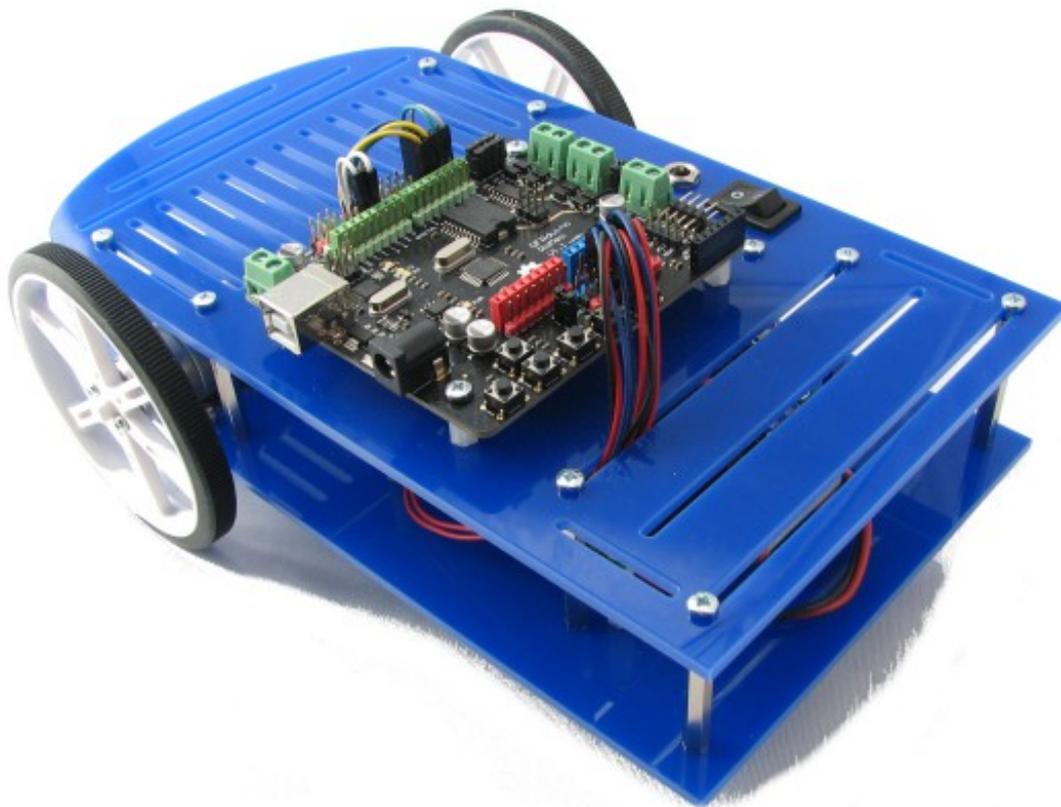




MOBILOS

DOCUMENTATION COMPLÈTE



Date de dernière mise à jour : 05/05/2014

Table des matières

1 - Introduction.....	<u>3</u>
2 - Matériel inclus.....	<u>3</u>
2.1 - Eléments constituant Mobilos.....	<u>4</u>
2.2 - Diagramme des échanges de signaux.....	<u>5</u>
3 - Conformité.....	<u>5</u>
4 - Installation de l'IDE Arduino.....	<u>6</u>
4.1 - Installation principale.....	<u>6</u>
4.2 - Installation de la bibliothèque complémentaire FlexiTimer2.....	<u>6</u>
4.3 - Installation de la bibliothèque complémentaire digitalWriteFast.....	<u>6</u>
4.4 - Installation de la bibliothèque complémentaire QTRSensors.....	<u>7</u>
4.5 - Téléchargement d'un programme sur la carte Arduino Romeo.....	<u>7</u>
5 - Précautions d'emploi.....	<u>8</u>
5.1 - Connexions d'alimentation sur la carte Romeo.....	<u>8</u>
5.2 - Recharge de la batterie.....	<u>8</u>
5.3 - Précautions d'utilisation.....	<u>9</u>
6 - Utilisation.....	<u>10</u>
6.1 - Mise en route.....	<u>10</u>
6.2 - Suivi de ligne.....	<u>11</u>
6.2.1 - Préparation.....	<u>11</u>
6.2.2 - Fonctionnement.....	<u>12</u>
6.2.3 - Asservissement.....	<u>12</u>
7 - Expériences additionnelles.....	<u>13</u>
7.1 - Commande des moteurs en tension.....	<u>13</u>
7.2 - Asservissement des moteurs en vitesse.....	<u>16</u>
8 - Autre documentation disponible.....	<u>20</u>
9 - Important.....	<u>21</u>
10 - ANNEXE 1 – Circuit de suivi de ligne.....	<u>22</u>

1 - Introduction

Mobilos est un robot ouvert et open-source, un concentré de technologie vous permettant de faire de nombreuses expériences. Il est basé sur une carte Romeo (compatible Arduino), différents capteurs et deux moteurs électriques permettant de le mettre en mouvement tout en assurant son maintien en équilibre.

Une documentation détaillée de la carte Arduino Romeo ainsi que les programmes associés au fonctionnement de Mobilos, intégrant tous les commentaires nécessaires à leur compréhension, sont téléchargeables sur notre site Web à l'adresse suivante:

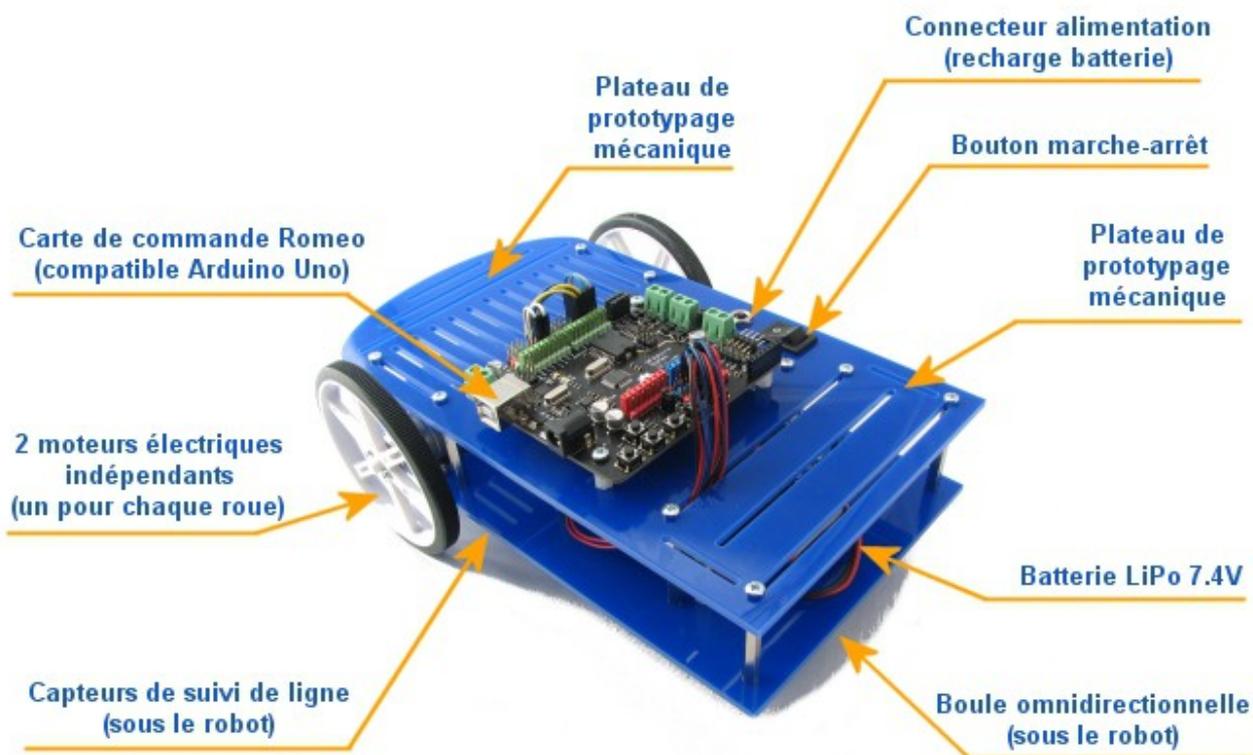
<http://www.3sigma.fr/telechargements>

2 - Matériel inclus

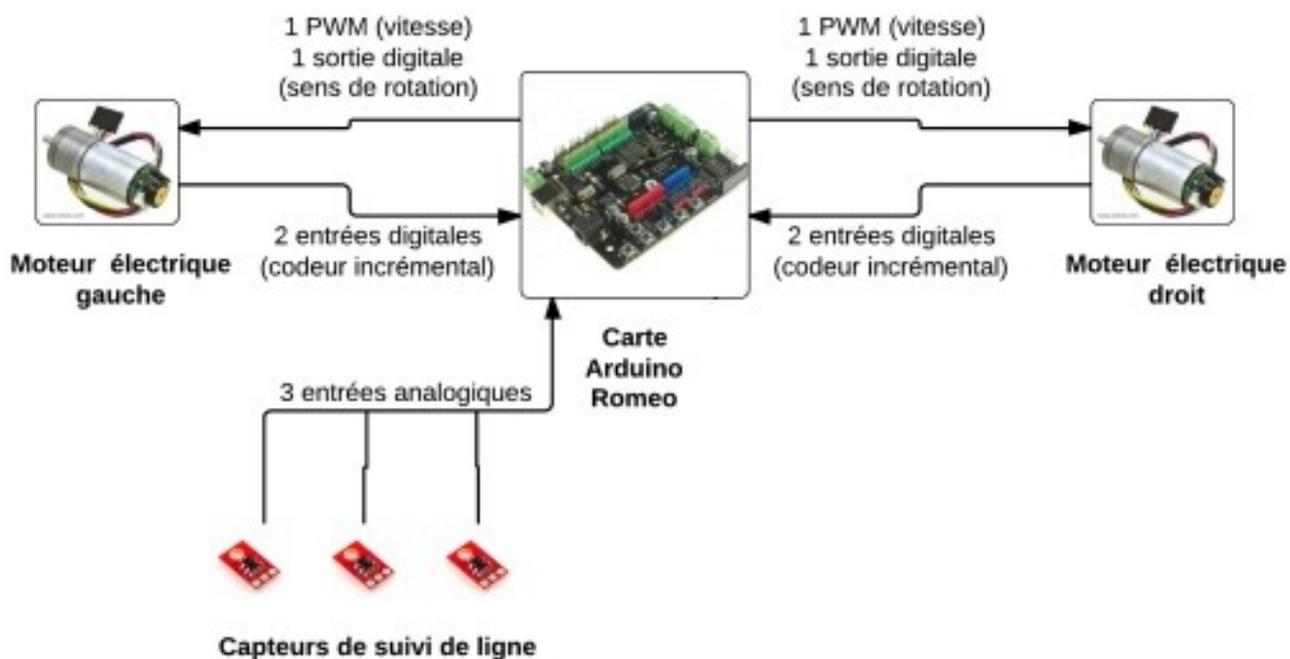
Mobilos est livré monté et **fonctionnel (programmé et testé par nos soins avant la livraison)**. Il est composé des éléments suivants:

- Le robot lui-même
- 1 câble USB A-B pour la programmation de la carte Romeo
- 1 alimentation 12V, 1A
- 1 clé allen pour visser ou dévisser les roues
- 1 platine de prototypage électronique vierge

2.1 - Éléments constituant Mobilos



2.2 - Diagramme des échanges de signaux



3 - Conformité

Le robot Mobilos, **dans sa configuration livrée aux clients**, est conforme à la directive 1999/EC.



4 - Installation de l'IDE Arduino

Mobilos intègre une carte Romeo, compatible Arduino Uno. **Le robot est déjà programmé à la livraison.** Cependant, si vous souhaitez télécharger de nouveaux programmes, vous devez au préalable installer l'IDE Arduino et quelques bibliothèques additionnelles.

4.1 - Installation principale

Télécharger et installer l'IDE Arduino (<http://arduino.cc/en/Main/Software>). Il suffit de décompresser l'archive téléchargée dans le répertoire de votre choix.

ATTENTION !

Avant de pouvoir compiler les programmes fournis avec Geeros, vous devez suivre les instructions suivantes pour installer les bibliothèques complémentaires **FlexiTimer2**, **digitalWriteFast**, **I2Cdev**, **MPU6050** et **SoftwareServo**.

4.2 - Installation de la bibliothèque complémentaire FlexiTimer2

Cette bibliothèque permet d'exécuter à cadence fixe une partie du programme Arduino.

Vous pouvez la télécharger à l'adresse suivante: <http://www.3sigma.fr/telechargements/FlexiTimer2.zip>.

Une fois téléchargée, décompressez-la dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries).

4.3 - Installation de la bibliothèque complémentaire digitalWriteFast

Cette bibliothèque permet de lire et d'écrire plus rapidement sur les entrées-sorties digitales de l'Arduino.

Vous pouvez la télécharger à l'adresse suivante:

<http://www.3sigma.fr/telechargements/digitalWriteFast.zip>.

Une fois téléchargée, décompressez-la dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries).

4.4 - Installation de la bibliothèque complémentaire QTRSensors

Cette bibliothèque permet de gérer les capteurs de suivi de ligne.

Vous pouvez la télécharger à l'adresse suivante: <http://www.3sigma.fr/telechargements/QTRSensors.zip>.

Une fois téléchargée, décompressez-la dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries).

ATTENTION !

L'environnement Arduino doit être redémarré après l'installation des bibliothèques complémentaires.

4.5 - Téléchargement d'un programme sur la carte Arduino Romeo

La procédure à suivre pour la programmation est la suivante:

- Mettre le robot sous tension en appuyant sur le bouton marche-arrêt
- Connecter le câble USB reliant l'ordinateur et la carte Romeo
- Lancer le téléchargement immédiatement

5 - Précautions d'emploi

Nous insistons sur le fait que Mobilos et un robot d'étude et de développement qui nécessite un certain nombre de précautions d'emploi.

5.1 - Connexions d'alimentation sur la carte Romeo

Il est impératif de faire très attention aux connexions de l'alimentation de la carte Romeo car celle-ci n'est pas protégée contre les inversions de polarité. Une erreur de connexion sur les bornes d'alimentation risque d'entraîner la destruction du sous-ensemble de gestion d'alimentation de la carte et de rendre celle-ci inutilisable. Le robot étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

5.2 - Recharge de la batterie

L'alimentation 12V, 1A fournie est destinée à recharger la batterie LiPo du robot en la connectant sur le connecteur jack situé à côté du bouton de marche-arrêt. Bien que cette batterie possède une protection intégrée contre les surcharges et les sous-charges, il n'est pas conseillé de la recharger avec une alimentation pouvant fournir un courant supérieur à 1A.

Si une autre alimentation devait être utilisée pour la recharger, il est important de vérifier que sa polarité est « positif au centre du connecteur ».

Il faut compter environ 2 heures pour une demi-recharge de la batterie et 8 heures pour une recharge complète.

5.3 - Précautions d'utilisation

Mobilos doit être utilisé dans les conditions suivantes:

- Au sol. Ne jamais l'utiliser en hauteur (sur une table, par exemple) à cause des risques de chute
- A l'intérieur. Les sols extérieurs présentent souvent des aspérités et des obstacles surdimensionnés par rapport à la taille du robot. Par ailleurs, Mobilos craint l'eau et l'humidité

Mobilos peut être reprogrammé à votre guise. Vous pouvez par exemple faire des programmes permettant d'étudier la commande des moteurs électriques. Mais attention: il est **fortement déconseillé** de faire des expériences de fonctionnement « rotor bloqué » avec une tension d'alimentation du moteur trop élevée. Ce type d'expérience peut générer des courants trop forts qui réduisent la durée des vie des éléments.

6 - Utilisation

6.1 - Mise en route

Le programme pré-chargé dans le robot à la livraison lui permet de réaliser une trajectoire pré-programmée sous forme de « 8 ».

La procédure à suivre est simple:

- Poser le robot au sol sans l'allumer
- Le mettre en route en appuyant sur le bouton marche-arrêt

Pour modifier cette trajectoire pré-programmée, il suffit de modifier de programme MobilosStandard.ino, téléchargeable à l'adresse <http://www.3sigma.fr/telechargements>. Ce programme est abondamment commenté, le passage concernant la programmation de trajectoire débute par le commentaire:

```
// Consigne
```

Ce programme modifié, vous pourrez ensuite le télécharger sur la carte Arduino Romeo en suivant la procédure décrite au paragraphe 4.5.

6.2 - Suivi de ligne

6.2.1 - Préparation

Une étape préalable est la réalisation d'un circuit. Vous pouvez pour cela imprimer les morceaux de lignes fournis en annexe (chapitre 10) pour construire un circuit personnalisé.

Les autres conditions préalables au bon fonctionnement sont les suivantes:

- Les capteurs de suivi de lignes doivent être présents (ils sont facilement démontables...)
- Vous devez avoir téléchargé sur la carte Arduino Romeo le programme MobilosSuiviLigne.ino, téléchargeable à l'adresse <http://www.3sigma.fr/telechargements>

Il est important de conserver en mémoire les éléments susceptibles de jouer sur la détection de la ligne:

- Le contraste entre la ligne et le fond. Les meilleurs résultats sont obtenus en réalisant un circuit à partir de morceaux de lignes noires imprimées sur du papier blanc (voir chapitre 10). Si vous avez un sol très clair ou très foncé, vous pouvez également réaliser un circuit directement par terre en utilisant du ruban adhésif pour tracer les lignes. Cependant, vous aurez dans ce cas du mal à tracer des courbes. Or, un angle sur une ligne correspond localement à un rayon de courbure nul, que le robot ne pourra pas suivre instantanément
- La distance entre les capteurs et le sol: il faut trouver un bon compromis car plus la distance est faible, plus la ligne sera détectée correctement, mais plus les capteurs risqueront de rencontrer des obstacles. Par exemple, si une feuille du circuit n'est pas très bien plaquée au sol, les capteurs du robot risquent de s'y accrocher. A la livraison du robot, la distance entre les capteurs et le sol est nominale et ne devrait pas être modifiée

6.2.2 - Fonctionnement

La réalisation d'un suivi de ligne se fait en deux étapes :

- Calibration : cette étape permet au robot de déterminer le contraste entre la ligne et le sol
 - Positionner le robot de telle sorte qu'il puisse croiser la ligne en moins d'1.5 s, sachant que sa vitesse est programmée à 10 cm/s
 - Démarrer le robot en appuyant sur le bouton marche-arrêt
 - Quand la led verte de la carte Romeo s'allume, reprendre le robot à la main sans l'arrêter (les roues continuent donc de tourner)
- Suivi de ligne :
 - Poser le robot centré sur la ligne
 - Mobilos tente alors de suivre cette dernière

6.2.3 - Asservissement

L'asservissement de suivi de ligne est du type proportionnel-dérivée, qui est un régulateur assez peu courant. La variable asservie est la distance entre le robot et la ligne ; la commande du système est une tension différentielle entre les deux moteurs, ce qui correspond directement à une commande de vitesse de rotation. C'est ce dernier point qui conduit à l'utilisation d'un régulateur PD.

L'action proportionnelle joue en fait le rôle de l'action intégrale dans un asservissement plus classique: si le robot roule parallèlement mais non centré sur la ligne, cette action va le ramener au centre.

L'action dérivée joue quant à elle le rôle de l'action proportionnelle d'un asservissement classique: lorsque le robot s'éloigne de la ligne, cette action le ramène sur cette dernière.

Il est important de conserver à l'esprit que la mesure de la distance entre le robot et la ligne est réalisée grâce à trois capteurs optiques analogiques mais qui fournissent dans les faits une information de type tout ou rien car en général les bords des lignes sont nets: le capteur est au-dessus de la ligne ou à côté, il n'y a pas vraiment d'intermédiaire. L'écart entre le robot et la ligne est donc une quantité discontinue qui, passée par l'action dérivée (même filtrée), conduit à des à-coups de rotation du robot assez brutaux.

7 - Expériences additionnelles

Mobilos est un système pédagogique multi-expériences qui permet non seulement d'étudier de près le fonctionnement d'un gyropode, mais également d'autres aspects plus spécifiques, comme la commande de moteurs électriques.

ATTENTION !

Veillez à sécuriser le robot lorsque vous réalisez les expériences décrites par la suite, pour qu'il ne risque pas de tomber d'une table, par exemple.

7.1 - Commande des moteurs en tension

Cette expérience permet de changer la vitesse de rotation des moteurs en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Arduino (commun aux robots de la famille Geeros, comme l'est Mobilos) peut être téléchargé à l'adresse suivante:

<http://www.3sigma.fr/Telechargements-Geeros.html>

Son nom est de la forme GeerosArduinoCommandeMoteursEnTension_x.y.zip (x.y correspond au numéro de version du programme).

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage des moteurs consiste à envoyer des signaux PWM sur le pont en H intégré à la carte Romeo afin de faire varier la tension d'alimentation des moteurs. Cette tension peut être modifiée interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

<http://www.3sigma.fr/Telechargements-Geeros.html>

Son nom est de la forme GeerosInterfaceCommandeMoteursEnTension_x.y.zip (x.y correspond au numéro de version du programme).

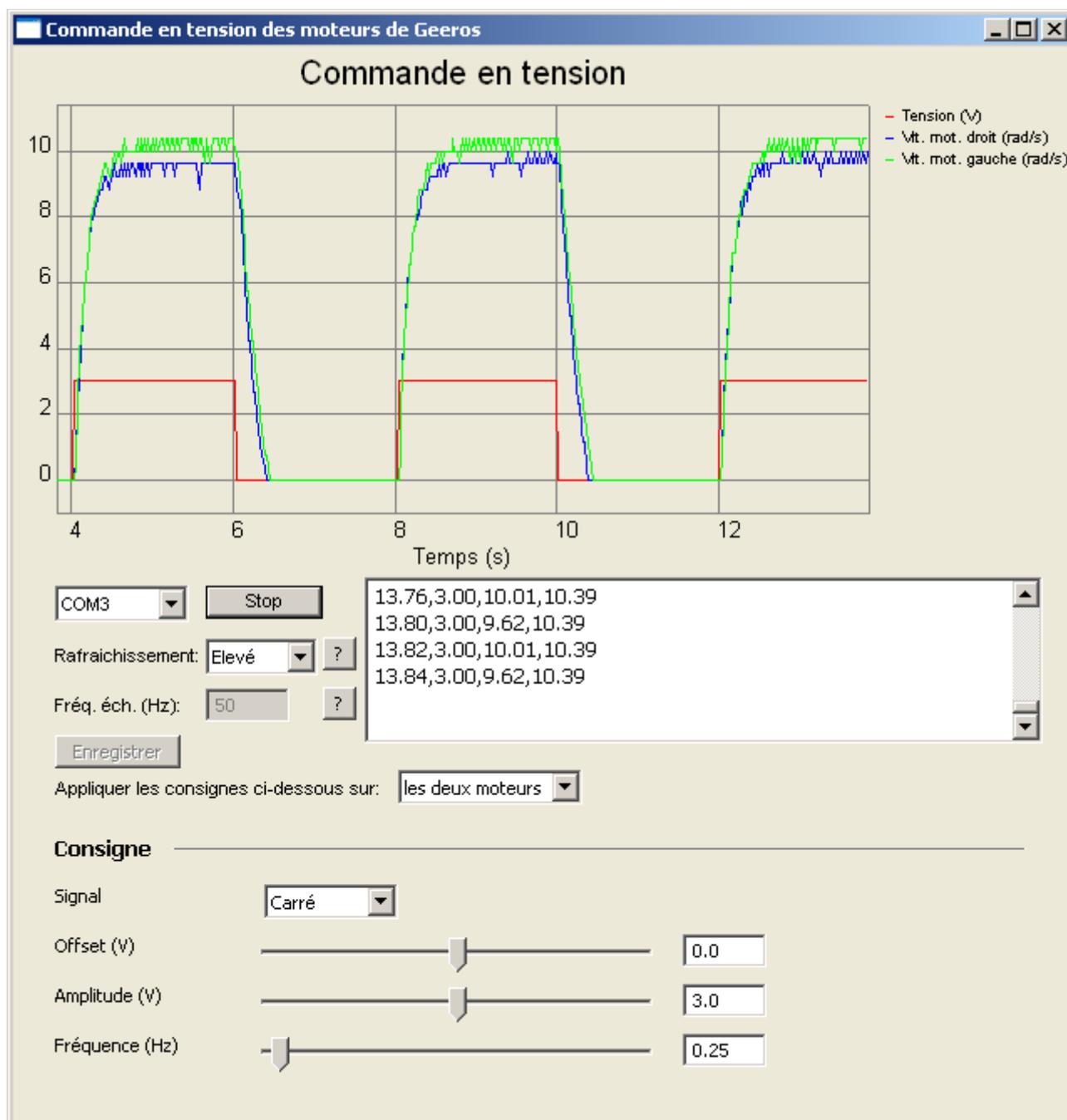
Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur GeerosInterfaceCommandeMoteursEnTension_x.y.exe.

Pour piloter la tension d'alimentation (et donc la vitesse) des moteurs depuis votre ordinateur, il vous suffit de suivre les étapes suivantes :

- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte Romeo à l'ordinateur avec le câble USB fourni

- télécharger le programme Arduino « GeerosArduinoCommandeMoteursEnTension » sur la carte Romeo
- lancer l'application « GeerosInterfaceCommandeMoteursEnTension »

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la tension de commande (V, en rouge)
 - les vitesses mesurées (rad/s, en bleu et vert)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte Romeo et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que les valeurs correspondantes dans le programme Arduino (CADENCE_MS et TSDATA) sont spécifiées en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{cadence (ms)}$$
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de tension avec les curseurs. La vitesse des moteurs varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel

IMPORTANT !

Si l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter les valeurs TSDATA et CADENCE_MS)

IMPORTANT !

Si vous comparez la vitesse de rotation des deux moteurs, vous obtiendrez probablement une valeur différente, bien que la tension de commande soit la même : c'est normal, ceci est dû à la disparité des caractéristiques des moteurs à courant continu.

D'un point de vue pédagogique, ce point permet de souligner la nécessité de réaliser un asservissement de vitesse : si l'on souhaite une vitesse de rotation précise, on ne peut pas se contenter d'une commande en boucle ouverte.

7.2 - Asservissement des moteurs en vitesse

Cette expérience permet d'asservir la vitesse de rotation des moteurs en appliquant une consigne de vitesse par l'intermédiaire d'une application qui s'exécute sur votre ordinateur.

Le programme Arduino (commun aux robots de la famille Geeros, comme l'est Mobilos) peut être téléchargé à l'adresse suivante:

<http://www.3sigma.fr/Telechargements-Geeros.html>

Son nom est de la forme GeerosArduinoAsservissementMoteursEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Il comporte de nombreux commentaires permettant de comprendre facilement son fonctionnement.

Le principe de pilotage des moteurs consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer aux moteurs (via commande PWM du pont en H intégré sur la carte Romeo) pour qu'il suive la consigne de vitesse spécifiée.

Notez que dans ce programme, la vitesse mesurée est en fait la moyenne glissante des 10 derniers échantillons de mesure de vitesse instantanée, ce qui permet d'avoir une mesure plus lisse. En effet, la résolution de la mesure instantanée est la suivante:

$$2*\pi/(Ts*CPR*ratio)$$

avec :

- Ts : cadence d'échantillonnage (0.01 s)
- CPR : nombre d'impulsions par tour du codeur (48)
- ratio : rapport de réduction du moteur (34)

Ceci donne une résolution de $2*\pi/(0.01*1632) = 0.4$ rad/s. Le moyennage permet d'améliorer cette valeur.

La consigne de vitesse peut se fixer interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

<http://www.3sigma.fr/Telechargements-Geeros.html>

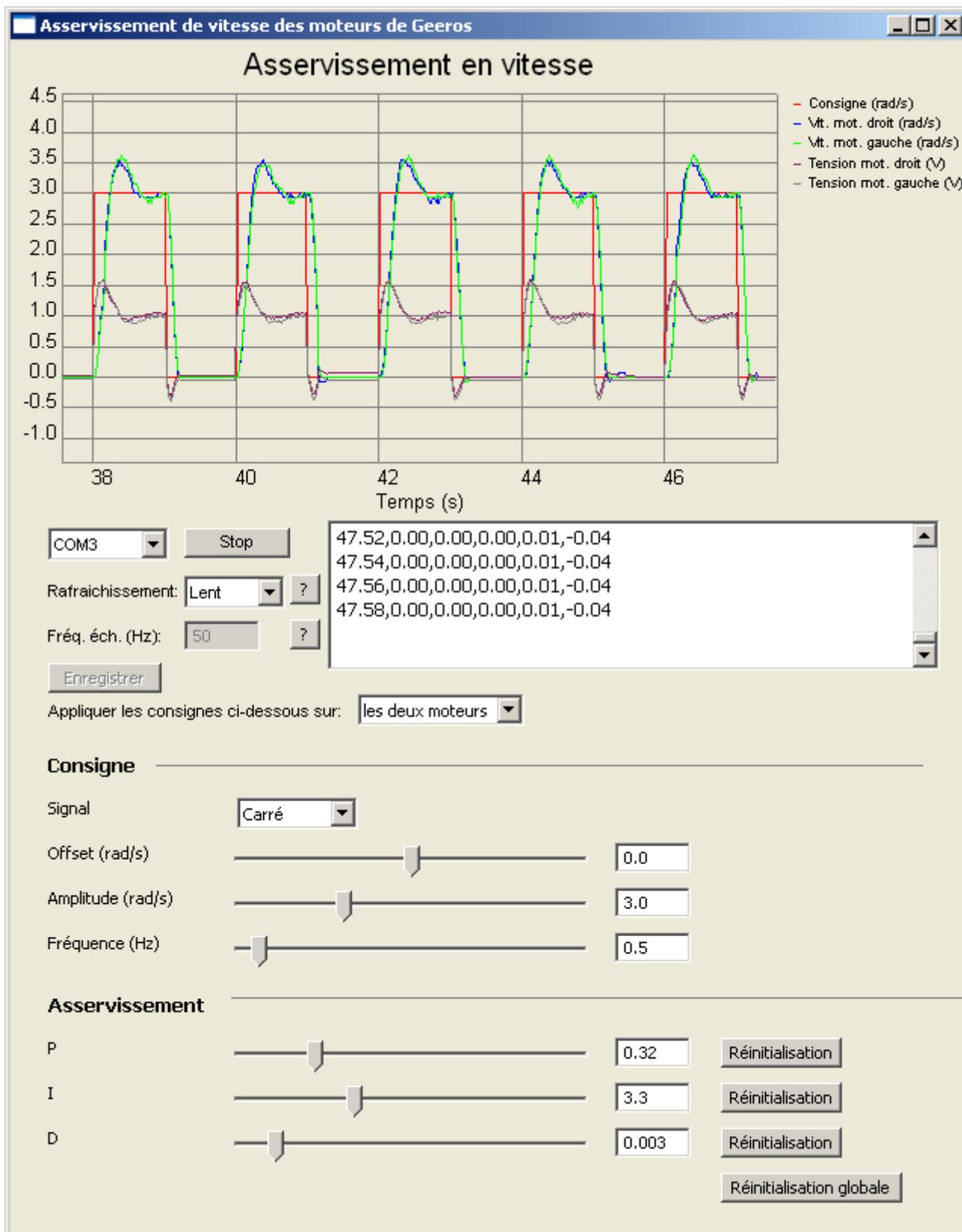
Son nom est de la forme GeerosInterfaceAsservissementMoteursEnVitesse_x.y.zip (x.y correspond au numéro de version du programme).

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur GeerosInterfaceAsservissementMoteursEnVitesse_x.y.exe.

Pour changer la consigne de vitesse du moteur depuis votre ordinateur, il vous suffit de suivre les étapes suivantes (remarque: n'exécutez pas de nouveau celles que vous avez déjà effectuées):

- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I »
- connecter la carte Romeo à l'ordinateur avec le câble USB fourni
- télécharger le programme Arduino « GeerosArduinoAsservissementMoteursEnVitesse » sur la carte Romeo
- lancer l'application « GeerosInterfaceAsservissementMoteursEnVitesse »

Voici une capture d'écran de l'interface de l'application de pilotage:



Les différents éléments sont les suivants (de haut en bas)

- courbes: l'interface permet de visualiser
 - la consigne de vitesse (rad/s, en rouge)
 - les vitesses mesurées (rad/s, en bleu et en vert)
 - les tensions de commande (V, en marron et en gris)
- zone de sélection du port série: choisir le port série sur lequel est connectée votre carte Romeo et cliquer sur le bouton « Connexion ». Vous verrez alors les courbes se mettre en jour automatiquement et défiler des valeurs numériques dans la zone d'affichage à droite
- « Rafraîchissement »: ceci pilote la fréquence de rafraîchissement des courbes. En fonction de la vitesse de votre ordinateur, vous pouvez choisir parmi les 4 valeurs « Minimum », « Lent », « Moyen » et « Rapide ». Plus la fréquence de rafraîchissement est élevée, moins le tracé est saccadé. Mais si votre ordinateur n'est pas très rapide, vous risquez d'observer un retard entre les consignes et l'affichage. Dans ce cas, il faut choisir un rafraîchissement plus lent
- « Fréq. éch. (Hz) »: ce paramètre correspond à la fréquence d'échantillonnage des mesures dans le programme Arduino. Attention: ce paramètre doit être spécifié en Hz, alors que la valeur correspondante dans le programme Arduino (TSDATA) est spécifiée en ms. La relation entre les deux est la suivante:
$$\text{freq (Hz)} = 1000 / \text{cadence (ms)}$$
- Bouton « Enregistrer »: il permet d'enregistrer les valeurs numériques affichées dans la zone de « log » vers un fichier texte pour une utilisation ultérieure avec n'importe quel logiciel permettant de lire des données séparées par des virgules dans un fichier texte.
Attention : ce bouton n'est actif qu'après une séquence de mesure. Il est grisé le reste du temps (au démarrage du programme et pendant une séquence de mesure)
- « Appliquer les consignes ci-dessous sur: »: permet d'appliquer les consignes sur le moteur gauche, le moteur droit ou les deux moteurs
- Zone « Consigne »: elle permet de modifier la consigne de vitesse avec les curseurs. La vitesse des moteurs varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel
- Zone « Asservissement »: elle permet de modifier les gains du régulateur PID pendant le fonctionnement du système. Cela permet de voir l'influence de ces gains sur les performances de l'asservissement et de régler précisément ce dernier.

IMPORTANT !

Si l'affichage ne suit pas les consignes même avec une fréquence de rafraîchissement très lente, veuillez diminuer la cadence d'échantillonnage dans le programme Arduino (augmenter la valeur TSDATA). Ne pas modifier la valeur CADENCE_MS, sous peine de déstabiliser l'asservissement.

8 - Autre documentation disponible

Cette documentation, bien que baptisée « complète », ne détaille pas certains points qui sont présents dans une autre documentation disponible sur simple demande à l'adresse support@3sigma.fr :

- Documentation pédagogique: elle contient tous les éléments relatifs à la mise en équations du robot ainsi que les documents Maple et modèles MapleSim associés. Cette documentation, utile dans le cadre de l'enseignement, n'est pas en accès libre car elle contient des informations qui peuvent être les solutions d'exercices proposés par les professeurs. Ceux-ci sont donc les seuls habilités à y avoir accès.

9 - Important

Mobilos est un produit « vivant » en constant développement pour 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).

Ne restez jamais bloqué sans nous contacter !

Pour tout problème ou toute requête, contactez-nous à l'adresse support@3sigma.fr.

10 - ANNEXE 1 – Circuit de suivi de ligne

Les deux pages suivantes peuvent être imprimées pour créer un circuit de suivi de ligne pour Mobilos. Vous pouvez créer d'autres branches de circuit avec n'importe quel logiciel de dessin, à condition de respecter les mêmes caractéristiques:

- l'épaisseur de la ligne est de 200 pixels, ce qui permet de couvrir la largeur des 3 capteurs de suivi de ligne du robot
- le rayon de courbure ne doit pas être trop petit. Le virage proposé ci-après peut être considéré comme la limite réalisable par le robot

L'impression doit être réalisée sur des feuilles de papier blanc, en échelle de gris (ce qui permet d'obtenir un « vrai » noir), avec une qualité suffisamment bonne.

Lors de la réalisation d'un circuit, il est important de bien joindre les feuilles (voir de les faire chevaucher) pour qu'il n'y ait pas d'espace entre elles, même sur les bords éloignés de la ligne. En effet, si le robot quitte la ligne et qu'il rencontre un espace entre deux feuilles, il risque de croire qu'il a retrouvé la ligne et partir définitivement dans la mauvaise direction.

Enfin, nous vous conseillons de relire le paragraphe 6.2, relatif à la mise en œuvre de Mobilos pour le suivi de ligne.

