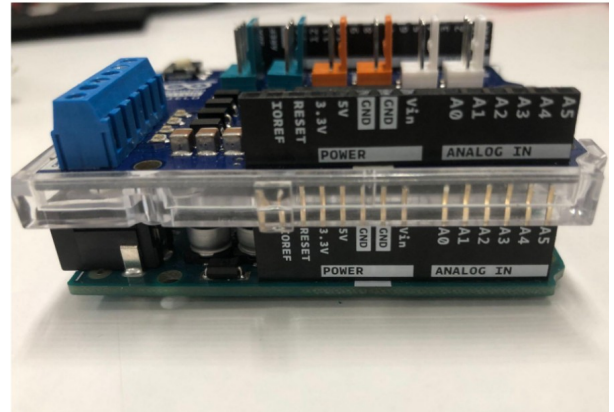
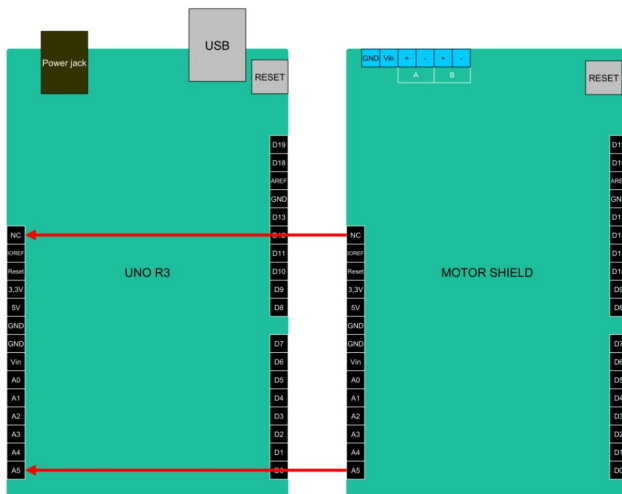
La datasheet nous indique que si on branche le moteur au canal A, plusieurs pin lui sont assignés pour le contrôler :

D12 : Choisir le sens de rotation du moteur
D3 : Signal PWM pour contrôler la vitesse de rotation
D9 : Activer/désactiver le frein moteur

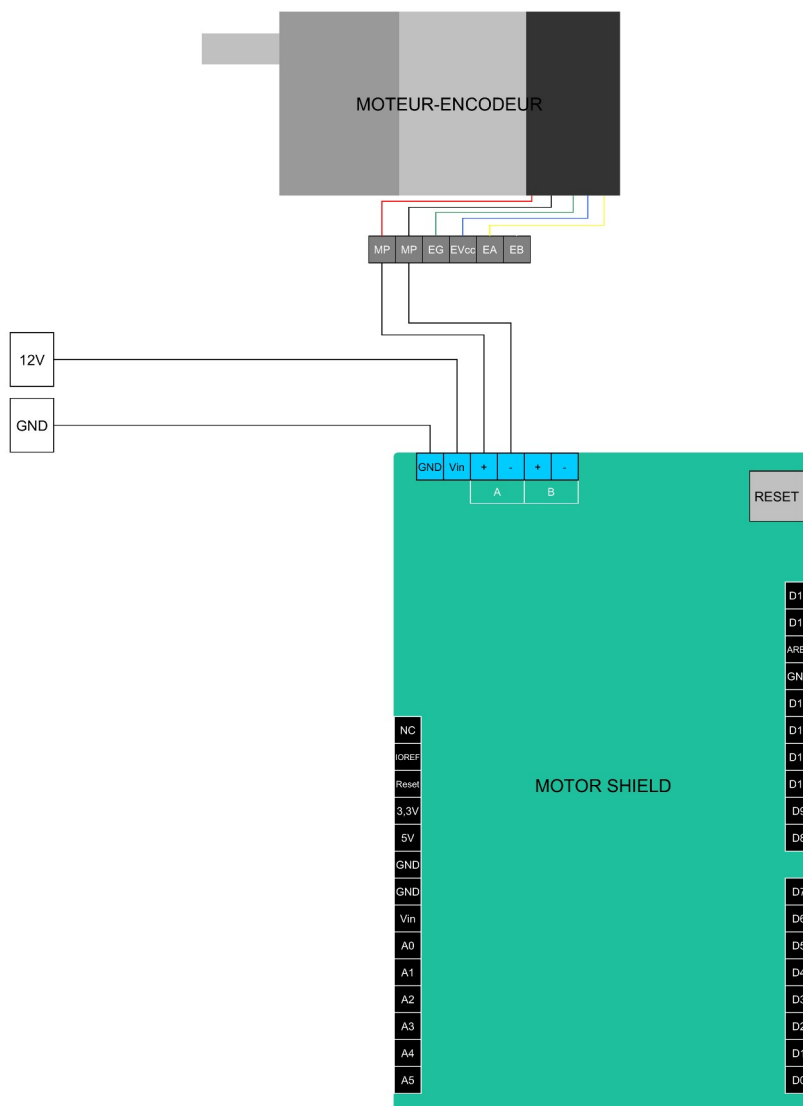
Function	pins per Ch. A	pins per Ch. B
<i>Direction</i>	D12	D13
<i>PWM</i>	D3	D11
<i>Brake</i>	D9	D8
<i>Current Sensing</i>	A0	A1

Nous pouvons déjà tester un programme pour faire tourner le moteur dans un sens si on envoie le caractère « a » dans un terminal série.

Commençons par assembler les 2 cartes, on assemble la carte de puissance motor shield sur la carte arduino en faisant correspondre les pins tel que :



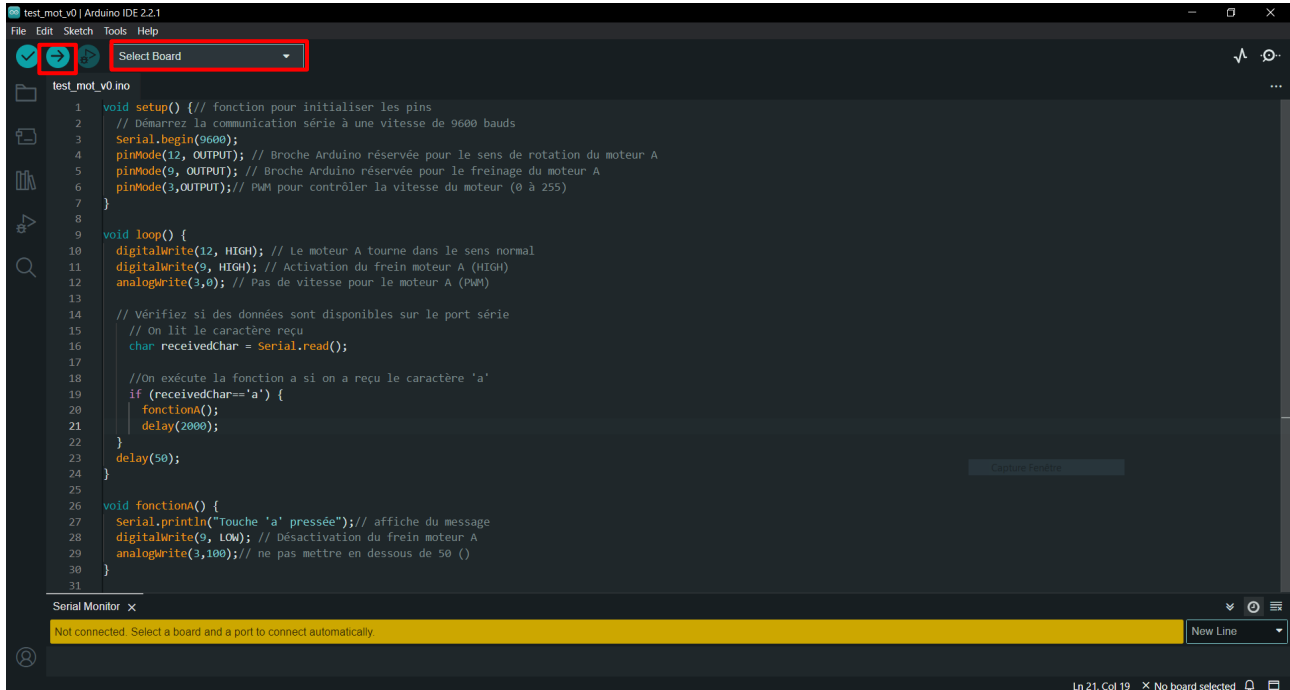
À présent, nous allons câbler le moteur et l'alimentation sur la carte, voici le schéma électrique (les schémas montreront uniquement la carte motor shield mais celle-ci doit rester montée sur la carte arduino uno) :



Test moteur :

Après avoir câblé, tu trouveras dans le dossier 3-code_arduino → test_mot_v0, le code à lancer. Tu devras ouvrir ce fichier avec l'IDE arduino (version 2.2.1).

Tu peux brancher l'arduino au port USB du PC.



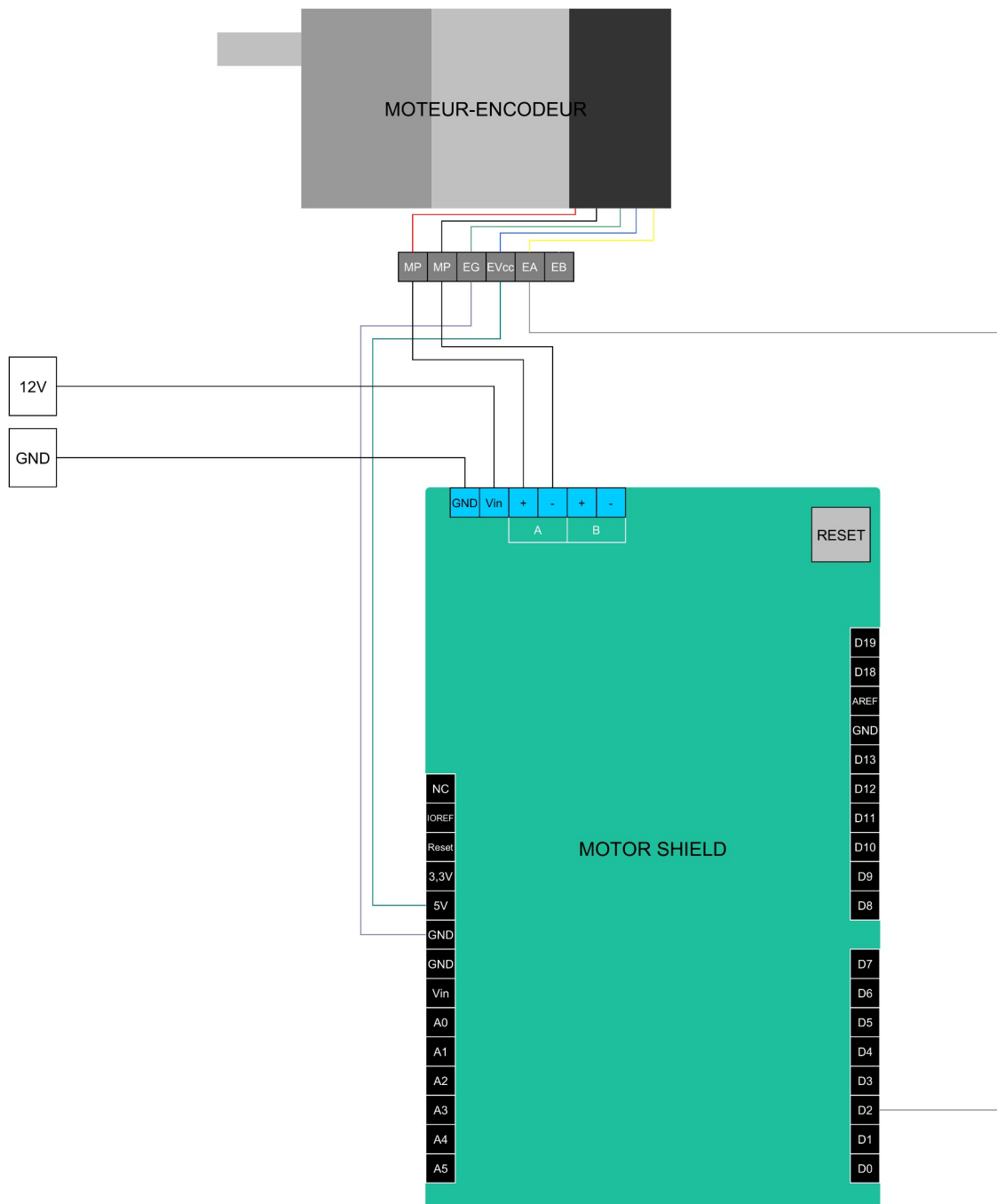
Sélectionne le port sur lequel se trouve la carte et transfère-le sur l'arduino.

Le moteur se met en route uniquement si nous envoyons le caractère a dans le terminal série, pour cela clique sur Tools dans la barre des menus et sur Serial Monitor.

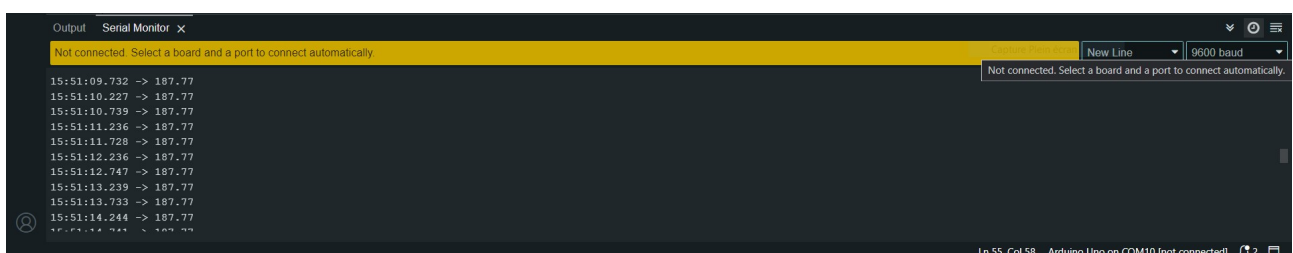
Appuie sur la touche a et sur la touche entrer, le moteur tourne pendant 2 secondes !

Test encodeur :

Voici le câblage de l'encodeur et du moteur :



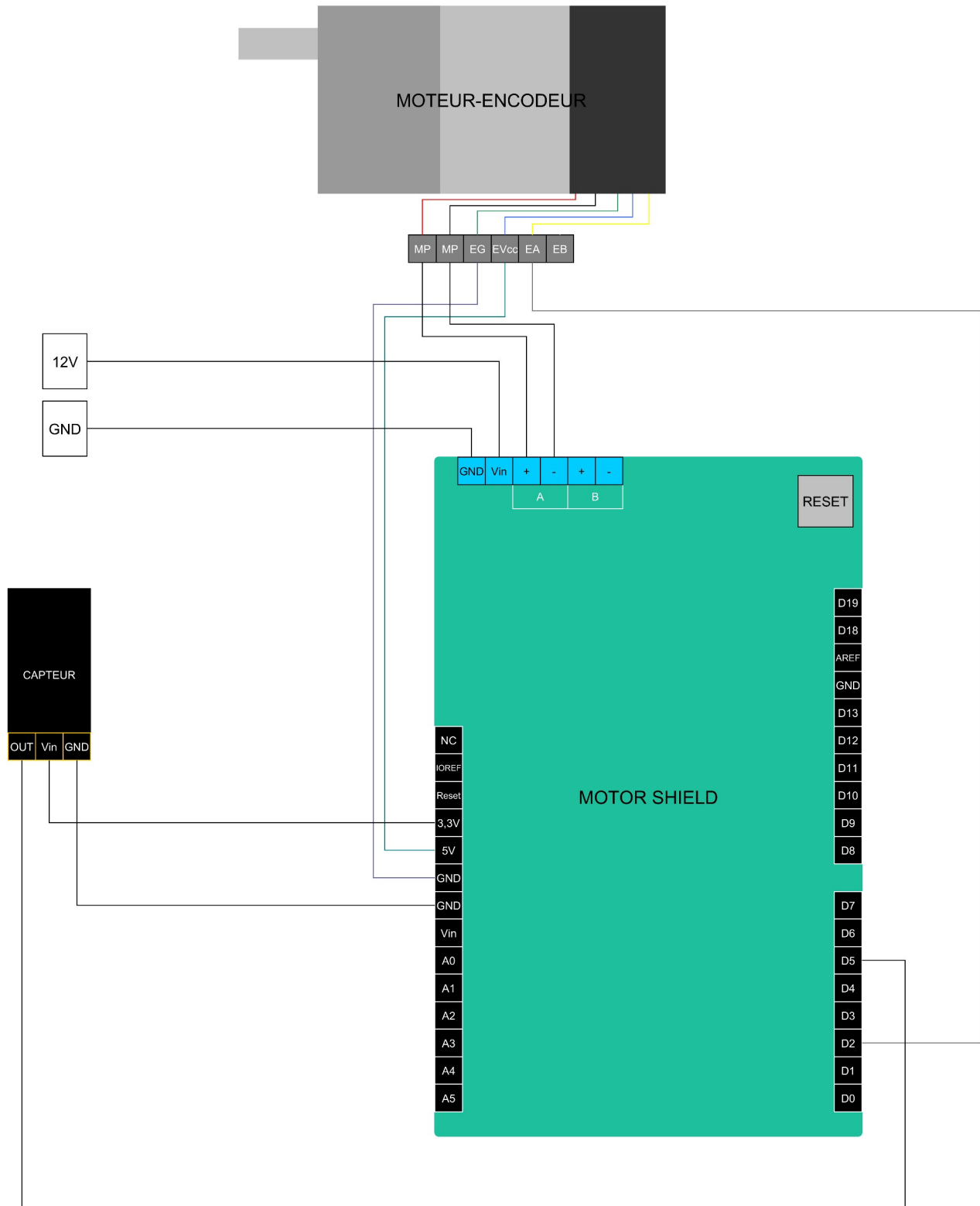
Après avoir effectué le montage tu peux ouvrir le fichier `test_mot_v1.ino` et le transférer dans l'arduino. Nous pouvons visualiser dans le terminal série, la vitesse du moteur tel que :



Test capteur :

À présent nous allons afficher la vitesse de l'encodeur et la distance du capteur dans le terminal série.

Voici le câblage électrique :



Ouvre le fichier *test_mot_v2.ino* et transfère-le dans la carte.

Cette fois-ci le programme affiche la vitesse du moteur ainsi que la distance de l'objet détecté.

Tu as désormais fini le câblage final.

Version finale :

Ouvres le fichier 3-code_arduino→code_final.ino et transfère-le dans la carte.

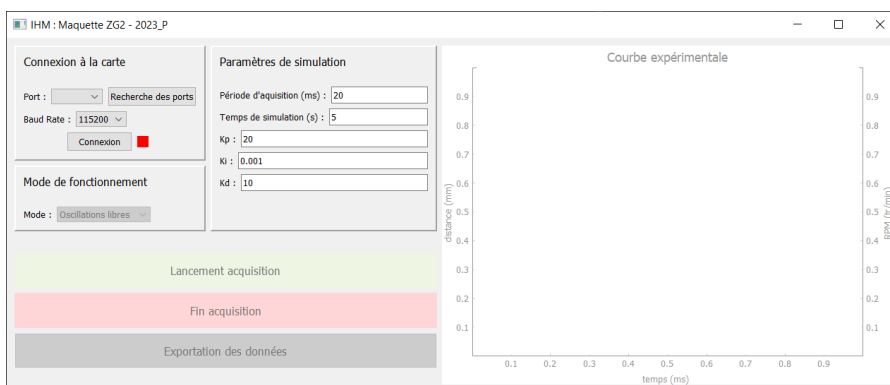
Précautions :

- **Ne pas débrancher de câble lorsque qu'une acquisition est en cours.**
- **Attendre quelques secondes entre chaque acquisition.**
- **Appuie sur le bouton vert de la plaque et vérifie que la led verte s'est allumée.**

Tu peux maintenant ouvrir l'IHM en ouvrant le 2-IHM→main.exe et attendre quelques instants le temps que l'IHM s'ouvre. Deux fenêtres apparaissent, une pour visualiser les données sur un terminal et l'autre l'IHM. Garde les 2 fenêtres ouvertes.

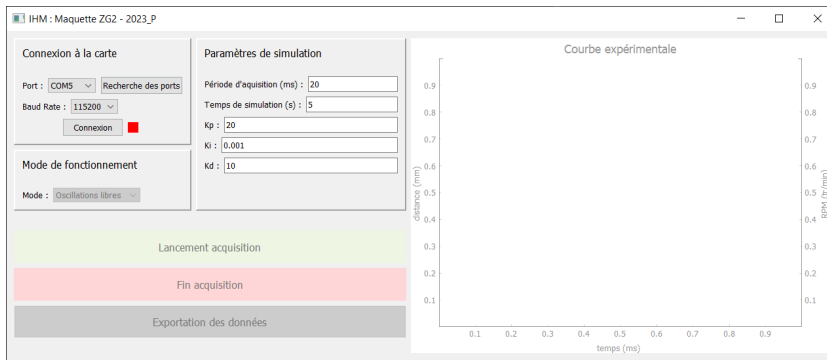
Voici les étapes à suivre :

Étape 1 :



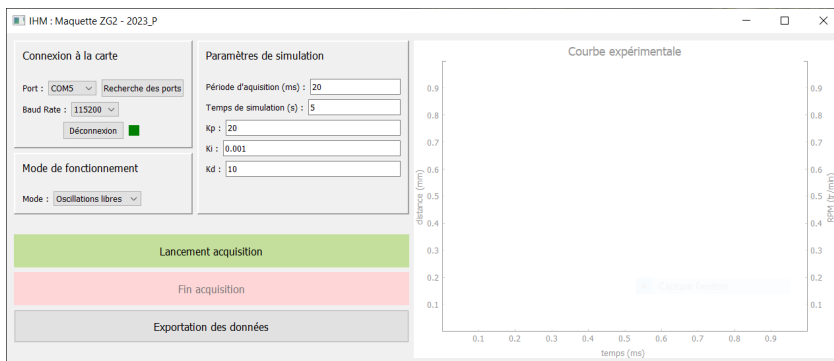
Pour que l'IHM se connecte sur un port série, appuie sur le bouton recherche des ports.

Étape 2 :



Un port contenant COMX doit apparaître dans la liste si tu es sur windows ou ACMX sur linux.

Étape 3 :

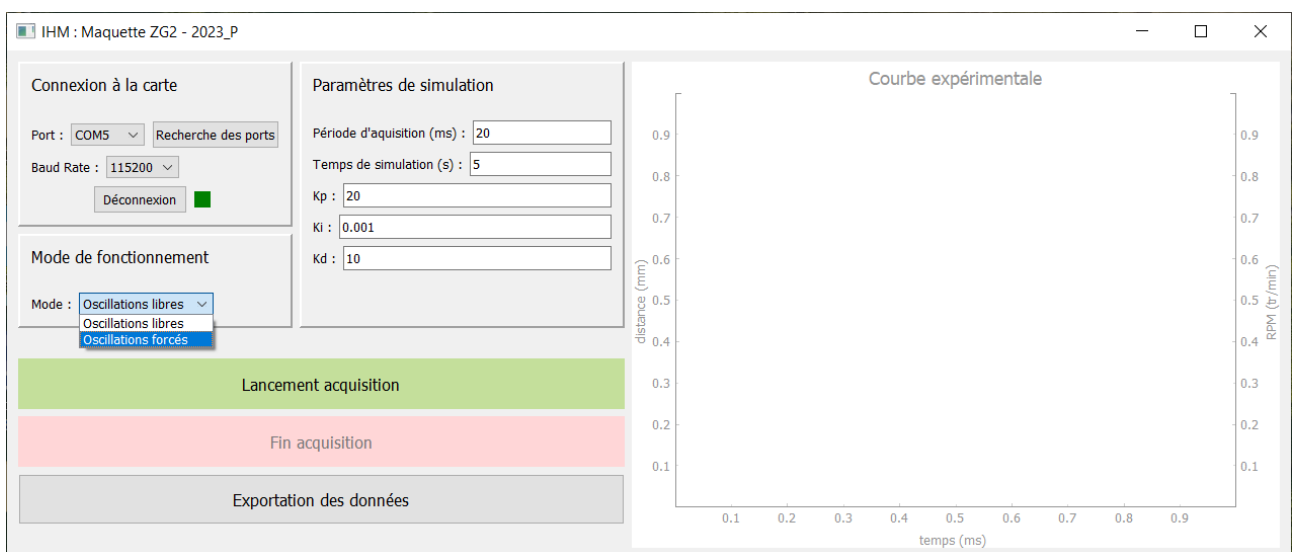


Appuie sur le bouton connexion et vérifie que le bouton est passé au vert.



Vérifie qu'une trame de données a bien été lue par l'IHM.

Par défaut, lorsqu'on lance l'IHM, le mode est sur Oscillation libre, pour la changer, clique sur la liste Mode et sélectionne Oscillation forcée pour changer de mode :



Tu peux également stopper une acquisition en cours en appuyant sur « Fin acquisition »

Réglages des différents paramètres :

Période d'acquisition : Ne pas descendre en dessous de 10ms

Coefficient du correcteur conseillé :

$k_p=20, k_i=0,1$ et $k_d=1$ **entre 15 et 120 tr/min**

Modifier les valeurs des coefficients **entre 120 et 150 tr/min**

Vitesse en tr/min : La vitesse du moteur doit être entre 15 et 150 tr/min

Tu peux aussi tester la version en BO pour comparer avec la BF.

Convertir un fichier .py en .exe sous windows (Pyinstaller)

- vous pouvez réaliser toutes le manip dans le terminal de VS Code
- vérifier que le code fonctionne correctement et que toutes les librairies nécessaires sont installées
- si besoin installer les librairies manquantes (ex: pyserial, pyqtgraph, PyQt5)
- vérifier que pip est installé (sinon modifier vos paramètre d'installation du logiciel python <https://www.youtube.com/watch?v=UZX5kH72Yx4&t=71s>)
- ouvrir l'application "invite de commande" en cherchant "cmd" dans la barre de recherche de windows puis installer pyinstaller grâce à la commande **pip install pyinstaller** (<https://pyinstaller.org/en/stable/installation.html#installing-from-the-source-archive>)
- se positionner dans le répertoire qui contient votre fichier .py grâce à la commande **cd + le chemin d'accès au répertoire**
- Convertire le fichier .py en fichier .exe grâce à la commande **pyinstaller --onefile main.py** (ici main.py est le nom du fichier à convertir, vous devez mettre le nom de votre fichier à convertir)