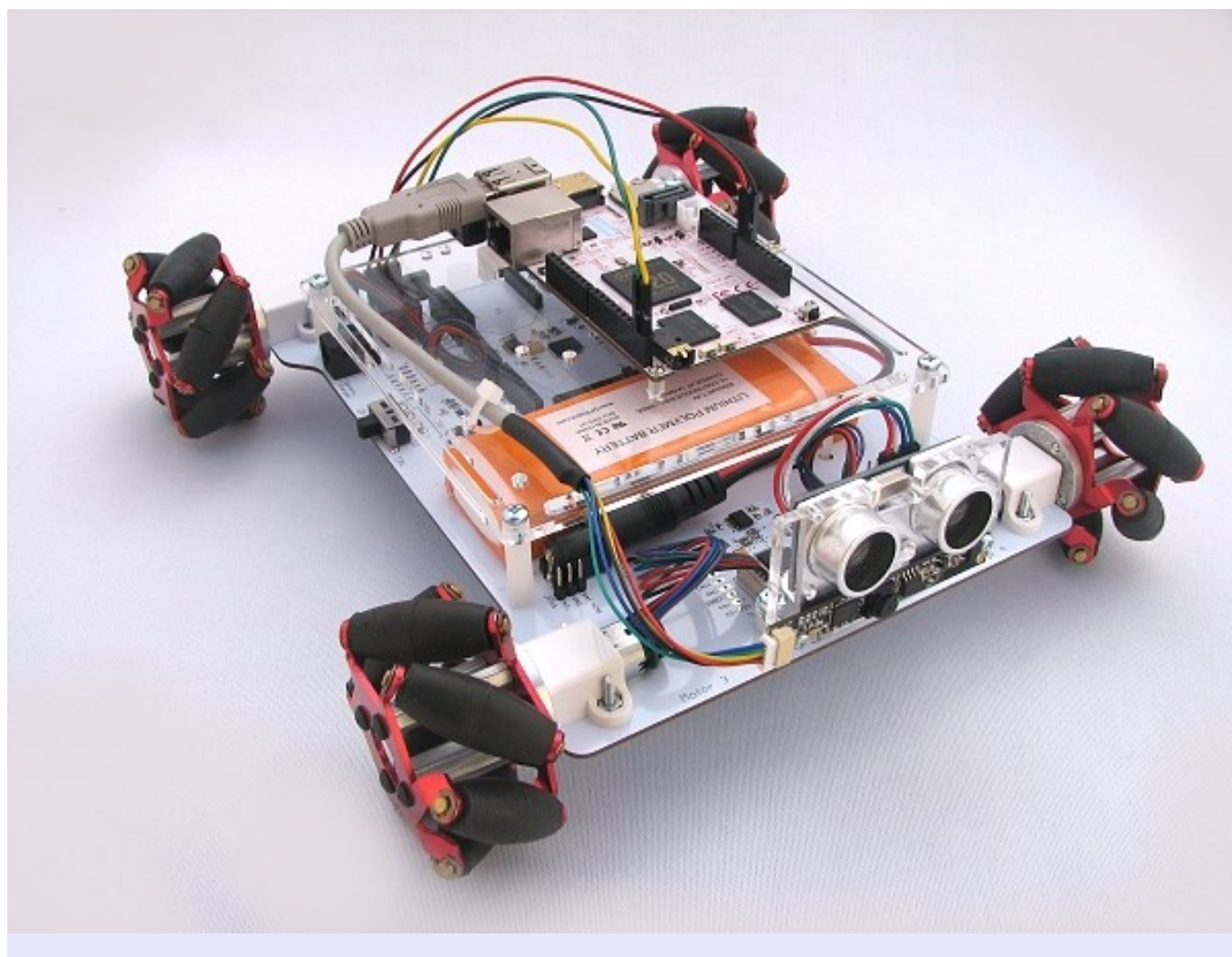




# T-QUAD

## COMMANDE DES MOTEURS EN TENSION



Date de dernière mise à jour : 15/12/2016

## Table des matières

---

<b>1 - Introduction.....</b>	<b><u>3</u></b>
<b>2 - Principes de fonctionnement.....</b>	<b><u>4</u></b>
2.1 - Calcul de la vitesse de rotation du moteur.....	<u>4</u>
2.2 - Comptage du nombre d'impulsions.....	<u>6</u>
<b>3 - Programme Arduino.....</b>	<b><u>7</u></b>
<b>4 - Programme Python.....</b>	<b><u>8</u></b>
<b>5 - Application MyViz de contrôle des moteurs en tension.....</b>	<b><u>9</u></b>
<b>6 - Important.....</b>	<b><u>13</u></b>

## 1 - Introduction

Cette documentation présente la procédure à suivre pour réaliser une commande en tension des moteurs à courant continu du robot T-Quad.

Les programmes et applications associés à cette activité sont téléchargeables ici :

<https://github.com/3sigma/T-Quad-Commande-Moteurs-en-Tension>

## 2 - Principes de fonctionnement

Cette activité permet de piloter les moteurs en boucle ouverte, en leur fournissant uniquement une tension de commande. Cependant, l'application MyViz de pilotage et de télémétrie permet également d'afficher la vitesse de rotation des moteurs, dont nous exposons le calcul ci-dessous.

### 2.1 - Calcul de la vitesse de rotation du moteur

Les moteurs à courant continu du robot T-Quad sont dotés de codeurs incrémentaux afin de mesurer la vitesse de rotation des moteurs.

Le codeur incrémental fournit deux signaux carrés en quadrature, comme sur la capture ci-dessous:



Ces deux signaux permettent de mesurer à la fois la vitesse et le sens de rotation. La mesure de la vitesse se fait simplement en comptant le nombre d'impulsions pendant un temps fixe. Les données du problème sont les suivantes :

- Le codeur est fixé à l'arbre moteur et non pas à l'arbre de sortie du réducteur (celui utilisé pour l'entraînement). Le rapport de réduction étant 100:1, l'arbre moteur fait 100 tours lorsque l'arbre « principal » en fait 1
- Le codeur génère 12 impulsions à chaque fois qu'il fait un tour
- La cadence d'échantillonnage utilisée pour l'asservissement sera de 0.01 s

Par conséquent, lorsque l'arbre principal fait un tour, le codeur génère :  
 $100 \times 12 = 1200$  impulsions.

Si  $N$  est le nombre d'impulsions comptées en 0.01 s, la vitesse est (en rad/s, l'unité standard, sachant qu'un tour fait  $2 \times \pi$  radians) :

$$\frac{2\pi N}{0.01 * 1200} \quad (2.1)$$

**ATTENTION !**

Bien que le codeur soit placé sur l'arbre moteur, le calcul ci-dessus donne la vitesse en sortie du réducteur.

Un point très important concerne la résolution de la mesure, c'est-à-dire la plus petite valeur qu'il est possible de calculer. La formule est la suivante (en rad/s) :

$$\frac{2\pi N}{Ts \text{ CPR } n} \quad (2.2)$$

avec :

- $T_s$  : cadence d'échantillonnage
- CPR : nombre d'impulsions par tour du codeur
- $n$  : rapport de réduction du moteur

Dans notre cas de figure, la résolution est la suivante :

$$\frac{2\pi}{0.01 * 1200} = 0.52 \text{ rad/s} \quad (2.3)$$

## 2.2 - Comptage du nombre d'impulsions

Compter le nombre d'impulsions du codeur revient à compter le nombre de fronts montants et descendants des signaux jaune et bleu représentés sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher les deux signaux (les fils bleu et blanc sur le codeur utilisé) sur deux entrées « interruption » de la carte Arduino. L'intérêt d'une ligne d'interruption est qu'elle permet, comme son nom l'indique, d'interrompre le déroulement des calculs sur le micro-contrôleur pour effectuer un traitement spécifique, en l'occurrence la mise à jour du compteur d'impulsions, avant de rendre la main à la boucle principale.

La seule « difficulté » est de savoir s'il faut incrémenter ou décrémenter le compteur dans le traitement de l'interruption. Il suffit pour cela d'observer les courbes ci-dessus, obtenues alors que le moteur tourne dans le sens positif. On constate que:

- Lorsque la voie A (en jaune) passe au niveau haut, la voie B (en bleu) est au niveau bas
- Lorsque la voie A passe au niveau bas, la voie B est au niveau haut

Quand le moteur tourne dans le sens négatif, lors d'une interruption sur la voie A, les niveaux de A et B sont donc inversés.

Noter que par manque de lignes d'interruption disponibles sur la carte Arduino Mega dans le contexte du robot T-Quad, seule la voie A du codeur est branchée sur une entrée de type interruption.

Par conséquent, la valeur à utiliser dans les calculs précédents doit être 600 et non pas 1200, cette dernière étant la valeur idéale dans le cas où chaque codeur est branché sur une ligne d'interruption du micro-contrôleur.

## 3 - Programme Arduino

Dans le cas où cette activité se fait avec un programme Python exécuté sur un éventuel mini-ordinateur ajouté au robot (pcDuino, Raspberry Pi ou autre), il faut charger sur la carte Arduino Mega le firmware Arduino, composé des fichiers téléchargeables à cette adresse :

[https://github.com/3sigma/T-Quad/tree/master/Arduino/Firmware\\_i2c](https://github.com/3sigma/T-Quad/tree/master/Arduino/Firmware_i2c)

Ils sont également contenus dans l'archive suivante :

<https://github.com/3sigma/T-Quad/archive/master.zip>

### ATTENTION !

Ce firmware est déjà chargé à la livraison du robot : si vous n'avez encore jamais reprogrammé l'Arduino Mega et s'il n'existe pas de nouvelle version de ce firmware, ce n'est pas nécessaire de le charger à nouveau.

Dans le cas contraire, la procédure à suivre pour la programmation est la suivante:

- Si ce n'est pas déjà fait, installez l'IDE Arduino et les bibliothèques additionnelles nécessaires (voir la documentation générale : [https://github.com/3sigma/T-Quad/raw/master/T-Quad\\_General.pdf](https://github.com/3sigma/T-Quad/raw/master/T-Quad_General.pdf))
- Ouvrez le programme Firmware\_i2c.ino téléchargé précédemment dans l'IDE Arduino
- Mettre le robot sous tension en appuyant sur le bouton marche-arrêt
- Connecter le câble USB reliant l'ordinateur et la carte Arduino Mega
- Lancer le téléchargement

## 4 - Programme Python

Dans le cas où cette activité se fait avec un programme Python exécuté sur un éventuel mini-ordinateur ajouté au robot (pcDuino, Raspberry Pi ou autre), ce programme est déjà présent sur la carte. Cependant, il peut être nécessaire de mettre ce programme à jour avec la dernière version, disponible à l'adresse suivante :

[https://github.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/tree/master/programmes\\_python](https://github.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/tree/master/programmes_python)

Il est également contenu dans l'archive suivante :

<https://github.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/archive/master.zip>

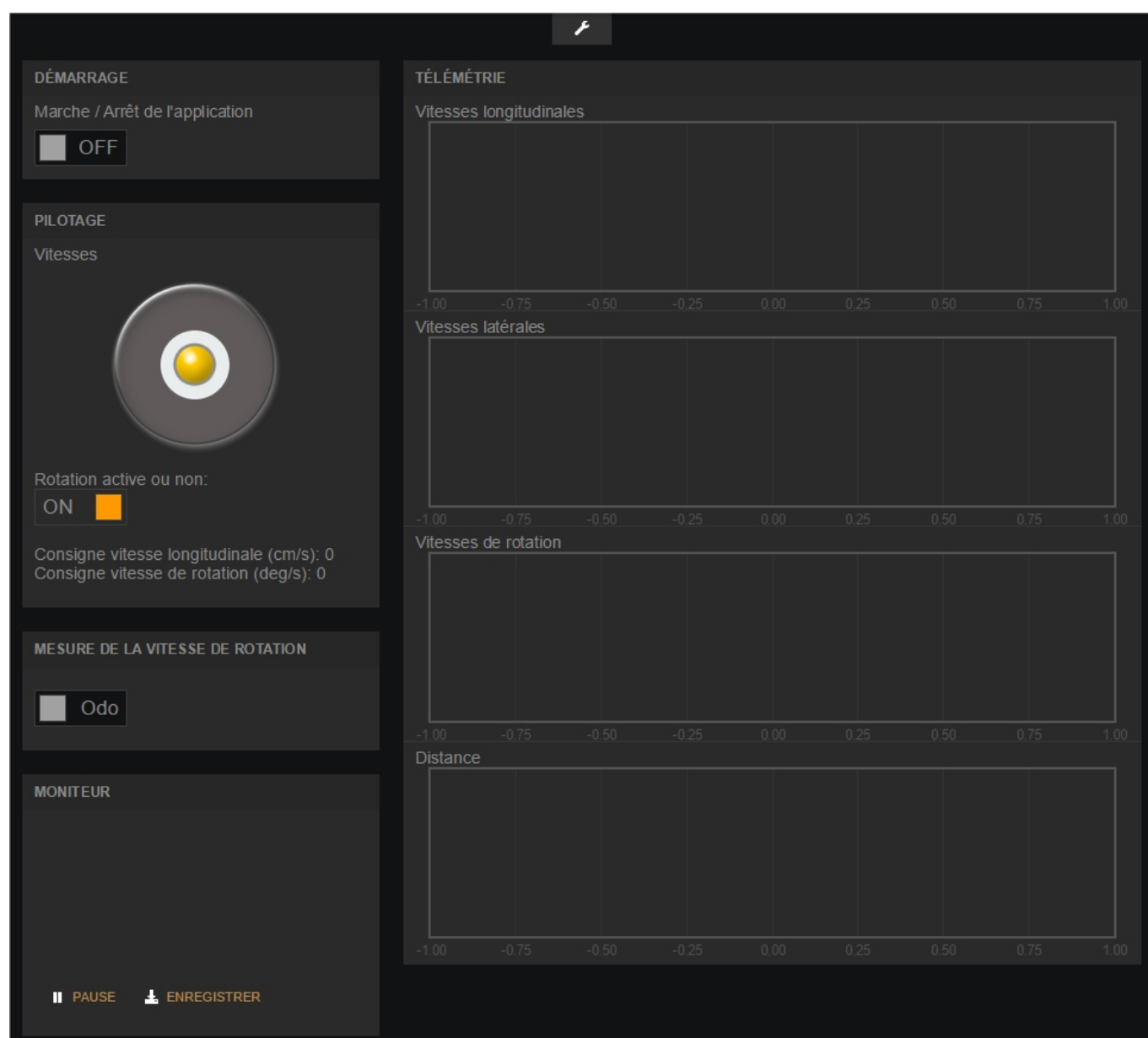
Noter que le comptage des interruptions permettant de mesurer la vitesse de rotation du moteur n'est pas réalisé par ce programme. En effet, le système d'exploitation standard de ce type de mini-ordinateur ne permet pas de faire ces manipulations avec suffisamment de performances. Cette tâche est donc sous-traitée à la carte Arduino Mega.



## 5 - Application MyViz de contrôle des moteurs en tension

Les activités réalisées sur le robot T-Quad utilisent en général le logiciel MyViz, très souple pour créer des tableaux de bord de pilotage et de visualisation de données.

Après l'avoir téléchargé (voir les liens sur la page de téléchargement consacrée à T-Quad : [http://www.3sigma.fr/Telechargements-T\\_Quad.html](http://www.3sigma.fr/Telechargements-T_Quad.html)) et installé, lancez son exécution. Le tableau de bord initialement affiché sera similaire à la capture d'écran ci-dessous :



Pour réaliser l'activité de contrôle des moteurs en tension, les conditions suivantes doivent être remplies :

- de préférence, le robot doit être « sur cale » (les moteurs doivent pouvoir tourner librement)
- le robot doit être allumé
- l'ordinateur doit être connecté en Wifi ou en Ethernet au robot
- le firmware Arduino doit être chargé (voir chapitre 3)

Charger ensuite le tableau de bord de contrôle des moteurs dans MyViz. Pour cela, il faut tout d'abord récupérer ce dernier sur votre ordinateur, à partir du lien suivant :

[https://raw.githubusercontent.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/master/MyViz/T-Quad\\_CommandeMoteurEnTension\\_Reseau.json](https://raw.githubusercontent.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/master/MyViz/T-Quad_CommandeMoteurEnTension_Reseau.json)

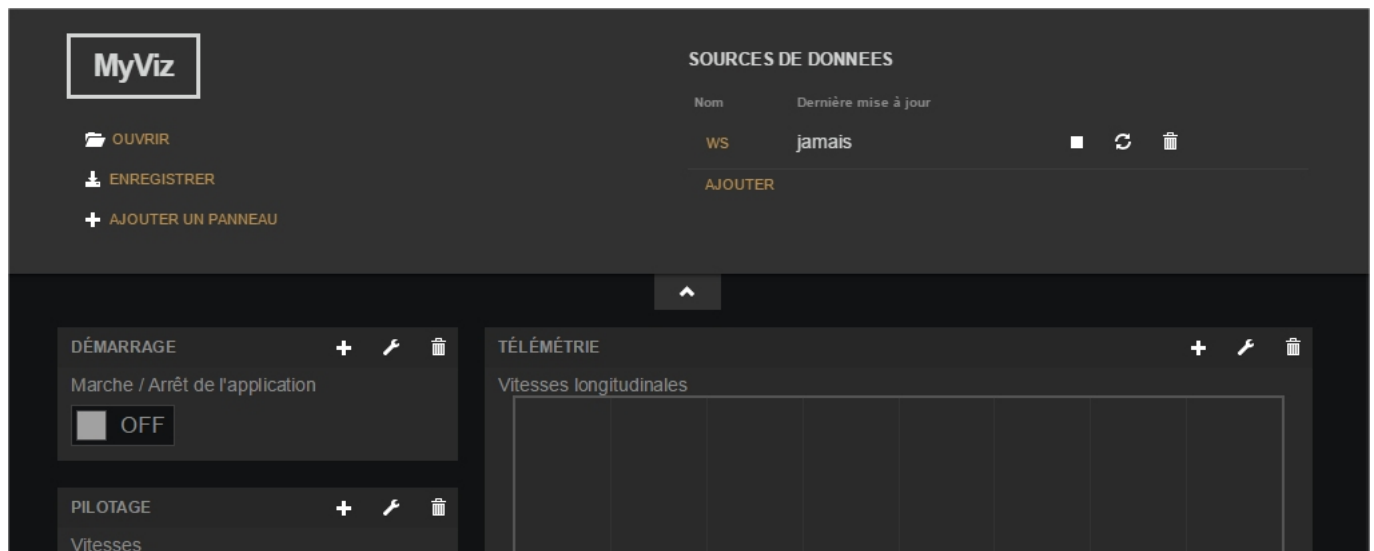
Il se trouve également dans l'archive suivante :

<https://github.com/3sigma/T-Quad-Commande-Moteurs-en-Tension/archive/master.zip>

Pour l'ouvrir dans MyViz, il suffit ensuite de cliquer sur la clé en haut de la fenêtre :

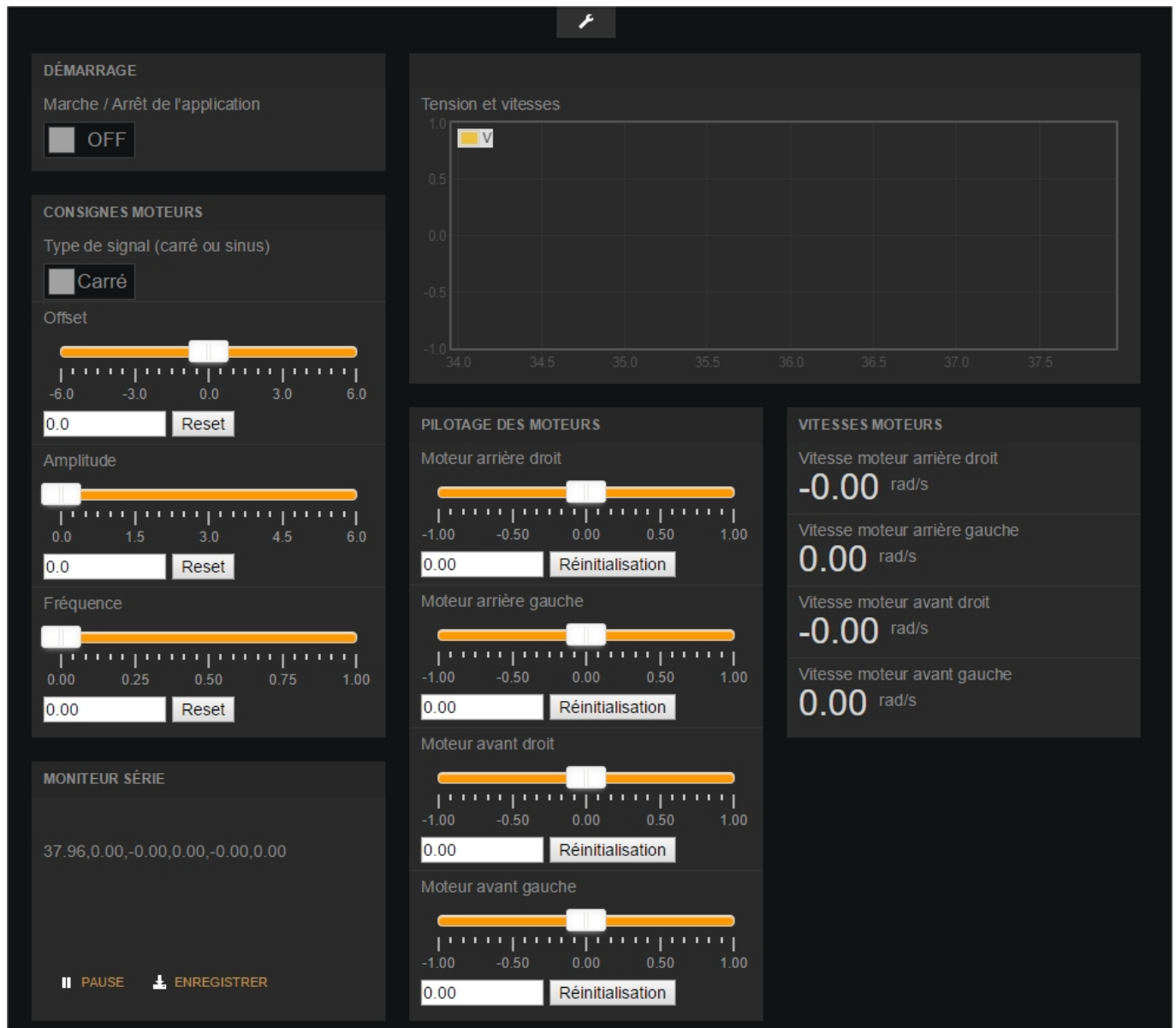


Ceci permet de déplier le panneau supérieur :



Cliquez sur « Ouvrir » et sélectionnez le fichier T-Quad\_CommandeMoteurEnTension\_Reseau.json que vous venez de télécharger.

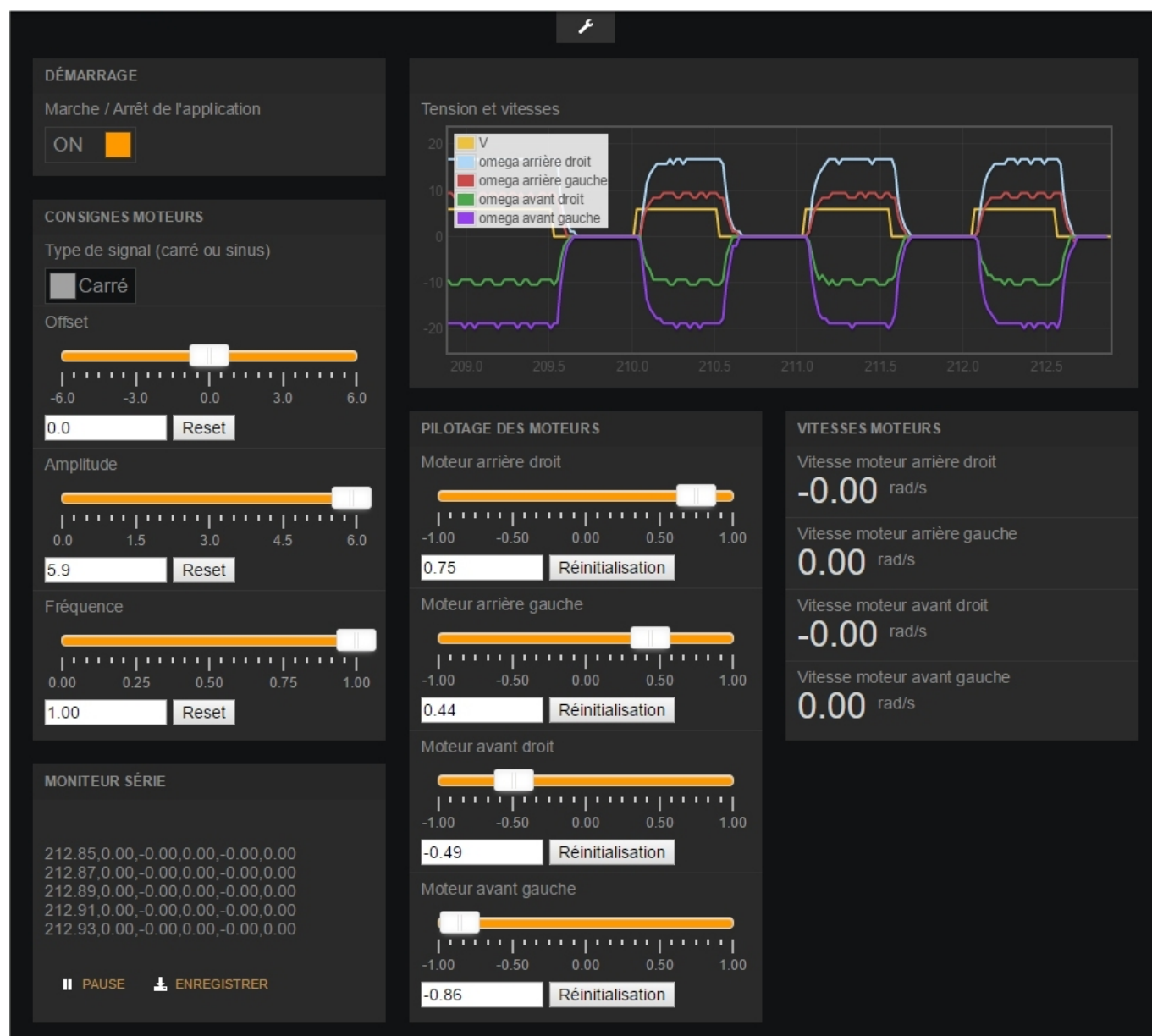
Le tableau de bord s'affiche alors :



Son utilisation est a priori intuitive. Il faut cependant noter les points suivants :

- le démarrage de l'application se fait via la bouton marche-arrêt en haut à gauche.  
**Attention** : il faut attendre quelques secondes avant de voir apparaître les courbes de télémétrie et de pouvoir piloter les tensions
- les curseurs de « Pilotage des moteurs » individuels se multiplient à la somme de l'offset et de l'amplitude généraux, placés dans la colonne de gauche
- la tension de commande envoyée aux moteurs est saturée par programme à  $\pm 6V$

En fonctionnement, ce tableau de bord peut avoir l'allure suivante :



## 6 - Important

T-Quad est un produit « vivant » en constant développement pour l'améliorer ou lui ajouter de nouvelles fonctionnalités. Si vous avez des idées ou des besoins pour des développements spécifiques, n'hésitez pas à nous contacter ([support@3sigma.fr](mailto:support@3sigma.fr)).

**Ne restez jamais bloqué sans nous contacter !**

Pour tout problème ou toute requête, contactez-nous à l'adresse [support@3sigma.fr](mailto:support@3sigma.fr)