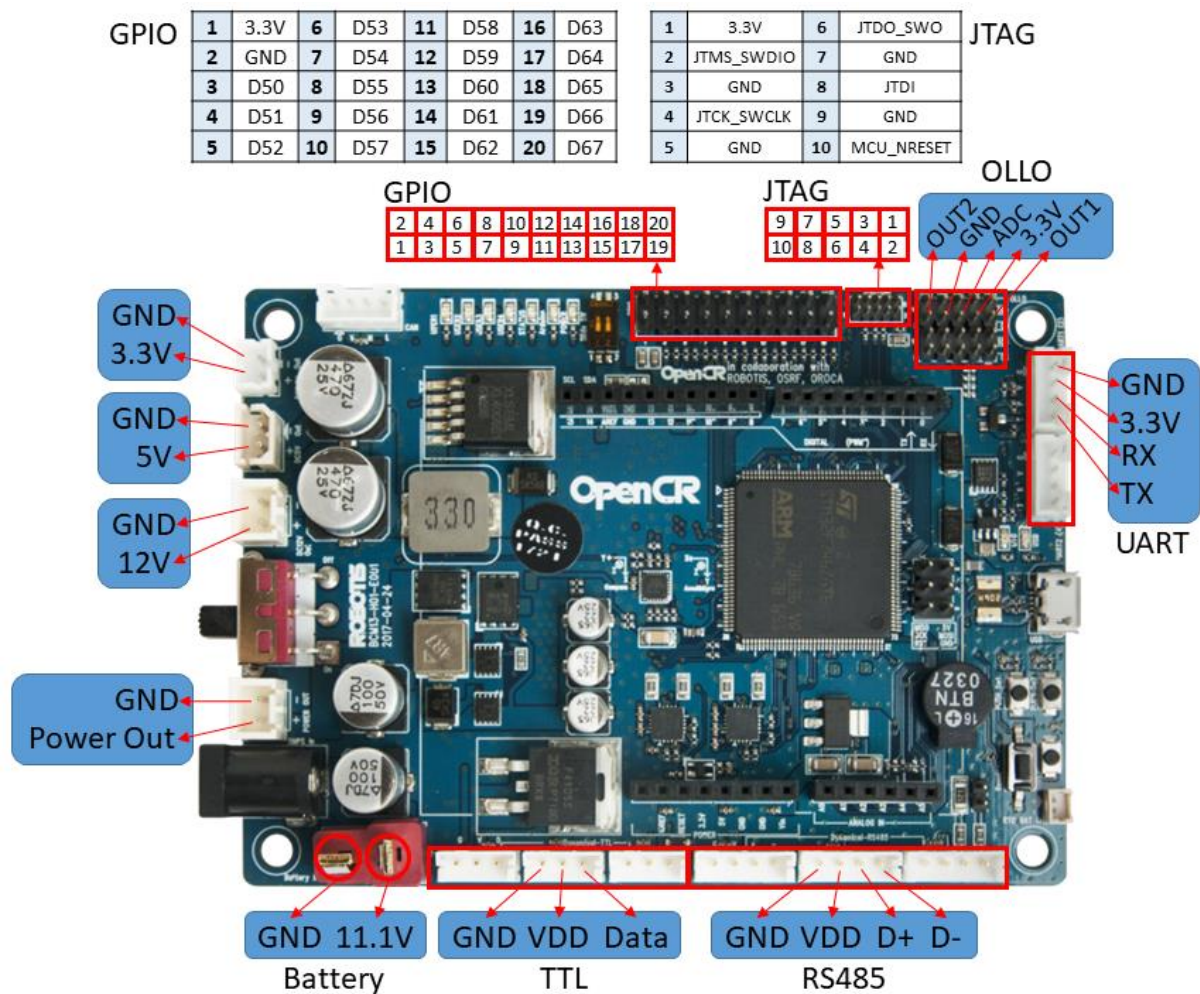


Doc Utilisation Moteur Saé

Table des matières

| | |
|--|---|
| Doc Carte Open CR Saé..... | 1 |
| Avec Arduino :..... | 2 |
| Ajouter la carte Open CR..... | 2 |
| Utilisation Pour STM32 Initialisation des moteurs..... | 7 |
| Utilisation avec une carte NUCLEO | 8 |

Pour une utilisation avec la carte STM32, vous pouvez aller directement à : Utilisation Pour STM32 Initialisation des moteurs



Pin Définition

Tout d'abord initialiser les moteurs pour faire la programmation de bas niveaux

Ne pas tenir compte si la carte Open CR n'est pas utilisé :

Open CR Test

D'abords faire l'initialisation de l'open CR et lancer le programme pour les moteurs et leurs initialisations

TurtleBot3 DYNAMIXEL setup instructions

Programmation de bas niveaux (directement sur l'open CR)

Avec Arduino :

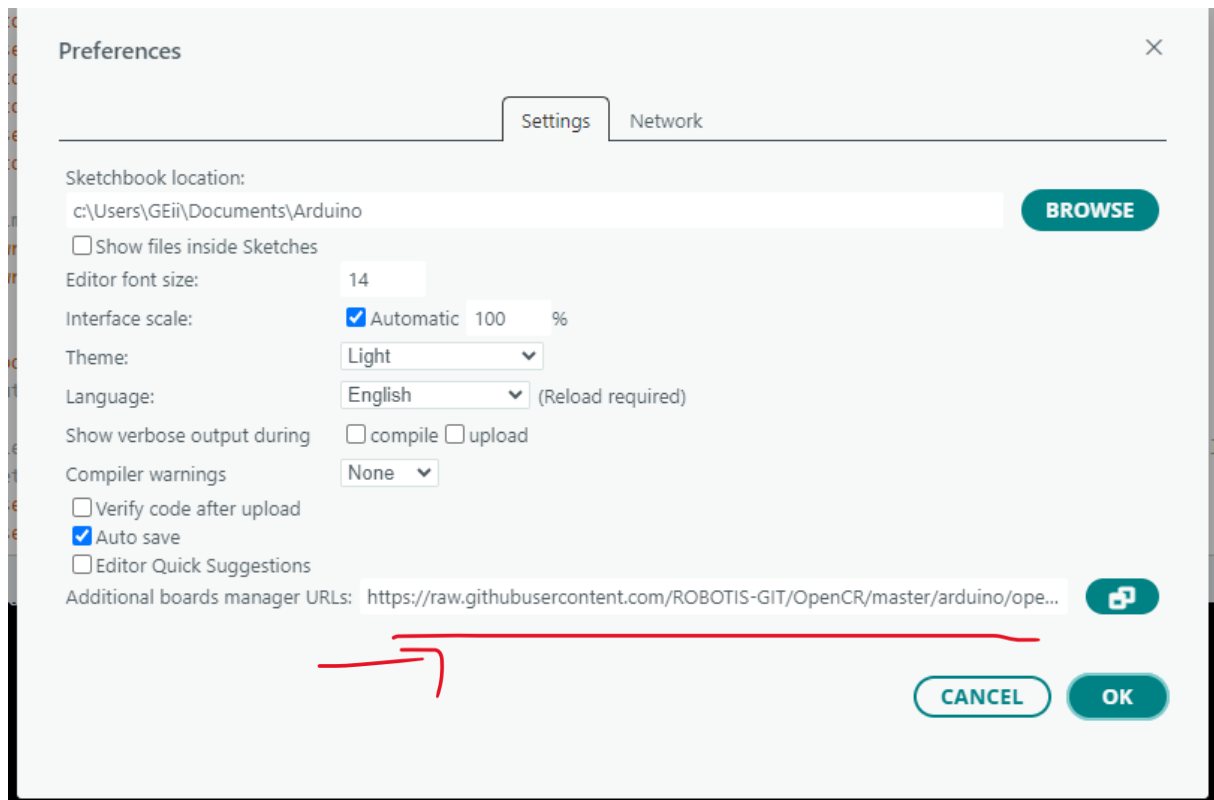
Installer Arduino IDE

<https://www.arduino.cc/en/Main/Software>

Ajouter la carte Open CR

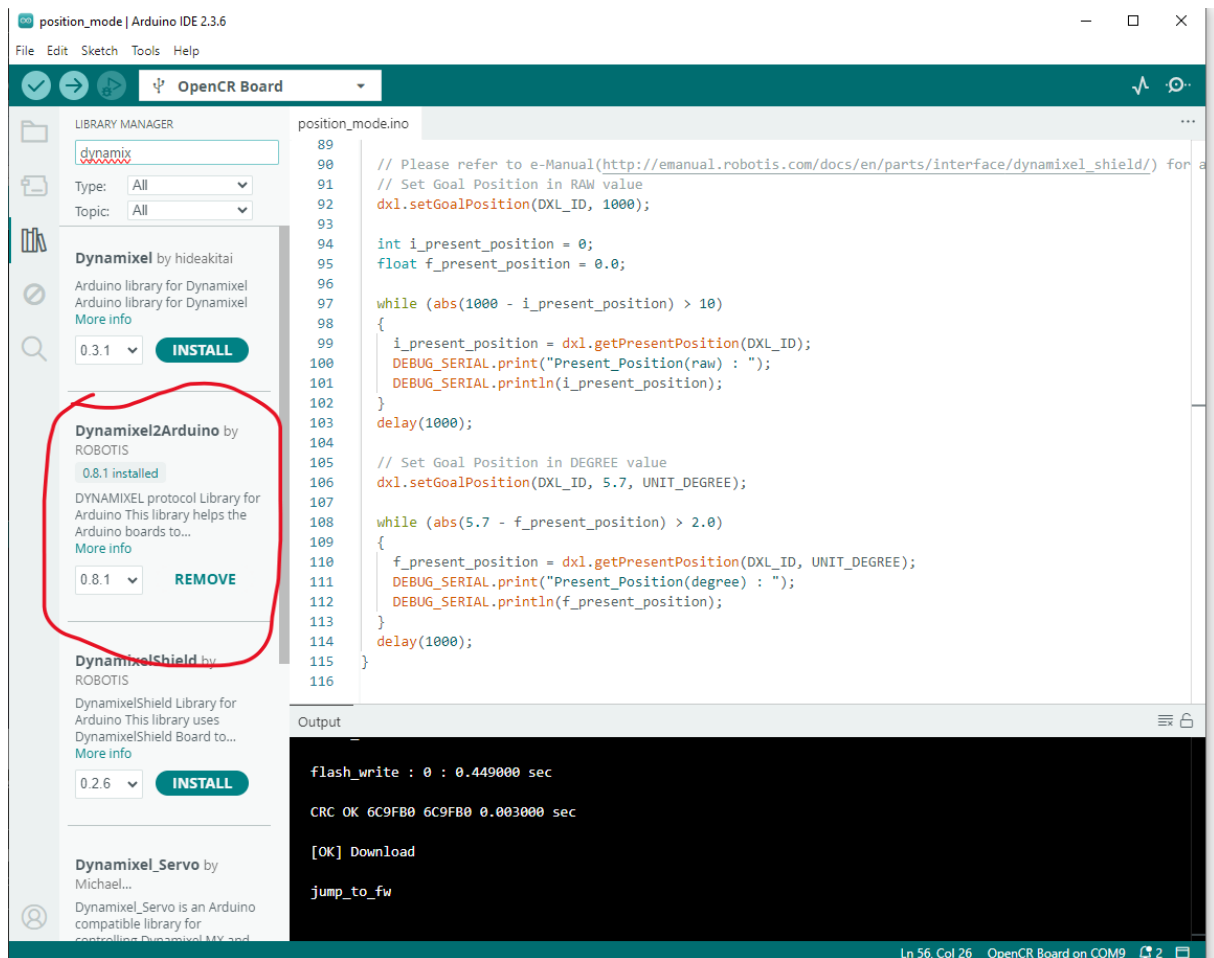
Dans File → Preferences, ajouter l'URL suivante dans *the Additional Boards Manager URLs* :

https://raw.githubusercontent.com/ROBOTIS-GIT/OpenCR/master/arduino/opencr_release/package_opencr_index.json

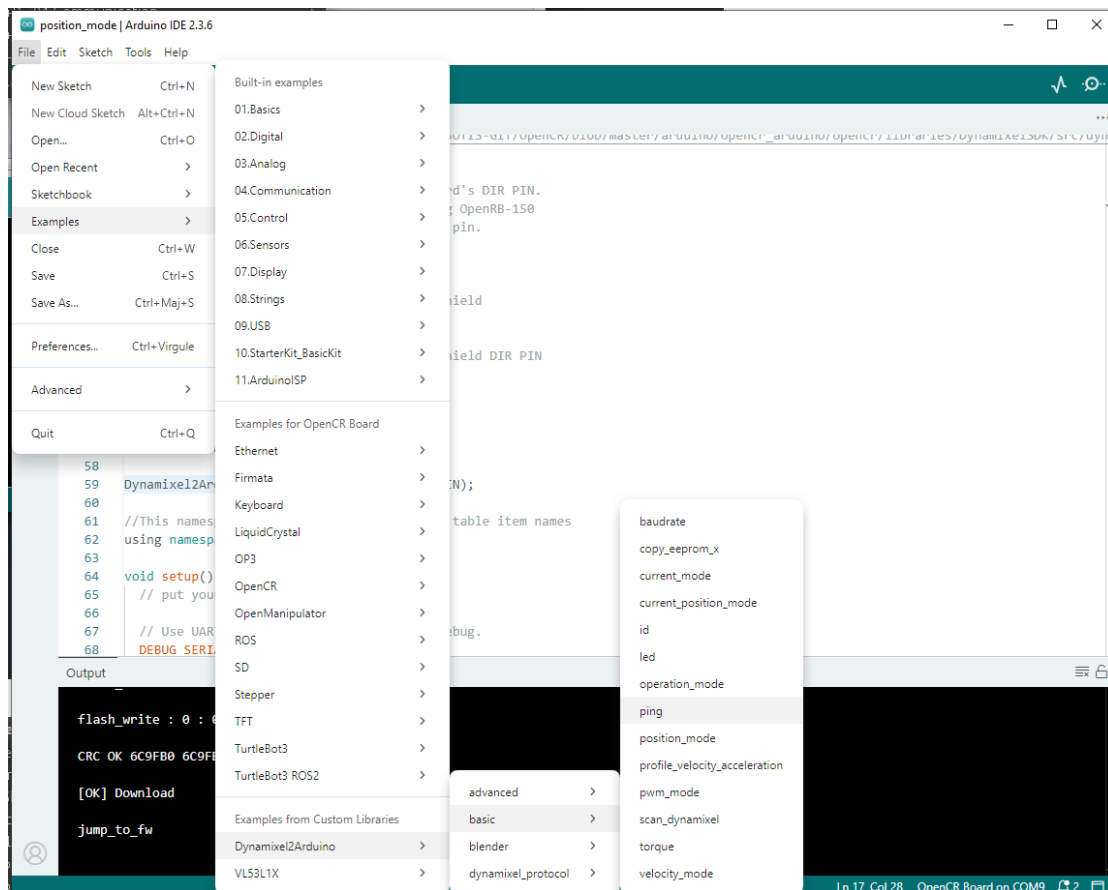


Utilisation de la bibliothèque Dynamixel2Arduino, à faire avec l'installation des moteurs.

A installer dans *library manager*



Utilisation des programmes exemples, ping ou autre pour vérifier fonctionnement des moteurs (bien pensé à changer le baud pour les moteurs Dynamixel)



Lancer le programme et vous devez voir normalement les deux moteurs avec les deux ID 1 et ID 2.

Lancer le programme position mode, pour avoir un aperçu simple de comment utiliser les moteurs simplement.

Fonction de base pour le fonctionnement des moteurs :

```
// Exemple : tourner les moteurs de 4096 ticks (≈ 1 tour complet)
dxl.setGoalPosition(DXL_ID_LEFT, 2048);
dxl.setGoalPosition(DXL_ID_RIGHT, 2048);

Pour utiliser des angles (en précisant le type d'unités)
dxl.setGoalPosition(DXL_IDD, 360, UNIT_DEGREE);
```

Pour faire avancer les moteurs à une certaine position et vitesse par défaut

Chaque roue a un codeur incrémental interne (≈ 4096 ticks/tour).
Tu peux calculer la distance :

- **Distance (m)** = $(\Delta \text{ticks} / 4096) \times (\text{périmètre roue})$.
- Périmètre roue ≈ 0.207 m (diamètre 66 mm).

Exemple lecture :

```
int32_t pos_left = dxl.getPresentPosition(DXL_ID_LEFT);
```

```
int32_t pos_right = dxl.getPresentPosition(DXL_ID_RIGHT);
```

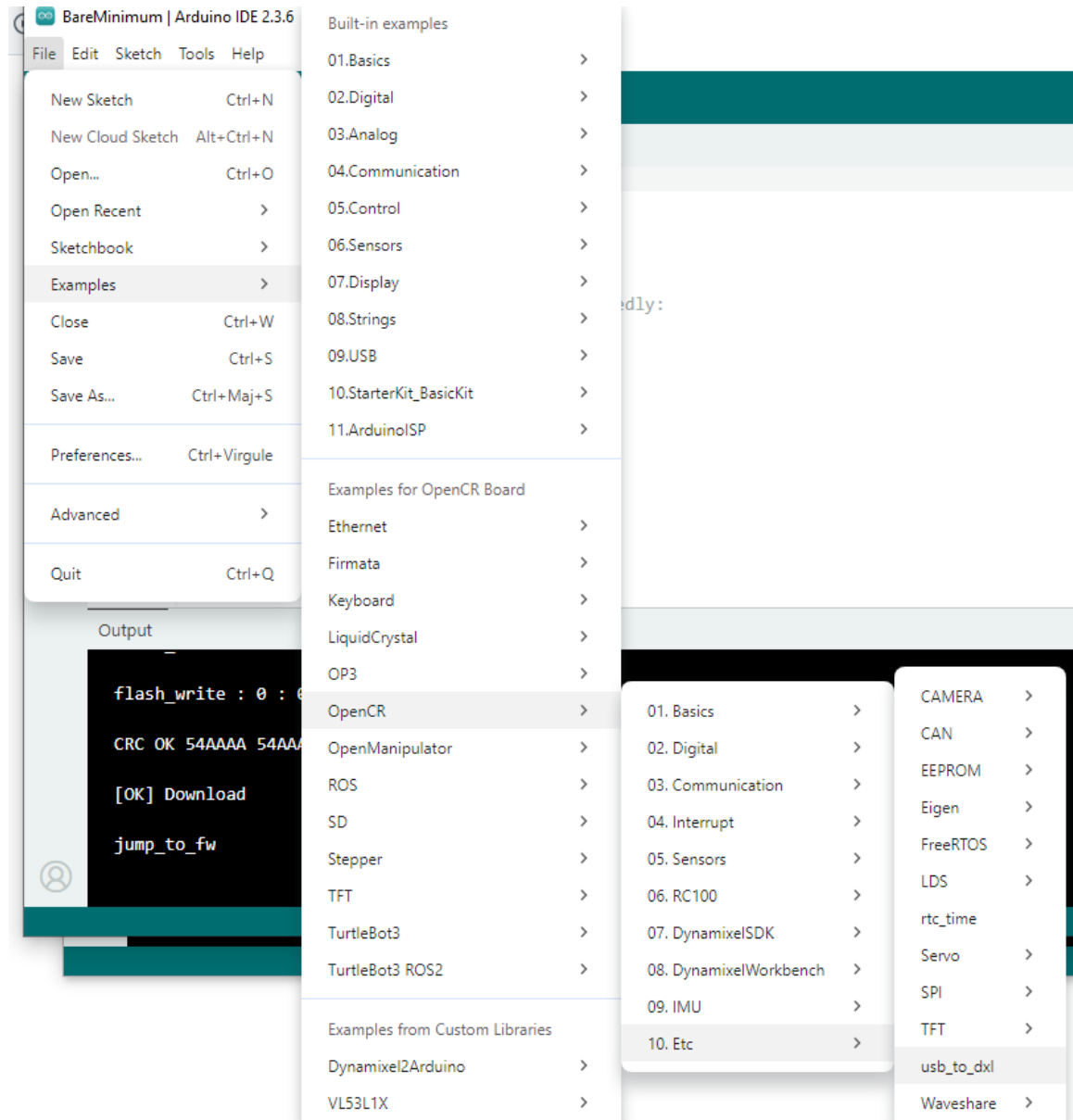
Pour plus de précision sur les fonctions, une vidéo est disponible :

<https://www.youtube.com/watch?v=iOBm5MJOLRo>

Utilisation Pour STM32 Initialisation des moteurs

Si ce n'a pas été fait, brancher avec les moteurs avec la carte OPEN CR.

Installation de dynamixel Wizard 2 en USB soit avec un adaptateur ou avec la carte OpenCR pour lié en USB et changer le baud rate des moteurs en 57600. Il est possible aussi d'exécuter le programme pour changer le BAUD.



Pour éviter les problèmes de communication UART, mettre la vitesse BAUD des moteurs à 57600 !

Une fois que les moteurs sont à la bonne vitesse, les erreurs de communication pourront être évités avec la carte NUCLEO.

Utilisation avec une carte NUCLEO

Créer un projet pour votre carte NUCLEO

USART1 Mode and Configuration

Mode

Mode

Hardware Flow Control (RS232)

Configuration

☒ NVIC Settings ☒ DMA Settings ☒ GPIO Settings

☒ Parameter Settings ☒ User Constants

GPIO mode

GPIO Pull-up/Pull-down

Maximum output speed

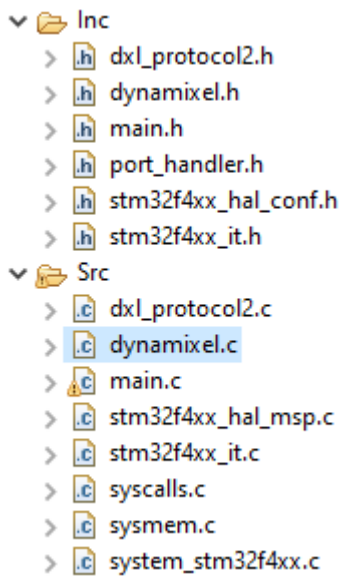
User Label

Voici les configurations du pin en TX pour USART 1 qui est relié au data du moteur

Mode Single Wire (Half-Duplex)

Baud rate : 57600 Bits/s

Voici l'arborescence du projet pour avoir la librairie pour le fonctionnement des moteurs



Pour installer cette librairie voici le lien :

<https://github.com/Myrton/STM32MotorDynamixelTurtleBot>

Une fois la bibliothèque correctement installée prendre le fichier main.c exemple et upload sur la carte NUCLEO et tester le programme.

Normalement moteurs sont censé tourné selon la séquence du programme du fichier main.c

Pour les délais se trouvant entre chaque commande des deux moteurs dans le fichier main.c, il y'a l'air d'avoir besoin d'un temps pour faire fonctionner les moteur « en même temps ». On met donc un délai qui peut possiblement être réduit, mais à 100ms cela semble fonctionner.

```
printf("Avancer\r\n");  
Dxl_Move(1, FORWARD);  
HAL_Delay(100);  
Dxl_Move(2, FORWARD);  
HAL_Delay(3000);
```

Montage des moteurs

Sachant que les moteurs se contrôlent en half-duplex, le pin doit être connecté au même moteur, l'identification se fera dans la trame avec la sélection de l'ID des moteurs, un seul pin suffit donc.

Utilisé du 12V pour alimenter les moteur, les alimentations de la carte nucléo ne suffiront pas.

Pensé bien à mettre la masse de la carte NUCLEO en commun avec la masse de l'alimentation 12V

4. 4. Connector Information

| Item | TTL | RS-485 |
|--------------------------|--|--|
| Pinout | <div> <div>1</div> GND <div>2</div> VDD <div>3</div> DATA </div> | <div> <div>1</div> GND <div>2</div> VDD <div>3</div> DATA+ <div>4</div> DATA- </div> |
| Diagram | | |
| Housing | JST EHR-03 | JST EHR-04 |
| PCB Header | JST B3B-EH-A | JST B4B-EH-A |
| Crimp Terminal | JST SEH-001T-P0.6 | JST SEH-001T-P0.6 |
| Wire Gauge for DYNAMIXEL | 21 AWG | 21 AWG |

Connector Information

Voici le connecteur du moteur qui est TTL, il y'a 3 broches GND, VDD (12V) et DATA (la broche de l'half-Duplex UART).

Bien vérifier le sens du connecteur avant de faire les branchements.

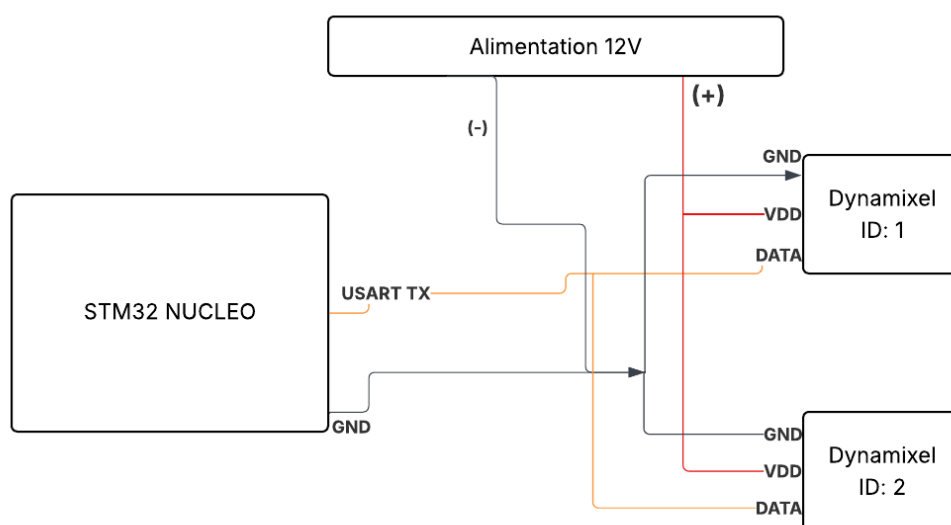


Schéma câblage du test des moteurs

Pour la connexion avec la batterie faire attention à ces étapes pour l'alimentation :

Ne pas brancher le cable usb avant d'avoir câbler l'alimentation externe et de l'avoir alimenté, et l'utilisé que pour debug et si pas beaucoup de courant est tirée

Connect the jumper between **pin 2 and pin 3 of JP5**

Check that JP1 is removed

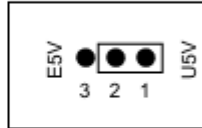
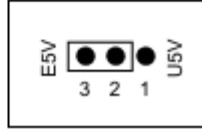
Connect the external power source to VIN or E5V

Power on the external power supply $7\text{ V} < \text{VIN} < 12\text{ V}$ to VIN, or 5 V for E5V

Check that LD3 is turned ON

Connect the PC to USB connector CN1

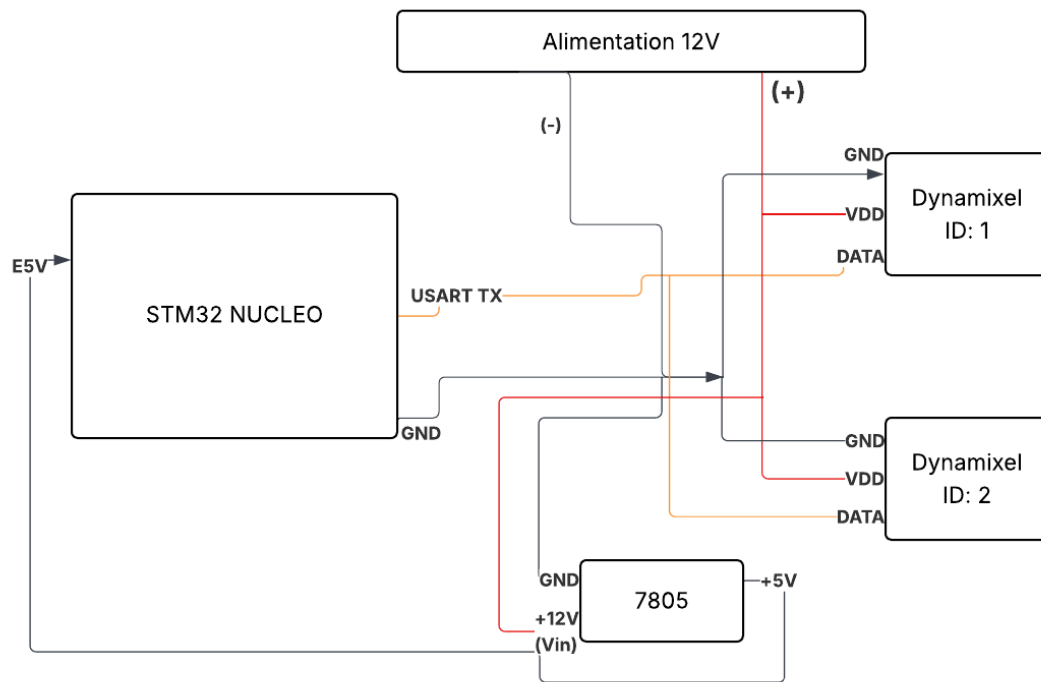
Table 8. Power-related jumper

| Jumper | Description |
|--------|---|
| JP5 | U5V (ST-LINK VBUS) is used as a power source when JP5 is set as shown below (Default setting)  |
| | VIN or E5V is used as a power source when JP5 is set as shown below.  |

Montage avec batterie

Pour utilisation avec une batterie, c'est possible à condition de bien alimenter le montage, pour pas qu'il n'y a des pertes de puissance et un manque de courant à délivrer.

Pour cela en se référant à la doc de la carte NUCLEO il faut donc utiliser du 5V sur le pin E5V pour avoir assez de courant pour contrôler les moteurs en alimentant la carte NUCLEO avec la batterie. On a une batterie de 12V il faut donc un régulateur de tension, on peut utiliser le régulateur 7805 pour avoir notre tension de 5V.



Montage avec batterie de la carte nucléo et des moteurs