



ENSEMBLE « COMMANDE DE MOTEUR À COURANT CONTINU »

DOCUMENTATION COMPLÈTE



Date de dernière mise à jour : 26/11/2012

Table des matières

1 - Introduction.....	3
2 - Matériel inclus.....	3
3 - Conformité.....	3
4 - Installation de l'IDE Arduino.....	4
4.1 - Installation principale.....	4
4.2 - Installation de la bibliothèque complémentaire FlexiTimer2.....	4
4.3 - Installation de la bibliothèque complémentaire digitalWriteFast.....	4
5 - Mise en œuvre de l'ensemble.....	5
5.1 - Utilisation standard.....	5
5.2 - Précautions d'emploi.....	5
5.2.1 - Connexions d'alimentation sur la carte Romeo.....	5
5.2.2 - Tension d'alimentation.....	5
5.2.3 - Utilisation.....	6
6 - Expériences de commande de moteur à courant continu.....	7
6.1 - Commande en tension.....	7
6.2 - Asservissement de vitesse.....	9

1 - Introduction

Cet ensemble permet de réaliser différentes expériences sur la base d'un moteur à courant continu avec codeur incrémental associé, commandé par une carte Romeo, compatible Arduino Uno.

Une documentation détaillée de la carte Romeo ainsi que les programmes associés sont téléchargeables sur notre site Web à l'adresse suivante:

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

2 - Matériel inclus

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

- une plaque de prototypage mécanique en plexiglas bleu, avec connecteur d'alimentation 5.5mm x 2.1mm et bouton marche-arrêt
- une carte Romeo compatible Arduino Uno
- un moteur à courant continu 6V, rapport de réduction 34:1, avec codeur incrémental 48 CPR (1632 impulsions par tour de l'arbre d'entraînement). Le moteur est assemblé sur la plaque de prototypage par l'intermédiaire d'un support de montage en équerre
- 1 câble USB A-B pour la programmation de la carte Romeo
- 1 alimentation 9V, 1A avec connecteur d'alimentation 5.5mm x 2.1mm
- divers éléments de montage mécanique (entretoises nylon 1 et 2 cm, vis, écrous)

3 - Conformité

L'ensemble « Commande de moteur à courant continu », **dans sa configuration livrée aux clients**, est conforme à la directive 1999/EC.

4 - Installation de l'IDE Arduino

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. Notez bien ce répertoire car vous aurez besoin de le retrouver si pour installer des librairies additionnelles.

ATTENTION !

Avant de pouvoir compiler les programmes fournis avec cet ensemble, vous devez suivre les instructions suivantes pour installer les deux bibliothèques complémentaires **FlexiTimer2** et **digitalWriteFast**.

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-là dans le sous-répertoire « libraries » de votre installation Arduino

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-là dans le sous-répertoire « libraries » de votre installation Arduino

5 - Mise en œuvre de l'ensemble

5.1 - Utilisation standard

La mise en œuvre de l'ensemble « Commande de moteur à courant continu » est très simple:

- brancher l'alimentation 9V fournie sur le connecteur jack du plateau portant la carte Romeo et le moteur
- commuter l'interrupteur sur la position « I »
- connecter la carte Romeo à votre ordinateur à l'aide du câble USB fourni

5.2 - Précautions d'emploi

Nous insistons sur le fait que cet ensemble est un matériel de développement qui nécessite un certain nombre de précautions d'emploi.

5.2.1 - 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. L'ensemble « Commande de moteur à courant continu » étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

5.2.2 - Tension d'alimentation

Cet ensemble est prévu pour fonctionner avec l'alimentation 9V, 1A fournie. Le moteur a une tension nominale de 6V et bien qu'il supporte sans problème des tensions jusqu'à 9V, le programme préchargé dans la carte (ainsi que tous les programmes téléchargeables sur notre site) empêche que la tension à ses bornes dépasse la valeur de 6V.

Il est possible d'utiliser un autre bloc d'alimentation à condition de bien respecter les points suivants:

- la tension ne doit pas dépasser 12V. Il est cependant recommandé de mettre éventuellement les programmes de la carte Arduino à jour si ceux-ci sont basés sur une valeur de tension différente
- la polarité doit être « positif au centre du connecteur »

- l'alimentation doit pouvoir fournir un courant suffisant, de préférence supérieur ou égal à 1A

5.2.3 - Utilisation

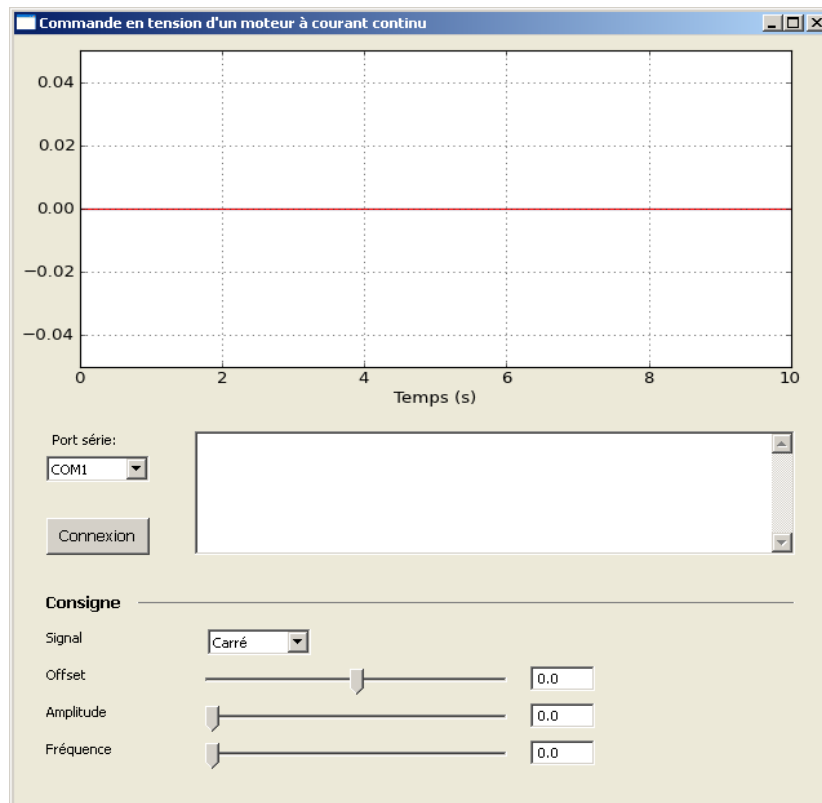
Il est fortement déconseillé de faire des expériences de fonctionnement « rotor bloqué » avec une tension d'alimentation du moteur trop élevée. Ce type d'expérience peut générer des courants trop forts qui réduisent la durée des vies des éléments.

6 - Expériences de commande de moteur à courant continu

Cet ensemble permet de réaliser différentes expériences de commande de moteur à courant continu.

6.1 - Commande en tension

Cette expérience (pré-chargée dans la carte Romeo à la livraison) permet de changer la vitesse de rotation du moteur en appliquant une tension variable par l'intermédiaire d'une application qui s'exécute sur votre ordinateur:



Le programme Arduino peut être téléchargé à l'adresse suivante:

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

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

Le principe de pilotage du moteur consiste à envoyer des signaux PWM sur le pont en H intégré à la carte Romeo afin de faire varier la tension d'alimentation du moteur. 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/CommandeMoteurEnTension.zip>

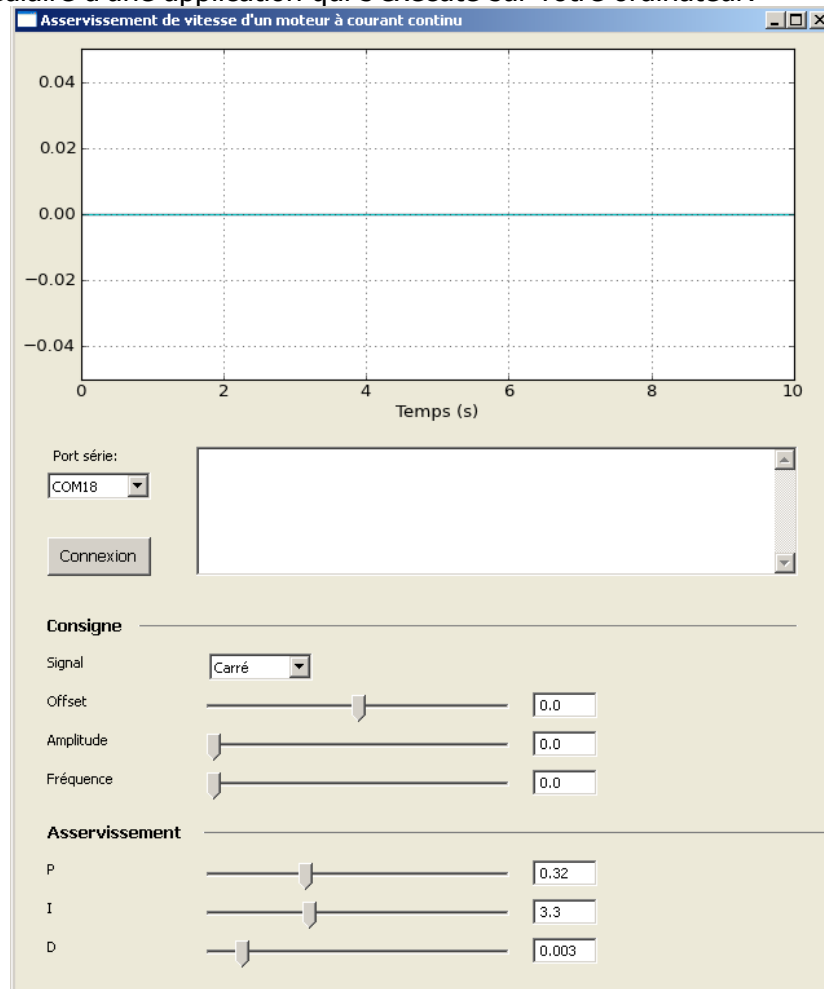
Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter, double-cliquer sur `CommandeMoteurEnTension.exe`.

Pour piloter la tension d'alimentation (et donc la 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):

- télécharger l'application de pilotage « `CommandeMoteurEnTension` » à l'adresse ci-dessus et l'installer
- télécharger et installer l'IDE Arduino (voir paragraphe 4)
- connecter l'alimentation 9V sur le connecteur jack du plateau supportant la carte Romeo et le moteur (voir paragraphe 5.1)
- connecter la carte Romeo à l'ordinateur avec le câble USB fourni (voir paragraphe 5.1)
- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I » (voir paragraphe 5.1)
- lancer l'application « `CommandeMoteurEnTension` »
- choisir le port série sur lequel est connectée votre carte Romeo et cliquer sur le bouton « Connexion »
- modifier la consigne de tension avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne, sur le graphique qui s'affiche en temps-réel

6.2 - Asservissement de vitesse

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



Le programme Arduino peut être téléchargé à l'adresse suivante:

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

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

Le principe de pilotage du moteur consiste à calculer, grâce à un régulateur de type PID, la tension de commande à appliquer au moteur (via commande PWM du pont en H intégré sur la carte Romeo) pour qu'il suive la consigne de vitesse spécifiée interactivement via une application qui s'exécute sur votre ordinateur. Elle peut se télécharger à l'adresse suivante:

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

Pour l'installer, il suffit de décompresser l'archive dans le répertoire de votre choix. Pour l'exécuter,

double-cliquer sur AsservissementMoteurEnVitesse.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):

- télécharger l'application de pilotage «AsservissementMoteurEnVitesse» à l'adresse ci-dessus et l'installer
- télécharger et installer l'IDE Arduino (voir paragraphe 4)
- connecter l'alimentation 9V sur le connecteur jack du plateau supportant la carte Romeo et le moteur (voir paragraphe 5.1)
- connecter la carte Romeo à l'ordinateur avec le câble USB fourni (voir paragraphe 5.1)
- mettre le système sous tension en positionnant le bouton marche-arrêt sur « I » (voir paragraphe 5.1)
- lancer l'application «AsservissementMoteurEnVitesse»
- choisir le port série sur lequel est connectée votre carte Romeo et cliquer sur le bouton « Connexion »
- modifier la consigne de vitesse avec les curseurs. La vitesse du moteur varie alors. Vous pouvez la visualiser, ainsi que la consigne et la tension de commande, sur le graphique qui s'affiche en temps-réel

Cette application permet également 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.