



X-Bot CHIP C / Python

Documentation complète

16/04/2018
Version 1.0

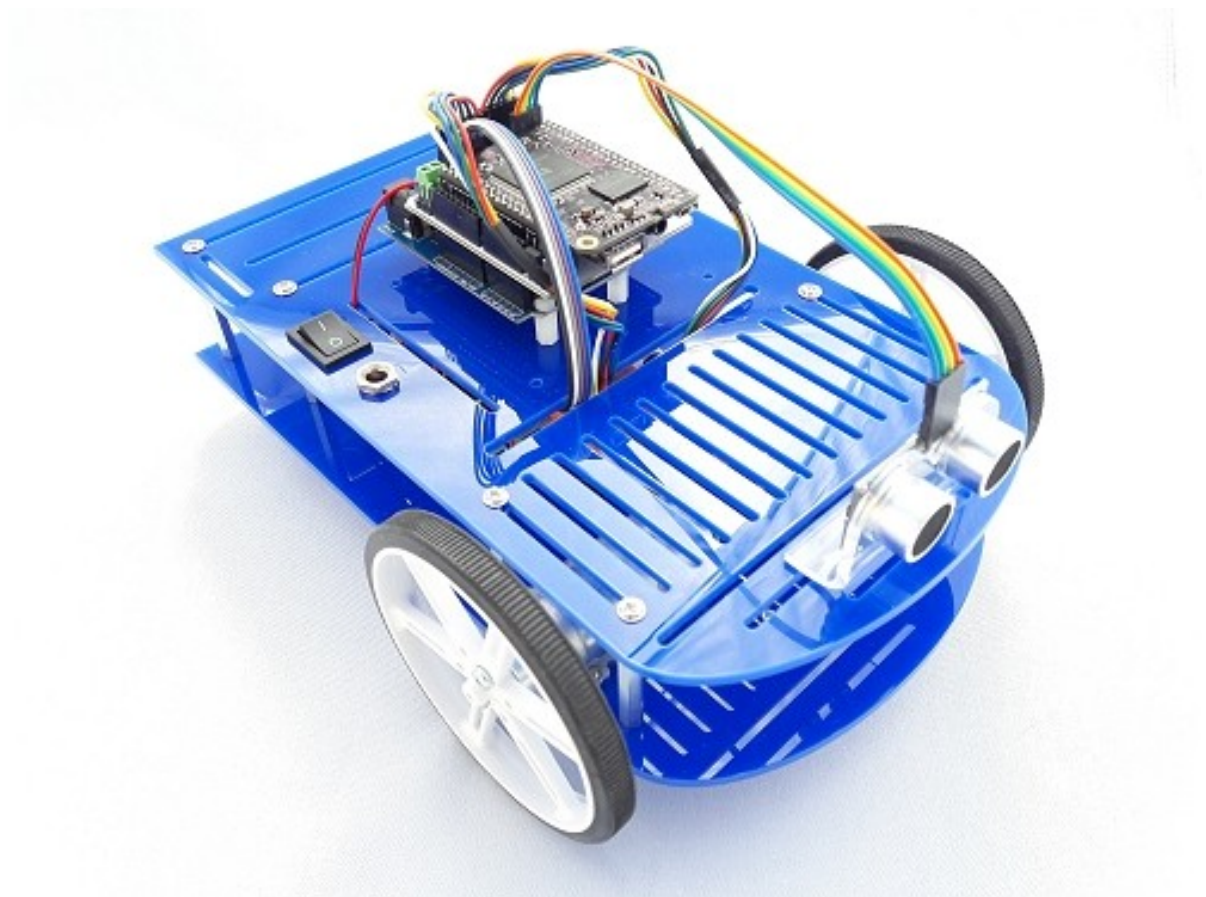


Table des matières

1 Introduction.....	4
2 Matériel inclus.....	5
3 Conformité.....	5
4 Présentation du matériel.....	6
4.1 Carte Iteduino Uno.....	9
4.2 Shield « robotique ».....	9
4.3 Moteurs électriques.....	10
4.3.1 Équations et fonctions de transfert.....	10
4.3.2 Calcul de la vitesse de rotation.....	11
4.3.3 Remarque sur le comptage des impulsions.....	13
4.4 Batterie LiPo.....	13
4.5 Communication Wifi.....	14
5 Installation des logiciels.....	15
5.1 Installation de l'environnement de développement Arduino.....	15
5.1.1 Installation principale.....	15
5.1.2 Installation de la bibliothèque complémentaire FlexiTimer2.....	16
5.1.3 Installation de la bibliothèque complémentaire digitalWriteFast.....	16
5.1.4 Installation de la bibliothèque complémentaire EnableInterrupt.....	16
5.1.5 Installation de la bibliothèque complémentaire NewPing.....	17
5.1.6 Choix de la carte Arduino dans l'IDE.....	17
5.1.7 Programmation de la carte Arduino.....	17
5.2 Installation et première utilisation du logiciel MyViz.....	18
6 Mise en œuvre de l'ensemble.....	20
6.1 Précautions d'emploi.....	20
6.1.1 Connexions d'alimentation sur le shield « robotique ».....	20
6.1.2 Recharge de la batterie.....	20
6.1.3 Précautions d'utilisation.....	21
6.2 Première utilisation.....	22
6.2.1 Pilotage par ordinateur.....	22
6.2.2 Pilotage par smartphone ou tablette.....	24

7 Activités réalisables avec le système en mode CHIP / Python.....	26
7.1 Contrôle de vitesse et pilotage en Wifi.....	27
7.1.1 Présentation.....	27
7.1.2 Expérimentations possibles.....	29
7.2 Suivi de ligne.....	30
7.2.1 Expérimentations possibles.....	32
7.3 Évitement d'obstacles.....	33
7.3.1 Expérimentations possibles.....	35
7.4 Commande des moteurs en tension.....	36
7.4.1 Présentation.....	36
7.4.2 Expérimentations possibles.....	38
7.5 Asservissement des moteurs en vitesse.....	40
7.5.1 Expérimentations possibles.....	42
8 Activités réalisables avec le système en mode Arduino / C.....	43
8.1 Trajectoire pré-programmée.....	45
8.1.1 Présentation.....	45
8.2 Commande des moteurs en tension.....	46
8.3 Asservissement des moteurs en vitesse.....	46
9 Support technique.....	47

1 Introduction

X-Bot CHIP C / Python est un robot ouvert et open-source, un concentré de technologie vous permettant de faire de nombreuses expériences. Il est basé sur une carte CHIP, une carte compatible Arduino, différents capteurs et deux moteurs électriques permettant de le mettre en mouvement.

Il intègre une liaison Wifi vous permettant de le piloter à distance à partir d'un ordinateur, d'un smartphone ou d'une tablette.

Les programmes associés à ce système sont téléchargeables sur notre site Web à l'adresse suivante :

http://www.3sigma.fr/xbotchip/fichiers/X-Bot_CHIP_C_Python.zip.

Cette archive contient la présente documentation et 3 sous-répertoires :

- Arduino : programmes Arduino permettant de réaliser des expériences utilisant cette carte exclusivement (voir chapitre 8) ou en liaison avec la CHIP (voir chapitre 7)
- MyViz : tableaux de bord des différentes expériences réalisables avec le robot (voir les chapitres 5.2 pour l'installation du logiciel, 7 pour les expériences utilisant la carte CHIP et 8 pour celles utilisant la carte compatible Arduino seule)
- programmes_python : programmes stockés sur la carte CHIP (dans le répertoire /root/programmes_python) pour les expériences mettant en œuvre le robot programmé en Python (voir chapitre 7)

2 Matériel inclus

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

- Le robot lui-même
- 1 chargeur de batterie LiPo 8.4V, 1A
- 1 câble mini USB pour la programmation de la carte compatible Arduino
- 1 clé allen pour visser ou dévisser les roues

3 Conformité

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

4 Présentation du matériel

Les différents éléments constituant X-Bot CHIP C / Python sont présentés ci-dessous :

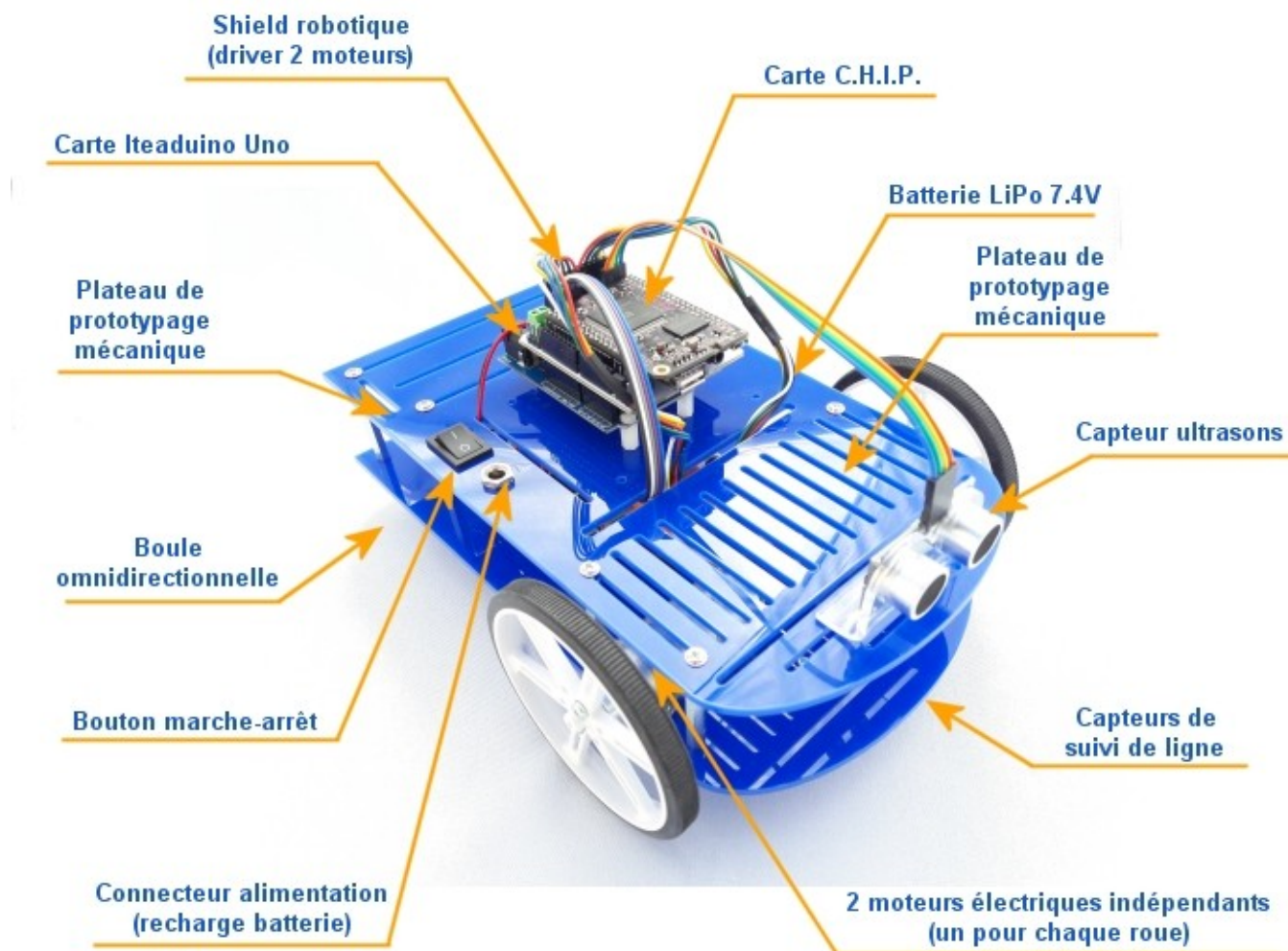
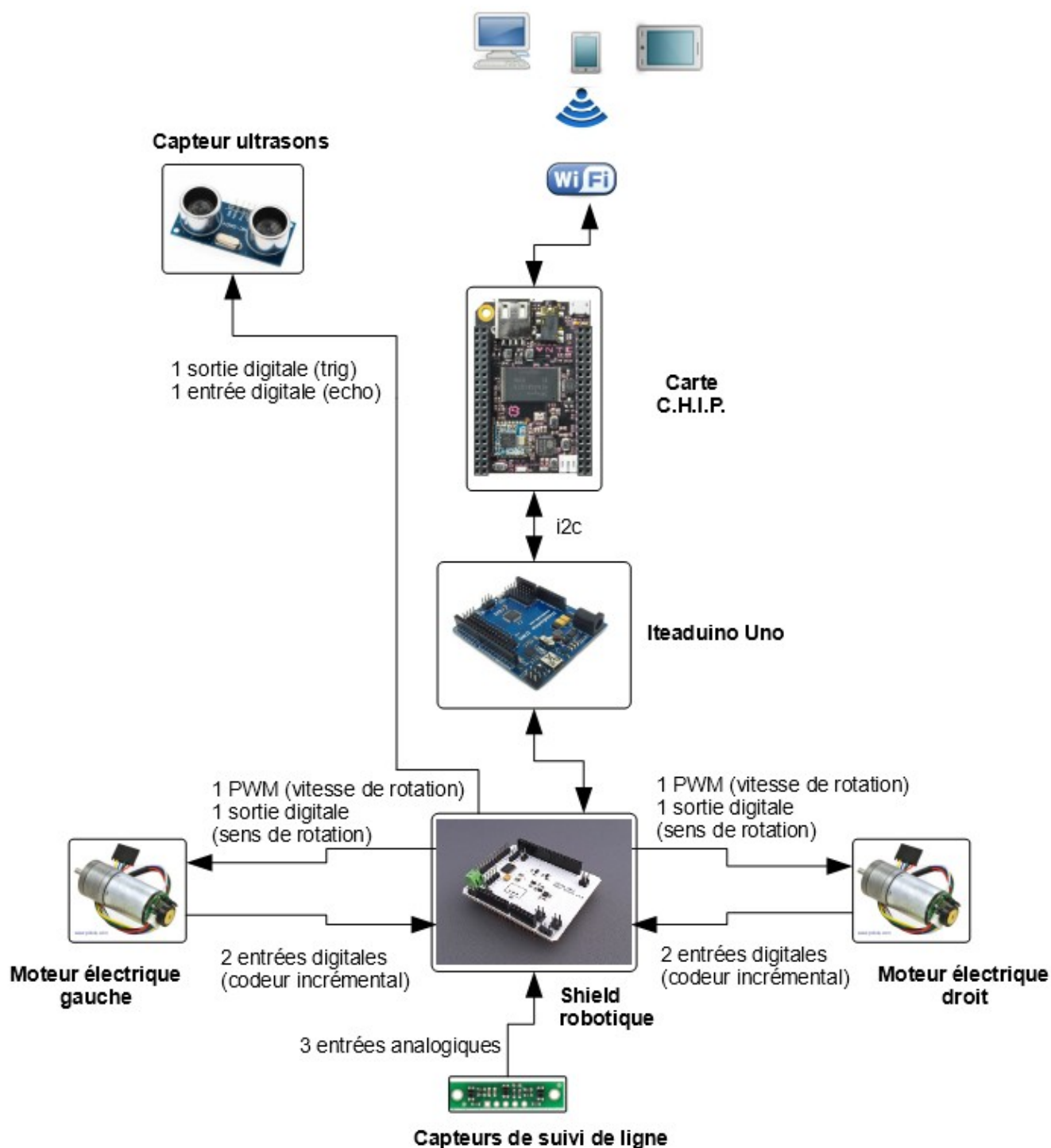


Diagramme des échanges de signaux :



Les références des principaux composants sont les suivantes :

- C.H.I.P. : <https://docs.getchip.com/chip.html#chip-hardware>
- Iteduino Uno : voir plus loin
- shield « robotique » : voir plus loin
- batterie Lipo :
<http://boutique.3sigma.fr/14-batterie-lipo-74-v-2200-mah-connecteur-jack-arduino.html>
- moteurs électriques :
<http://boutique.3sigma.fr/12-moteur-%C3%A0-courant-continu-r%C3%A9ducteur-341-codeur-incr%C3%A9mental-48-cpr.html>
- supports moteurs :
<http://boutique.3sigma.fr/16-paire-de-support-de-moteur-pololu-25d.html>
- roues :
<http://boutique.3sigma.fr/18-paire-de-roues-blanches-pololu-90x10mm.html>
- moyeux :
<http://boutique.3sigma.fr/17-paire-de-moyeux-aluminium-universels-pololu-m3-pour-arbre-de-4mm.html>
- capteurs de suivi de ligne :
<http://boutique.3sigma.fr/81-capteurs-de-suivi-de-ligne.html>
- capteur ultrasons :
<http://boutique.3sigma.fr/127-capteur-de-distance-ultrason-hc-sr04.html>

Certains éléments ou points techniques sont présentés plus en détails ci-dessous.

4.1 Carte Iteaduino Uno

Cette carte est 100 % compatible Arduino Uno. Elle a été choisie à la place de cette dernière pour les raisons suivantes :

- elle est bi-tension et permet par conséquent de fonctionner en 3.3V. Elle peut ainsi être directement connectée sur la carte CHIP
- elle possède un convertisseur de tension très efficace permettant d'alimenter la CHIP en 5V / 1A

Cette carte a deux utilisations :

- elle est utilisée pour le mode « programmation en C / Arduino »
- elle est également utilisée en mode « programmation Python » pour le comptage des impulsions des codeurs incrémentaux des moteurs. En effet, la carte CHIP ne peut pas le faire d'une façon aussi performante et fiable (ce n'est pas un système temps-réel, contrairement à un micro-contrôleur dédié). La communication entre les cartes Iteaduino Uno et CHIP (la première envoie à la seconde le nombre d'impulsions codeur de chaque moteur toutes les 10 ms) se fait par i2c.

4.2 Shield « robotique »

Un shield est une carte possédant des connecteurs dont l'emplacement permet un branchement direct sur la carte Arduino principale. X-Bot utilise un shield développé par 3Sigma pour ses robots et dont les caractéristiques qui nous intéressent ici sont les suivantes :

- compatible avec les cartes Arduino 5V ou 3.3V
- tension d'alimentation comprise entre 2.5V et 13.5V
- alimentation par bornier à vis
- permet d'alimenter le système sur lequel il est connecté (la tension d'alimentation doit alors être supérieure à 7 V)
- pont diviseur de tension (rapport 3/13) pour la mesure de la tension d'alimentation sur la broche A0
- intègre un driver double moteur à courant continu Toshiba TB6612FNG (courant maximal en pic: 3.2 A)
- PWM sur les broches D5 et D6
- variation de vitesse et changement de direction grâce au pilotage des deux broches PWM
- fréquence PWM max: 100 kHz
- connecteurs directement compatibles avec les moteurs Pololu 25D avec codeur incrémental (signaux « codeur A » routés sur les broches D2 et D3 et signaux « codeur B » routés sur les broches D8 et D9)
- branchement direct des capteurs de suivi de ligne (routés sur les broches analogiques A1, A2 et A3)
- branchement direct d'un capteur ultrasons (signaux « trig » et « echo » routés sur les broches D10 et D13)

4.3 Moteurs électriques

4.3.1 Équations et fonctions de transfert

Ce robot embarque deux moteurs à courant continu 6V, de rapport de réduction 34:1, avec codeur incrémental 48 CPR (Counts Per Revolution).

Les équations d'un moteur sont les suivantes:

$$J_m n^2 \frac{d}{dt} \omega_m(t) + d \omega_m(t) - n K i_m(t) = Tr(t) \quad (4.1)$$

$$L \frac{d}{dt} i_m(t) + R_m i_m(t) = V(t) - K n \omega_m(t) \quad (4.2)$$

Avec :

- R_m : résistance électrique interne : 2.5Ω
- L : inductance des enroulements : 3 mH
- J_m : moment d'inertie du rotor : $3 \cdot 10^{-6} \text{ kg.m}^2$
- K : constante de couple = constante de fem : 0.01 N.m/a
- d : coefficient de frottement visqueux : 0.005 N.m.s/rad
- ω_m : vitesse de rotation de l'arbre de sortie du réducteur
- i_m : courant dans le moteur
- V : tension d'alimentation
- Tr : couple résistant
- n : rapport de réduction (34)

Ces paramètres ont été identifiés via notre système de commande de moteur électrique (<http://boutique.3sigma.fr/23-commande-de-moteur-%C3%A9lectrique.html>).

La fonction de transfert entre l'entrée $V(t)$ et la sortie $\omega_m(t)$ est la suivante :

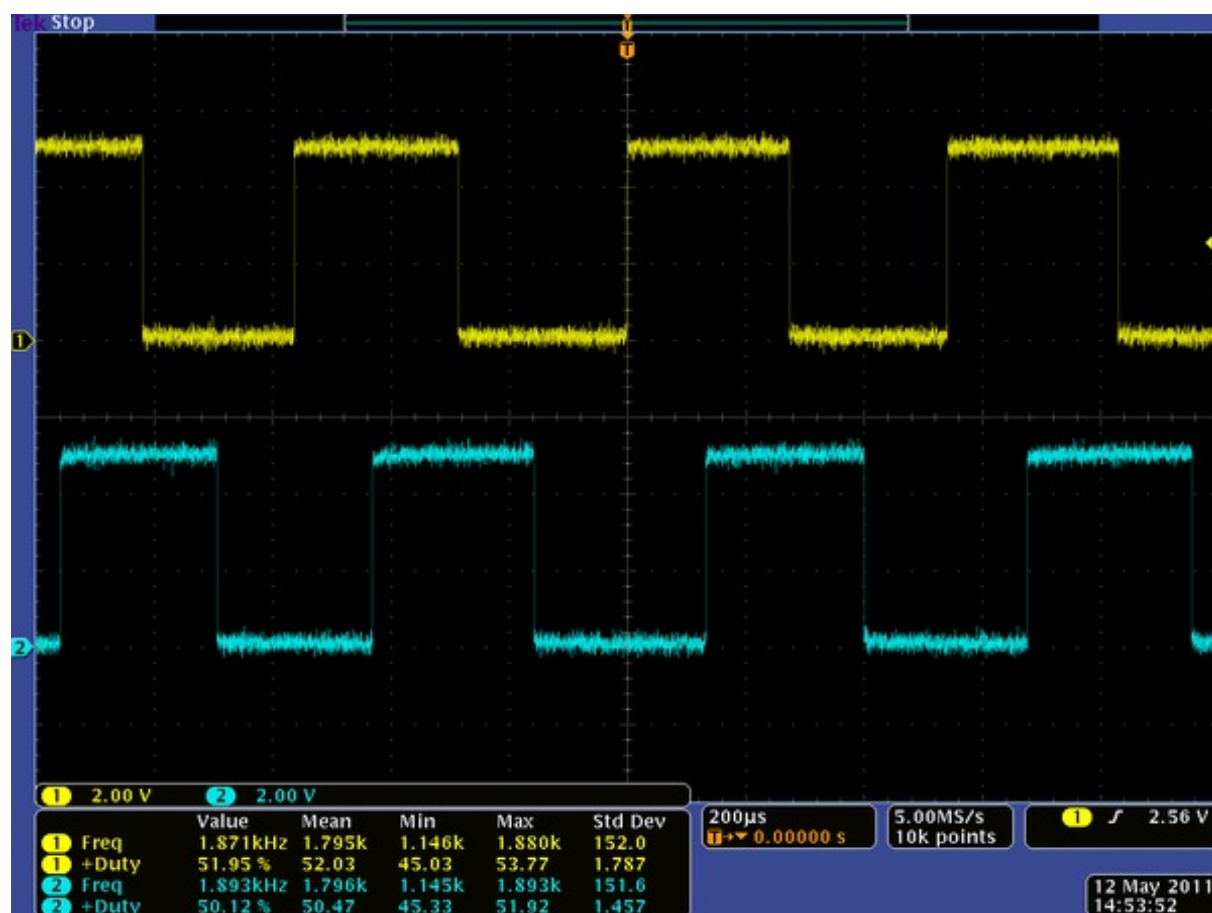
$$\frac{Kn}{JLn^2s^2 + (JRn^2 + Ld)s + K^2n^2 + Rd} \quad (4.3)$$

La fonction de transfert entre l'entrée $V(t)$ et la sortie $i_m(t)$ est la suivante :

$$\frac{Jn^2s + d}{JLn^2s^2 + (JRn^2 + Ld)s + K^2n^2 + Rd} \quad (4.4)$$

4.3.2 Calcul de la vitesse de rotation

Les moteurs à courant continu du robot 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 34:1, l'arbre moteur fait 34 tours lorsque l'arbre « principal » en fait 1
- Le codeur génère 48 impulsions à chaque fois qu'il fait un tour. Cependant, la carte Iteduino qui gère le comptage des impulsions ne dispose que de deux lignes d'interruptions matérielles (une pour chaque moteur). Par conséquent, on ne prend en compte qu'une interruption sur 2
- La cadence d'échantillonnage utilisée pour l'asservissement est de 0.01 s

Par conséquent, lorsque l'arbre principal fait un tour, le codeur génère : $34 \times 24 = 816$ 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) :

$$\frac{2\pi N}{0.01 * 816} \quad (4.5)$$

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) :

$$\frac{2\pi N}{T_s \text{ CPR } n} \quad (4.6)$$

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 :

$$\frac{2\pi}{0.01 * 816} = 0.77 \text{ rad/s} \quad (4.7)$$

Enfin, pour l'améliorer, nous faisons une moyenne sur les 10 derniers échantillons. Ceci introduit un retard (à prendre en compte dans les asservissements) mais permet d'avoir une résolution de l'ordre de **0.08 rad/s**.

4.3.3 Remarque sur le comptage des impulsions

Le comptage des impulsions revient à compter le nombre de fronts montants du signal jaune représenté sur l'image ci-dessus. Pour ce faire, la seule méthode viable consiste à brancher ce signal (le fil jaune du codeur utilisé) sur une des deux entrées « interruption matérielle » de la carte Iteduino Uno. Le fil blanc est branché sur une entrée digitale classique et les deux derniers fils (bleu et vert) seront respectivement branchés sur le 3.3 V et sur la masse du système.

L'intérêt d'une ligne d'interruption est qu'elle permet, comme son nom l'indique, d'interrompre le déroulement des calculs sur le processeur 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 positif, lors d'une interruption sur la voie A, les niveaux de A et B sont donc inversés. Pour connaître le sens de rotation du moteur, il faut donc mesurer le niveau de B lorsque l'interruption survient.

4.4 Batterie LiPo

Contrairement à ce que l'on trouve couramment en modélisme, cette batterie est « protégée » pour qu'elle ne puisse pas délivrer des courants trop forts.

4.5 Communication Wifi

X-Bot embarque une carte CHIP avec Wifi intégré lui permettant d'être piloté par certains appareils possédant une connectivité Wifi.

Lorsqu'il est démarré, la connexion est disponible après environ une minute. X-Bot se comporte alors comme point d'accès Wifi autonome: un nouveau réseau Wifi devient alors visible sur vos appareils, avec le SSID « X-Bot-CHIP-xxx ». Ce réseau est sécurisé, le mot de passe est « xbotchip ».

Notez les points suivants:

- Un appareil donné ne peut pas en général se connecter en même temps à deux réseaux Wifi différents: la connexion au réseau X-Bot entraînera la déconnexion du réseau auquel vous étiez éventuellement connecté au préalable. Vous n'aurez alors plus accès à Internet sur votre appareil, sauf si celui-ci est
 - un ordinateur connecté en parallèle en filaire (liaison Ethernet). Attention cependant: si ce réseau Ethernet utilise la même plage d'adresse (192.168.0.xxx), vous devrez probablement le désactiver pour vous connecter à l'adresse 192.168.0.199 de X-Bot en Wifi
 - un smartphone ou une tablette connecté à un réseau 3G en parallèle
- X-Bot fonctionne comme un point d'accès « maître ». Il ne peut pas, dans sa configuration de base, se connecter comme client du réseau Wifi de votre organisation
- L'adresse de X-Bot est fixée à 192.168.0.199. Pour que votre appareil puisse accéder au robot, il doit accepter l'adressage automatique (adresse fournie par serveur DHCP). Si votre appareil n'accepte pas l'adressage automatique, cela signifie que vous lui avez donné une adresse IP fixe. Dans ce cas, celle-ci doit commencer par 192.168.0. (elle doit donc être de la forme 192.168.0.xxx). Si ce n'est pas le cas, vous ne pourrez pas communiquer avec X-Bot.

Enfin, le système CHIP est totalement accessible via une connexion SSH (vous pouvez utiliser sur Windows PuTTY pour un accès en ligne de commande ou WinSCP pour un accès via un explorateur graphique). Les informations de connexion sont les suivantes :

- login : root
- mot de passe : xbot

5 Installation des logiciels

Les expériences réalisables avec ce robot utilisent la carte Iteaduino (compatible Arduino) Uno seule ou en association avec la carte CHIP. Elles sont par ailleurs chacune basées sur une interface graphique sous la forme d'un tableau de bord exécuté par le logiciel MyViz.

5.1 Installation de l'environnement de développement Arduino

Cette section présente les étapes à mettre en œuvre (une fois pour toute) pour mettre en place sur votre ordinateur l'environnement de programmation de la carte Iteaduino Uno.

Attention : si les possibilités de programmation de la carte Arduino ne vous intéressent pas, c'est-à-dire si seules les expériences programmées en Python sur la carte CHIP vous intéressent, vous pouvez sauter ce qui suit et aller directement au paragraphe 5.2. En effet, la carte Iteaduino est déjà programmée lorsque vous recevez le robot.

5.1.1 Installation principale

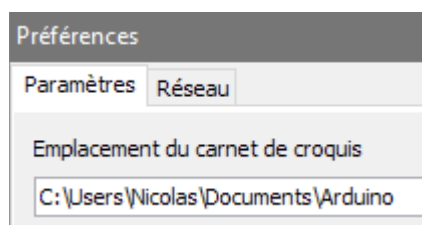
L'IDE Arduino en version supérieure ou égale à 1.8 doit tout d'abord être téléchargé (<http://arduino.cc/en/Main/Software>) et installé (<http://arduino.cc/en/Guide/HomePage>).

5.1.2 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 répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries sur Windows). Pour identifier ce répertoire sur votre ordinateur, lancez l'IDE Arduino et sélectionnez **Fichier → Préférences**, l'information recherchée se trouve en haut de la fenêtre qui s'ouvre alors (« Emplacement du carnet de croquis »). Par exemple :



5.1.3 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 répertoire des librairies de votre installation Arduino (voir les explications détaillées ci-dessus)

5.1.4 Installation de la bibliothèque complémentaire EnableInterrupt

Cette bibliothèque permet de gérer les interruptions.

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

Une fois téléchargée, décompressez-là dans le répertoire des librairies de votre installation Arduino (voir les explications détaillées ci-dessus)

5.1.5 Installation de la bibliothèque complémentaire NewPing

Cette bibliothèque permet de gérer le capteur ultrasons.

Vous pouvez la télécharger à l'adresse suivante:

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

Une fois téléchargée, décompressez-là dans le répertoire des librairies de votre installation Arduino (voir les explications détaillées ci-dessus)

ATTENTION !

L'environnement Arduino doit être redémarré après l'installation d'une bibliothèque complémentaire.

5.1.6 Choix de la carte Arduino dans l'IDE

L'IDE Arduino permet de programmer un très grand nombre de cartes. Il faut par conséquent le configurer pour fonctionner avec la carte compatible Uno. Dans le menu **Outils → Type de carte**, choisir Arduino / Genuino Uno.

Lorsque la carte l'arduino est reliée à l'ordinateur via le câble mini USB fourni, il est également possible de choisir le port de cet ordinateur dans l'IDE Arduino en sélectionnant, dans la liste **Outils → Ports**, celui correspondant à l'arduino.

5.1.7 Programmation de la carte Arduino

Enfin, la programmation de la carte se fait en ouvrant (depuis l'IDE Arduino) le programme que vous souhaitez utiliser et en sélectionnant **Croquis → Téléverser**. La programmation est faite quand il s'affiche « Téléversement terminé » dans la zone bleue en bas de l'écran :

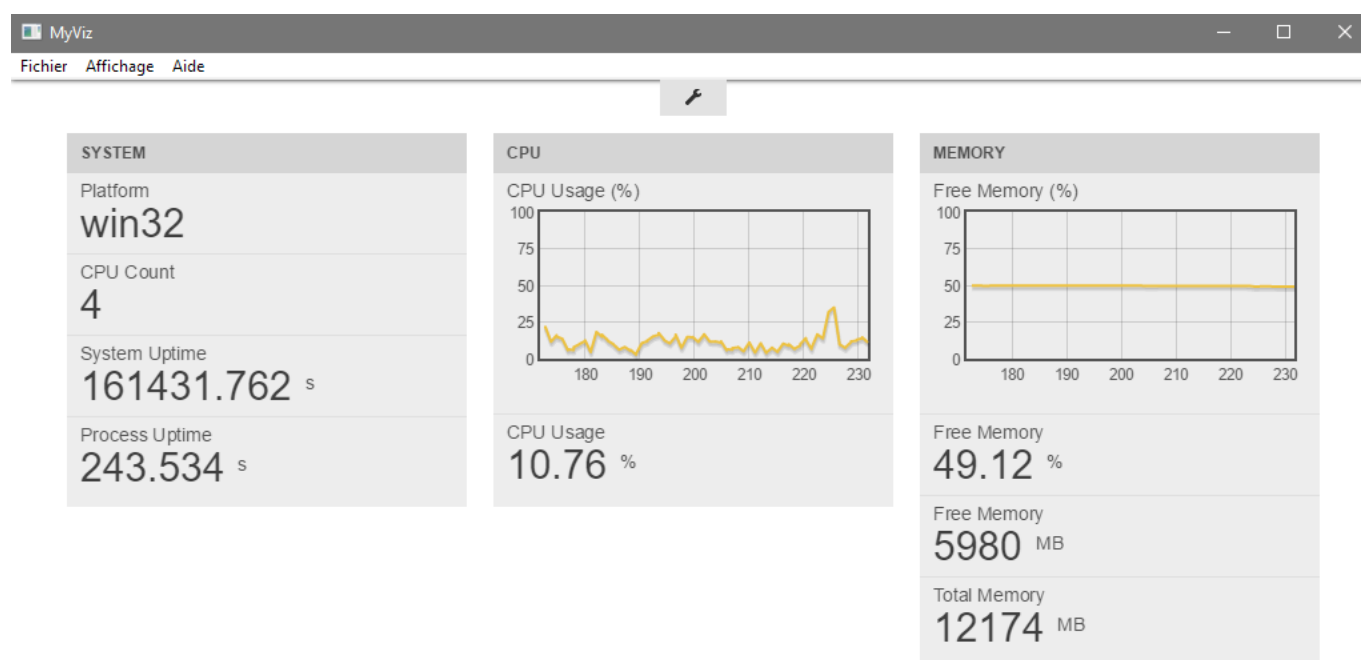


5.2 Installation et première utilisation du logiciel MyViz

MyViz s'installe très facilement après l'avoir téléchargé depuis la page suivante :

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

Le tableau de bord initialement affiché sera similaire à la capture d'écran ci-dessous :



Ce tableau de bord, qui affiche les ressources utilisées de votre système, est un exemple de ce qui peut être réalisé avec MyViz. Bien sûr, ce n'est pas ce que vous souhaitez faire ici !

Le robot est livré avec des tableaux de bord tous prêts que vous pouvez directement utiliser sans lire la fin de ce chapitre. Néanmoins, il est intéressant de connaître le principe de création ou de modification d'une interface graphique :

- La conception d'un tableau de bord se fait en mode édition, après avoir cliqué sur la clé en haut de l'écran pour déplier le panneau supérieur :



- Cela permet d'ouvrir le panneau supérieur du mode "édition" :



- Vous pouvez alors définir dans cette zone :
 - Une ou plusieurs sources de données accessible(s) via un canal de communication. Dans l'exemple ci-dessus, la source de données contient les ressources de votre système, directement accessibles sur l'ordinateur. Il n'y a donc pas réellement de canal de communication. Mais dans le système du robot, les tableaux de bord reposent sur des données envoyées par la carte Iteaduino Arduino via une liaison série ou par la carte CHIP via Wifi (communication par websockets)
 - Un ou plusieurs panneaux organisant sur la fenêtre un ou plusieurs « widgets » graphiques permettant de visualiser les variables de la sources de données et d'envoyer au système des commandes via le canal de communication

Vous pouvez ainsi créer de nouveaux tableaux de bord assez facilement si vous le souhaitez. Nous vous recommandons cependant de commencer d'abord par comprendre le fonctionnement d'une interface existante et de voir comment la modifier avant de faire des choses plus ambitieuses.

6 Mise en œuvre de l'ensemble

6.1 Précautions d'emploi

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

6.1.1 Connexions d'alimentation sur le shield « robotique »

Il est impératif de faire très attention aux connexions de l'alimentation du shield « robotique » car celui-ci n'est pas protégé contre les inversions de polarité. Une erreur de connexion sur les bornes d'alimentation risque d'entraîner la destruction de certains éléments. Le robot étant livré connecté et fonctionnel, il est préférable de ne pas modifier les branchements sur les connecteurs d'alimentation.

6.1.2 Recharge de la batterie

Le chargeur 8.4V, 1A fourni est destiné à recharger la batterie LiPo du robot en le 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. Lorsque la batterie est en charge, la LED rouge du chargeur est allumée. Lorsque la batterie est chargée, la LED verte du chargeur s'allume.

6.1.3 Précautions d'utilisation

Le robot 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 sur-dimensionnés par rapport à la taille du robot. Par ailleurs, celui-ci craint l'eau et l'humidité

Le robot 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.

6.2 Première utilisation

Le programme pré-chargé dans le robot à la livraison lui permet de tenir en équilibre sur ses deux roues et de recevoir des consignes de mouvement (transmises par liaison Wifi) provenant d'un ordinateur, d'un smartphone ou d'une tablette.

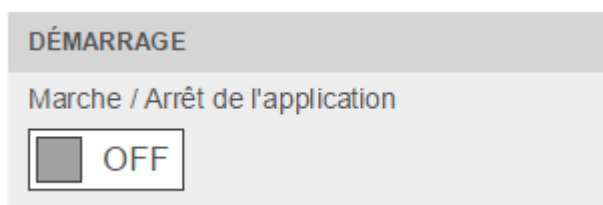
La procédure à suivre est simple:

- Poser le robot au sol
- Mettre le robot sous tension en basculant l'interrupteur sur « I »
- Attendre environ une minute

6.2.1 Pilotage par ordinateur

Pour piloter le robot, se connecter en Wifi sur le réseau de SSID « X-Bot-CHIP-xxx ». Le mot de passe Wifi est « xbotchip », puis :

- Lancer le logiciel MyViz
- Ouvrir le tableau de bord "X-Bot_3_Reseau.json" qui se trouve dans le répertoire "MyViz" de l'archive "X-Bot_CHIP_C_Python.zip" précédemment téléchargée (voir chapitre 1), en sélectionnant dans MyViz: Fichier → Ouvrir
- Démarrer l'interaction en cliquant sur le bouton "ON / OFF" en haut à gauche du tableau de bord:



- Vous pouvez alors interagir avec le système grâce au joypad de la zone « Pilotage » du tableau de bord:



Ce joypad permet de donner:

- une consigne de vitesse de rotation uniquement, lorsque le centre de la boule jaune se trouve dans la moitié supérieure du disque blanc
- une consigne de vitesse de rotation et une consigne de vitesse de translation lorsque le centre de la boule jaune se trouve dans la moitié supérieure de l'anneau gris
- une consigne de vitesse de translation (négative) uniquement, lorsque le centre de la boule jaune se trouve dans la moitié inférieure de l'anneau gris
- aucune consigne lorsque le centre de la boule jaune se trouve dans la moitié inférieure du disque blanc

Noter par ailleurs que ce contrôle est doté d'un « ressort de rappel virtuel » : quand vous « lâchez » la boule avec la souris, celle-ci revient automatiquement au centre et ne donne plus de consigne.

Enfin, si vous souhaitez arrêter le programme de contrôle de mouvement, positionnez le bouton sur OFF en haut à gauche du tableau de bord.

6.2.2 Pilotage par smartphone ou tablette

Pour pouvoir piloter X-Bot en Wifi grâce à votre smartphone ou votre tablette (nous utiliserons par la suite le nom générique « appareil »), les conditions suivantes doivent être respectées:

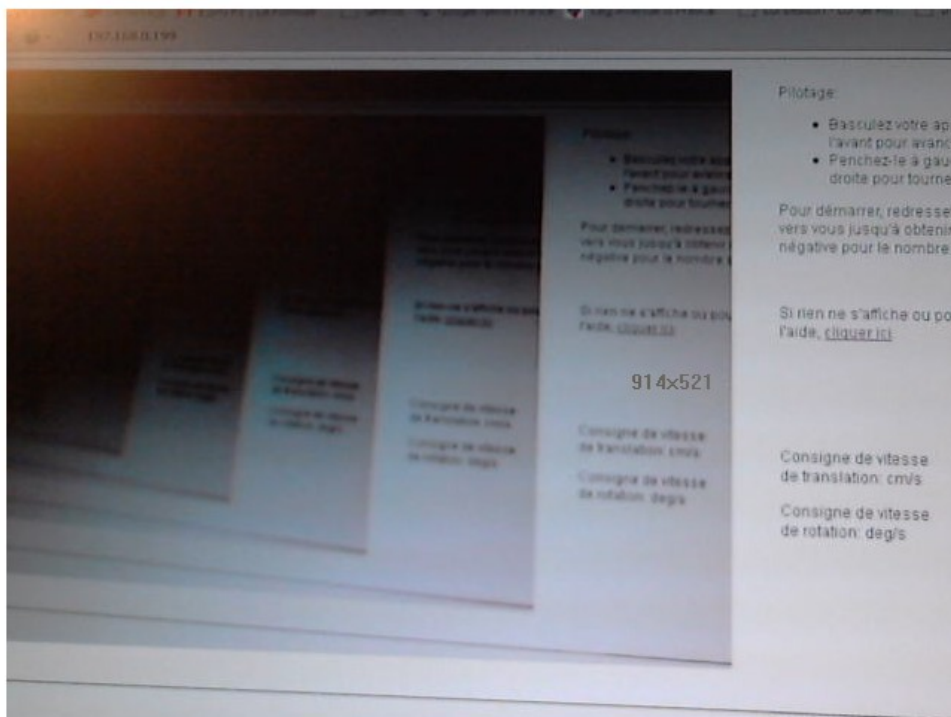
- Votre appareil doit posséder une interface Wifi
- Il doit posséder un accéléromètre
- Il doit être orienté en mode « paysage », comme ceci:



- Vous devez avoir rejoint le réseau Wifi « X-Bot-xxx » (voir plus haut)

Normalement, tout appareil récent respectant les conditions ci-dessus devrait fonctionner.

Ouvrez ensuite le navigateur Web de votre appareil à l'adresse 192.168.0.199. Vous obtenez tout d'abord la page suivante:



Pilotage:

- Basculez votre appareil vers l'avant pour avancer
- Penchez-le à gauche ou à droite pour tourner

Pour démarrer, redressez l'appareil vers vous jusqu'à obtenir une valeur négative pour le nombre suivant:

Si rien ne s'affiche ou pour avoir de l'aide, [cliquer ici](#).

Consigne de vitesse de translation: cm/s

Consigne de vitesse de rotation: deg/s

Cette page permet de tester si votre appareil est compatible. Si tout se passe bien, vous devez voir, à droite de la caméra, un nombre changer en fonction de l'inclinaison de votre appareil. Ce nombre correspond à l'accélération de la pesanteur mesurée par votre appareil, normalisée entre 0 et 1. Si vous ne voyez pas ce nombre, cela signifie:

- que votre appareil ne possède pas d'accéléromètre
- ou bien qu'il n'est pas compatible avec cette application Web de pilotage (iPhone, iPad ou iPod avec iOS < 4.2, par exemple)
- ou bien que le navigateur utilisé n'est pas compatible

Dans ce cas, merci de nous contacter à l'adresse support@3sigma.fr en nous indiquant les caractéristiques précises du smartphone ou de la tablette que vous utilisez. Une solution personnalisée est peut-être possible.

Si vous voyez l'accélération évoluer en fonction de l'inclinaison de votre appareil, tout va bien. Avant d'accéder au pilotage à proprement parler, vous devez activer l'application. C'est très simple, il suffit de redresser votre appareil jusqu'à obtenir une accélération nulle. Une fois ceci fait, l'interface change légèrement : l'accélération n'est plus visible, seules restent les consignes de vitesse de translation et de rotation (en plus de la transmission vidéo temps-réel de la Webcam, à gauche de l'écran). Piloter X-Bot est alors très facile:

- Lorsque vous basculez votre appareil vers l'avant, vous modifiez la consigne de vitesse de translation (entre 0 et 50 cm/s)
- Lorsque vous inclinez votre appareil sur la gauche ou sur la droite, vous modifiez la consigne de vitesse de rotation (entre -180 et 180 deg/s), comme avec un volant. Si l'inclinaison est trop forte à gauche ou à droite, l'application détecte que votre appareil est en mode portrait et annule les consignes

Remarque : Dans ce mode de pilotage par smartphone ou tablette, il est impossible de faire reculer le robot car il est impossible de lui donner une consigne de vitesse négative.

7 Activités réalisables avec le système en mode CHIP / Python

Dans ce mode, les différentes applications sont programmées en Python et sont exécutées par la carte CHIP. Les programmes sont pré-chargés sur la carte, celle-ci n'a donc pas besoin d'être « programmée » pour chaque nouvelle application.

La carte Iteaduino (compatible Arduino) Uno est également sollicitée pour le comptage des impulsions générées par les codeurs incrémentaux. Pour que l'ensemble fonctionne, il faut téléverser un programme spécifique sur cette carte Iteaduino : **Firmware_1.ino**

Ce programme est pré-chargé sur la carte Iteaduino à la livraison du robot. Si vous commencez l'exploitation de ce dernier par des programmes Python, vous n'avez pas besoin de le téléverser de nouveau.

Les différentes applications du système en mode CHIP / Python sont présentées ci-dessous. Nous indiquons pour chacune le programme Python utilisé sur la carte CHIP : n'hésitez pas à le modifier en fonction de vos besoins.

7.1 Contrôle de vitesse et pilotage en Wifi

7.1.1 Présentation

Cette expérience correspond au programme exécuté par le robot à sa mise sous tension dans sa configuration initiale (en sortie du carton de livraison). Elle permet de piloter en Wifi les mouvements du robot et d'avoir une télémétrie des données mesurées.

Le programme Python et le tableau de bord MyViz sont :

- Déjà présent sur la carte CHIP
/root/programmes_python/X-Bot2.py
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_Reseau.json

Voici une capture d'écran du tableau de bord MyViz :



Cette interface possède les fonctionnalités suivantes :

- Visualisation de la vitesse longitudinale (consigne et mesure) mesurée à partir de la vitesse de rotation des roues (odométrie)
- Visualisation de la vitesse de rotation (consigne et mesure) mesurée à partir de la vitesse de rotation des roues (odométrie)
- Pilotage du robot via un joystick virtuel (voir paragraphe 6.2 pour l'utilisation de ce dernier)
- Modification des gains du PID d'asservissement de mouvement longitudinal pendant le fonctionnement
- Modification des gains du PID d'asservissement en rotation pendant le fonctionnement
- Visualisation de la distance mesurée par le capteur ultrasons
- Visualisation de la tension de la batterie
- Visualisation textuelle des données mesurées avec possibilité de sauvegarde sur l'ordinateur hôte

Attention : les commandes de pilotages ne sont pas actives pendant les 5 premières secondes.

7.1.2 Expérimentations possibles

Au-delà de son aspect ludique, cette expérience permet de visualiser les caractéristiques des asservissements de mouvement du robot. Elle permet également d'appréhender l'influence de la charge sur un moteur. En effet, dans les expérimentations avec asservissement de vitesse et roues libres (robot non posé sur le sol), les moteurs doivent juste mettre les roues (de faible inertie) en mouvement. Mais si le robot est posé au sol, les moteurs doivent mettre en mouvement l'inertie totale du robot et, de plus, compenser la résistance au roulement.

Il est intéressant de constater la différence de temps de réponse sur la rotation des moteurs si on donne une consigne de vitesse avec le robot « en l'air » ou posé au sol. On peut ainsi régler les gains dans les deux cas pour que les deux temps de réponses soient similaires.

7.2 Suivi de ligne

3 capteurs optiques sont situés sous le robot et permettent de réaliser des expériences de suivi d'une ligne noire sur fond blanc.

Le programme Python et le tableau de bord MyViz sont :

- Déjà présent sur la carte CHIP
/root/programmes_python/X-Bot_SuiviLigne2.py
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_SuiviLigne_Reseau.json

On applique aux moteurs du robot une tension de commande dépendant du nombre de capteurs détectant la ligne :

- Si tous les capteurs détectent la ligne, la tension de commande est identique sur les deux moteurs
- Si 2 capteurs sur 3 détectent la ligne, la tension de commande est différente sur les deux moteurs, pour que le robot tourne doucement afin de ramener le 3^{ème} capteur sur la ligne
- Si un seul capteur détecte la ligne, la différence de tension entre les deux moteurs est plus forte
- Si aucun capteur ne détecte la ligne, la commande des moteurs est identique à la dernière valeur utilisée

Le tableau de bord (voir ci-après) permet par ailleurs de modifier le seuil de détection de la ligne pour s'adapter à des conditions de luminosité changeantes.

Voici une capture d'écran du tableau de bord MyViz :



Cette interface possède les fonctionnalités suivantes :

- Visualisation de l'état de chaque capteur (sur ligne ou hors ligne)
- Visualisation de la valeur analogique de chaque capteur superposée au seuil de détection
- Visualisation de la vitesse de rotation de chaque moteur
- Modification du seuil de détection pendant le fonctionnement
- Modification de la valeur absolue de tension de commande des moteurs
- Visualisation textuelle des données mesurées avec possibilité de sauvegarde sur l'ordinateur hôte

Attention : les commandes de pilotages ne sont pas actives pendant les 5 premières secondes.

7.2.1 Expérimentations possibles

Cette expérience est intéressante pour définir des logiques de type « si – alors – sinon » en Python.

Elle permet par ailleurs de bien visualiser l'aspect « continu » des informations renvoyées par un capteur analogique et de montrer comment on peut transformer cette information en un signal booléen par l'intermédiaire d'une comparaison avec un seuil.

Enfin, il est intéressant de tester différentes valeurs de tension de commande des moteurs avec différents type de circuits et notamment différents rayons de courbure des virages.

7.3 Évitement d'obstacles

Dans cette expérience, X-Bot est programmé pour éviter automatiquement les obstacles détectés par le capteur ultrasons situé à l'avant du robot.

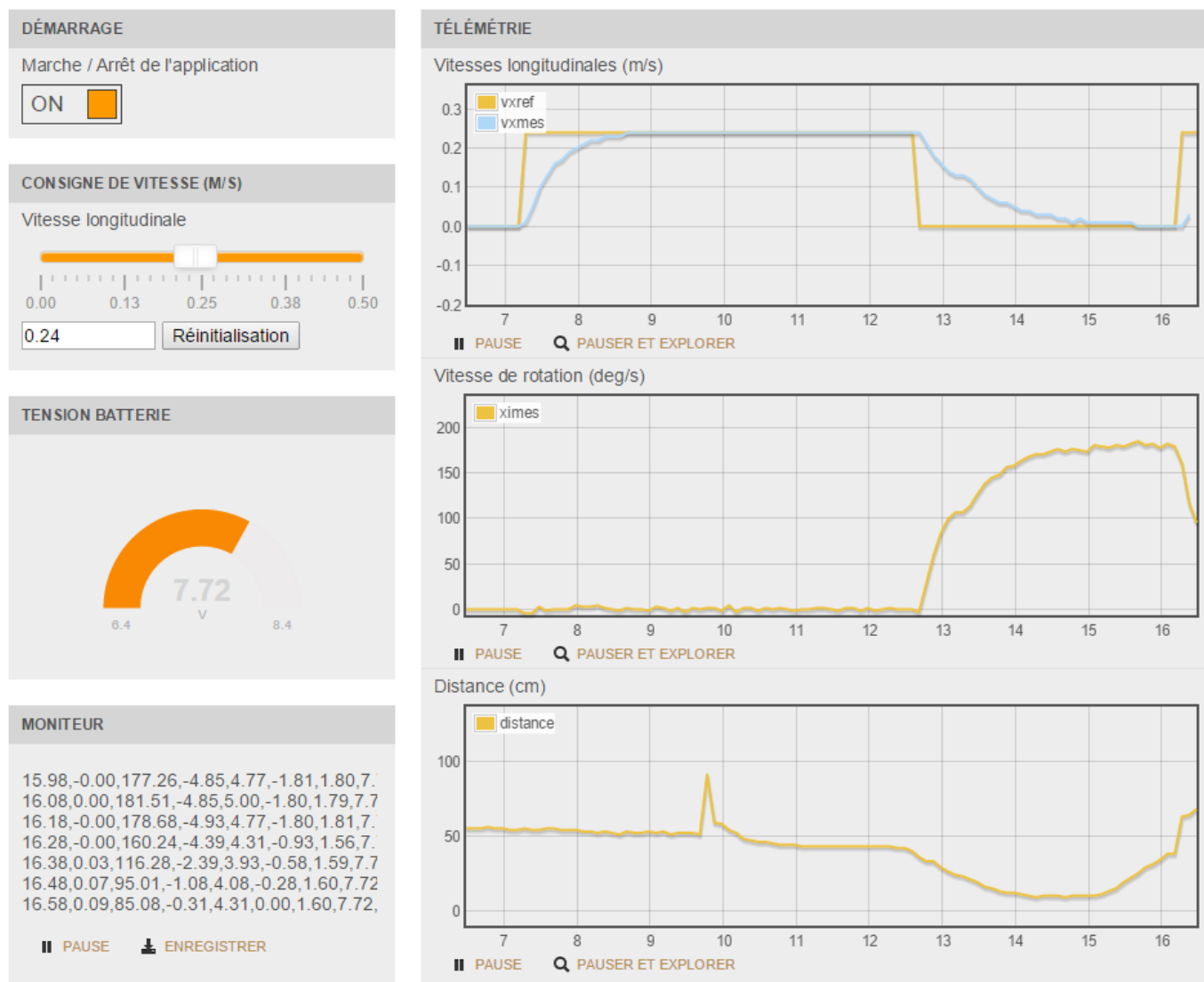
Le programme Python et le tableau de bord MyViz sont :

- Sur la carte CHIP
/root/programmes_python/X-Bot_Obstacle2.py
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_Obstacle_Reseau.json

Le programme fonctionne de la façon suivante :

- Si la distance mesurée est inférieure à 40 cm, le robot tourne à droite à la vitesse de 180 deg/s
- Sinon, le robot va tout droit à la vitesse de consigne donnée dans le tableau de bord

Voici une capture d'écran du tableau de bord MyViz :



Cette interface possède les fonctionnalités suivantes :

- Visualisation de la vitesse longitudinale (consigne et mesure) mesurée à partir de la vitesse de rotation des roues (odométrie)
- Visualisation de la vitesse de rotation (consigne et mesure) mesurée à partir de la vitesse de rotation des roues (odométrie)
- Visualisation de la distance mesurée par le capteur ultrasons
- Pilotage de la consigne de vitesse longitudinale
- Visualisation de la tension de la batterie
- Visualisation textuelle des données mesurées avec possibilité de sauvegarde sur l'ordinateur hôte

Attention : les commandes de pilotages ne sont pas actives pendant les 5 premières secondes.

7.3.1 Expérimentations possibles

Cette expérience permet d'utiliser le robot en mode autonome, sans qu'il soit nécessaire de le piloter manuellement. Le scénario d'évitement d'obstacle écrit en Python peut être enrichi pour, par exemple, tourner alternativement à droite et à gauche (au lieu de toujours à droite) lorsqu'un obstacle se présente.

7.4 Commande des moteurs en tension

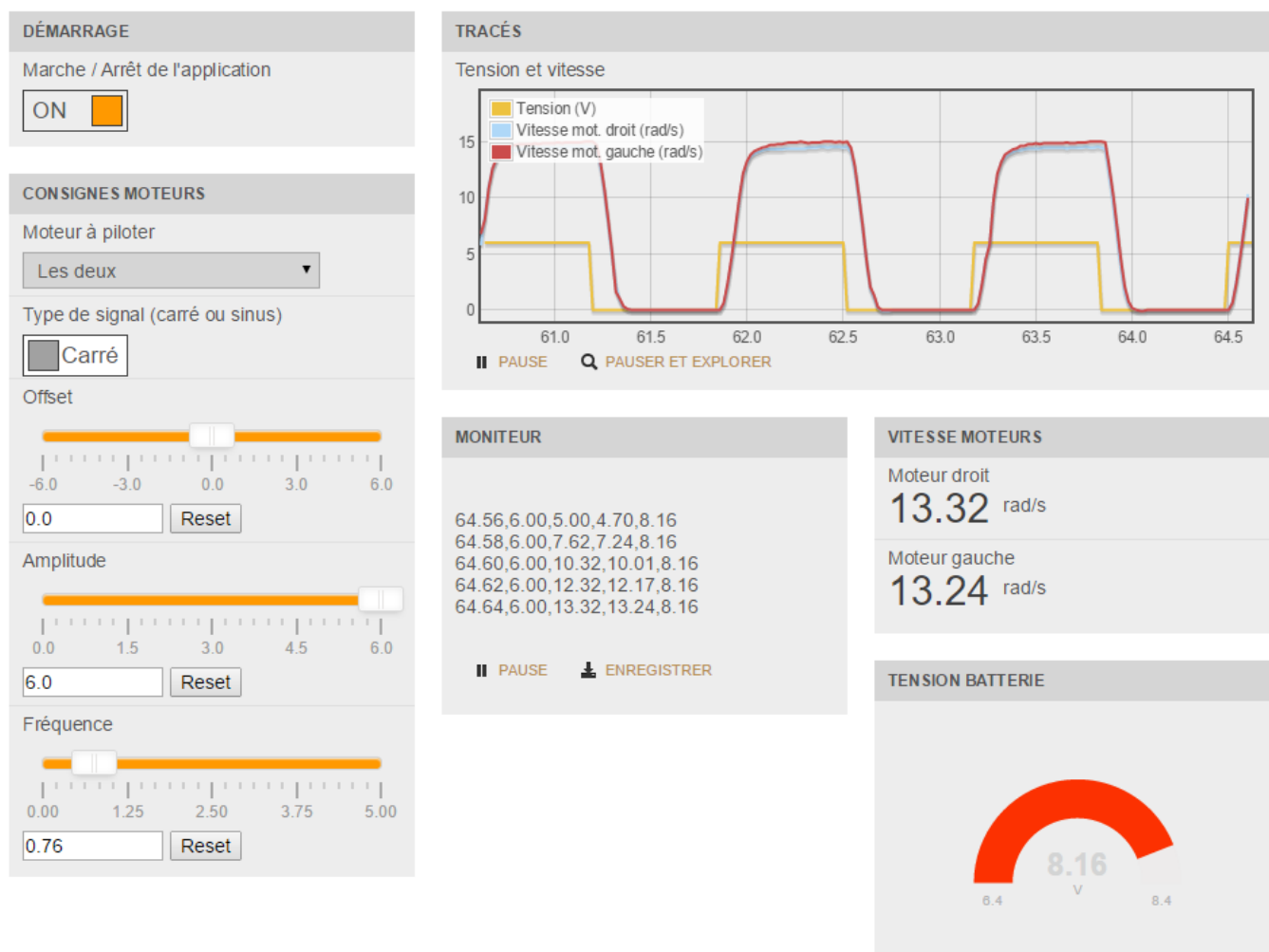
7.4.1 Présentation

Cette expérience permet de donner une consigne de tension aux moteurs du robot et d'avoir une télémétrie des données mesurées.

Le programme Python et le tableau de bord MyViz sont :

- Déjà présent sur la carte CHIP
/root/programmes_python/CommandeEnTension2.py
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_CommandeMoteursEnTension_Reseau.json

Voici une capture d'écran du tableau de bord MyViz :



Cette interface possède les fonctionnalités suivantes :

- Visualisation de la tension de commande et des vitesses mesurées
- Génération d'une consigne de tension sous la forme d'un signal carré ou sinusoïdal d'amplitude et de fréquence variable, avec ou sans offset (remarque : la somme offset + amplitude est saturée en interne à + / - 6 V)
- Visualisation de la tension de la batterie
- Visualisation textuelle des données mesurées avec possibilité de sauvegarde sur l'ordinateur hôte

Attention : les commandes de pilotages ne sont pas actives pendant les 5 premières secondes.

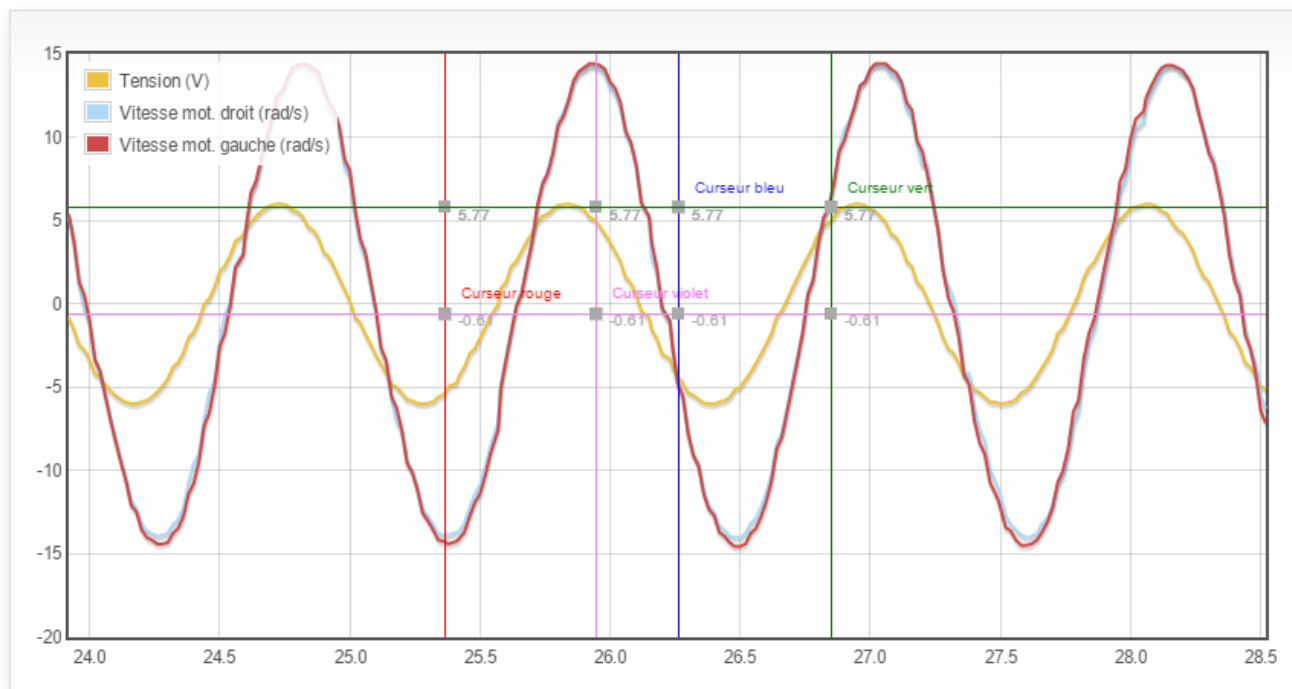
7.4.2 Expérimentations possibles

Cette expérience permet d'estimer les caractéristiques des moteurs en boucle ouverte :

- Le temps de réponse peut être mesuré grâce à un essai de réponse à un échelon :
 - en changeant avec la souris l'offset de commande
 - ou en générant un signal carré
- Il est possible de construire un diagramme de Bode point par point de la façon suivante :
 - définir une amplitude de consigne de tension sinusoïdale constante pour tous les points de mesure
 - définir la fréquence du point de mesure
 - cliquer sur « Cliquer et explorer » sous le graphique pour ouvrir un agrandissement de ce dernier
 - mesurer le gain et le déphasage entre la commande et la mesure
 - (à l'extérieur de MyViz) stocker les différents points dans un tableau. Quand celui-ci est rempli, le diagramme de Bode peut être tracé

La capture d'écran ci-dessous montre un exemple d'exploitation de la fenêtre d'exploration d'un graphique :

Tension et vitesse



Utiliser le bouton gauche de la souris pour zoomer. Double-cliquer pour dézoomer.

Utiliser le bouton droit de la souris pour déplacer les curseurs.

- **Curseur rouge**
 - x:25.3663 y: -0.6100 (Tension (V))
 - x:25.3663 y: 5.7700 (Vitesse mot. droit (rad/s))
 - x:25.3663 y: 5.7700 (Vitesse mot. gauche (rad/s))
- **Curseur violet**
 - x:25.9472 y: -0.6100 (Tension (V))
 - x:25.9472 y: 5.7700 (Vitesse mot. droit (rad/s))
- **Curseur bleu**
 - x:26.2640 y: -0.6100 (Tension (V))
 - x:26.2640 y: 5.7700 (Vitesse mot. droit (rad/s))
 - x:26.2640 y: 5.7700 (Vitesse mot. gauche (rad/s))
- **Curseur vert**
 - x:26.8537 y: -0.6100 (Tension (V))
 - x:26.8537 y: 5.7700 (Vitesse mot. droit (rad/s))

Remarque : le bas de la capture d'écran est tronqué

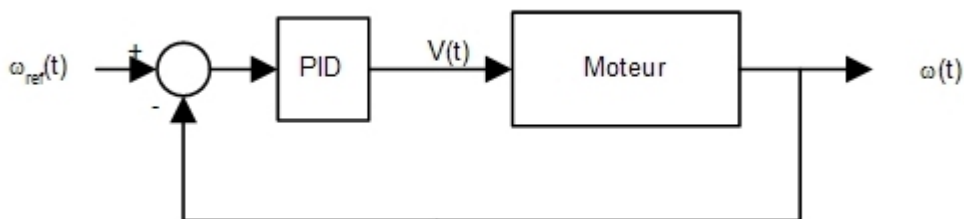
7.5 Asservissement des moteurs en vitesse

Cette expérience permet de piloter les moteurs du robot en boucle fermée et d'avoir une télémétrie des données mesurées. L'asservissement de vitesse est par ailleurs réglable pendant le fonctionnement du système.

Le programme Python et le tableau de bord MyViz sont :

- Déjà présent sur la carte CHIP
/root/programmes_python/AsservissementVitesse2.py
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_AsservissementMoteursEnVitesse_Reseau.json

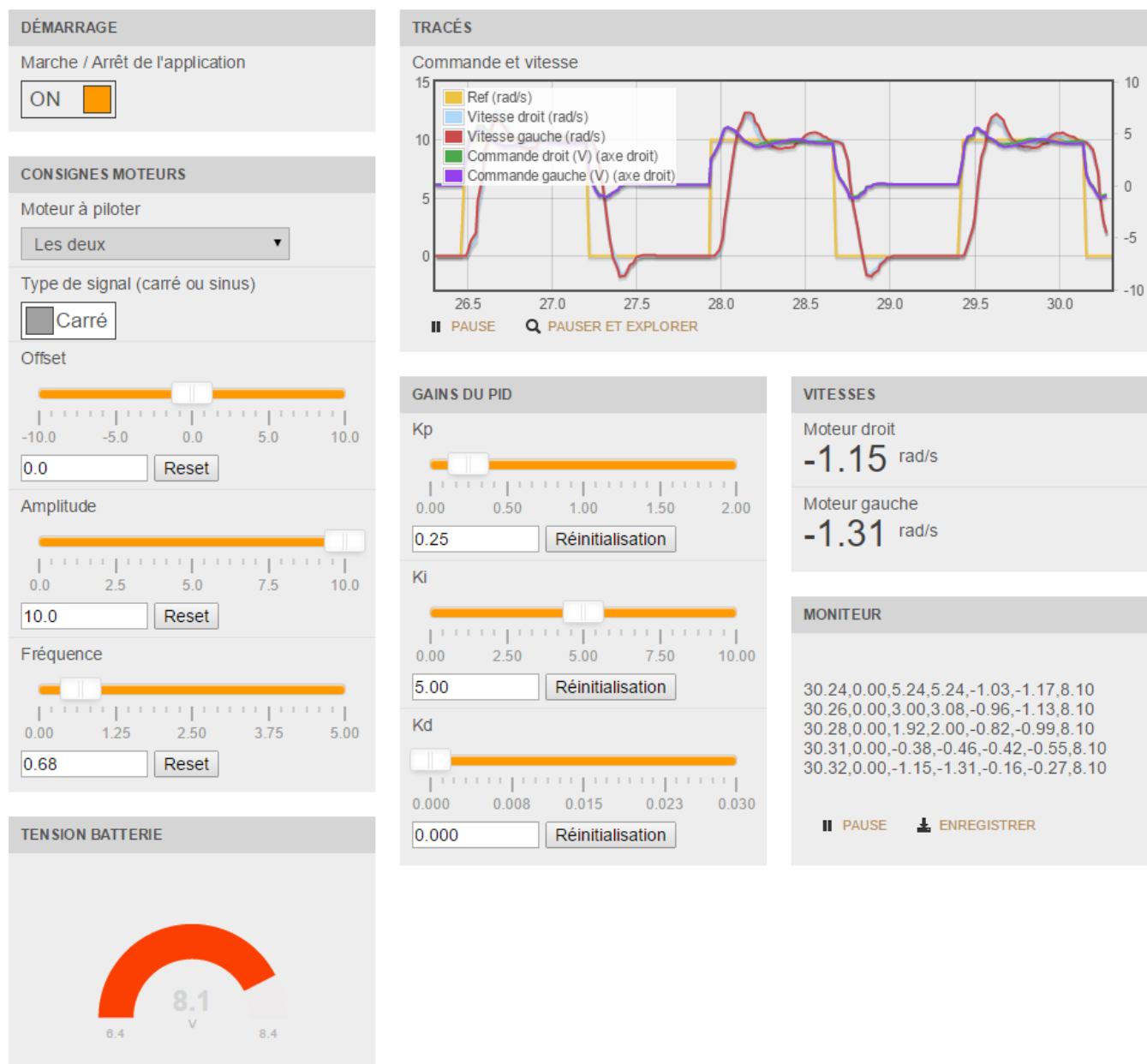
Le schéma de l'asservissement est le suivant :



La fonction de transfert du PID est la suivante :

$$K_p + \frac{K_i}{p} + \frac{K_d p}{T_f p + 1} \quad (7.1)$$

Voici une capture d'écran du tableau de bord MyViz :



Cette interface possède les fonctionnalités suivantes :

- Visualisation de la consigne de vitesse, des tensions de commande issues de l'asservissement et des vitesses mesurées
- Génération d'une consigne de vitesse sous la forme d'un signal carré ou sinusoïdal d'amplitude et de fréquence variable, avec ou sans offset
- Modification des gains du PID pendant le fonctionnement
- Visualisation de la tension de la batterie
- Visualisation textuelle des données mesurées avec possibilité de sauvegarde sur l'ordinateur hôte

Attention : les commandes de pilotages ne sont pas actives pendant les 5 premières secondes.

7.5.1 Expérimentations possibles

Cette expérience permet de comprendre l'intérêt et le fonctionnement d'un asservissement et de tester l'effet des 3 gains d'un régulateur PID :

- Si une des deux roues est freinée, son asservissement va tout faire pour suivre malgré tout la consigne de vitesse
- On constate alors que la tension de commande de la roue freinée est plus grande que celle de la roue « libre » : en effet, le travail de l'asservissement est de calculer la tension de commande permettant de suivre la consigne de vitesse
- La modification des gains du PID permet de constater leurs effets respectifs

8 Activités réalisables avec le système en mode Arduino / C

Dans ce mode, les différentes applications sont programmées en C (Arduino) et sont exécutées par la carte Iteduino Uno. Il est nécessaire de reprogrammer cette dernière pour chaque application.

Les différentes applications du système dans ce mode sont très similaires à celles présentées dans le mode CHIP / Python. Le choix d'utiliser un mode plutôt que l'autre est essentiellement une question de préférence de langage de programmation ou d'utilisation de plate-forme (micro-contrôleur ou mini-ordinateur).

Par conséquent, nous indiquons par la suite uniquement les différences par rapport aux applications présentées au chapitre précédent et en particulier le programme Arduino à charger sur la carte Iteduino pour chacune d'elle. N'hésitez pas à le modifier en fonction de vos besoins.

Important : la philosophie de MyViz implique que l'identifiant du port série utilisé soit stocké dans chaque tableau de bord et non pas dans le logiciel lui-même. C'est la raison pour laquelle il est nécessaire de modifier les tableaux de bord livrés et d'enregistrer la modification réalisée sur le numéro du port utilisé.

Cette modification doit être réalisée dans chaque tableau de bord. L'enregistrement permet de le faire une bonne fois pour toute et non pas à chaque nouvelle utilisation.

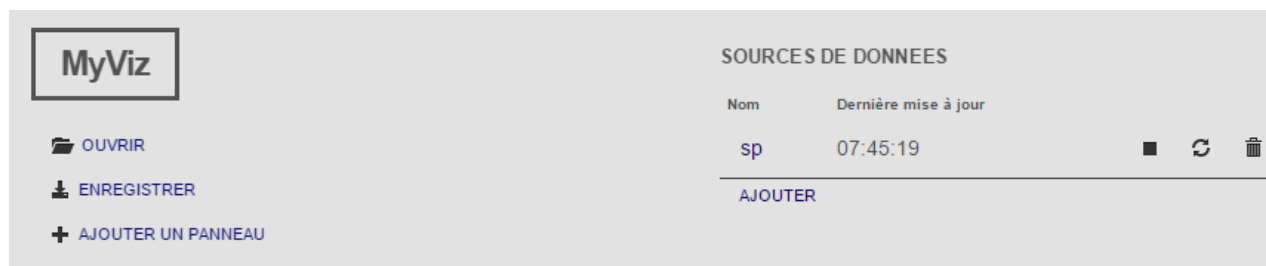
La mise en œuvre de ce mode et le changement du numéro du port se fait en suivant ces différentes étapes :

- Allumer le robot
- Brancher le câble mini USB fourni sur le connecteur mini USB de la carte Iteduino
- Lancer le logiciel MyViz
- Charger le tableau de bord de l'application que vous souhaitez réaliser en sélectionnant dans MyViz: Fichier → Ouvrir


- Cliquer sur la clé en haut de la fenêtre:




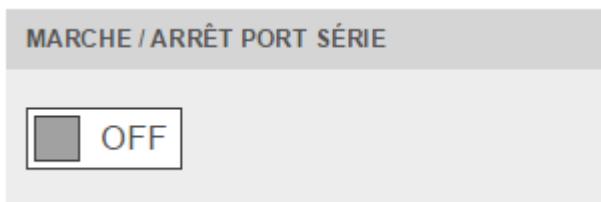
- Cela permet d'ouvrir le panneau supérieur du mode "édition" afin de sélectionner le port série de votre ordinateur sur lequel est branché le robot



- Pour ce faire, cliquer sur la source de données nommée "sp" afin d'ouvrir le panneau de configuration du port série:



- Sélectionner l'identifiant du port correspondant à la carte Iteaduino Uno (comme ci-dessus) et cliquer sur le bouton "Enregistrer" en bas de la fenêtre
 - Quitter le mode édition en cliquant sur la flèche située sous le panneau supérieur:
- 
- Sauvegarder le tableau de bord modifié en sélectionnant Fichier → Enregistrer
 - Démarrer l'interaction en cliquant sur le bouton "ON / OFF" en haut à gauche du tableau de bord:



- Les acquisitions doivent démarrer très rapidement. Vous pouvez alors interagir avec le système grâce aux contrôles prévus à cet effet.

8.1 Trajectoire pré-programmée

8.1.1 Présentation

Contrairement au mode CHIP / Python (voir 7.1), on ne fait pas ici de pilotage à distance et on n'utilise pas de tableau de bord car le robot n'est pas libre de ses mouvements s'il est connecté à l'ordinateur. L'objectif dans cette version « Arduino » est de programmer cette carte micro-contrôleur avec des manœuvres pré-déterminées.

Le programme Arduino est :

XBotDefinitionMouvement_1.ino

La programmation de manœuvres se fait en modifiant la partie « Consignes » du programme. Par exemple, le code suivant permet de réaliser une suite de virages :

```
if (temps > Tmax) {
    Tmax = temps + 6.; // Le 8 dure 6 s
}
else if (temps < (Tmax-5.5)) { // La ligne droite dure 0.5 s
    vxref = 0.3;
    xiref = 0.;
}
else if (temps < (Tmax-3.)) { // Le virage dure 2.5 s
    vxref = 0.3;
    xiref = 2.;
}
else if (temps < (Tmax-2.5)) { // La ligne droite dure 0.5 s
    vxref = 0.3;
    xiref = 0.;
}
else { // // Le virage dure 2.5 s
    vxref = 0.3;
    xiref = -2.;
}
```

Important : dans la partie « Setup » du programme, il est nécessaire d'attendre au moins 30 secondes pour que le CHIP ait démarré et ne perturbe pas l'Arduino.

8.2 Commande des moteurs en tension

Cette expérience (analogue à celle du paragraphe 7.4) permet de donner une consigne de tension aux moteurs du robot et d'avoir une télémétrie (via liaison série) des données mesurées.

Le programme Arduino et le tableau de bord MyViz sont :

- A téléverser sur la carte Iteaduino :
CommandeMoteursEnTension_1.ino
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_CommandeMoteursEnTension_Serie.json

8.3 Asservissement des moteurs en vitesse

Cette expérience (analogue à celle du paragraphe 7.5) permet de piloter les moteurs du robot en boucle fermée et d'avoir une télémétrie (via liaison série) des données mesurées. L'asservissement de vitesse est par ailleurs réglable pendant le fonctionnement du système.

Le programme Arduino et le tableau de bord MyViz sont :

- A téléverser sur la carte Iteaduino :
AsservissementMoteursEnVitesse_1.ino
- A charger dans MyViz sur votre ordinateur :
X-Bot_3_AsservissementMoteursEnVitesse_Serie.json

9 Support technique

Pour toute question ou problème, veuillez nous contacter à l'adresse support@3sigma.fr.

Par ailleurs, ce système est un produit « vivant ». N'hésitez pas à vérifier régulièrement la présence de nouvelles versions de la documentation ou des programmes et tableaux de bord.