

POPI

Prototype RICA IV

L'équipe ayant participé à ce projet est composée de :

Aharram Gutton Anaïs

Brottet Karla

Combacal Rémi

Graton Clémence

Labarussiat Lucas

Oddon Yannis

Pelloux-Prayer Jean

Peres Olivier

Rougé Guillaume

Thomaso Clément

Tichit Pascal



SOMMAIRE

| | | |
|-------|--|----|
| 1 | PREAMBULE..... | 4 |
| 2 | DESCRIPTION GENERALE..... | 4 |
| 2.1 | DESCRIPTION PHYSIQUE DU ROBOT..... | 4 |
| 2.2 | DESCRIPTION DES ACCESSOIRES..... | 5 |
| 2.2.1 | SOCLE..... | 5 |
| 2.2.2 | POIGNEES..... | 5 |
| 2.2.3 | PIEDS..... | 5 |
| 2.3 | DESCRIPTION DU FONCTIONNEMENT DU ROBOT..... | 5 |
| 2.4 | CARACTERISTIQUES TECHNIQUES DU ROBOT POPI..... | 6 |
| 3 | PARTIE MECANIQUE..... | 6 |
| 3.1 | DESCRIPTIF DE L'EQUIPEMENT..... | 6 |
| 3.1.1 | ACTIONNEUR EPAULE..... | 7 |
| 3.1.2 | ACTIONNEUR AILE..... | 7 |
| 3.1.3 | ACTIONNEUR COUDE..... | 8 |
| 3.2 | PRINCIPE DE MONTAGE..... | 8 |
| 4 | PARTIE ELECTRONIQUE..... | 8 |
| 4.1 | INTELLIGENCE EMBARQUEE..... | 8 |
| 4.2 | CAPTEURS..... | 9 |
| 4.3 | ALIMENTATION..... | 9 |
| 4.3.1 | PLATINE D'ALIMENTATION..... | 9 |
| 4.3.2 | DISTRIBUTION DE L'ALIMENTATION..... | 10 |
| 4.4 | GESTION DU CABLAGE..... | 11 |
| 4.4.1 | CODE COULEUR..... | 11 |
| 4.4.2 | AFFECTATION DES BORNES..... | 12 |
| 4.5 | MONTAGE DE L'ELECTRONIQUE..... | 13 |
| 5 | RESEAU..... | 13 |
| 6 | ARCHITECTURE LOGICIELLE..... | 13 |
| 6.1 | PRINCIPE DE FONCTIONNEMENT..... | 13 |
| 6.2 | GENERATION DE TRAJECTOIRE..... | 14 |
| 6.3 | SIMULATION..... | 14 |
| 6.4 | MONITORING..... | 14 |
| 6.5 | ENVOI DE TRAJECTOIRE SUR POPI..... | 15 |
| 6.6 | ARBORESCENCE DES DOSSIERS..... | 16 |
| 7 | NOTICE D'INSTRUCTION..... | 17 |
| 7.1 | INSTALLATION..... | 17 |
| 7.2 | RACCORDEMENT / ALIMENTATION..... | 18 |
| 7.3 | GENERATION DE TRAJECTOIRES..... | 19 |
| 7.3.1 | UTILISATION DE TOWR..... | 19 |
| 7.3.2 | ENREGISTREMENT DES COMMANDES EN ROSBAG..... | 20 |
| 7.3.3 | MODIFIER LES COMMANDES SUR UN TABLEUR..... | 22 |
| 7.4 | UTILISATION DE LA MANETTE..... | 23 |
| 7.5 | SIMULATION..... | 24 |
| 7.6 | MONITORING..... | 24 |
| 7.7 | ENVOI DE TRAJECTOIRE SUR POPI..... | 26 |
| 7.7.1 | MODE OPERATOIRE DE MANUTENTION..... | 26 |
| 7.7.2 | PREMIERE CONNEXION..... | 26 |
| 7.7.3 | UTILISATION DES TRAJECTOIRES..... | 29 |
| 7.7.4 | TRAJECTOIRES MISES A DISPOSITION..... | 30 |
| 7.7.5 | MISE EN POSITION REPOS..... | 31 |
| 8 | SECURITE..... | 31 |

| | | |
|--|--|----|
| 8.1 | PRESRIPTION GENERALE DE SECURITE | 31 |
| 8.2 | TRANSPORT | 32 |
| 8.3 | INSTALLATION / DESINSTALLATION | 32 |
| 8.4 | MAINTENANCE..... | 32 |
| 8.4.1 | INSTRUCTIONS DE MAINTENANCE PREVENTIVE..... | 34 |
| 8.4.2 | INSTRUCTIONS DE MAINTENANCE CORRECTIVE | 34 |
| ANNEXE 1 : PLANS DE MONTAGE ELECTRONIQUE | | 37 |
| ANNEXE 2 : NOTICE DE MONTAGE MECANIQUE..... | | 41 |

TABLE DES FIGURES

| | |
|--|----|
| Figure 1 : Dimensions du robot POPI | 4 |
| Figure 2 : Les différents types d'actionneurs | 5 |
| Figure 3 : Axe de symétrie POPI..... | 6 |
| Figure 4 : Electronique de la partie commande | 8 |
| Figure 5 : Schéma de la platine d'alimentation | 9 |
| Figure 6 : Distribution de l'alimentation | 10 |
| Figure 7 : Architecture réseau du robot..... | 13 |
| Figure 8 : Schéma de fonctionnement logiciel de POPI | 15 |
| Figure 9 : Arborescence des dossiers | 16 |
| Figure 10 : Branchement de l'ombilic..... | 18 |
| Figure 11 : Interface utilisateur Towr | 19 |
| Figure 12 : Interface Rviz | 20 |
| Figure 13 : Sélection des topics sur rqt_bag | 21 |
| Figure 14 : Fenêtre rqt_bag pendant l'enregistrement | 22 |
| Figure 15 : Configuration de la manette..... | 23 |
| Figure 16 : Interface rqt_gui – message publisher (1), topic monitor (2) et dynamic reconfigure (3) .. | 24 |
| Figure 17 : Interface rqt_plot..... | 25 |
| Figure 18 : Utilisation des poignées de manutention..... | 26 |
| Figure 19 : Page de paramétrage réseau | 26 |
| Figure 20 : Page de paramétrage de la connexion filaire | 27 |
| Figure 21 : Onglet détail de la page de paramétrages de la connexion filaire | 27 |
| Figure 22 : POPI en position repos | 29 |
| Figure 23 : POPI en position nominale | 30 |
| Figure 24 : Positionnement du socle | 31 |
| Figure 25 : Ajout d'une connexion par pont..... | 35 |
| Figure 26 : Sélection de l'interface à asservir..... | 36 |
| Figure 27 : Vérification de paramètres réseaux | 36 |

TABLE DES TABLEAUX

| | |
|---|----|
| Tableau 1 : Caractéristiques du robot POPI..... | 6 |
| Tableau 2 : Caractéristiques de l'ensemble épaule | 7 |
| Tableau 3 : Caractéristiques de l'ensemble aile | 7 |
| Tableau 4 : Caractéristiques de l'ensemble coude | 8 |
| Tableau 5 : Code couleur des fils électriques du robot..... | 11 |
| Tableau 6 : Affectation des bornes BeagleBoneBlack Gauche | 12 |
| Tableau 7 : Affectation des bornes BeagleBoneBlack Droite..... | 12 |
| Tableau 8 : Nomenclature des composants électronique | 38 |

1 Préambule

Le produit POPI est proposé par les MKX3 - Groupe 2 en réponse au cahier des charges rédigé par MM. P. GIRONES & V. CHAPURLAT, daté du 19/01/2016, réf : DPAD/CP2NAPM/CC/2016-01b.

Le produit est un robot quadrupède dont les caractéristiques techniques ont été choisies dans le cadre de la montée en gamme du RICA III. Le produit POPI a été conçu pour franchir un palier et une fosse de 30cm, ainsi qu'une configuration escalier (Giron 28cm).

Dans l'état, POPI est un prototype de niveau TRL 6 (*Démonstration* du modèle système / sous-système ou du prototype dans un environnement significatif) du sous-système base mobile du robot RICA.

Le robot RICA a pour cible des missions de collecte de données radiologiques, dimensionnelles et visuelles dans des milieux sévères. L'interface opérateur du prototype a été développée afin de rendre POPI pilotable par un technicien de niveau BAC+2. Dans ce dossier constructeur, vous retrouverez toutes les informations nécessaires à la mise en fonctionnement, à l'utilisation, au transport et à la maintenance du produit POPI.

Note : l'ensemble de nos travaux, comprenant ce dossier constructeur, les plans électroniques et mécaniques ainsi que les codes sources, sont disponibles en open-source (sous licence BSD-3) sur le [GitHub](#) du projet.

2 Description générale

2.1 Description physique du robot

Le produit POPI est un robot quadrupède. L'alimentation du robot se fait via un ombilic, relié à un tableau d'alimentation qui est raccordé au secteur.

Le poids de l'ensemble est de 70 kg,

Le robot POPI, présenté en Figure 1, en position nominale, possède les dimensions générales suivantes : 1017mm x 662mm x 620mm (PxLxH).

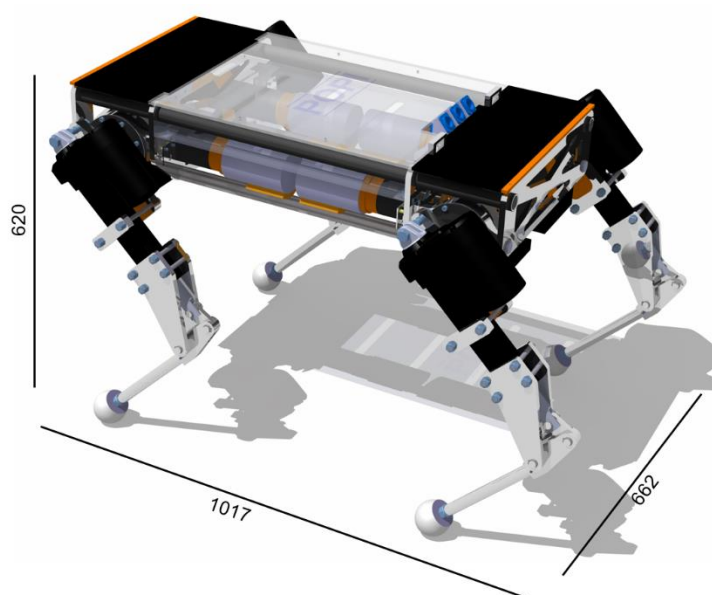


FIGURE 1 : DIMENSIONS DU ROBOT

2.2 Description des accessoires

2.2.1 Socle



Un support à roulettes a été conçu afin de faciliter le déplacement du robot POPI sur de courtes distances et sur un même niveau.

C'est aussi son support de stockage. Lorsque le robot est éteint, il doit être positionné sur ce socle.

2.2.2 Poignées



Lorsque le robot doit être déplacé alors qu'il n'est pas sur son socle, des poignées sont à disposition.

Voir 7.7.1 Mode opératoire de manutention

2.2.3 Pieds



Plusieurs types de pieds sont disponibles, en fonction de l'état du sol. Les pieds du premier sont souples, pour permettre une surface de contact maximale. Les pieds du second lot sont rigides, il est adapté à un sol abrasif. Ils sont très facilement interchangeables.

2.3 Description du fonctionnement du robot

L'ensemble des choix de conception ont été réalisés dans le but de satisfaire aux exigences de franchissement définies dans le cahier des charges. Pour rappel, ces exigences sont le franchissement d'un palier de 30cm, d'une fosse de 30 cm ainsi qu'un escalier avec un giron de 28cm.

Le produit POPI est composé de 12 actionneurs, 3 par patte (voir partie 3.1 Descriptif de l'équipement.).

L'association des actionneurs « épaule » et « coude » permet la marche linéaire du robot. L'action de « l'aile » permet au robot de tourner.

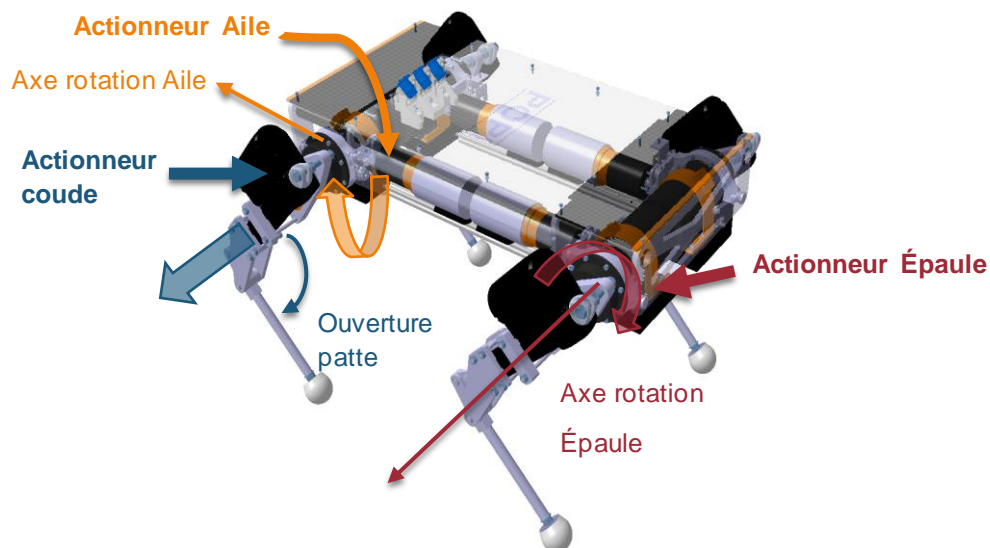


FIGURE 2 : LES DIFFERENTS TYPES D'ACTIONNEURS

2.4 Caractéristiques techniques du robot POPI

| | | |
|---|--------------------------------------|----------------------------|
| Vitesse | Marche statique : 0,2 m/s | Marche dynamique : 0,5 m/s |
| Poids | 70 kg | |
| Charge utile transportable | 20kg | |
| Encombrement (PxLxH) | 1017mm x 662mm x 620mm | |
| Raccordement électrique | 1 x 220V / 50 Hz / 16A | |
| Consommation électrique max | 2000W | |
| Portée de l'ombilic | 10 m | |
| Nombre d'actionneurs par patte | 3 | |
| Degrés de liberté | 18 DDL | |
| Boîtier de commande | Manette de Xbox & ordinateur externe | |
| Alimentation interne des actionneurs | 2x24V avec limitation 2x40A | |
| Niveau sonore | 60dB | |
| Matériaux | Aluminium, Acier, PLA, ABS, Silicone | |
| Température de fonctionnement | -20 → 45° C | |

TABEAU 1 : CARACTERISTIQUES DU ROBOT POPI

3 Partie mécanique

3.1 Descriptif de l'équipement

Le robot POPI possède un axe de symétrie longitudinal. Il possède 4 pattes, identiques 2 à 2.

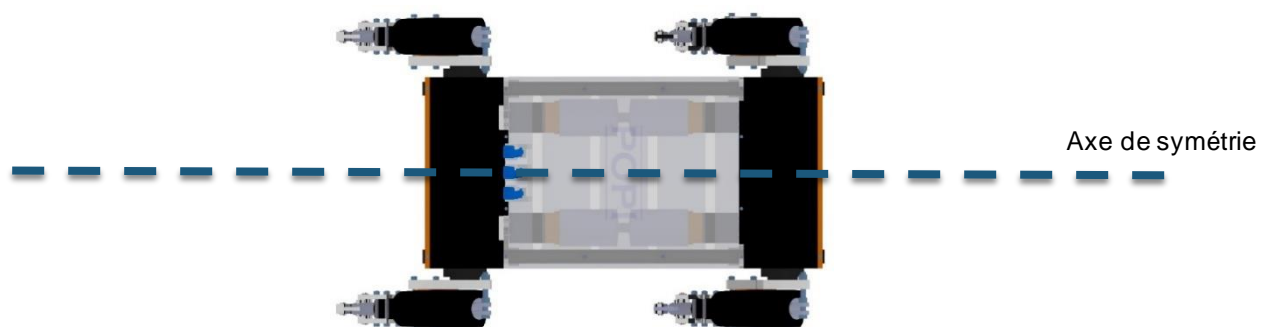


FIGURE 3 : AXE DE SYMETRIE POPI

Chaque patte est composée d'une articulation de type « épaule », d'une articulation de type « aile » et d'une articulation de type « coude ». Chacune de ces articulations correspond à un actionneur.

On compte donc 12 actionneurs :

- Épaule : 4 moteurs (couplés à des réducteurs)
- Aile : 4 motoréducteurs
- Coude : 4 vérins

Pour une nomenclature complète des pièces, se référer à l' *Annexe 2 : Notice de montage mécanique*.

3.1.1 Actionneur épaule

| Moteur – Réducteur épaule | |
|-------------------------------|--------------------------------------|
| Référence | MY1016 - CSMOTO |
| Alimentation | 24 V DC |
| Puissance | 350 W |
| Courant max | 18,7 A |
| Réducteur | PLFE 90 - NEUGART |
| Rapport de réduction | 100 |
| Vitesse max sans charge | 34,5 tr/min |
| Couple max sans charge | 300 Nm |
| Vitesse max avec charge | 27,5 tr/min |
| Couple max avec charge | 122 Nm |
| Plages d'articulations épaule | [0; 180] degré, soit $[0 ; \pi]$ rad |

TABLEAU 2 : CARACTERISTIQUES DE L'ENSEMBLE EPAULE

3.1.2 Actionneur aile

| Motoréducteur aile | |
|-----------------------------|---|
| Référence | MR 977 – CBF MOTOR |
| Alimentation | 24 V DC |
| Puissance | 130 W |
| Courant max | 20 A |
| Rapport de réduction | 96 |
| Longueur | 229 mm |
| Vitesse max sans charge | 30 tr/min |
| Vitesse max avec charge | 26 tr/min |
| Couple max avec charge | 120 Nm |
| Plages d'articulations aile | $[-22; 22]$ degré soit $[-0,4 ; 0,4]$ rad |

TABLEAU 3 : CARACTERISTIQUES DE L'ENSEMBLE AILE

3.1.3 Actionneur coude

| Vérin coude | |
|-----------------------------------|--------------------------------------|
| Référence | MA2 - TiMOTION |
| Alimentation | 24V DC |
| Puissance | 200 W |
| Vitesse max sans charge | 28,5 mm/s |
| Vitesse max avec charge | 22,0 mm/s |
| Charge de poussée | 2000 N |
| Force de verrouillage automatique | 2600 N |
| Duty cycle | 25 % |
| Courant sans charge | 2,2 A |
| Courant avec charge | 9.0 A |
| Plages d'articulations coude | [15; 125] degré soit [0,2 ; 2,2] rad |

TABEAU 4 : CARACTERISTIQUES DE L'ENSEMBLE COUDE

3.2 Principe de montage

Une nomenclature détaillée et une notice de montage sont présentées en Annexe 2 : Notice de montage mécanique.

4 Partie électronique

L'ensemble des éléments électroniques est fixé sur une plaque en plexiglas disposée au-dessus dans le corps du robot. On retrouve dans le positionnement des éléments une symétrie droite / gauche comme pour la partie mécanique. On distingue la partie commande (au centre de la plaque), de la partie puissance (en dehors de la plaque). Le lien entre ces deux parties se fait par les cartes « contrôleur » Cytron 13A et 20A placées aux extrémités avant et arrière de la plaque.

4.1 Intelligence embarquée

Le robot embarque une carte Raspberry PI 3B+ et deux cartes BeagleBoneBlack (BBB). La Raspberry PI est utilisée pour sa puissance de calcul tandis que les BBB sont utilisées pour leur nombre d'entrées / sorties. Chaque BBB s'occupe de la lecture des capteurs et de l'écriture des commandes

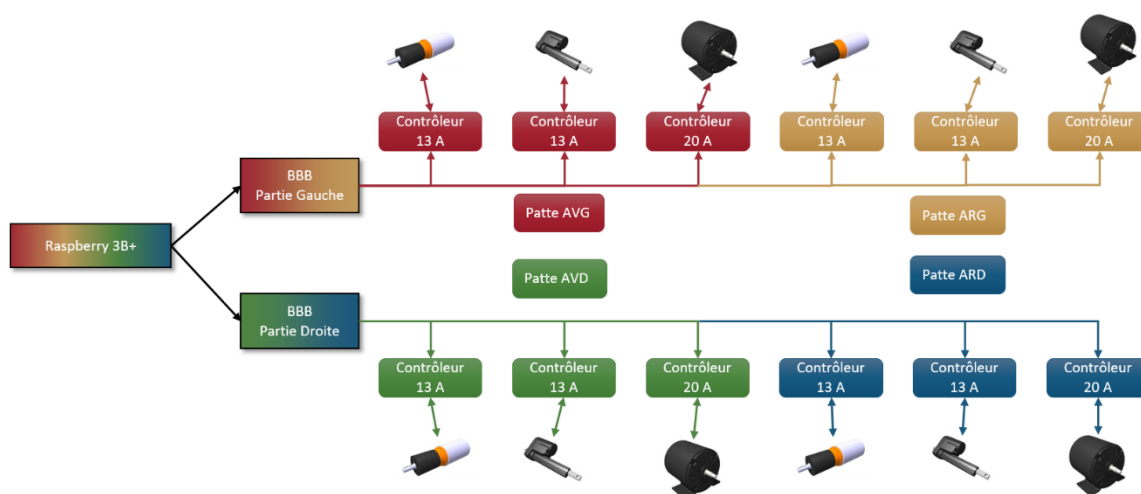


FIGURE 4 : ELECTRONIQUE DE LA PARTIE COMMANDE

moteurs PWM pour l'un des deux côtés du robot. Les actionneurs de l'aile et du coude sont commandés par des cartes contrôleur 13A et les actionneurs de l'épaule par des cartes contrôleurs 20A.

4.2 Capteurs

25 capteurs sont présents sur le robot POPI. On retrouve tout d'abord une centrale inertielle, située sur la plaque en plexiglas supportant la partie électronique.

Ensuite sont situés sur chaque patte :

- Une roue codeuse placée entre le moteur et le réducteur de l'épaule. C'est un capteur de position relative, qui nous permet de connaître l'angle de l'épaule. Il est contrôlé par les QEC.
- Un capteur de position situé au niveau de l'aile.
- Une sortie capteur provenant du vérin.

Pour chaque actionneur il y a un capteur de courant, qui permet d'être informé en cas d'appel de courant trop important.

4.3 Alimentation

4.3.1 Platine d'alimentation

La platine d'alimentation, présentée en Figure 5, fournit l'alimentation au robot. Elle permet l'alimentation directe en 230V AC, et la transformation du 230 V AC en 24V DC par deux sources distinctes situées à l'arrière de la platine.

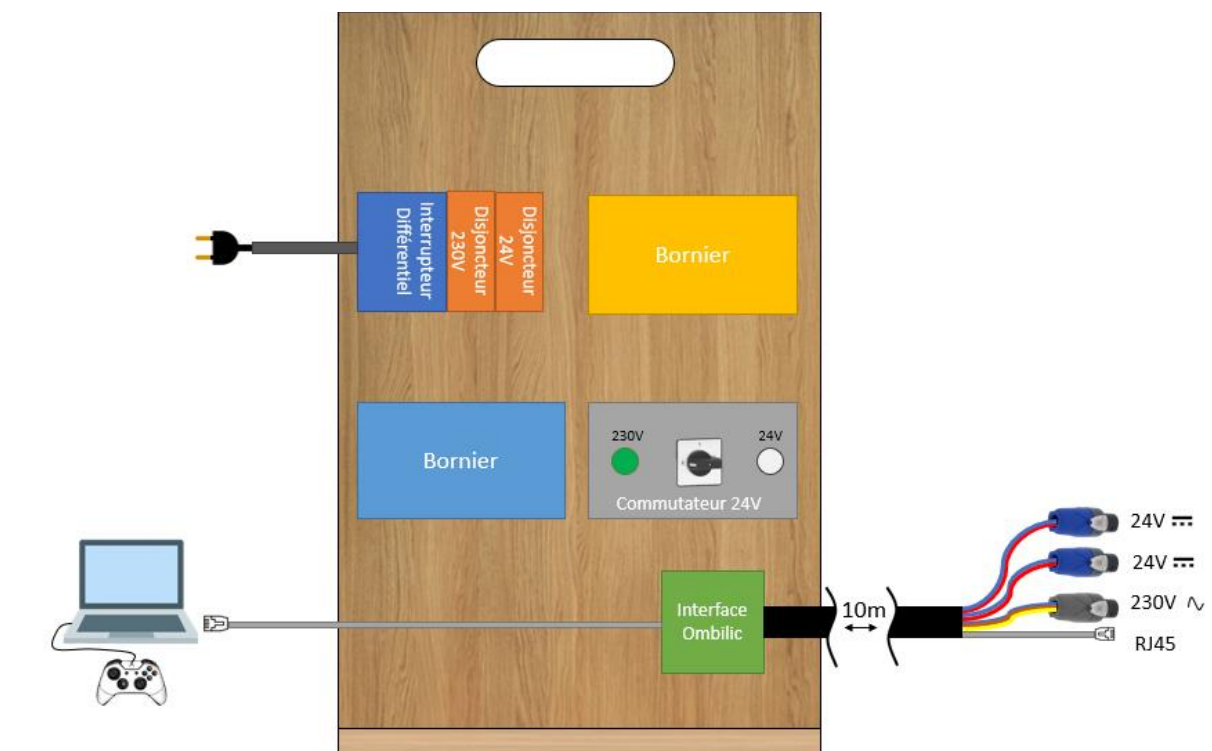


FIGURE 5 : SCHEMA DE LA PLATINE D'ALIMENTATION

L'ombilic, qui regroupe les différents câbles d'alimentation et de communication dans une seule gaine est relié d'une part à la platine d'alimentation et de l'autre part au robot.

4.3.2 Distribution de l'alimentation

A l'extrémité de l'ombilic coté robot, on trouve :

- 2 connecteurs d'alimentation 24V DC
- 1 connecteur d'alimentation 230V AC
- Un câble RJ45
- 2 mousquetons

Il permet d'alimenter les actionneurs, les cartes électroniques, les capteurs. Il permet aussi la communication avec le PC utilisateur et possède deux mousquetons pour éviter de tirer sur les câbles. Les mousquetons peuvent aussi permettre de tirer le robot hors de sa zone de mission si nécessaire.

Le 24V DC est dédié à la partie puissance. Il est relié aux cartes « contrôleur » qui délivrent la puissance aux actionneurs.

Le 230V AC est transformé en 5V DC et est distribué à la Raspberry PI, aux BBB, aux capteurs de courant et aux cartes Quadrature Encoder Counter (QEC) via un hub USB.

Les deux BeagleBoneBlack fournissent une tension de 1,8V utilisée pour alimenter les capteurs de positions de type potentiomètre du coude et de l'aile.

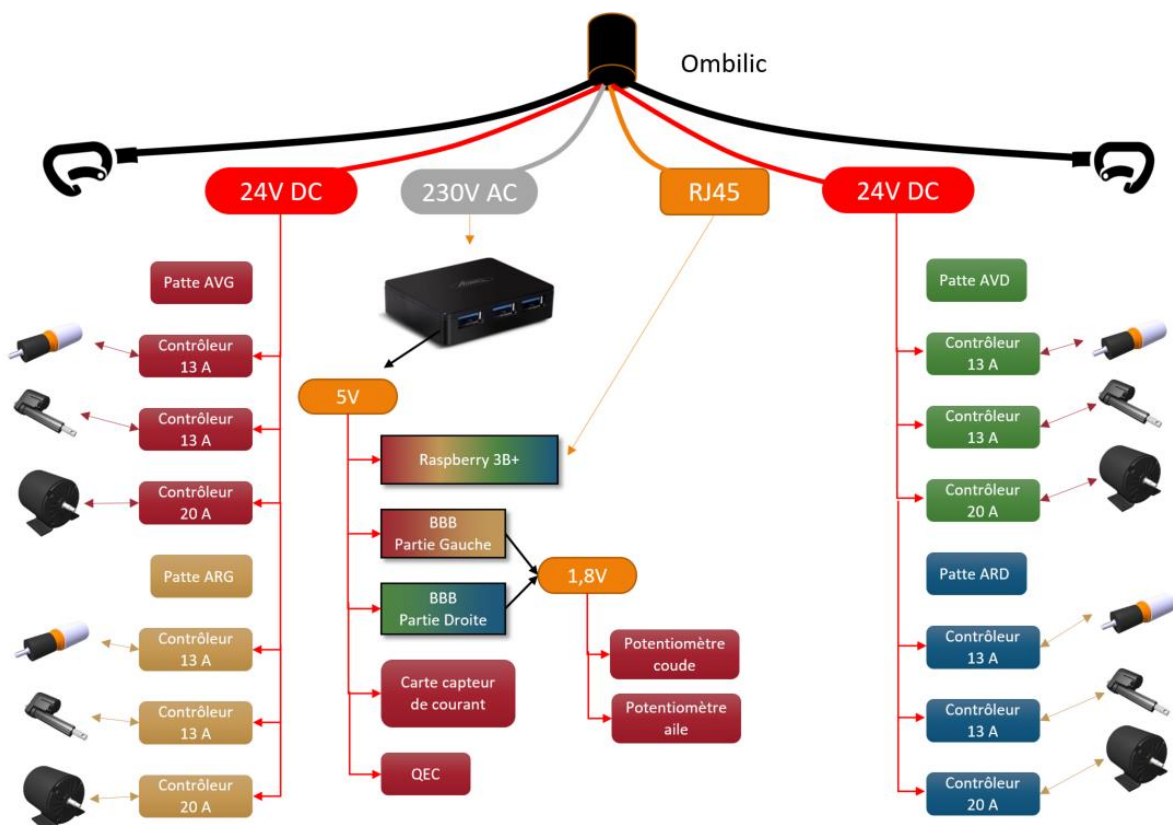


FIGURE 6 : DISTRIBUTION DE L'ALIMENTATION

4.4 Gestion du câblage

4.4.1 Code couleur

| COULEUR | | |
|-------------------|--------------|----------------|
| Fils fins | | |
| Orange foncé | | PWM |
| Jaune | | Pin TOR |
| Orange clair | | Pin Analogique |
| Rouge | | 5V |
| Rouge/Blanc | | 1,8V |
| Noir | | GND |
| Blanc | | S.Clock |
| Vert – Vert/blanc | | Chip select |
| Bleu – Bleu/blanc | | Liaison série |
| Fils épais | | |
| Rouge – Marron | | 24V |
| Bleu | | 0V |
| Vert / jaune | | Terre |
| Gaine noir | Marron foncé | Ligne – 230V |
| | Bleu | Neutre |

TABEAU 5 : CODE COULEUR DES FILS ELECTRIQUES DU ROBOT

4.4.2 Affectation des bornes

BeagleBoneBlack Gauche

| P8 | | P9 | |
|----------------------|----------------------|------------------------|------------------------|
| 46 | 45 | 46 | 45 |
| 44 | 43 | 44 | 43 |
| 42 | 41 | 42 | 41 |
| 40 | 39 | ANA-Aile-Potar-AVG 40 | ANA-Aile-Potar-ARG 39 |
| 38 | 37 | ANA-Coude-Potar-AVG 38 | ANA-Coude-Potar-ARG 37 |
| 36 | 35 | ANA-Aile-Cour-ARG 36 | ANA-Epaule-Cour-ARG 35 |
| 34 | 33 | 34 | ANA-Coude-Cour-ARG 33 |
| 32 | 31 | 1,8V 32 | SPI-SCK-QECG 31 |
| 30 | 29 | SPI-MOSI-QECG 30 | SPI-MISO-QECG 29 |
| 28 | 27 | SPI-CS2-QECG 28 | SPI-CS1-QECG 27 |
| TOR-Courant-1-AVG 26 | 25 | 26 | 25 |
| 24 | 23 | 24 | 23 |
| 22 | 21 | PWM-Aile-AVG 22 | PWM-Aile-ARG 21 |
| 20 | PWM-Coude-ARG 19 | 20 | 19 |
| TOR-Courant-1-ARG 18 | TOR-Courant-2-AVG 17 | 18 | 17 |
| TOR-Courant-2-ARG 16 | TOR-Courant-3-AVG 15 | PWM-Epaule-ARG 16 | 15 |
| TOR-Courant-3-ARG 14 | PWM-Coude-AVG 13 | PWM-Epaule-AVG 14 | 13 |
| DIR-Coude-ARG 12 | DIR-Aile-ARG 11 | 12 | 11 |
| DIR-Epaule-ARG 10 | DIR-Coude-AVG 9 | 10 | 9 |
| DIR-Aile-AVG 8 | DIR-Epaule-AVG 7 | 8 | 7 |
| 6 | 5 | 6 | 5V 5 |
| 4 | 3 | 4 | 3 |
| 2 | 1 | 2 | GND 1 |

TABEAU 6 : AFFECTATION DES BORNES BEAGLEBONEBLACK GAUCHE

BeagleBoneBlack Droite

| P8 | | P9 | |
|----------------------|----------------------|------------------------|------------------------|
| 46 | 45 | 46 | 45 |
| 44 | 43 | 44 | 43 |
| 42 | 41 | 42 | 41 |
| 40 | 39 | ANA-Aile-Potar-AVD 40 | ANA-Aile-Potar-ARD 39 |
| 38 | 37 | ANA-Coude-Potar-AVD 38 | ANA-Coude-Potar-ARD 37 |
| 36 | 35 | ANA-Aile-Cour-AVD 36 | ANA-Epaule-Cour-AVD 35 |
| 34 | 33 | 34 | ANA-Coude-Cour-AVD 33 |
| 32 | 31 | 1,8V 32 | SPI-SCK-QECD 31 |
| 30 | 29 | SPI-MISO-QECD 30 | SPI-MOSI-QECD 29 |
| 28 | 27 | SPI-CS1-QECD 28 | SPI-CS2-QECD 27 |
| TOR-Courant-1-AVD 26 | 25 | 26 | 25 |
| 24 | 23 | 24 | 23 |
| 22 | 21 | PWM-Aile-AVD 22 | PWM-Aile-ARD 21 |
| 20 | PWM-Coude-ARD 19 | 20 | 19 |
| TOR-Courant-1-ARD 18 | TOR-Courant-2-AVD 17 | 18 | 17 |
| TOR-Courant-2-ARD 16 | TOR-Courant-3-AVD 15 | PWM-Epaule-ARD 16 | 15 |
| TOR-Courant-3-ARD 14 | PWM-Coude-AVD 13 | PWM-Epaule-AVD 14 | 13 |
| DIR-Coude-ARD 12 | DIR-Aile-ARD 11 | 12 | 11 |
| DIR-Epaule-ARD 10 | DIR-Coude-AVD 9 | 10 | 9 |
| DIR-Aile-AVD 8 | DIR-Epaule-AVD 7 | 8 | 7 |
| 6 | 5 | 6 | 5V 5 |
| 4 | 3 | 4 | 3 |
| 2 | 1 | 2 | GND 1 |

TABEAU 7 : AFFECTATION DES BORNES BEAGLEBONEBLACK DROITE

4.5 Montage de l'électronique

Une nomenclature détaillée et des plans de montages sont présentés en Annexe 1 : Plans de montage électronique.

5 Réseau

Le réseau a été configuré de la façon suivante :

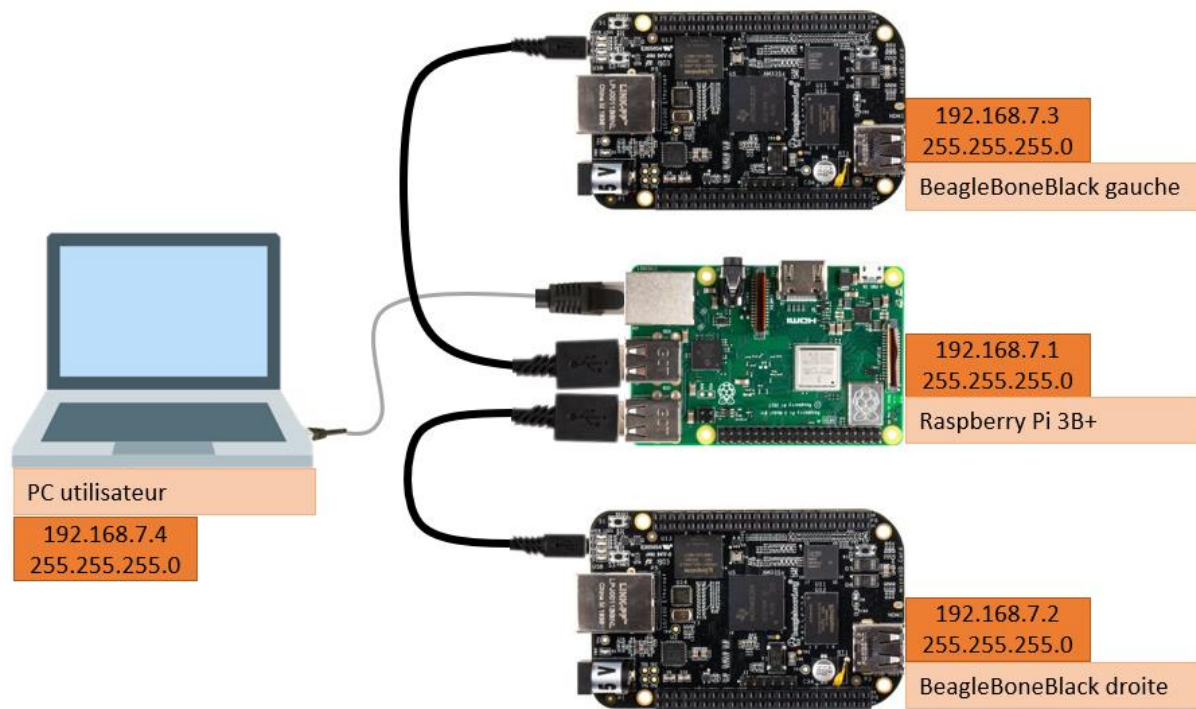


FIGURE 7 : ARCHITECTURE RESEAU DU ROBOT

Toutes les adresses IP sont configurées en statique. Les ports USB de la Raspberry et micro USB des BeagleBoneBlack sont utilisés comme des ports Ethernet. Un pont réseau relie les 3 interfaces réseau de la Raspberry (2 USB et 1 Ethernet) sous l'adresse 192.168.7.1.

Sur chaque carte des serveurs SSH ont été ouverts.

6 Architecture logicielle

L'ensemble de l'architecture logicielle de POPI a été créée sur ROS. ROS, pour *Robot Operating System*, est un *middleware* qui est devenu un standard dans la conception robotique. C'est lui qui assure la communication entre l'ensemble de nos logiciels, ou plutôt l'ensemble de nos nœuds, à savoir des programmes (en Python ou C++) que nous avons transformés en exécutables.

6.1 Principe de fonctionnement

Pour le moment, le produit fonctionne en deux temps qu'il se faut de bien distinguer :

- La génération de la trajectoire de marche
- L'envoi de cette trajectoire sur le robot

En effet, le produit ne dispose pas à l'heure actuelle d'une génération de trajectoire de marche à la volée. Ce fonctionnement en deux étapes distinctes permet de s'assurer que les commandes envoyées sur le robot soient correctes et ainsi de diminuer le risque de chute du robot.

Le produit est fourni avec des trajectoires de marche qui ont été testées et validées par nos équipes, mais aussi avec l'ensemble des outils nécessaires à la génération de vos propres trajectoires. En cela, POPI se veut être une plateforme de développement ouverte pour le client.

Dans la suite de cette partie, nous présenterons ces différents outils. Si vous souhaitez connaître immédiatement les instructions pour les utiliser, vous pouvez vous référer à la partie [7 Notice d'instruction](#).

6.2 Génération de trajectoire

C'est durant cette étape que vont être calculées l'ensemble des commandes à envoyer sur les 12 actionneurs pour réaliser une marche. Pour ce faire, nous utilisons le package ROS [Tower](#) développé par Alexander W. Winkler.

Les modèles cinématique et dynamique de notre robot y ont été ajoutés, indiquant notamment les positions nominales des pieds dans le repère de la base et leur écart maximal à cette position nominale, mais aussi le poids de la base et les composantes de sa matrice d'inertie (selon le modèle du [Single Rigid Body Dynamics](#)).

À partir de ces informations et des paramètres de marche définis dans l'application (terrain, position et orientation d'arrivée, démarche souhaitée), le programme calcule les trajectoires de chaque pied au cours de la marche. Avec celles-ci, en utilisant le modèle géométrique inverse du robot, défini dans le package *Xpp* qui est nécessaire au fonctionnement de *Tower*, nous calculons les trajectoires que doivent suivre les actionneurs : on obtient nos commandes.

Ces commandes peuvent par la suite être enregistrées en format *rosvbag* à l'aide de l'outil [rqt bag](#) et éventuellement modifiées sur un tableur, puis simulées avant d'être envoyées sur le robot.

6.3 Simulation

Xpp permet la visualisation 3D sur [RViz](#) du robot lors de la marche, mais également d'autres éléments comme les forces de contact ou encore les cônes de friction.

Le modèle [URDF](#) (*Unified Robot Description Format*) indispensable à toute visualisation ou simulation du robot fait partie des éléments livrés avec POPI. Celui-ci a été généré directement à partir du modèle 3D mécanique du robot afin de s'assurer de sa représentativité. Il contient, sous forme de texte (XML), l'ensemble des informations des corps composants le robot (forme, dimensions, poids, etc.) et de leurs liaisons (type, limites, élasticité, friction).

Les liaisons ont ensuite été simplifiées pour conserver les mouvements fonctionnels, alléger la simulation et passer outre les problèmes générés par les chaînes cinématiques fermées qui transforment le mouvement de translation du vérin en rotation du coude. En effet, le format URDF ne permet pas l'utilisation d'une structure cinématique en graph, simplement en arbre.

Ce modèle est également utilisé avec l'outil de simulation physique [Gazebo](#), qui supporte les [moteurs physiques](#) ODE, Dart, Bullet et Simbody et utilise le [moteur de rendu](#) OGRE. Il permet également d'intégrer au modèle des capteurs et des interfaces utilisateur. Cela en fait aujourd'hui l'un des simulateurs 3D de robotique les plus utilisés.

Nous avons donc créé sur Gazebo un monde virtuel dont nous avons paramétré la physique afin d'assurer une représentativité proche du robot réel. Avant d'envoyer une trajectoire sur le robot réel, vous pouvez donc vous assurer de son bon fonctionnement. De même, tous vos tests d'intégration continue de programmes peuvent être réalisés sans risquer d'endommager le robot.

6.4 Monitoring

Lors de la simulation comme lors du fonctionnement réel, il est possible de suivre sur l'outil [rqt gui](#) l'évolution de toutes les données échangées entre nos différents programmes, notamment les consignes de position et les valeurs renvoyées par les capteurs, mais aussi par exemple la fréquence de la boucle de contrôle.

6.5 Envoi de trajectoire sur POPI



Mises en garde :

- Dans sa version actuelle, le robot n'a pas de connaissance de son environnement immédiat. Il revient donc à l'utilisateur de s'assurer que le terrain sur lequel il évolue est similaire à celui utilisé pour générer la trajectoire de marche.
- Malgré une simulation représentative, il convient également de garder à l'esprit que le comportement du robot réel peut différer légèrement.
- Enfin, puisque sur *Towr* le robot démarre en position nominale, les trajectoires sont générées à partir de cette position : il faut donc prendre garde à le placer dans cette position avant d'envoyer une trajectoire. Pour cela, nous avons au préalable recours à une manette de Xbox branchée au PC utilisateur, paramétrée de sorte à permettre facilement l'envoi de la position nominale ou de la position de repos (décrit en 7.7.3).

La *Figure 8* décrit le fonctionnement logiciel de POPI lors d'une marche réelle.

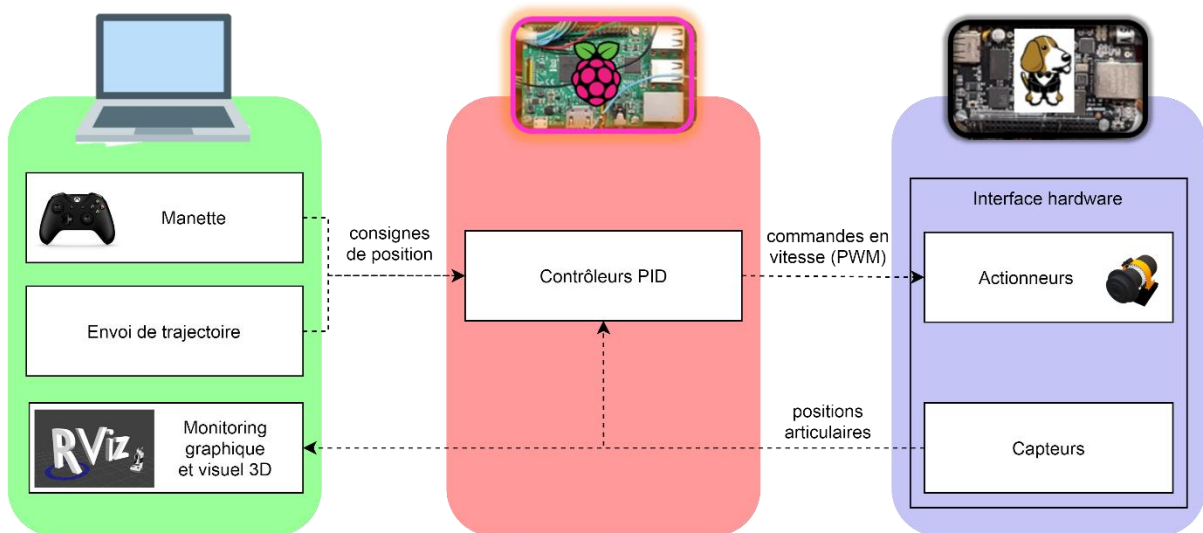


FIGURE 8 : SCHEMA DE FONCTIONNEMENT LOGICIEL DE POPI

6.6 Arborescence des dossiers

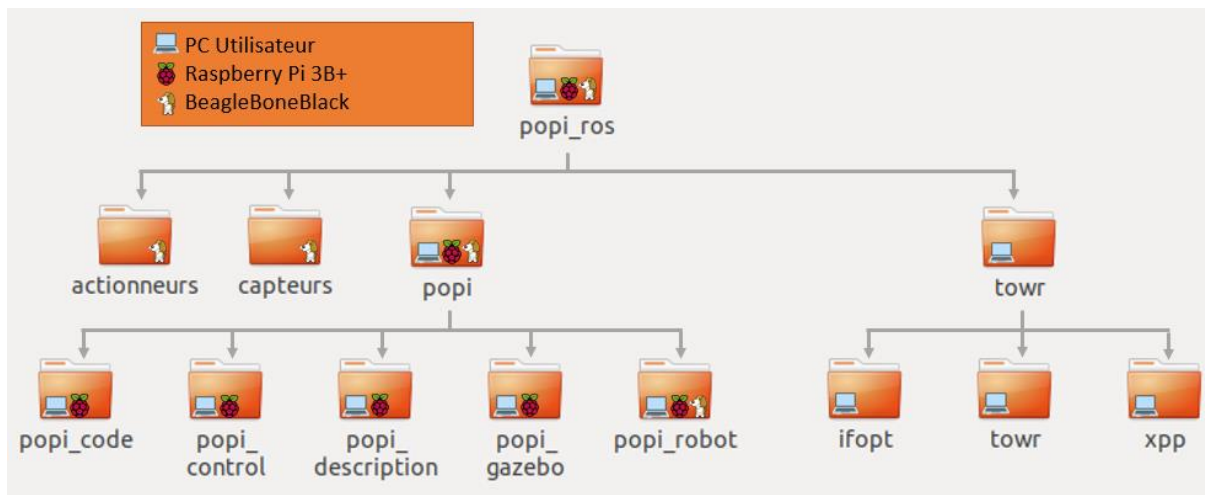


FIGURE 9 : ARBORESCENCE DES DOSSIERS

- **Popi_code** : Contient différents codes source C++ générés automatiquement.
- **Popi_control** : Contient la description des contrôleurs utilisés pour asservir les actionneurs ainsi que les routines permettant de faire lever le robot et de le mettre au repos.
- **Popi_description** : Contient le modèle virtuel du robot POPI ainsi qu'un fichier launch pour le visualiser sur Rviz.
- **Popi_gazebo** : Contient tous les fichiers de paramétrage de l'environnement de simulation ainsi qu'un fichier launch pour visualiser le robot sur Gazebo.
- **Popi_robot** : Contient les définitions des messages et des services utilisés pour communiquer l'état du robot et envoyer les commandes aux moteurs.
- **Ifopt** : Package permettant à Towr d'utiliser un solveur haute performance.
- **Towr** : Package contenant l'interface graphique permettant de générer des trajectoires pour robot à patte.
- **Xpp** : Package permettant la visualisation de trajectoire pour robot à patte sur Rviz. Il contient aussi des plug-ins affichant les zones et forces de contact au sol.
- **Actionneurs** : Contient les nœuds ROS qui communiquent avec les moteurs du robot.
- **Capteurs** : Contient les nœuds ROS qui communiquent avec les capteurs du robot.

7 Notice d'instruction



Cette notice d'instructions sera amenée à évoluer en même temps que le code source. Une simplification des procédures d'utilisation est à l'étude.

Note : Les mots de passe des cartes sont « ubuntu » pour la Raspberry Pi et « tempPWD » pour les BeagleBone. Ils peuvent être modifiés par l'utilisateur si besoin.

7.1 Installation

Vous trouverez ci-dessous les étapes d'installation à réaliser sur le PC utilisateur afin d'utiliser les outils logiciels de POPI :

1. Disposer d'une session Ubuntu 18.04 (en [machine virtuelle](#) ou en [dual boot](#))
2. Installer [ROS melodic](#) full desktop (cette version contient nativement les outils tels que RViz, Gazebo et rqt)
3. [FACULTATIF] Installer le logiciel de traitement de texte [Sublime text](#). Pour cela, ouvrir un terminal (**CTRL+ALT+T**) et taper la commande suivante :

```
01. sudo apt-get install sublime-text
```

Un mot de passe vous sera demandé pour passer en super utilisateur ("sudo"), par défaut, c'est "ubuntu", autrement il s'agit du mot de passe que vous avez défini lors de la création de votre session Ubuntu. Si vous êtes habitués à d'autres outils de traitement de texte, cette étape n'est pas nécessaire. Veillez, le cas échéant, à remplacer l'instruction « *subl* » des commandes par l'instruction appropriée (e.g. « *nano* »).

4. Installer les *packages* nécessaires :

```
01. sudo apt-get install ros-melodic-gazebo-ros ros-melodic-ros-control
ros-melodic-gazebo-ros-control ros-melodic-gazebo-ros-pkgs
ros-melodic-ros-controllers ros-melodic-robot-controllers ros-melodic-robot-state-publisher
ros-melodic-joy git cmake libeigen3-dev coinor-libipopt-dev libncurses5-dev xterm
```

5. Nous utilisons la version de Gazebo 9.12.0. Il est possible que l'installation de ROS melodic ne vous fournisse que la version 9.0.0. Vous pouvez le vérifier en tapant la commande :

```
01. gazebo -v
```

Si vous n'avez pas la version 9.12.0, procédez comme suit :

```
01. sudo subl /etc/apt/sources.list.d/gazebo-stable.list
```

Dans le fichier texte qui vient de s'ouvrir, vérifiez que l'adresse est bien : *deb http://packages.osrfoundation.org/gazebo/ubuntu-stable bionic main*, remplacez-la si ce n'est pas le cas.

6. Enfin, mettez à jour l'ensemble de vos sources :

```
01. sudo apt-get update
02. sudo apt-get upgrade
```

7. Récupérer puis compiler le dossier *popi_ros* présent sur le GitHub du projet :

```
01. mkdir -p ~/catkin_ws/src
02. cd ~/catkin_ws/src
03. git clone https://github.com/popi-mkx3/popi_ros.git
```

```
04. cd ~/catkin_ws
05. catkin_make_isolated
```

8. Ouvrez votre fichier `.bashrc` (les instructions qu'il contient seront effectuées automatiquement à chaque ouverture de terminal) :

```
01. sudo subl ~/.bashrc
```

Ajoutez, à la fin de ce fichier, les quatre lignes suivantes, puis sauvegardez le fichier :

```
source /opt/ros/melodic/setup.bash
source ~/catkin_ws/devel_isolated/setup.bash
#export ROS_IP=192.168.7.4
#export ROS_MASTER_URI=http://192.168.7.1:11311
```

Important : pour les parties 7.3 à 7.6, il n'est pas nécessaire de disposer de la plateforme POPI ni d'y être branché. Toutes les instructions s'effectuent sur le PC utilisateur. Assurez-vous que les deux dernières lignes dans votre fichier `.bashrc` sont commentées (présence du #), sans quoi les instructions de ces trois parties ne fonctionneront pas.

À l'inverse, il faudra penser à décommenter ces deux lignes (enlever les #) lors de l'utilisation de la plateforme POPI (partie 7.70).

7.2 Raccordement / alimentation

Pour utiliser le robot il faut dans un premier temps le raccorder à la platine d'alimentation. Avant de commencer, assurez-vous que la platine soit branchée, et que tous les interrupteurs soient placés sur off : l'interrupteur différentiel, les deux disjoncteurs et le commutateur 24V (voir *Figure 5*).

Branchez ensuite le câble RJ45 sur le port Ethernet de la Raspberry. Installez le carter en plexiglass et fixez-le avec ses vis. Branchez les trois câbles restant sur les bons connecteurs et attachez les mousquetons comme présenté sur la *Figure 10*.

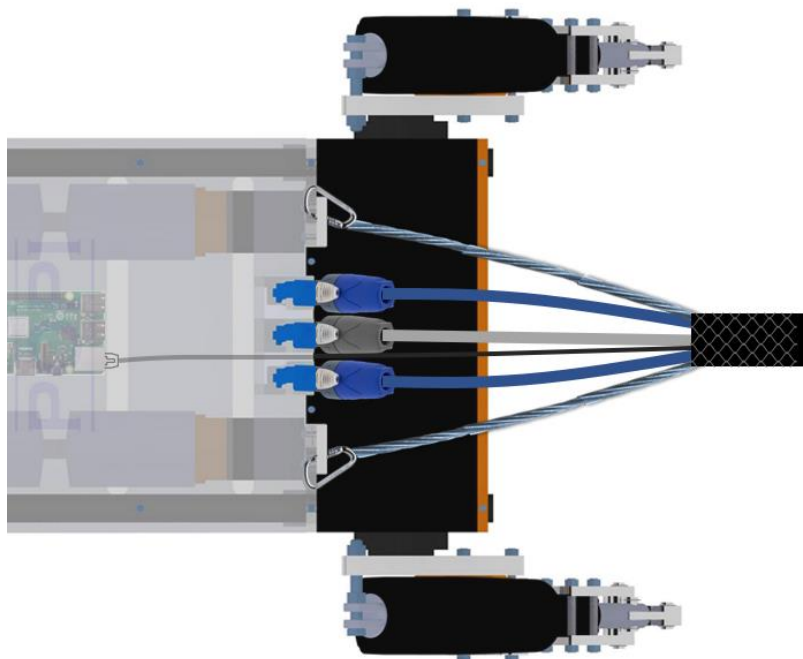


FIGURE 10 : BRANCHEMENT DE L'OMBILIC.

Une fois tous les câbles correctement branchés, positionnez sur ON les interrupteurs suivants : interrupteur différentiel, disjoncteur 230V, disjoncteur 24V. Connectez le PC utilisateur de l'autre côté de l'ombilic au niveau de la platine d'alimentation. Essayez de ping les adresses IP des trois cartes jusqu'à ce que la communication soit opérationnelle. Une fois que les cartes ont démarré correctement, la plateforme POPI est prête à l'emploi.



Suivre toutefois les précautions suivantes :

- Avant d'activer le 24V, veiller qu'aucune commande ne soit envoyée sur les actionneurs
- Ne pas activer le 24V si le ping des cartes Raspberry et BBB ne fonctionne pas.
- Le robot est équipé de deux sécurités majeures en cas d'envoi d'une commande trop importante : les alimentations se mettent en défaut durant 3 secondes lors d'un appel de courant trop important, les cartes contrôleurs possèdent des condensateurs capables d'absorber d'importants appels de courants.
- Si une alimentation se met en défaut, appuyer rapidement sur l'arrêt d'urgence ou positionner le commutateur 24V sur OFF et analyser les commandes envoyées aux actionneurs avant de le réactiver.
- Lorsque le commutateur 24V est positionné sur OFF, attendre 45s que les condensateurs des alimentations se déchargent avant toute intervention sur le robot.
- Si aucun des voyants de la platine ne s'allume, vérifier que le bouton d'arrêt d'urgence ne soit pas enclenché.

7.3 Génération de trajectoires

Pensez à vérifier votre fichier bashrc comme indiqué en fin de partie 7.1.

7.3.1 Utilisation de Tower

1. Sur le PC utilisateur, lancez l'interface *Tower* :

```
01. roslaunch tower_ros tower_ros_popi.launch
```

Cette commande permet l'ouverture de l'interface utilisateur de *Tower* (Figure 11) et de *Rviz* (Figure 12).

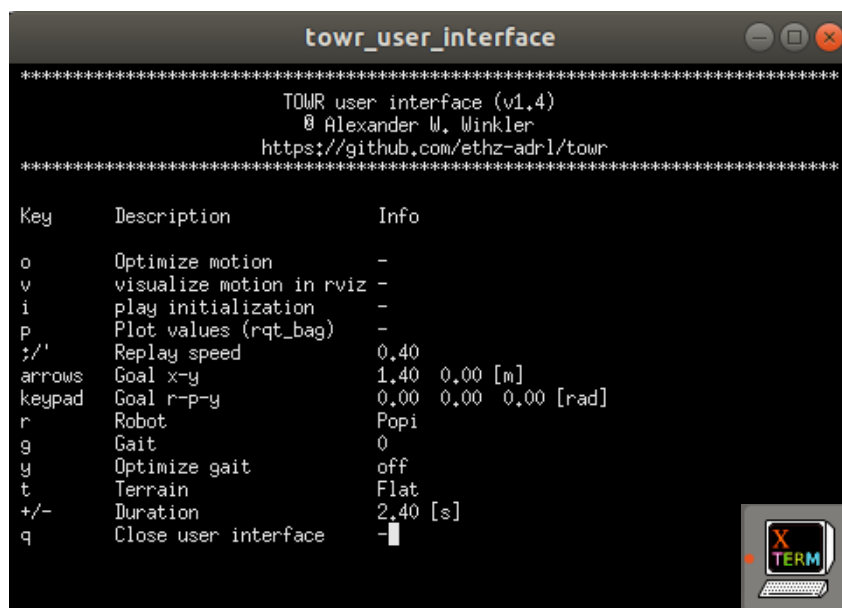


FIGURE 11 : INTERFACE UTILISATEUR TOWER

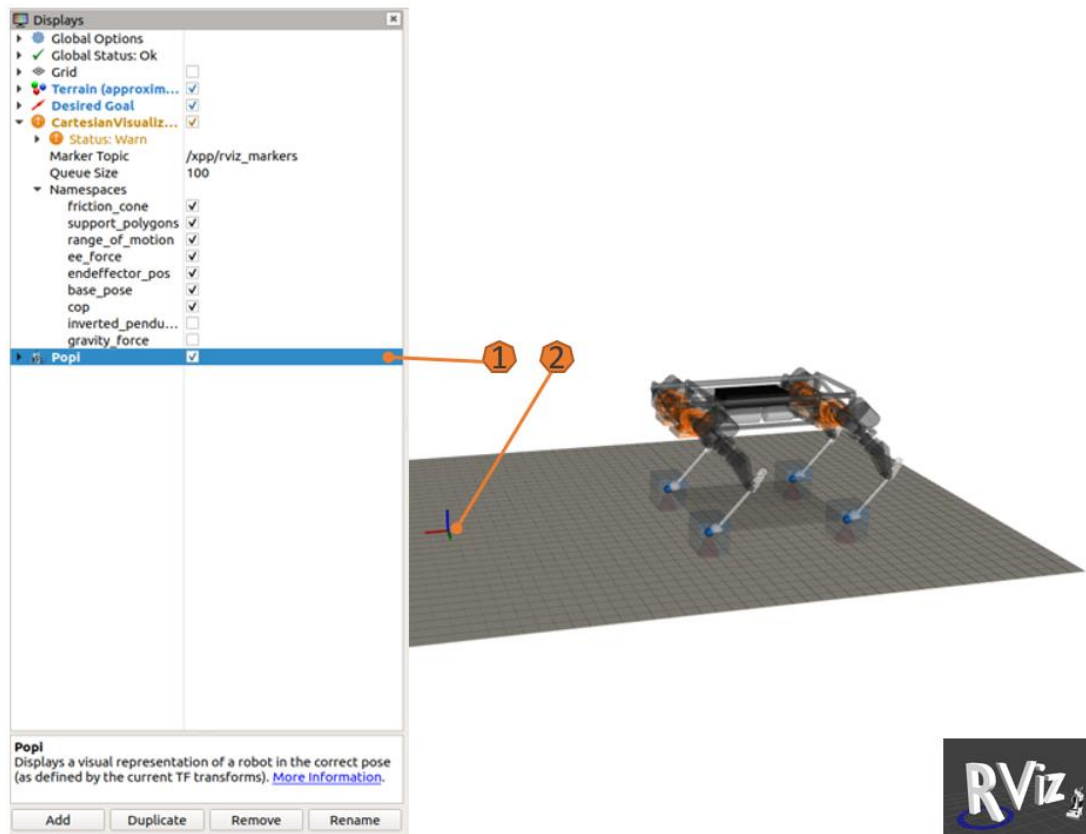


FIGURE 12 : INTERFACE RVIZ

Pour afficher le visuel 3D sur *RViz*, cochez la case comme indiqué en *Figure 12* (1), sélectionnez l'interface utilisateur de *Towr* et cliquez sur « i » pour lancer la séquence d'initialisation.

L'interface utilisateur de *Towr* est très intuitive et facile à prendre en main, les touches sont indiquées dans la colonne key.

2. Déplacez l'objectif à atteindre avec POPI (représenté par le trièdre en 2) (flèches directionnelles et pavé numérique).
3. Choisissez le terrain désiré (« t »).
4. Choisissez la démarche à adopter, les seules ayant été testées en réel sont la 0 et la 1 (« g ») :
 - 0 : démarche statique : pattes gauches puis droites
 - 1 : démarche dynamique : les pattes diagonales bougent ensemble
 - 2 : démarche dynamique : pattes gauches puis droites
 - 3 : démarche dynamique : pattes avant arrière
 - 4 : démarche statique : pattes avant puis arrière
5. Lancez le calcul de la trajectoire (« o »)
6. Vous pouvez par la suite revoir le mouvement du robot en appuyant sur « v » (en appuyant de nouveau sur « o », un nouveau calcul serait effectué).
7. Ajustez avec précaution la vitesse du mouvement avec les touches « ; » et « ' ». La vitesse de visualisation correspondra également à la vitesse à laquelle vos commandes seront envoyées sur le robot réel !

7.3.2 Enregistrement des commandes en rosbag

1. Une fois satisfait de la trajectoire que vous avez créée, il vous faut l'enregistrer. Pour ce faire, ouvrez `rqt_bag` :

```
01. cd ~/catkin_ws/src/popl_ros/popl/popl_robot/trajectories/bags
02. rosrn rqt_bag rqt_bag
```

2. Pour démarrer l'enregistrement, cliquez sur le cercle rouge situé en haut à gauche de la fenêtre. La nouvelle fenêtre en *Figure 13* apparaît. Comme sur l'image, sélectionnez les 12 topics `/popi/.../command` qu'on souhaite surveiller puis appuyez sur *Record*. Choisissez un nom pour votre fichier (e.g. « *ma_marche* ») et cliquez sur *Save*.

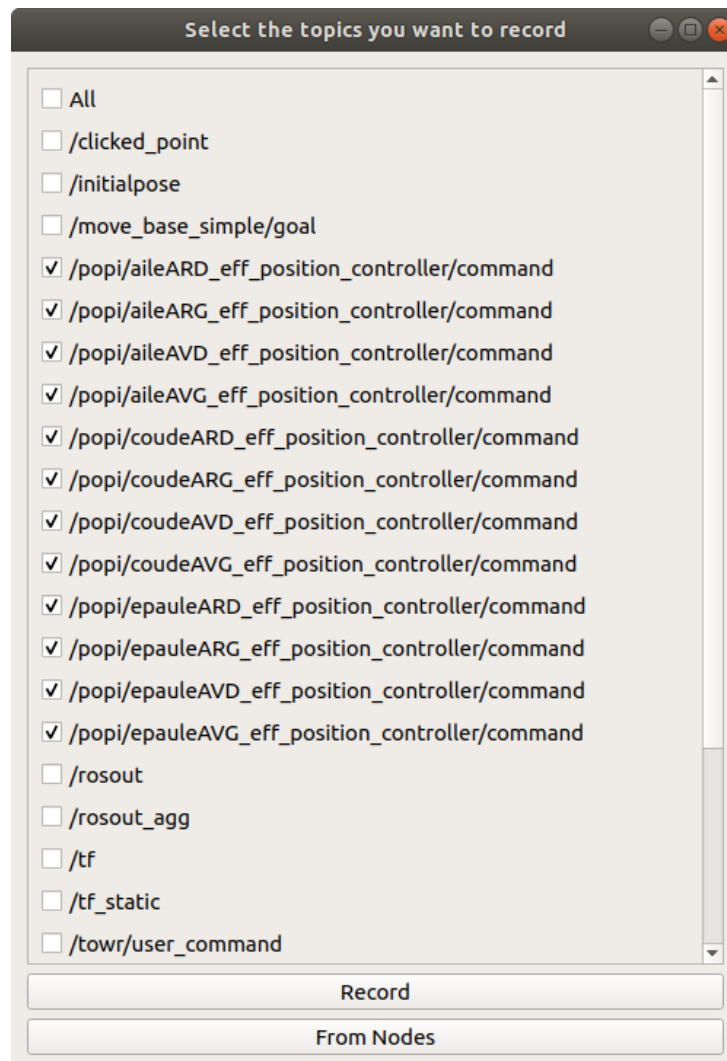


FIGURE 13 : SELECTION DES TOPICS SUR RQT_BAG

3. Vous pouvez à présent revenir sur l'interface utilisateur *Towr* et taper « v ». Sur *rqt_bag* s'affichent les trames enregistrées comme sur la *Figure 14*.

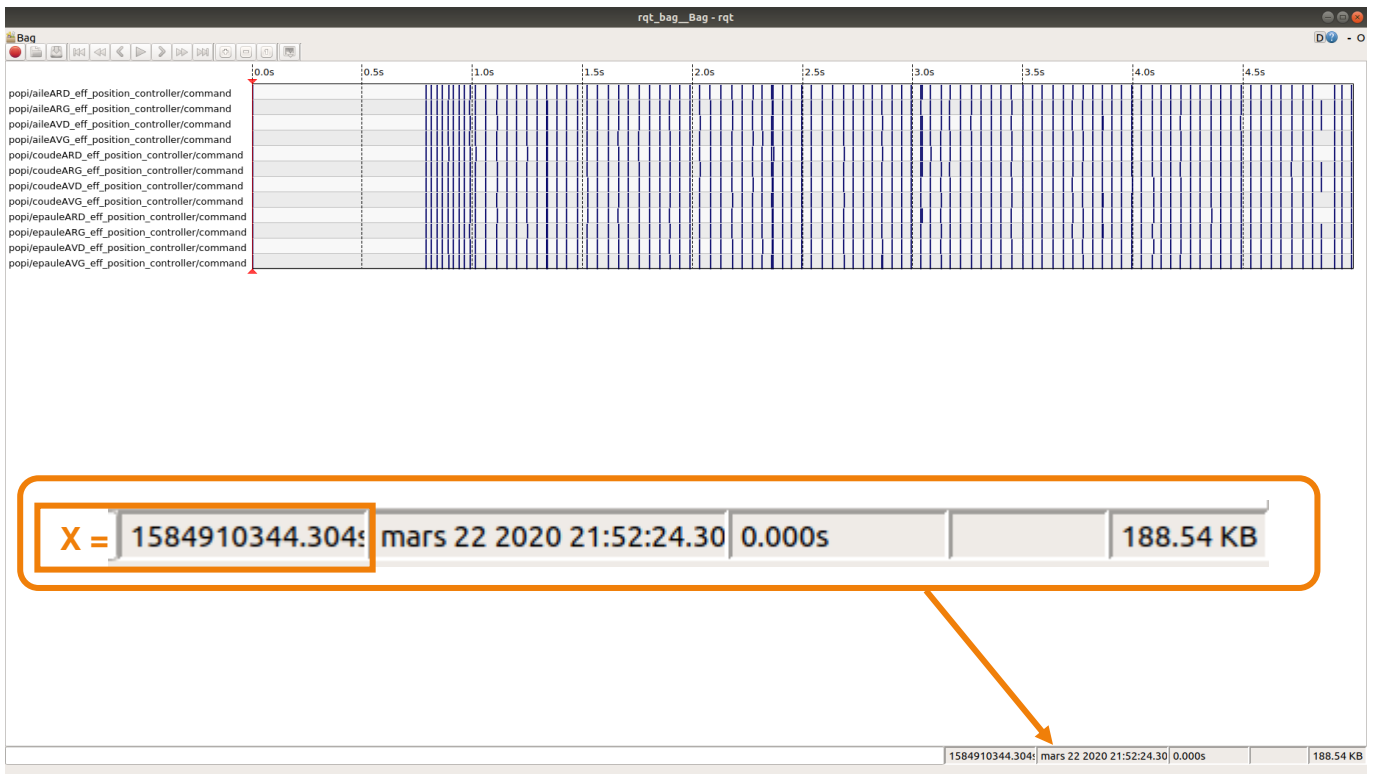


FIGURE 14 : FENETRE RQT_BAG PENDANT L'ENREGISTREMENT

4. Lorsque votre trajectoire est terminée, vous pouvez à nouveau cliquer sur le bouton rouge en haut à gauche pour mettre fin à l'enregistrement.

7.3.3 Modifier les commandes sur un tableur

1. Afin de modifier les commandes sur un tableur (e.g. Excel), il faut convertir le fichier *rosvbag* que nous avons créé en format csv. On ne peut le faire qu'un topic à la fois pour le moment, et on va donc commencer par créer 12 fichiers csv différents.

```
01. cd ~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/bags
02. rostopic echo -b ma_marche.bag -p
03. /pop_i/aileAVD_eff_position_controller/command > ../csv/aileAVD.csv
04. rostopic echo -b ma_marche.bag -p
05. /pop_i/aileAVG_eff_position_controller/command > ../csv/aileAVG.csv
06. rostopic echo -b ma_marche.bag -p
07. /pop_i/aileARD_eff_position_controller/command > ../csv/aileARD.csv
08. rostopic echo -b ma_marche.bag -p
09. /pop_i/aileARG_eff_position_controller/command > ../csv/aileARG.csv
10. rostopic echo -b ma_marche.bag -p
11. /pop_i/epauleAVD_eff_position_controller/command > ../csv/epauleAVD.csv
12. rostopic echo -b ma_marche.bag -p
13. /pop_i/epauleAVG_eff_position_controller/command > ../csv/epauleAVG.csv
14. rostopic echo -b ma_marche.bag -p
15. /pop_i/epauleARD_eff_position_controller/command > ../csv/epauleARD.csv
16. rostopic echo -b ma_marche.bag -p
17. /pop_i/epauleARG_eff_position_controller/command > ../csv/epauleARG.csv
18. rostopic echo -b ma_marche.bag -p
19. /pop_i/coudeAVD_eff_position_controller/command > ../csv/coudeAVD.csv
20. rostopic echo -b ma_marche.bag -p
21. /pop_i/coudeAVG_eff_position_controller/command > ../csv/coudeAVG.csv
22. rostopic echo -b ma_marche.bag -p
23. /pop_i/coudeARD_eff_position_controller/command > ../csv/coudeARD.csv
24. rostopic echo -b ma_marche.bag -p
25. /pop_i/coudeARG_eff_position_controller/command > ../csv/coudeARG.csv
```


2. Une fois les 12 fichiers csv, vous pouvez les importer sur votre tableur et les assembler (ils se trouvent dans le dossier `~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/csv`).
3. Les *timestamps* utilisés par `rqt_bag` sont particuliers : afin d'obtenir votre temps en secondes écoulées depuis le début de l'enregistrement, il faut effectuer le calcul suivant :

$$t = \frac{\%time}{10^9} - X$$

Où X est tel qu'indiqué en *Figure 14* sur **votre** rosbag.

4. Une fois vos modifications effectuées sur le tableur, il faut effectuer la conversion inverse. Pour ce faire, il faut d'abord organiser votre tableau ainsi : en première ligne, les titres de colonnes doivent être, dans l'ordre : `%time`, `aileAVD`, `aileAVG`, `aileARD`, `aileARG`, `epauleAVD`, `epauleAVG`, `epauleARD`, `epauleARG`, `coudeAVD`, `coudeAVG`, `coudeARD`, `coudeARG`.
5. Une fois le modèle respecté, enregistrez votre fichier, par exemple sous le nom « `ma_marche_modifiee.csv` ». Vérifiez en ouvrant ce fichier avec un éditeur de texte (e.g. *Notepad++*) que votre tableur a créé un format csv identique à ceux des fichiers initiaux (utilisant des « , » et pas des « ; », etc.)
6. Enfin, ouvrez un terminal et tapez les commandes suivantes :

```
01. cd ~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/csv
02. ./csv_to_rosbag.py ma_marche_modifiee
03. mv ma_marche_modifiee.bag ../bags/
```

7.4 Utilisation de la manette

Comme expliqué dans la partie 6.5, il est nécessaire de placer POPI en position nominale avant de pouvoir utiliser une trajectoire. Cela vaut pour la simulation comme pour l'envoi sur le robot. Pour ce faire, nous avons développé un programme se reposant sur une manette, qui offre également d'autres possibilités. La configuration de la manette se trouve en *Figure 15*.

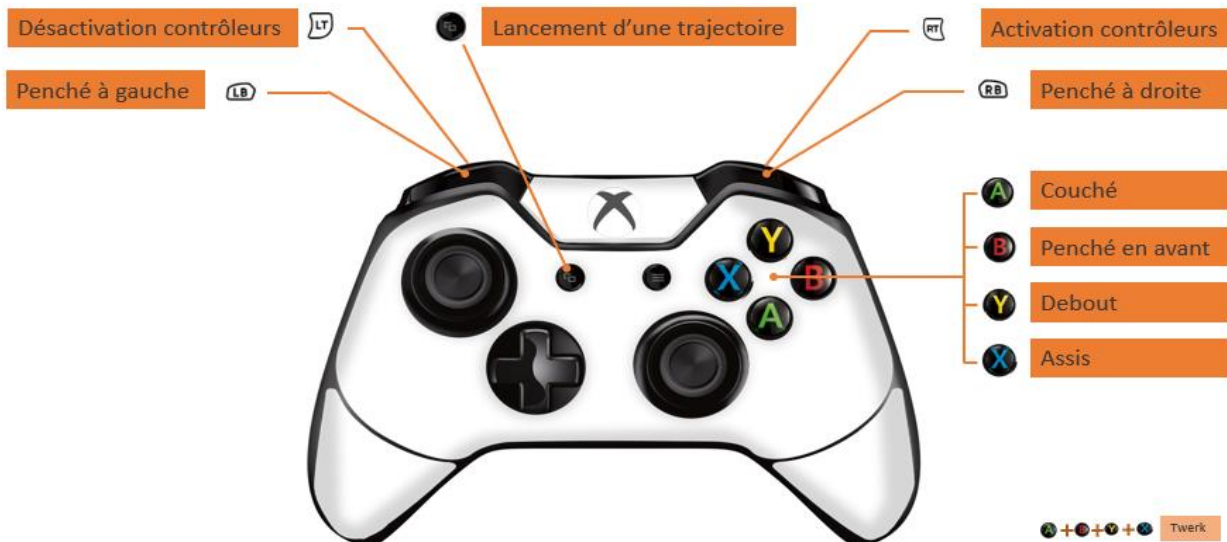


FIGURE 15 : CONFIGURATION DE LA MANETTE

En cas de problème d'utilisation de la manette, vous pourrez obtenir de plus amples renseignements sur la page du package `teleop_joy` que nous utilisons.

7.5 Simulation

Pensez à vérifier votre fichier `bashrc` comme indiqué en fin de partie 7.1.

Maintenant, il vous est possible de vérifier ce que donne la trajectoire créée sur *Gazebo*.

1. Ouvrez un terminal et entrez la commande suivante :

```
01. roslaunch popi_robot popi_simu.launch
```

2. Utilisez la touche  de la manette pour mettre POPI en position nominale.

3. Dans un nouveau terminal :

```
01. cd ~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/bags
02. rosbag play ma_marche.bag
```

7.6 Monitoring

Pour assurer le monitoring des données échangées entre les programmes, ouvrir un terminal et taper la commande :

```
01. rosrn rqt_gui rqt_gui
```

Une fenêtre qui contient les *Figure 16* et *Figure 17* s'ouvre :

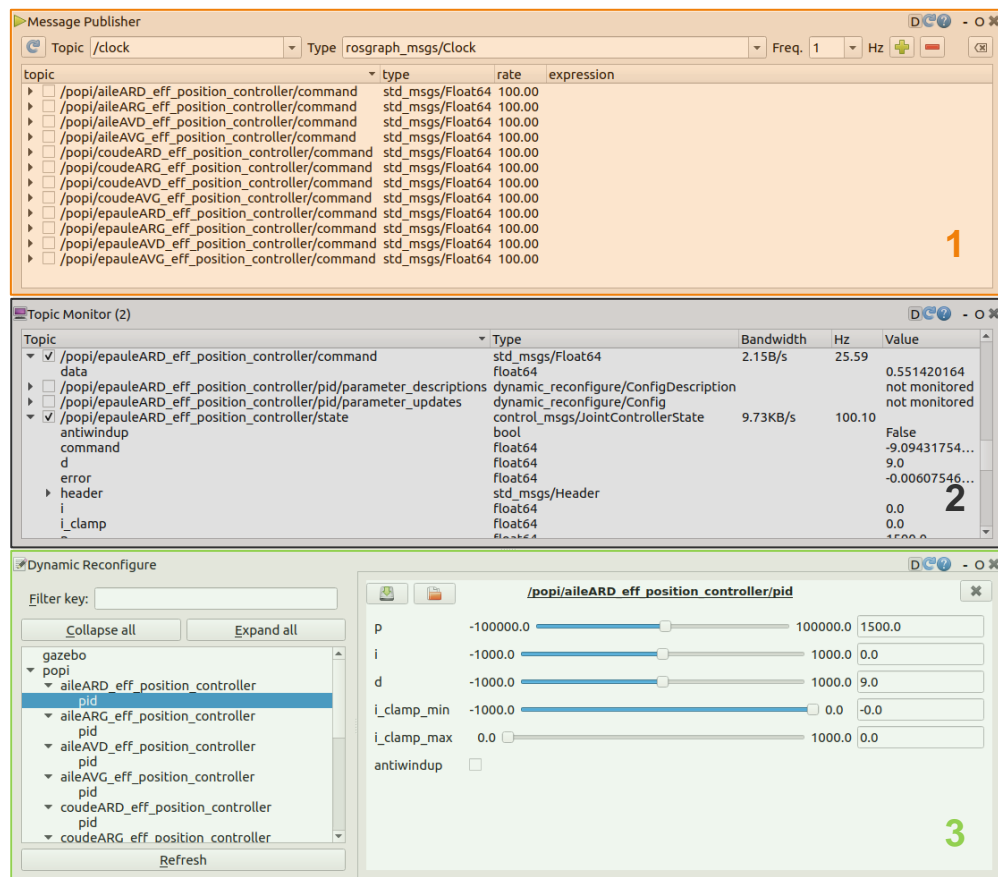


FIGURE 16 : INTERFACE RQT_GUI – MESSAGE PUBLISHER (1), TOPIC MONITOR (2) ET DYNAMIC RECONFIGURE (3)

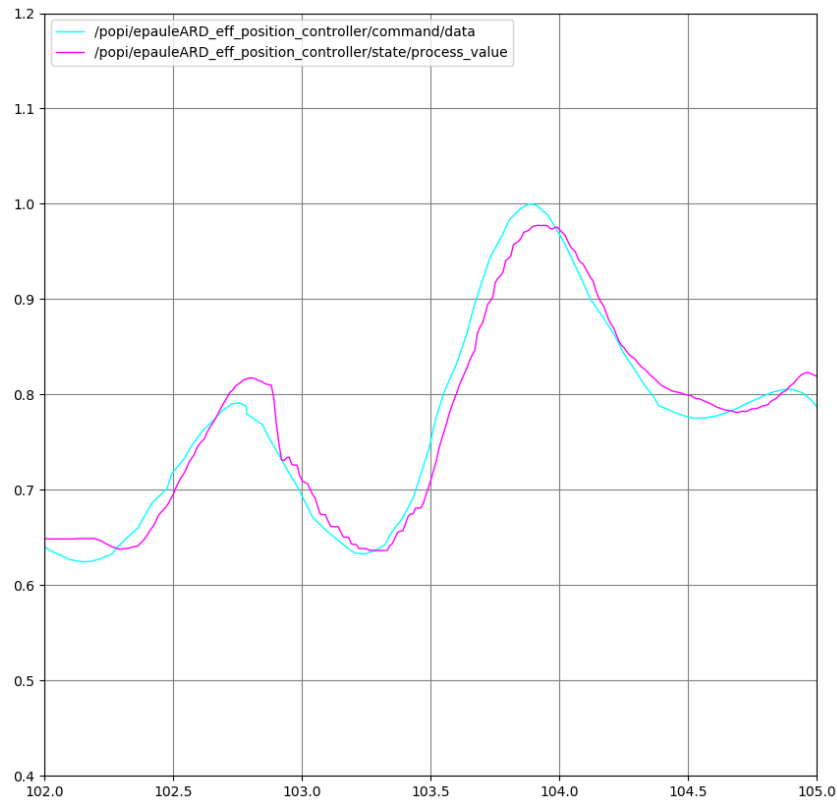


FIGURE 17 : INTERFACE RQT_PLOT

Explication des outils :

1. **Message publisher** : cet outil permet d'envoyer depuis l'interface un message, par exemple d'écrire la commande « 1.0 » en radian sur l'épaule arrière droite. On le retrouve dans *Plugins/Topics*
2. **Topic monitor** : cet outil permet de sélectionner les topics que l'on veut surveiller en « temps réel ». Il suffit de les cocher. On le retrouve également dans *Plugins/Topics*.
3. **Dynamic Reconfigure** : il autorise le changement dynamique des valeurs de notre système, dont notamment les valeurs des coefficients des PID. Il se trouve dans *Plugins/Configuration*.
4. Enfin, l'outil *plot* permet comme le *topic monitor* de surveiller des topics, mais avec un tracé graphique. On peut glisser/déposer depuis le *topic monitor* dans le *plot* les topics voulus.

Rqt_gui est un outil aux multiples possibilités. Le plus efficace est de le prendre en mains pour apprendre à s'en servir.

7.7 Envoi de trajectoire sur POPI

7.7.1 Mode opératoire de manutention

Si le robot est en position de repos, il peut être déplacé directement sur son socle à roulette. S'il n'est pas en mode de repos, le robot POPI doit être déplacé par deux utilisateurs avec les poignées fournies, en les positionnant comme en *Figure 18*.

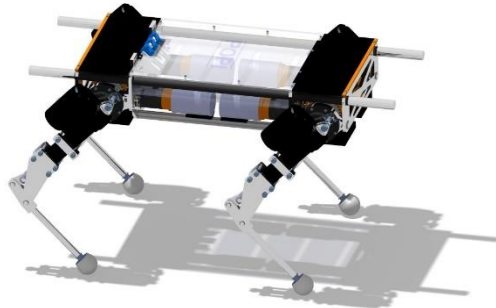


FIGURE 18 : UTILISATION DES POIGNEES DE MANUTENTION

7.7.2 Première connexion

Lors de la première connexion du PC utilisateur au câble Ethernet de l'ombilic, vous devrez effectuer certaines opérations afin de l'intégrer correctement au réseau de POPI.

PARAMETRAGE DE L'ADRESSE IP

Dans un premier temps il faut donner au PC l'adresse IP 192.168.7.4 :

1. Ouvrez les paramètres réseau et éditez la connexion filaire.

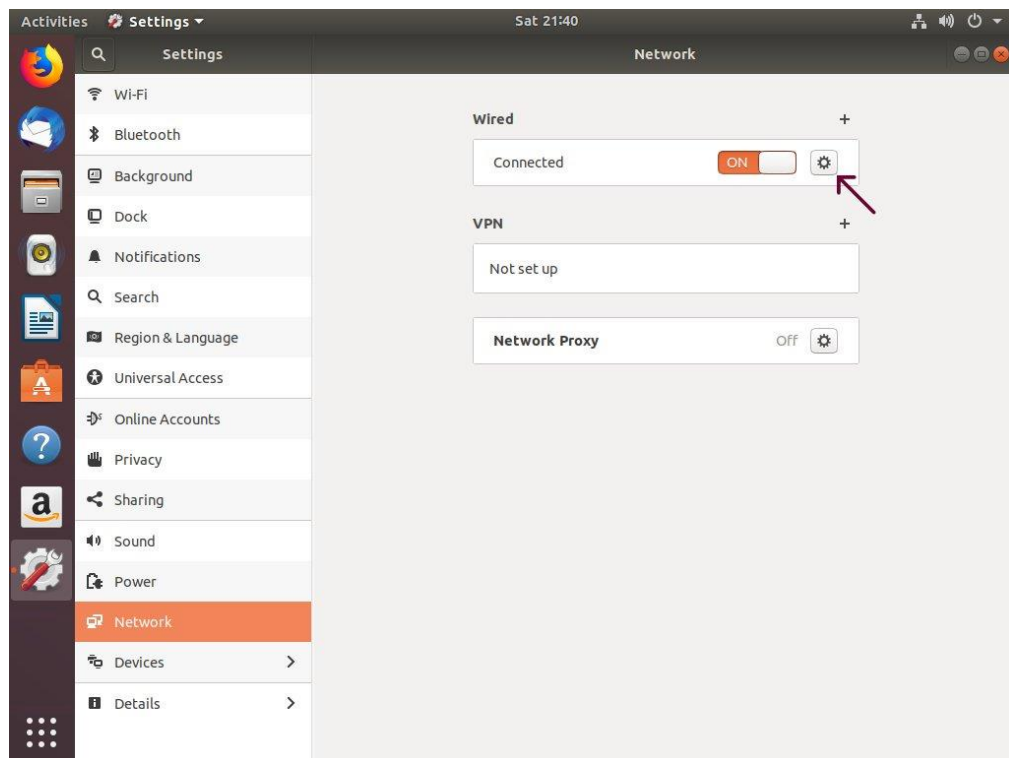


FIGURE 19 : PAGE DE PARAMETRAGE RESEAU

2. Dans l'onglet IPV4, sélectionnez la méthode manuelle et paramétrez la partie adresse comme indiqué sur l'image ci-dessous.

The screenshot shows the 'eth0' network configuration window. The 'IPv4 Method' is set to 'Manual'. The 'Addresses' table is highlighted with an orange oval and contains the following data:

| Address | Netmask | Gateway |
|-------------|---------------|-------------|
| 192.168.7.4 | 255.255.255.0 | 192.168.7.1 |
| | | |

Below the addresses, the 'DNS' and 'Routes' sections are visible, both with 'Automatic' toggles turned 'ON'. The 'Routes' section also has a table with columns for Address, Netmask, Gateway, and Metric.

FIGURE 20 : PAGE DE PARAMETRAGE DE LA CONNEXION FILAIRE

3. Avant de valider, vérifiez dans l'onglet détails que les cases suivantes sont bien cochées.

The screenshot shows the 'Details' tab of the network configuration window. The following options are checked:

- ☒ Connect automatically
- ☒ Make available to other users
- ☐ Restrict background data usage (Appropriate for connections that have data charges or limits.)

FIGURE 21 : ONGLET DETAIL DE LA PAGE DE PARAMETRAGES DE LA CONNEXION FILAIRE

AJOUT DES NOMS D'HOTES

Pour ne pas avoir à saisir à chaque fois les adresses IP, on utilise des noms d'hôtes qui permettent de se substituer à celles-ci.

1. Ouvrez le fichier /etc/hosts :

```
01. sudo subl /etc/hosts
```

2. Ajoutez les lignes suivantes à la fin du fichier.

```
01. 192.168.7.1 popi-raspi
02. 192.168.7.2 beagle-droite
03. 192.168.7.3 beagle-gauche
```

ROS_IP ET ROS_MASTER_URI

1. Ouvrez le fichier .bashrc :

```
01. sudo subl ~/.bashrc
```

2. Décommentez les deux lignes suivantes (supprimer le #), puis sauvegardez le fichier :

```
#export ROS_IP=192.168.7.4
```

```
#export ROS_MASTER_URI=http://192.168.7.1:11311
```

PARAMETRAGE DES CONNEXIONS SSH

1. Pour commencer, il faut ouvrir un serveur SSH sur le PC :

```
01. sudo apt-get update
02. sudo apt-get install openssh-server
```

Le serveur SSH s'active automatiquement.

Il faut maintenant établir une première connexion SSH aller et retour avec toutes les cartes :

2. Établir la connexion depuis votre PC vers la Raspberry PI :

```
01. ssh ubuntu@popi-raspi -oHostKeyAlgorithms='ssh-rsa'
```

Le mot de passe de la raspberry est : ubuntu

3. Établir la connexion depuis la Raspberry vers votre PC :

```
01. ssh user@pc -oHostKeyAlgorithms='ssh-rsa'
```

Remplacer « user » par votre nom d'utilisateur. Le mot de passe à saisir est celui de votre PC.

4. Établir la connexion depuis votre PC vers la BeagleBoneBlack droite :

```
01. ssh debian@beagle-droite -oHostKeyAlgorithms='ssh-rsa'
```

Le mot de passe des beaglebone est : temppwd

5. Établir la connexion depuis la BeagleBoneBlack droite vers votre PC :

```
01. ssh user@pc -oHostKeyAlgorithms='ssh-rsa'
```

Remplacer « user » par votre nom d'utilisateur. Le mot de passe à saisir est celui de votre PC.

6. Établir la connexion depuis votre PC vers la BeagleBoneBlack gauche :

```
01. ssh debian@beagle-gauche -oHostKeyAlgorithms='ssh-rsa'
```

Le mot de passe est : temppwd

7. Établir la connexion depuis la BeagleBoneBlack gauche vers votre PC :

```
01. ssh user@pc -oHostKeyAlgorithms='ssh-rsa'
```

Remplacer « user » par votre nom d'utilisateur. Le mot de passe à saisir est celui de votre PC.

7.7.3 Utilisation des trajectoires

Merci de reprendre connaissance des mises en garde données en partie 6.5.

1. Commencez par placer manuellement les épaules à leur « zéro » initial, autrement dit à l'horizontale. L'utilisation d'un niveau à bulles est recommandée. Cette étape n'est pas facultative : lors du lancement de la commande suivante, le programme enregistre la position actuelle des épaules comme leur « zéro ». Si elles n'y sont pas réellement, toutes les commandes envoyées seront interprétées avec le décalage de l'initialisation ! Dans l'idéal, POPI devrait être dans une position proche de celle en *Figure 22*.



FIGURE 22 : POPI EN POSITION REPOS

2. Ouvrir un terminal et entrez la commande suivante :

```
01. roslaunch popi_robot beagles.launch
```

3. Dans un nouveau terminal, entrez la commande :


```
01. ssh ubuntu@192.168.7.1
```


Une fois connecté sur la Raspberry Pi (mot de passe *ubuntu*), tapez :

```
01. roslaunch popi_robot hw_interface.launch
```

4. Dans un troisième terminal:

```
01. roslaunch popi_robot popi_reel.launch
```

5. Utilisez la touche  de la manette pour vous assurer que les commandes envoyées soient celles de la position de repos.
6. Allumez le 24V en tournant le commutateur sur la platine d'alimentation (voir *Figure 5*).

7. Utilisez la touche  de la manette pour démarrer les contrôleurs. Vous pouvez vérifier dans l'onglet *Dynamic Reconfigure* de *rqt_gui* que les valeurs des coefficients « P » des contrôleurs ne sont plus à 0.



Lors de cette étape, s'il y avait un écart entre la position actuelle de POPI et les dernières commandes envoyées au robot, les actionneurs vont bouger. Pour ne pas entraîner de mouvements brusques des actionneurs, nous privilégions des commandes de type rampes. Cet écart peut au contraire résulter en échelon important ! Soyez donc vigilants.


8. Utilisez la touche  de la manette pour mettre POPI en position nominale, comme en *Figure 23*.




FIGURE 23 : POPI EN POSITION NOMINALE

9. Retirez le socle.

10. Dans un quatrième terminal, envoyez la trajectoire voulue :

```
01. cd ~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/bags
02. rosbag play ma_marche.bag
```

En cas de problème, vous pouvez immédiatement mettre fin à une marche en appuyant sur la touche  de la manette. Cela a pour effet de remettre tous les coefficients « P » des contrôleurs à 0, ce qui implique que les moteurs ne reçoivent plus aucune commande.



Attention, si le robot n'est pas dans une position équilibrée, cette action présente un risque de chute ! Dans le cas contraire, POPI tient mécaniquement sur ses quatre pattes sans avoir besoin de lui envoyer des commandes.

11. Les trajectoires générées par *Towr* ne finissent généralement pas en position nominale.

Pensez à repositionner POPI (voir étape 8) pour assurer son bon équilibre.

Lorsque la trajectoire envoyée est terminée, vous pouvez immédiatement en envoyer une autre sans refaire l'ensemble des étapes. Gardez néanmoins à l'esprit qu'une utilisation intensive des vérins peut provoquer leur détérioration du fait de leur *duty cycle* limité (voir *Tableau 4*).

7.7.4 Trajectoires mises à disposition

Cinq trajectoires sont mises à disposition par l'équipe. Pour y accéder, vous pouvez vous rendre dans le dossier `~/catkin_ws/src/pop_i_ros/pop_i/pop_i_robot/trajectories/bags`. Ces trajectoires sont les suivantes :

- *short_static_walking.bag* : POPI avance en marche statique sur sol plat, sur une distance de 80cm.
- *short_dynamic_walking.bag* : POPI avance en marche dynamique sur sol plat, sur une distance de 80cm.
- *long_dynamic_walking.bag* : POPI avance en marche statique sur sol plat, sur une distance de 4m.
- *step_20cm.bag* : POPI franchit, en marche statique, une marche de 20cm de haut.
- *pit_30cm.bag* : POPI franchit, en marche statique, une fosse de 30cm de large.

Il est possible d'obtenir un aperçu vidéo de la configuration des terrains et des trajectoires réalisées sur le [GitHub](#) du projet.



Les deux dernières trajectoires n'ont jamais été testées en réalité. L'équipe POPI se dégage de toute responsabilité en cas de dégâts survenus suite à l'utilisation d'une de ces trajectoires ou d'une autre trajectoire qui n'a pas été validée par ses soins.


7.7.5 Mise en position repos

Pour pouvoir mettre POPI en position repos, il faut que le robot soit alimenté et qu'il soit en l'état de recevoir des commandes. Lorsque vous n'utilisez pas POPI, merci de le replacer en position de repos sur son socle :

1. Positionnez le socle à roulette en-dessous du robot comme sur la *Figure 24*.



FIGURE 24 : POSITIONNEMENT DU SOCLE

2. Utilisez la touche  de la manette pour mettre POPI en position de repos sur son socle. Il est conseillé que 2 manutentionnaires se munissent des poignées pour accompagner le mouvement du robot.
3. Une fois le robot sur son socle, vous pouvez fermer les processus en cours dans vos différents terminaux (**CTRL+C**) dans l'ordre inverse de celui dans lequel vous les avez ouverts. Enfin, vous pouvez couper l'alimentation.

8 Sécurité

8.1 Prescription générale de sécurité

Cette notice d'instructions ne prend en compte que les comportements raisonnablement prévisibles. Le produit POPI est conçu et réalisé en considérant toujours la sécurité de l'opérateur, toute utilisation autre que celle indiquée sur la présente notice d'instructions peut constituer un danger.



Suivre les précautions suivantes :

- Avant toute utilisation, vérifier que la machine est montée correctement dans son ensemble.
- Conserver une zone de travail propre et ordonnée.
- Positionner la machine dans une zone de travail suffisamment éclairée.
- Ne pas endommager le câble d'alimentation électrique.
- Ne jamais tirer sur le câble d'alimentation électrique pour le retirer de la prise de courant.
- Maintenir le câble d'alimentation électrique éloigné des sources de chaleur, des parties grasses et/ou des bords tranchants.
- Protéger le câble d'alimentation électrique contre l'humidité et tous risques éventuels de dégradations.
- Vérifier périodiquement le câble d'alimentation électrique et s'il est endommagé, le faire réparer par un réparateur agréé.
- Les protections mécaniques et/ou électriques ne doivent pas être enlevées ou shuntées.
- Aucune modification et/ou reconversion ne doit être effectuée.

- Seul le personnel compétent et qualifié est autorisé à réparer ou remplacer les pièces endommagées.
- Ne pas toucher le robot en mouvement.
- Pour toute opération présentant des risques de pincement, happement, enroulement, écrasement, arrêter la machine et porter des gants de protection.
- Déconnecter l'alimentation électrique pour toute opération plus importante (maintenance, entretien ...).
- Ne pas nettoyer le robot ou le système embarqué lorsqu'ils sont en mouvement.
- Ne pas laver la machine avec un jet d'eau sous pression, car risque de faire pénétrer celle-ci dans la partie électrique.
- Ne pas utiliser de solvant ou de détergents agressifs.
- Couper l'alimentation et vérifier que les parties mobiles sont bloquées, lors du déplacement.
- Ne pas utiliser la machine sans une formation suffisante.

8.2 Transport

Le transport et le déplacement du robot, lorsqu'il est à l'arrêt, doivent se faire sur son support à roulette ; le robot doit être en position de repos, les vérins doivent être sortis de façon à ce que l'angle formé par les pattes fasse 15°.

Si le robot n'est pas positionné sur son socle, le robot POPI peut être déplacé grâce aux poignées fournies.



Le robot pèse 70kg, il ne peut pas être transporté par une personne seule. Il faut au minimum 2 personnes pour le déplacer.

8.3 Installation / Désinstallation

Lors de l'installation du robot dans son environnement, commencer par vérifier l'état général du produit. Lors de la réception, le produit POPI est sur son socle de repos. Le produit POPI doit être démarré dans cette position, et remis dans cette position à chaque fin d'utilisation.

Pour cela, il faut que le robot POPI soit relié à sa platine d'alimentation.

Les opérations électriques doivent être effectuées par un personnel qualifié et habilité aux travaux électriques basse tension. S'assurer que la tension d'alimentation de l'installation correspond à celle nécessaire par la platine d'alimentation. Contrôler que l'installation électrique sur laquelle le robot POPI est branché est bien reliée à la terre conformément aux normes de sécurité en vigueur.

Connecter les câbles d'alimentation du produit POPI au tableau d'alimentation



L'usage de l'appareil avec un câble endommagé est rigoureusement interdit. Vérifier régulièrement l'état des câbles d'alimentation.

8.4 Maintenance



Avant toute opération de maintenance ou d'entretien, déconnecter l'alimentation électrique. Débrancher l'ombilic et attendre 45 secondes afin de s'assurer que le produit POPI est complètement hors tension. Désactiver toute autre source d'énergie connectée au robot et à la platine d'alimentation. Prendre les précautions nécessaires pour empêcher que d'autres personnes activent le système pendant la période de réparation.

Vérifier la connexion de terre avant de remettre le système sous tension.



N'utilisez jamais d'air comprimé pour nettoyer le robot, car cela pourrait endommager les composants électriques.

INSTRUCTION DE MAINTENANCE

L'acquisition de deux pattes supplémentaires du robot quadrupède POPI est vivement conseillée afin de pallier rapidement les problèmes mécaniques pouvant survenir.

L'acquisition d'un double des cartes électroniques du robot quadrupède POPI est vivement conseillée, il n'est pas garanti que les cartes électroniques soient toujours produites à l'avenir, des versions supérieures pourront être installées sous réserve de leurs intégrations.

8.4.1 Instructions de maintenance préventive

Le prototype de robot quadrupède POPI a une durée de fonctionnement de 500h. Afin d'atteindre et de prolonger cette limite, veuillez effectuer les actions de maintenance préventive ci-dessous :

Note : Certaines actions ci-dessous nécessitent le démontage/montage du robot. Pour se faire, se référer à la notice de montage/démontage.

Toutes les 5h de fonctionnement :

| Libellé | Fréquence | Détail |
|-----------------------|--------------------------------|---|
| Vérification visuelle | Toute les 5h de fonctionnement | Vérification visuelle du robot par un utilisateur du robot. |

Toutes les 50h de fonctionnement :

| Libellé | Fréquence | Détail |
|----------------|----------------------------------|--|
| Graissage | Toutes les 50h de fonctionnement | Démontage intégral du robot et remontage avec graissage des liaisons avec un produit adapté aux métaux (ex. : WD40). |
| Serrage | Toutes les 50h de fonctionnement | Vérification des couples de serrages à la clef dynamométrique. Resserrage si nécessaire. |
| Électronique 1 | Toutes les 50h de fonctionnement | Vérification des tensions des cartes électroniques et du bruit. |

Toutes les 500h de fonctionnement :

| Libellé | Fréquence | Détail |
|----------------|---------------------------------|---|
| Électronique 2 | Tous les 500h de fonctionnement | Changement de toutes les cartes électroniques. |
| Vernissage | Tous les 500h de fonctionnement | Démontage intégral du robot. Vernissage de toutes les pièces métalliques avec un vernis incolore mat (ou équivalent). |
| Mécanique | Tous les 500h de fonctionnement | Changement des axes de liaisons pivots et changement des clavettes moteurs. |

8.4.2 Instructions de maintenance corrective

Maintenance corrective des pièces mécaniques :

Dans le cas d'un défaut mécanique sur l'une des pièces. Veuillez faire usiner la pièce aux côtes spécifiées dans les plans mécaniques. Pour les pièces imprimées en 3D, veuillez les réimprimer avec une imprimante frittage (ou équivalent).

Maintenance corrective de l'électronique :

Dans le cas d'un défaut sur l'une des cartes électroniques de puissance ou de commande, veuillez la remplacer.

Si besoin, les images des cartes SD des BeagleBoneBlack et de la Raspberry PI sont disponibles sur le [GitHub](#) du projet et peuvent être flashées sur une nouvelle carte (prévoir au moins 16 Go) à l'aide par exemple du logiciel [BalenaEtcher](#).

Lors du remplacement de la Raspberry PI, une manipulation réseau est nécessaire pour assurer le bon fonctionnement du pont réseau br0. En effet, ce sont les adresses physiques des ports qui sont utilisées pour identifier les interfaces à relier en pont, et ces adresses physiques changent avec la carte. Pour ajouter les nouvelles adresses dans le pont, il faut connecter un écran sur le port HDMI de la Raspberry PI et redémarrer la carte. Dans la partie droite de la barre des tâches cliquer sur l'icône réseau puis sur « connections information ». Cette fenêtre permet d'identifier l'adresse physique des ports connectés aux adresses IP 192.168.7.2, 192.168.7.3 et 192.168.7.4. Il faut ensuite à nouveau cliquer sur l'icône réseau et ouvrir la fenêtre « edit connections » puis double cliquer sur le pont br0 (tout en bas) afin de l'éditer. Il faut alors s'assurer que les interfaces identifiées précédemment apparaissent bien sous le pont.

Si ce n'est pas le cas il faut les ajouter de la manière suivante :

1. Cliquer sur « add » pour ajouter une connexion.

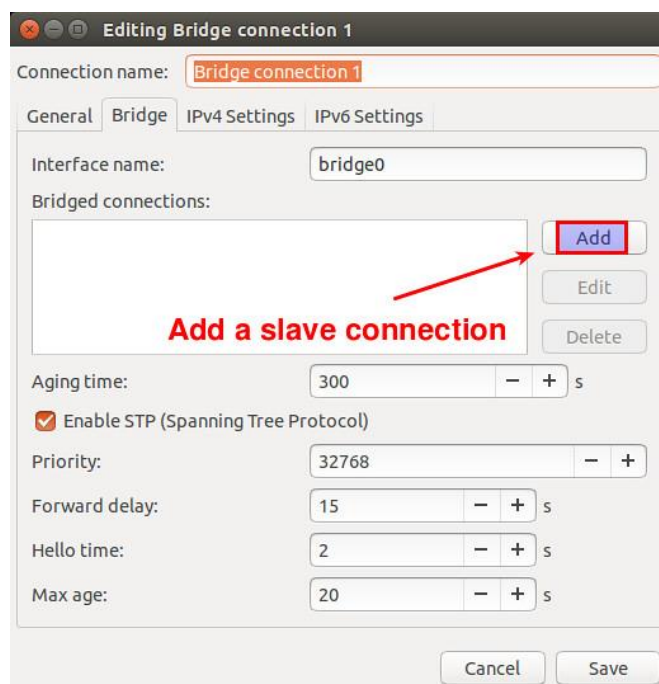


FIGURE 25 : AJOUT D'UNE CONNEXION PAR PONT.

2. Sélectionner le type de connexion « Ethernet ».
3. Sélectionner l'interface à ajouter dans le pont.

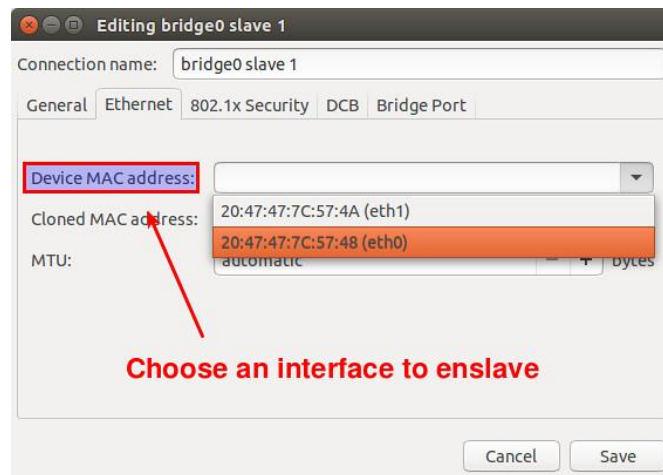


FIGURE 26 : SELECTION DE L'INTERFACE A ASSERVIR

4. Sélectionner l'onglet « general » et vérifier que les cases suivantes sont bien cochées puis cliquer sur « save ».

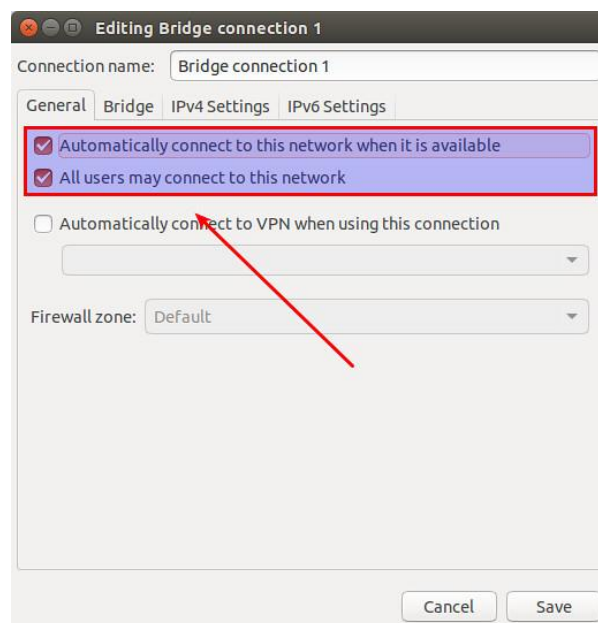
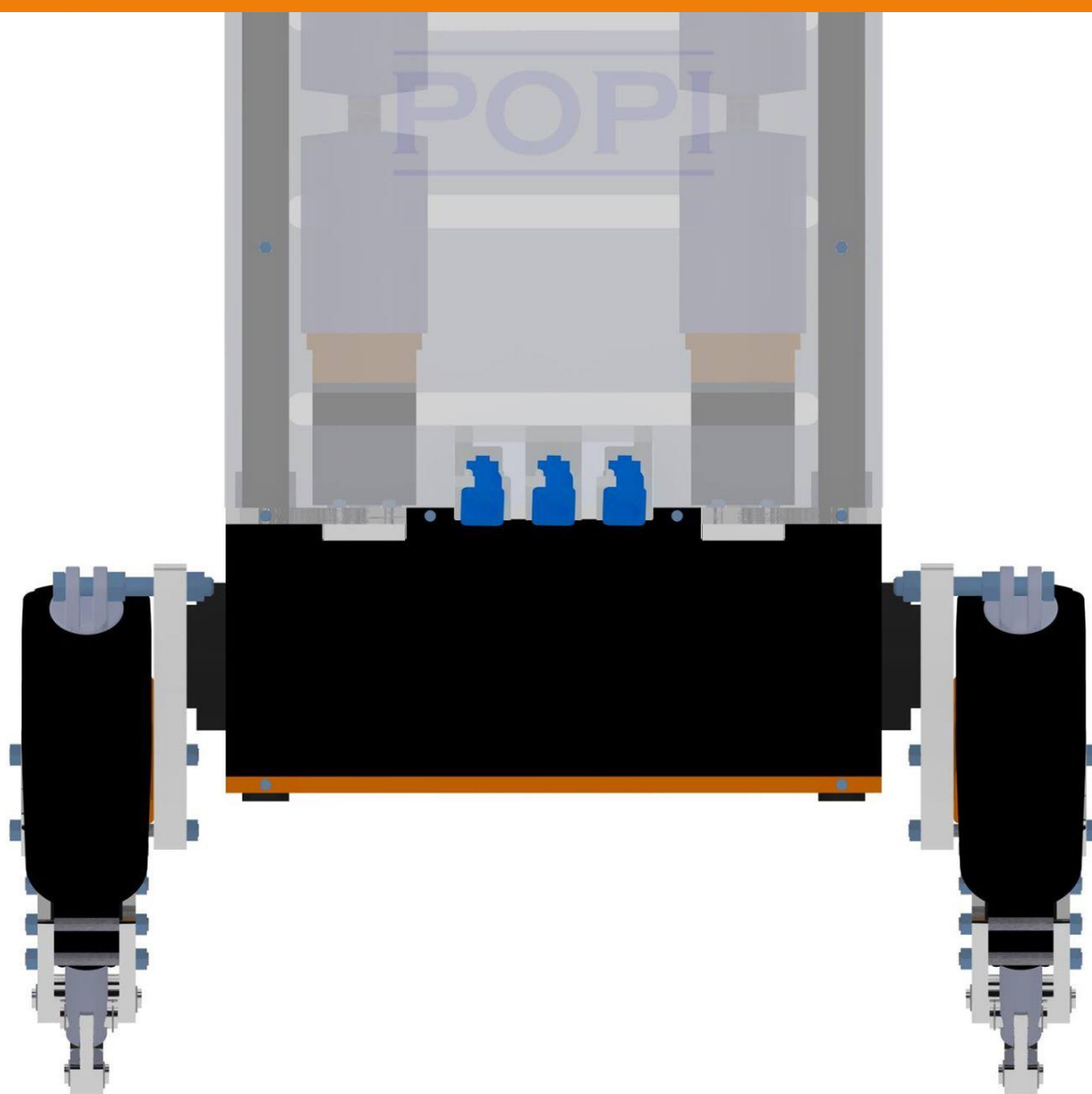


FIGURE 27 : VERIFICATION DE PARAMETRES RESEAUX

Afin de vérifier que les connexions sont rétablies, un test de ping pourra être effectué depuis le PC vers la Raspberry PI puis vers les deux BeagleBoneBlack.

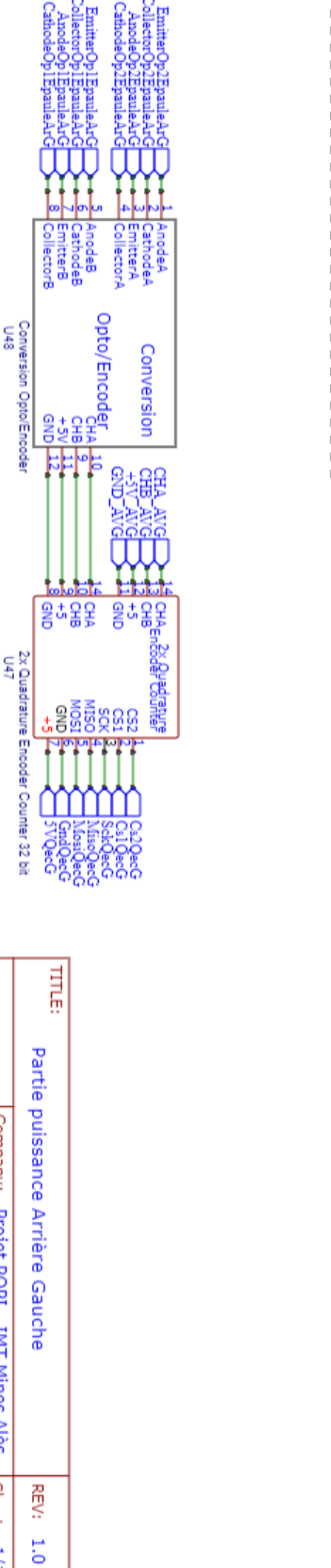
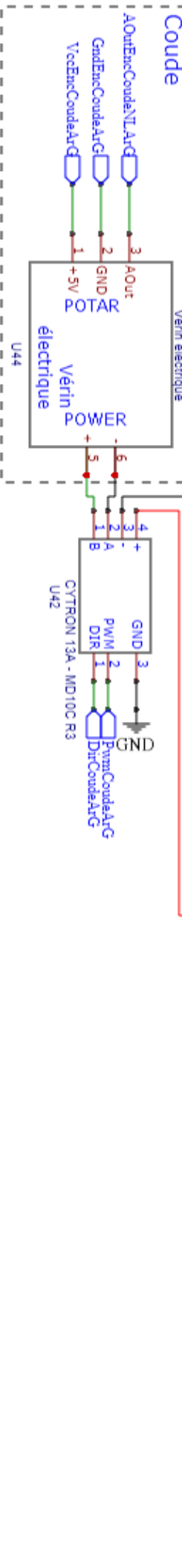
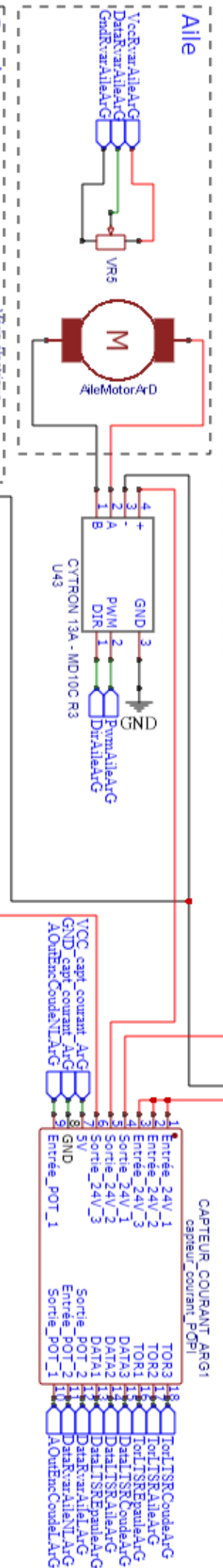
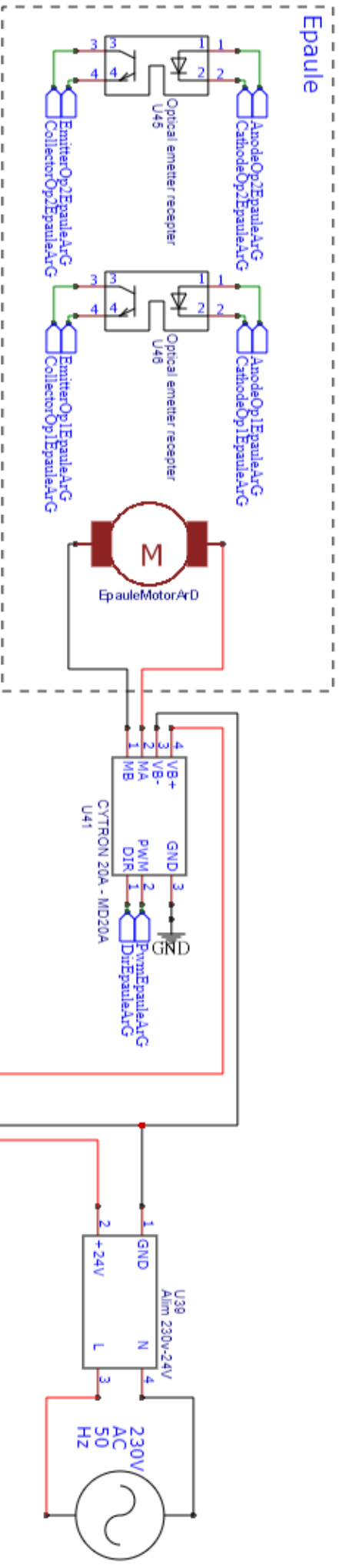


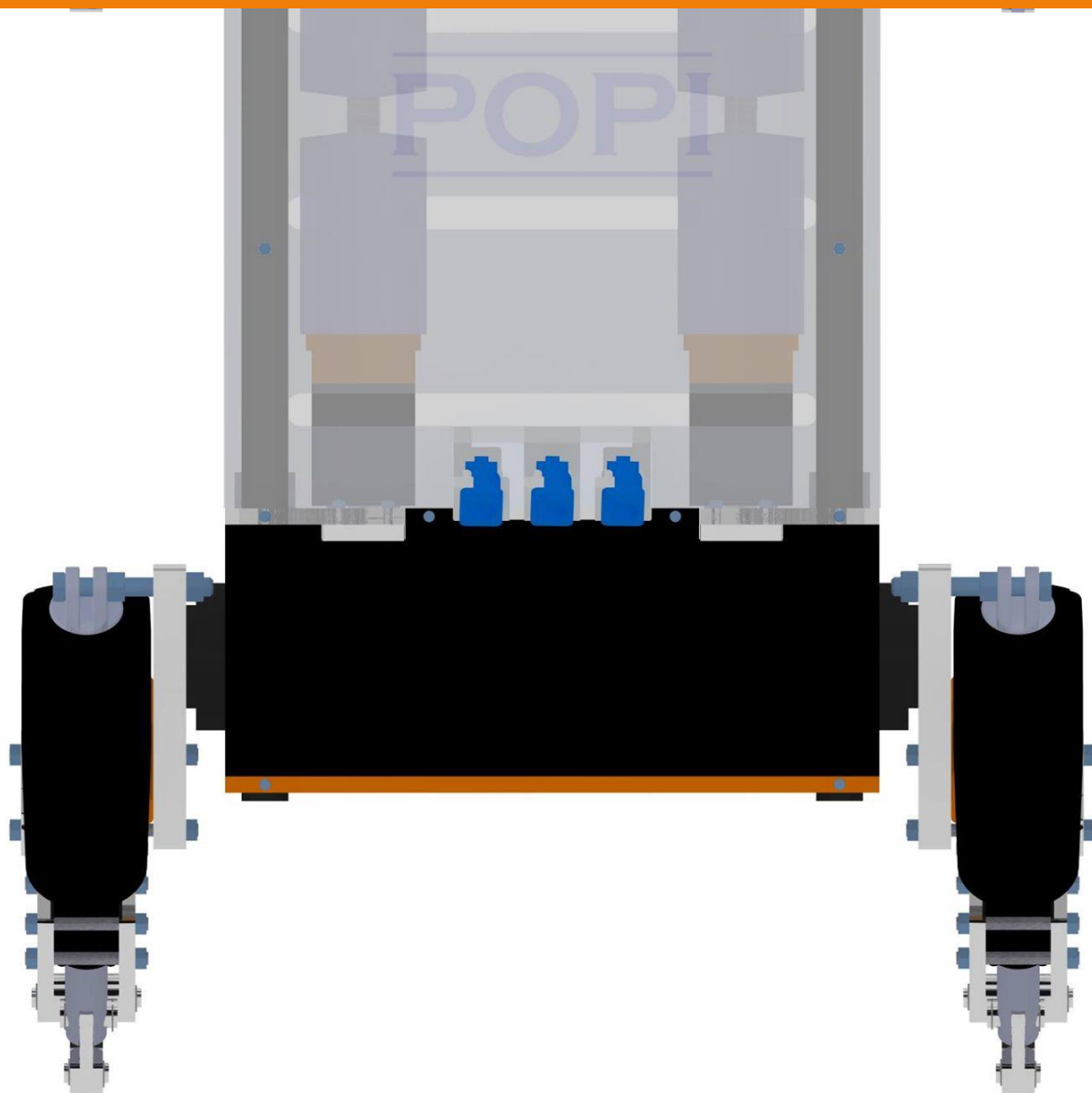
POPI

Annexe 1 : Plans de montage électronique

| Qté | Nom | N° |
|-----|--|----|
| | Partie intelligence | |
| 1 | Raspberry pi 3B+ | 1 |
| 2 | BeagleBoneBlack | 2 |
| | Partie distributions | |
| 4 | Contrôleur moteur Cytron 24V 20A | 3 |
| 8 | Contrôleur moteur Cytron 24V 13A | 4 |
| 2 | Bornier de distribution 1.8V, 5V et Ground | 5 |
| 2 | Bornier 24V | 6 |
| 1 | Bornier 230V | 7 |
| 1 | HUB USB 60W | 8 |
| 2 | Alimentation 24V 40A 1000W | 9 |
| 1 | Boitier arrêt d'urgence | 10 |
| 2 | Disjoncteur de type courbe C | 11 |
| | Partie capteurs | |
| 4 | Potentiomètres 1k Ω | 12 |
| 4 | Carte capteurs courants | 13 |
| 4 | Capteur de positions vérin | 14 |
| 4 | Carte roues codeuses | 15 |
| 2 | Quadrature encoder counter | 16 |
| 1 | Centrale inertielle | 17 |
| | Partie actionneurs | 18 |
| 4 | Moteurs MY1016 480W | 19 |
| 4 | Moteurs MR997 300W | 20 |
| 4 | Vérins MA2 300W | 21 |

TABLEAU 8 : NOMENCLATURE DES COMPOSANTS ELECTRONIQUE



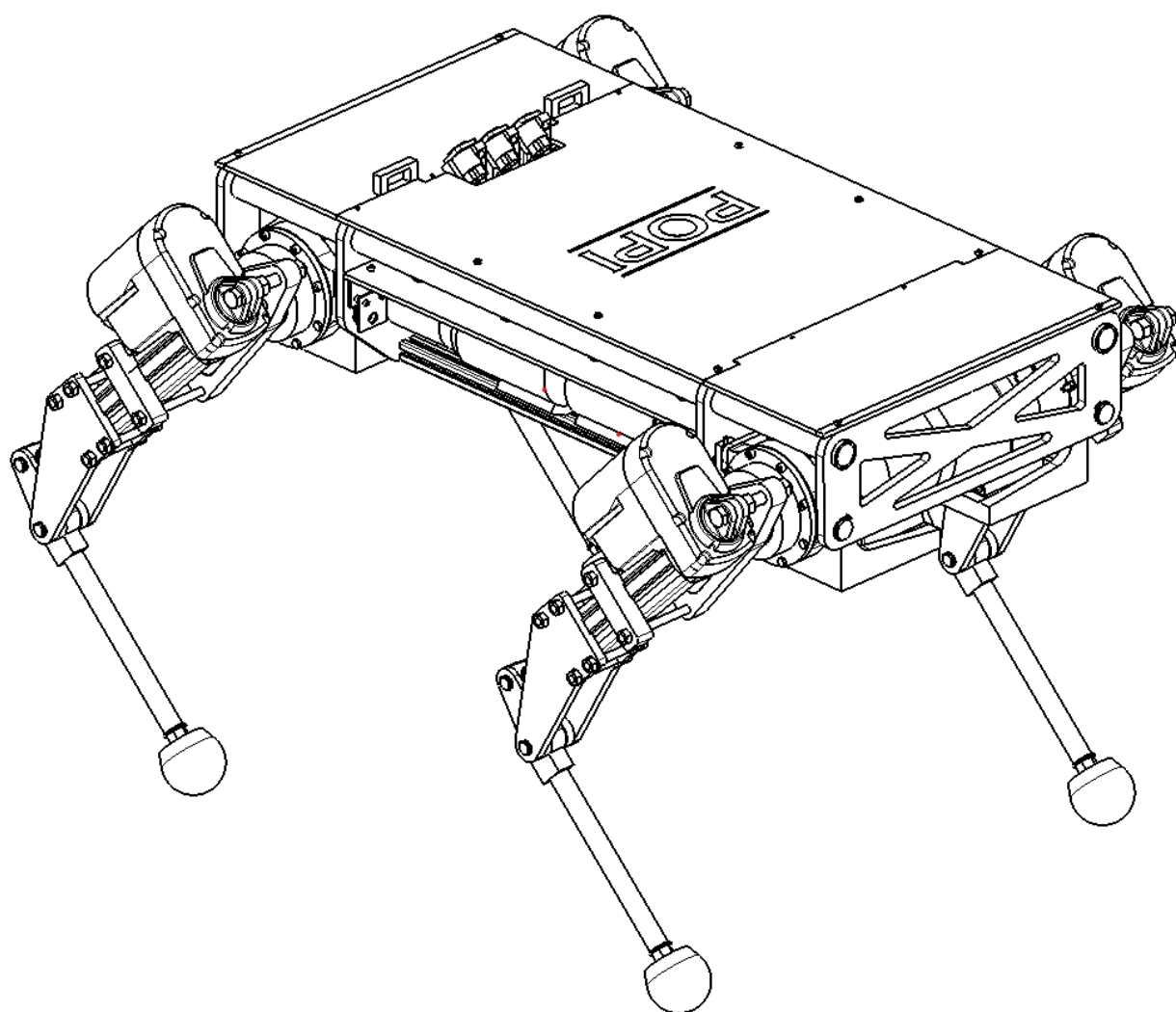


POPI

Annexe 2 : Notice de montage mécanique

POPI

NOTICE DE MONTAGE



MKX 3

PIECES

| N° | Pièce | Quantité |
|----------|--|----------|
| 1 | Rail Elcom | 2 |
| 2 | Fixations Elcom | 4 |
| 3 | Support Moteur Aile | 4 |
| 4 | Plaque Verticale Interne | 2 |
| 5 | Tube Structure | 2 |
| 6 | Moteur Aile - <i>MR977 63 1/96</i> | 4 |
| 7 | Clavette Moteur Aile | 4 |
| 8 | Porte Réducteur | 4 |
| 9 | Biellette Aile | 4 |
| 10 | Liaison Moteur Aile | 4 |
| 11 | Plaque Verticale Extérieure | 2 |
| 12/13 | Porte Vérin Haut A / <i>Symétrique</i> | 2+2 |
| 14/15 | Porte Vérin Haut B / <i>Symétrique</i> | 2+2 |
| 16 | Réducteur – <i>PLE060</i> | 4 |
| 17 | Vérin – <i>TiMOTION MA2</i> | 4 |
| 18/19 | Porte Vérin Haut C / <i>Symétrique</i> | 2+2 |
| 20 | Pied | 4 |
| 21 | Tube Patte | 4 |
| 22 | Double Pivot | 4 |
| 23 | Biellette Coude | 4 |
| 24/25 | Porte Vérin Bas / <i>Symétrique</i> | 4+4 |
| 26 | Entretoise Coude | 8 |
| 27 | Cale Réducteur | 4 |
| 28 | Porte Moteur Épaule | 4 |
| 29 | Moteur Épaule – <i>MY1016 350W</i> | 4 |
| 30 | Œillet Moteur | 4 |
| 31 | Roue Codeuse | 4 |
| 32 | Bague Réducteur | 4 |
| 33 | Cale Bague Réducteur | 4 |
| 34 | Support Capteur Épaule | 4 |
| 35 | Plaque Électronique | 1 |
| 36 | Équerre | 4 |
| 37 | Entretoise Potentiomètre | 8 |
| 38 | Support Potentiomètre | 4 |
| 39/40/41 | Bornier Type A/B/C | 1+1+1 |
| 42 | Maintien Bornier | 3 |
| 43 | Accroche Ombilic | 2 |
| 44 | Plaque de Protection | 1 |
| 45/46 | Cache Moteurs | 1+1 |
| 47 | Poignée | 4 |

VISSERIE

| N° | Type | Quantité |
|-----|---------------------------------------|----------|
| 101 | M3x10 Cruciform | 8 |
| 102 | M5x16 Slotted Cheese Head | 20 |
| 103 | M4x10 Hex Socket Set with Flat Point | 20 |
| 104 | M6x25 Hexagon Socket Countersunk Head | 28 |
| 105 | M12x90 Hexagon Head | 4 |
| 106 | M8x115 Threaded Rod | 8 |
| 107 | M8x75 Threaded Rod | 16 |
| 108 | M5x40 Hexagon Socket Head Cap | 16 |
| 109 | M5x60 Hexagon Head | 16 |
| 110 | M4x16 Hexagon Socket Head Cap Special | 4 |
| 111 | M6x35 Hexagon Socket Head Cap | 16 |
| 112 | M3x25 Hexagon Socket Head Cap | 8 |
| 113 | M5x25 Hexagon Socket Head Cap | 4 |
| 114 | M5x50 Hexagon Socket Head Cap | 3 |
| 115 | M4x14 Hexagon Socket Head Cap | 4 |
| 116 | M4x16 Hexagon Socket Head Cap | 16 |

Prévoir un nombre important d'écrous et rondelles M4/M5/M6/M8 en conséquence ainsi que 8 écrous et rondelles M12. Respecter idéalement les couples de serrage de la norme E25-030.

AXES

| N° | Type | Quantité |
|-----|---|----------|
| 201 | Stub D10 L20 | 4 |
| 202 | Stub D10 L20 taraudé M5 | 4 |
| 203 | Axe D20 L29.5 épaulement + rainure circlips | 4 |
| 204 | Axe D20 L28 épaulement + rainure circlips | 4 |
| 205 | Axe D12 L27.2 + 2 rainures circlips | 4 |
| 206 | Axe D12.8 L35 + 2 rainures circlips | 4 |
| 207 | Axe D12 L66.8 + 2 rainures circlips | 4 |

CIRCLIPS

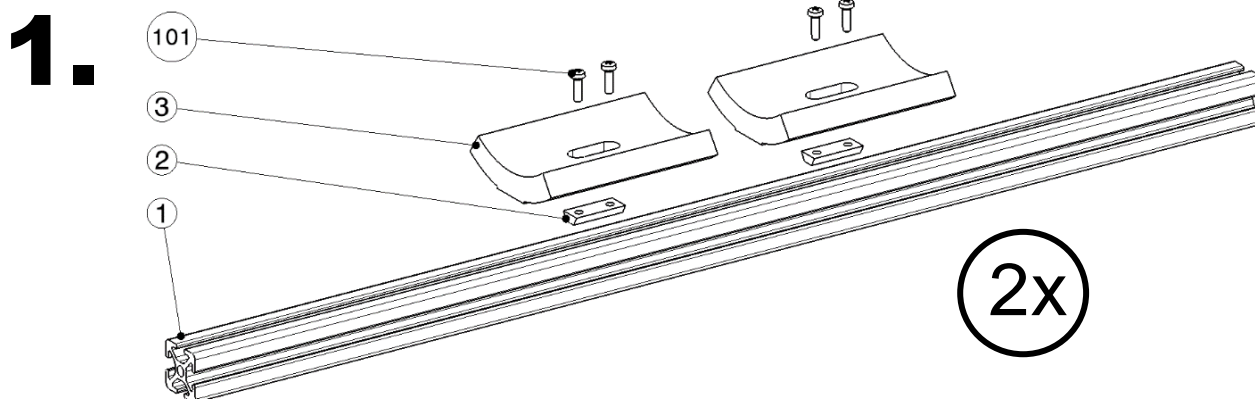
| N° | Type | Quantité |
|-----|-----------|----------|
| 301 | D3=20 Ep1 | 8 |
| 302 | D3=12 Ep1 | 24 |

OUTILS NÉCESSAIRES

- Clé de 8, 13 et 16
- Clé à pipe de 8, 13 et 16
- Clé allen de 2, 2.5, 4 et 5
- Tournevis plat de différentes tailles (de petite à grande empreinte)
- Pince circlips extérieur
- Extracteur arrache moyeu
- Maillet

Dans l'assemblage du robot, l'assemblage du corps (étapes 1 à 7) et l'assemblage de la patte (étapes 8 à 12) peuvent être effectuées en parallèle par 2 à 4 personnes.

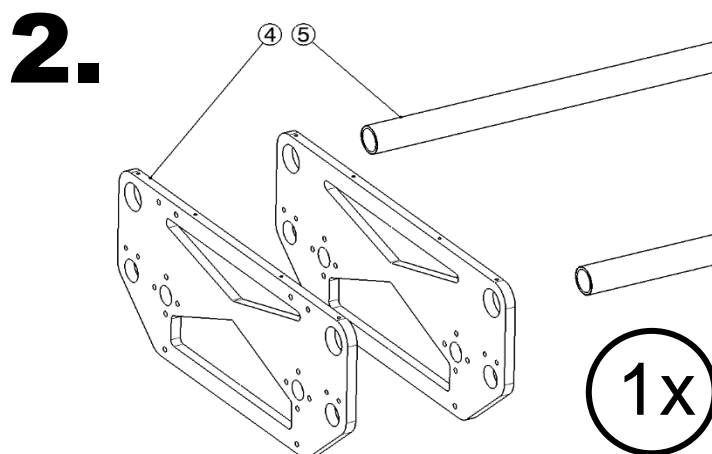
ASSEMBLAGE DU CORPS



Insérer 2 dans 1 par les rails.

Laisser idéalement un espace de 35mm entre les deux supports 3 autour du centre.

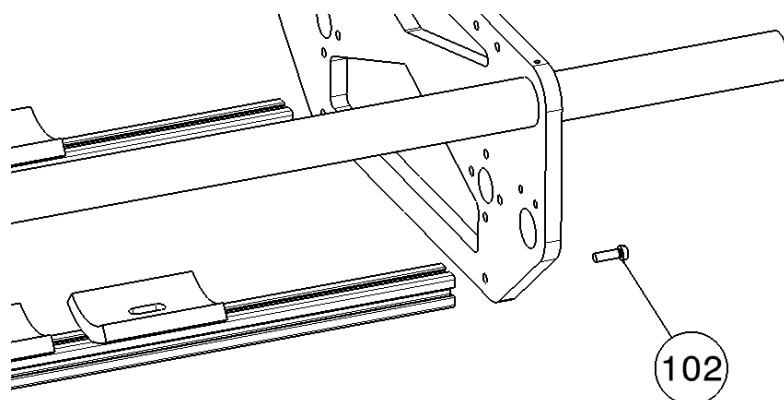
Visser jusqu'à ce que les 3 ne puissent plus translater sur la barre.



Insérer 5 dans 4 sans forcer, progressivement.

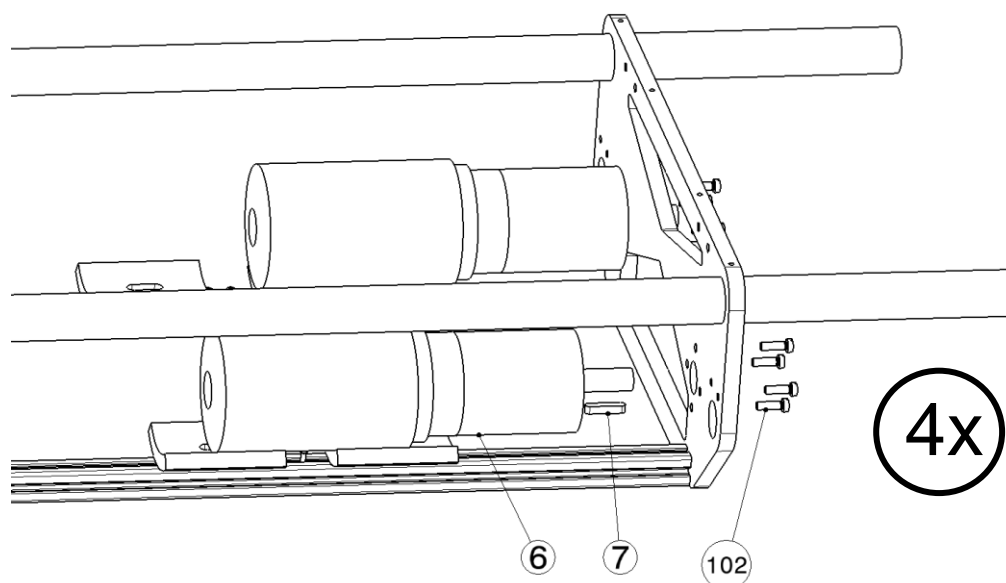
Laisser un espace de 17cms entre le bout du tube 5 et la 1^{ère} plaque 4 ainsi qu'un espace d'au moins 48cms entre les deux plaques 4 une fois celles-ci insérées.

3.



Fixer 1 à 4 par la vis 102 de chaque côté des 2 barres avec une rondelle M5.

4.



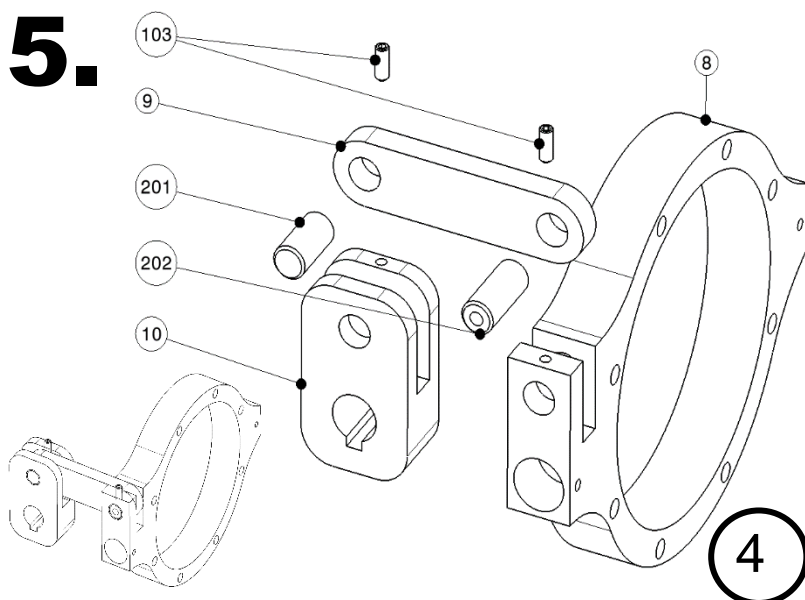
Encastrer la clavette 7 dans l'arbre du moteur 6.

Déposer du frein faible sur les vis 102 (type GISS Colles).

Fixer le moteur 6 à 4 par les vis 102, en orientant la clavette 7 vers le bas.

Il est conseillé d'ajouter des écrous entre 6 et 4 sur les vis 102 pour un meilleur maintien.

5.



Il est possible que ce sous-ensemble ne soit pas démonté et déjà disponible pour le montage.

Disposer 9 dans 10.

Encastrer 201 au travers de 9 et 10.

Disposer 9 dans 8.

Encastrer 202 au travers de 8 et 10.

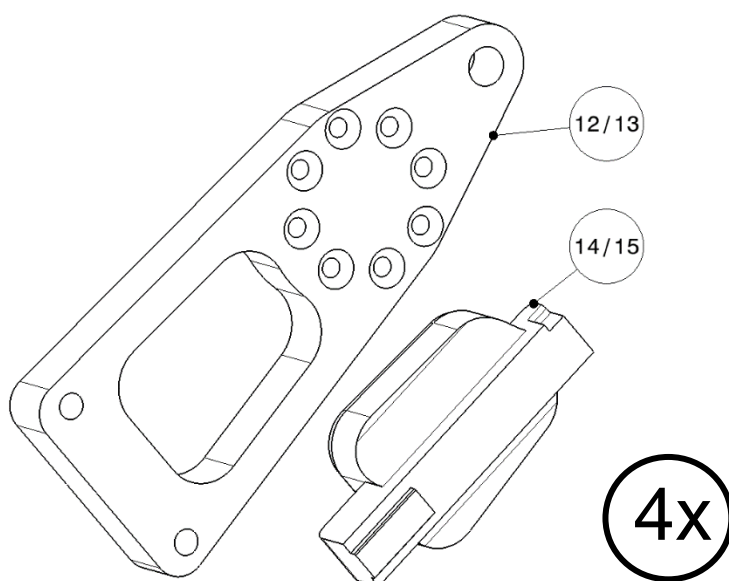
Visser 103 dans 8 et 9 afin de bloquer les axes en rotation par rapport à 8 et 9.

Serrer fortement **203** dans **8** avec **103**.

Serrer fortement **204** dans **8** avec **103**.

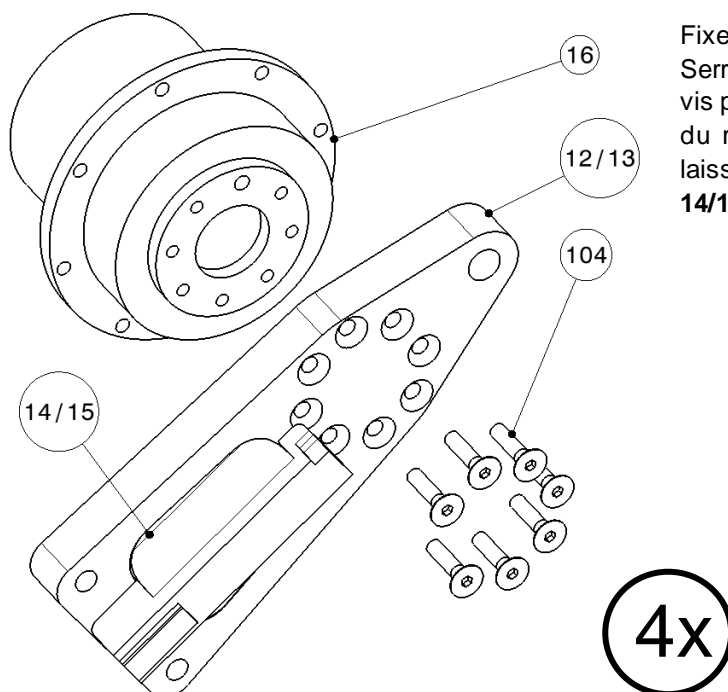
ASSEMBLAGE DE LA PATTE

8.



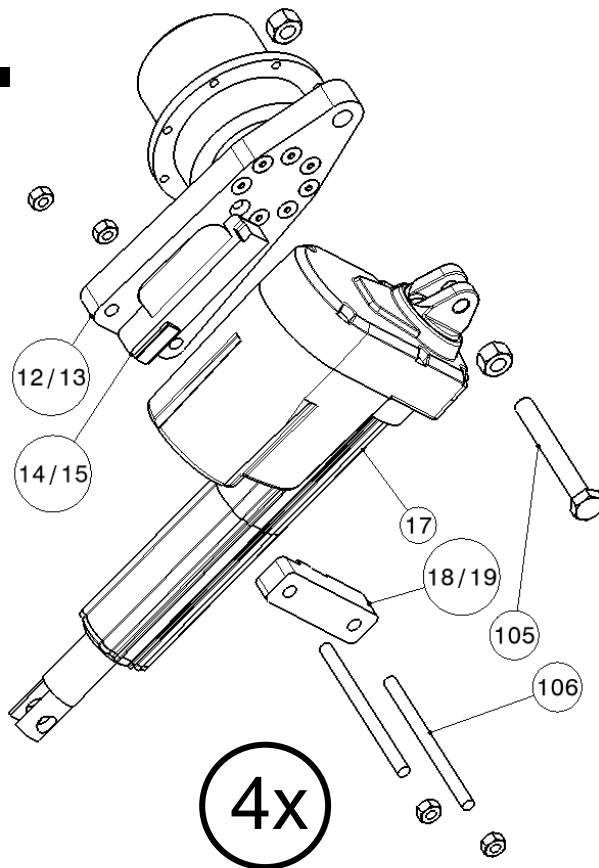
Insérer **14/15** dans **12/13**, attention, les deux pièces ont leurs *symétries* différents, bien observer la forme de la poche de **12/13** et du contour de **14/15** afin de former les bonnes paires. Cette opération peut s'effectuer à l'aide d'une presse si nécessaire, la pièce **14/15** doit être insérée complètement.

9.



Fixer **16** sur **12/13** avec les vis **104**. Serrer fortement. Mettre seulement **7** vis pour les **8** trous, car un des trous du réducteur **16** n'est pas taraudé, laisser ce trou sans vis sous la pièce **14/15**.

10.



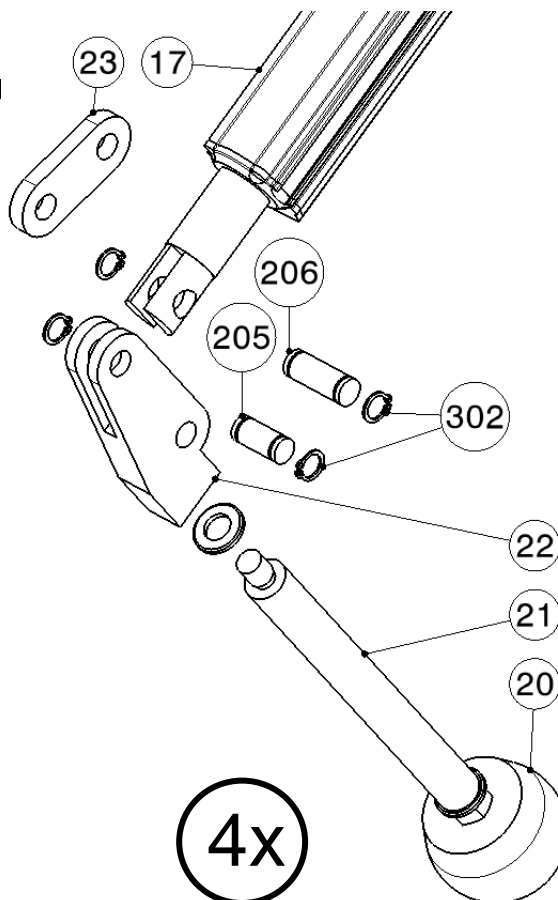
Fixer le haut du vérin **17** par **105** au travers de **12/13** et **17**. Positionner un écrou sur **105** entre **17** et **12/13** ainsi qu'un autre derrière **12/13**. Ajouter une rondelle devant chaque écrou. Ne pas serrer la vis dans un premier temps.

Positionner la rainure de **18/19** sur les rails de **17**, attention **18/19** a son *symétrique* différent, deux d'entre elles une poche de plus pour laisser passer le câble des vérins qui seront côté droit du robot.

Fixer le bas de **17** par les vis **106** au travers de **18/19** et **12/13**. Ajouter une rondelle M8 sous chaque écrou. Serrer fortement.

Serrer fortement **105** et ses écrous.

11.

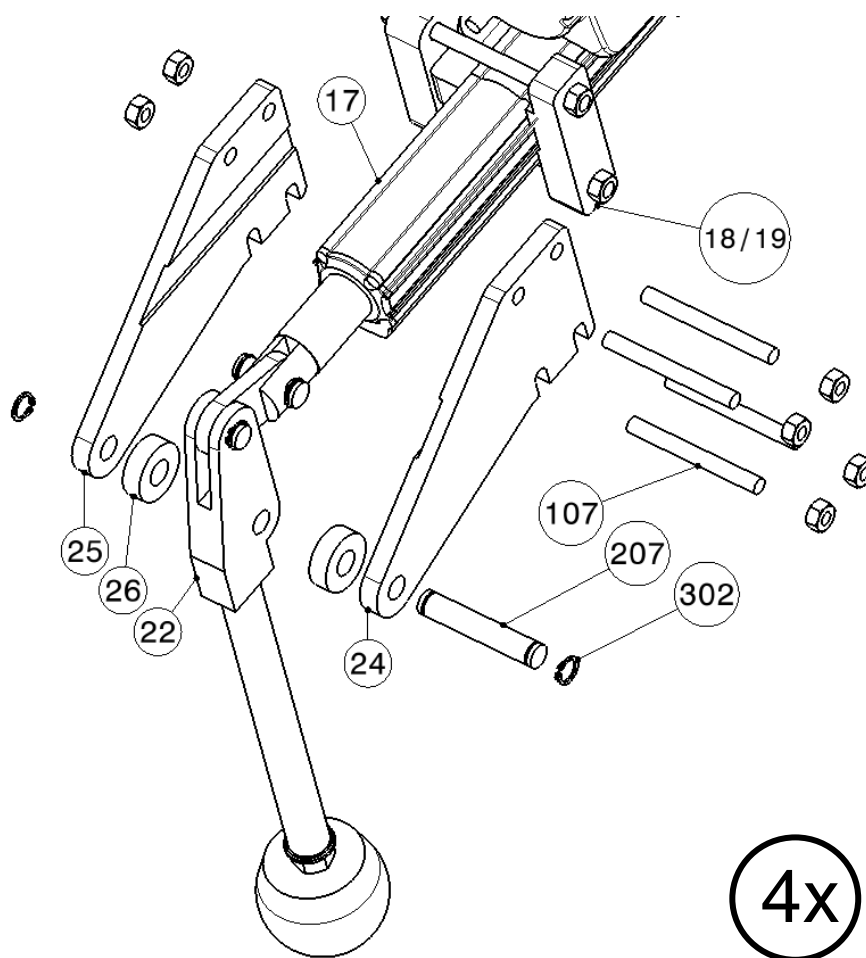


Fixer le pied **20** au tube **21**. La morphologie du pied peut changer suivant les besoins, le pied type actuel est vissé dans **21** avec une rondelle M12 intermédiaire.

Fixer **21** dans **22** selon la même méthode. Attention, **22** peut se fixer dans les deux sens mais seul le sens représenté ici est valide (arrondi vers l'intérieur du coude, découpe en biais vers l'extérieur).

Assembler **23** dans **22** et **17** avec respectivement **205** et **206**. Mettre les circlips **302** de chaque côté des axes pour bloquer la translation.

12.

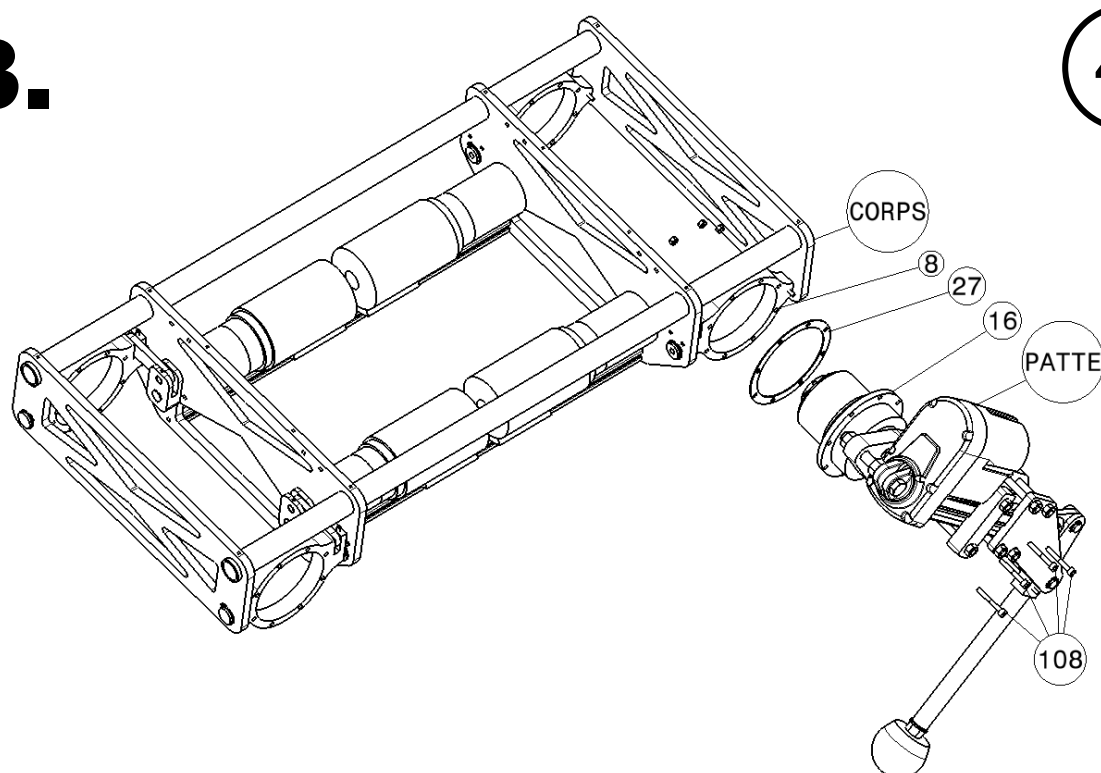


Insérer **207** au travers de respectivement **24**, **26**, **22**, **26** et **25**. Mettre les circlips **302** de chaque côté des axes pour bloquer la translation.
Positionner les rainures de **24** et **25** dans les rails de **17**.

Fixer **24** et **25** à **17** par les tiges filetées **107** et leurs écrous. Peu serrer pour pouvoir ensuite coulisser **24** et **25** le long des rails de **17** jusqu'à ce que les faces de **24** et **18/19** qui se font face, soient distantes de $55,5\text{mm}$ (mesurer au réglet pendant l'ajustement). Serrer progressivement les différents écrous en conservant le décalage de $55,5\text{mm}$. Ajouter une rondelle M8 sous chaque écrou.

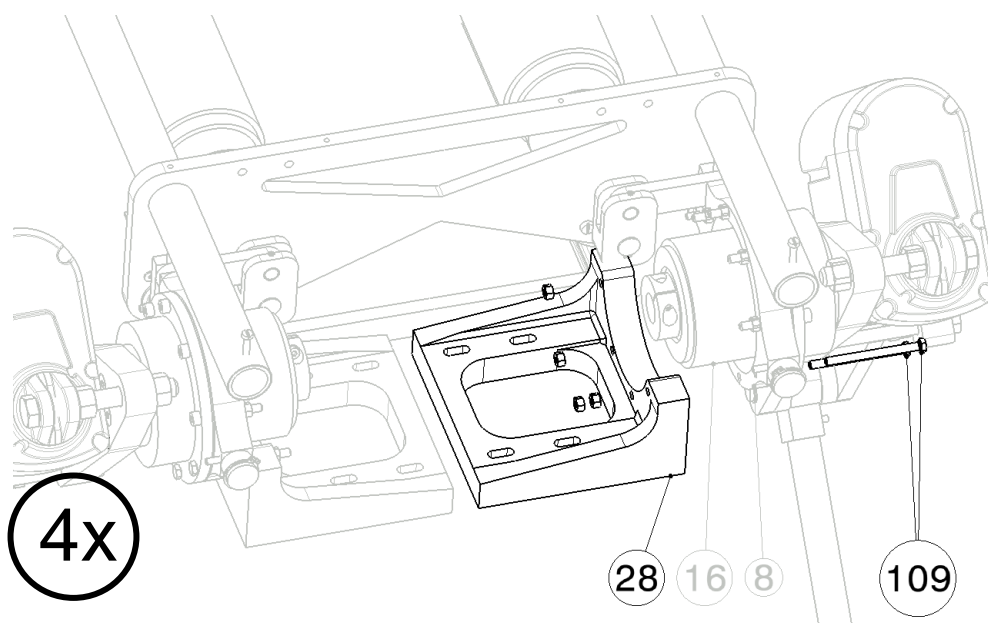
ASSEMBLAGE FINAL

13.



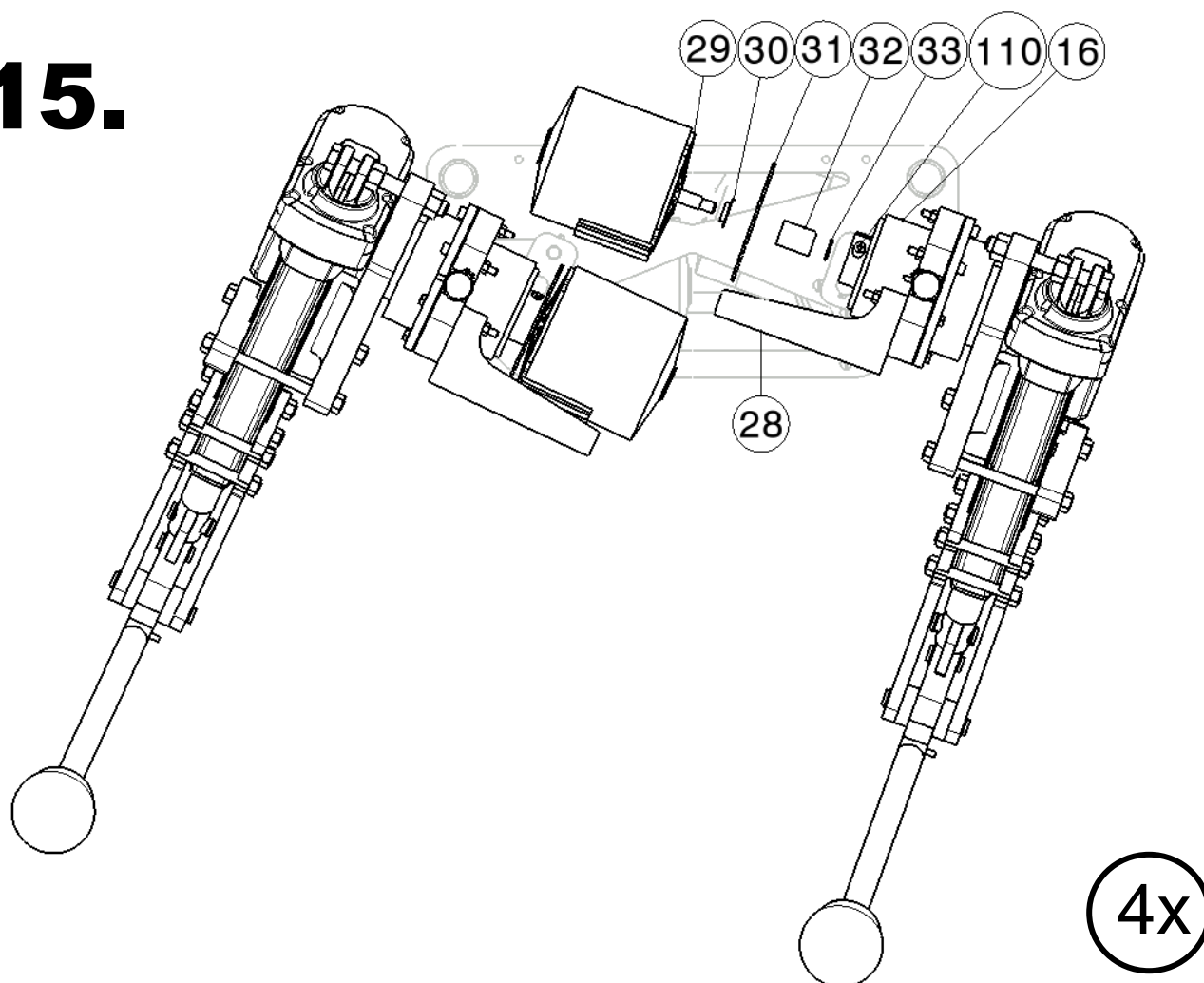
Fixer la patte au corps par les vis **108** et leurs écrous au travers de **16**, **27** et **8**, ajouter une rondelle M5 pour la tête de vis ainsi qu'une autre côté écrou.
Les 4 vis **108** doivent être positionnées sur les 4 trous les plus haut des 8 trous présents sur **8**.

14.



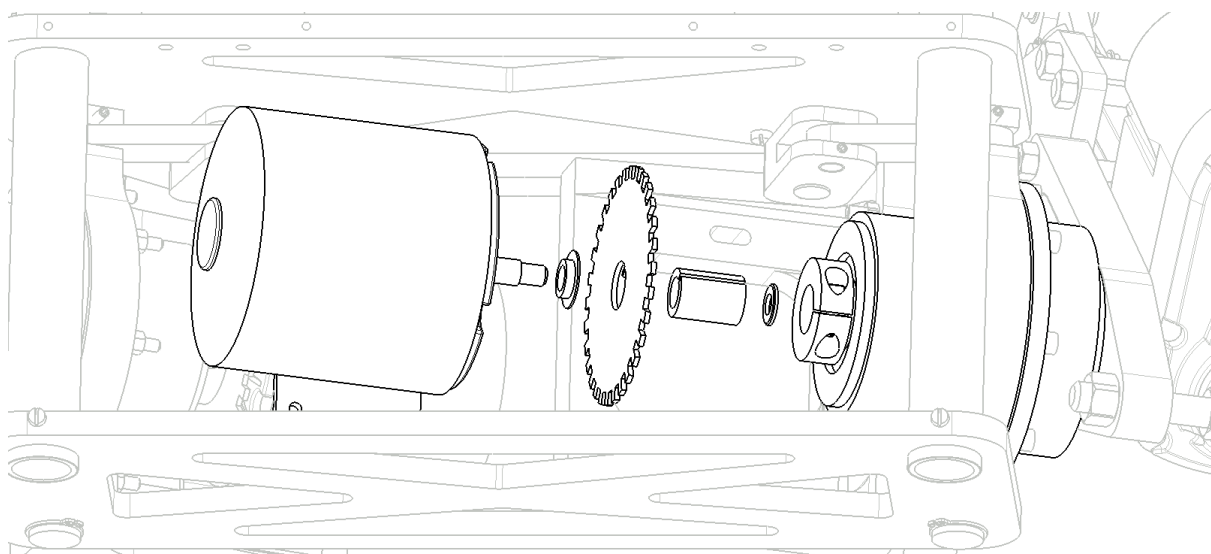
Fixer le porte moteur **28** à **8** par les vis **109** et leurs écrous, ajouter une rondelle M5 pour la tête de vis ainsi qu'une autre côté écrou.

15.

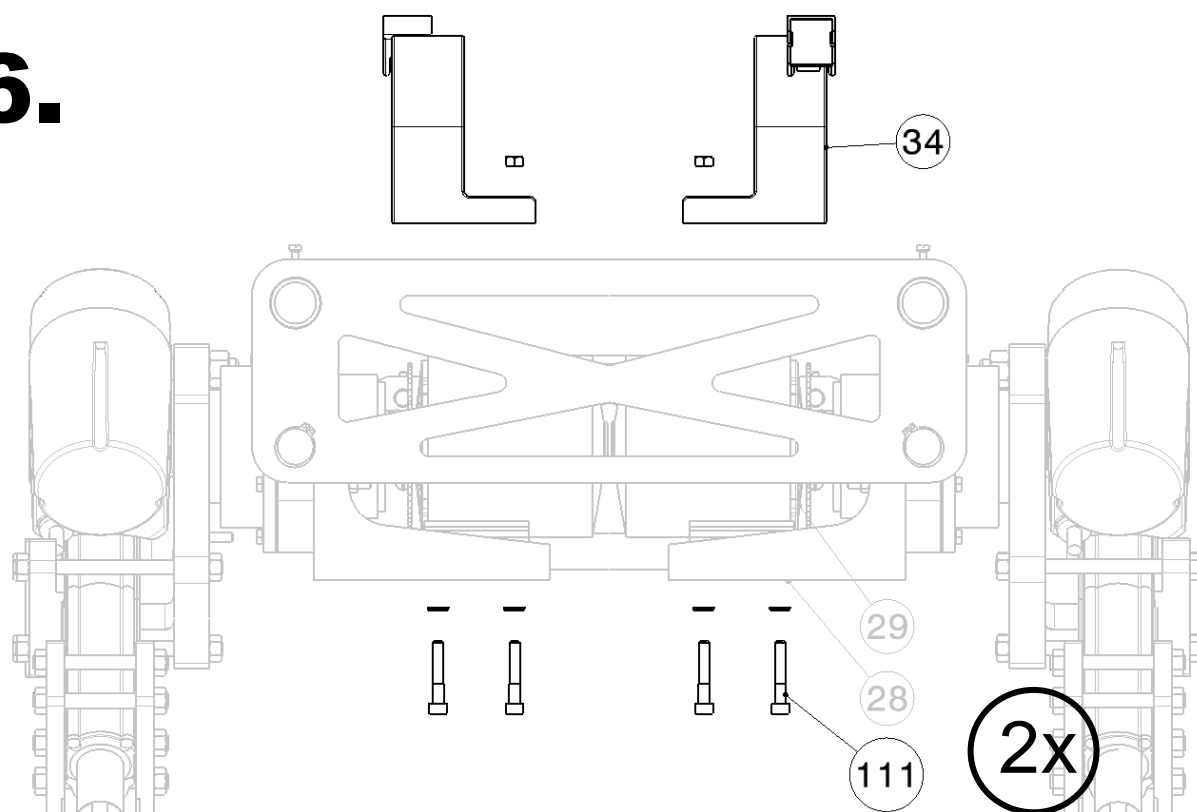


Insérer l'œillet **30** jusqu'au bout de l'arbre moteur **29**, petit diamètre d'abord.

Positionner la cale **33** à l'intérieur du réducteur **16**, insérer ensuite la bague par-dessus. La bague dépassera légèrement. Insérer la roue **31** sur la bague **32** en positionnant l'ergot dans la fente de la bague. Maintenir ainsi **31** dans **32** face contre **16** afin qu'elle ne tombe pas. Insérer l'arbre **29** dans **16** au travers de **31**, **32** et **33** sans forcer. Pour faciliter l'insertion on peut maintenir plaqué en forçant **29** contre **28** pendant l'insertion. Serrer fortement **110** dans son logement.

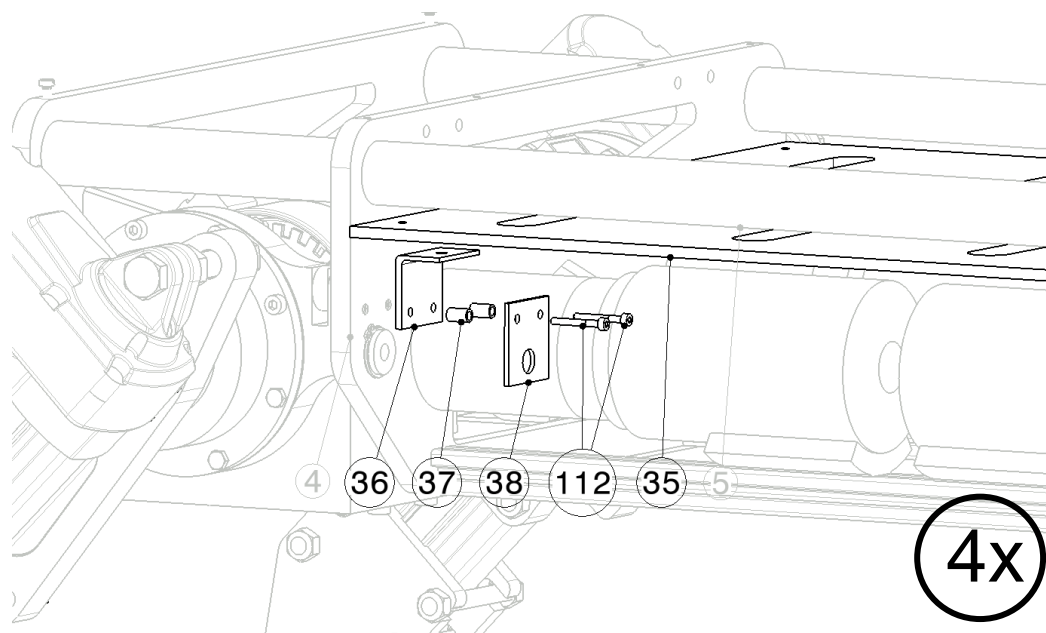


16.



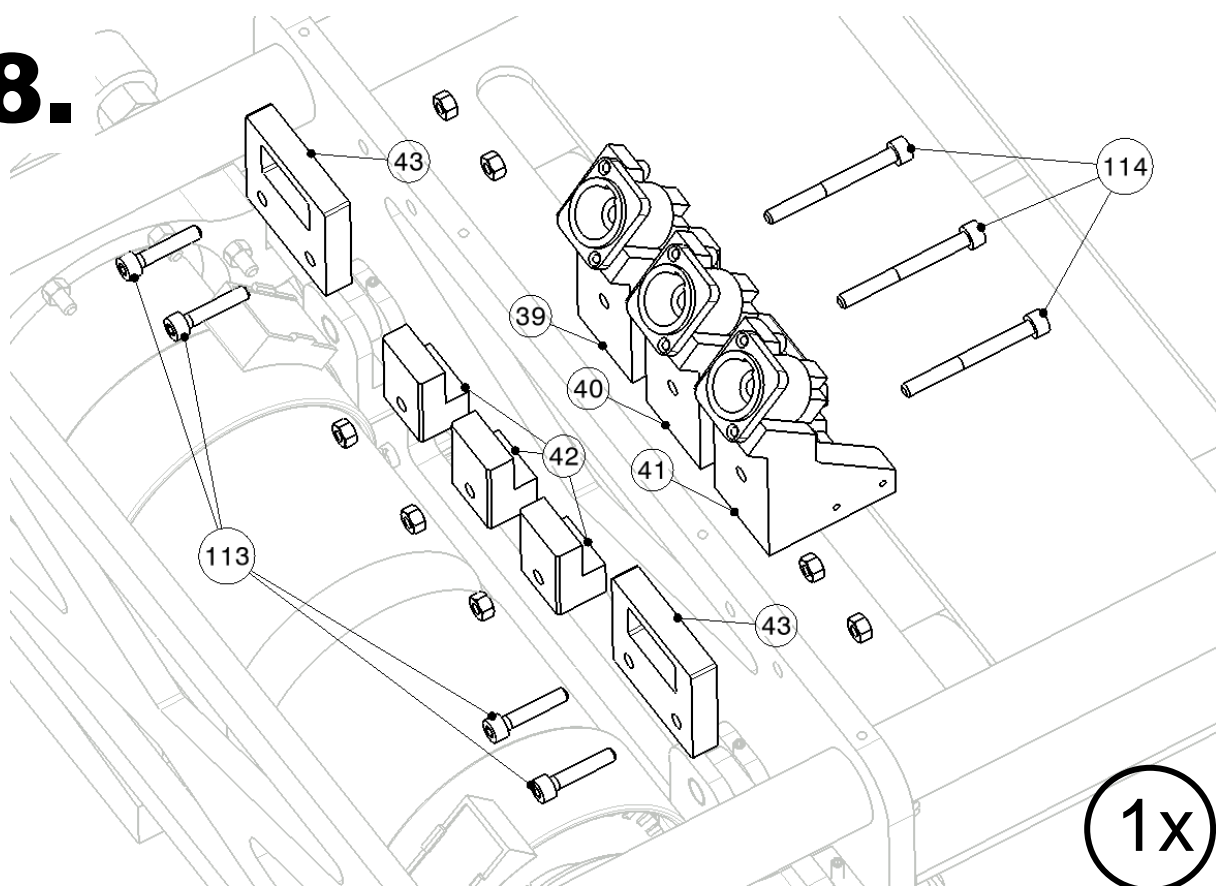
Positionner **34** autour de **29** en utilisant l'élasticité de **34**. La roue codeuse doit se retrouver dans le U formé par le capteur présent sur **34**. Fixer **34** à **29** par les vis **111** et son écrou à travers **28**, **29** et **34**. Seulement 2 vis sont nécessaires au bon maintien de **34**, mais elles doivent être positionnées en diagonale l'une de l'autre. Ajouter une rondelle M5 du côté de la tête de vis. Serrer faiblement.

17.



Insérer la plaque électrique **35** sous les tubes **5** (on ne peut pas monter **35** après fixation de l'équerre **36**). Fixer **36** sur la plaque **4** par les vis **112** à travers **38**, **37** et **36**. Serrer modérément de manière à ne pas trop déformer les entretoises **37**.

18.



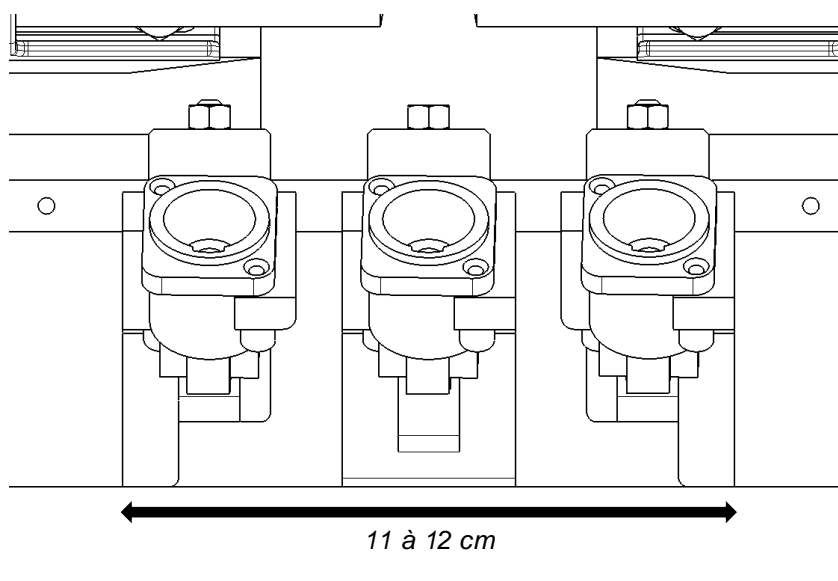
Positionner les borniers à plats contre les plaques. Attention les borniers étant tous différents, reproduire la mise en place ci-dessous.

Fixer les borniers **39**, **40** et **41** par les vis **114** au travers de **39/40/41** et **42**.

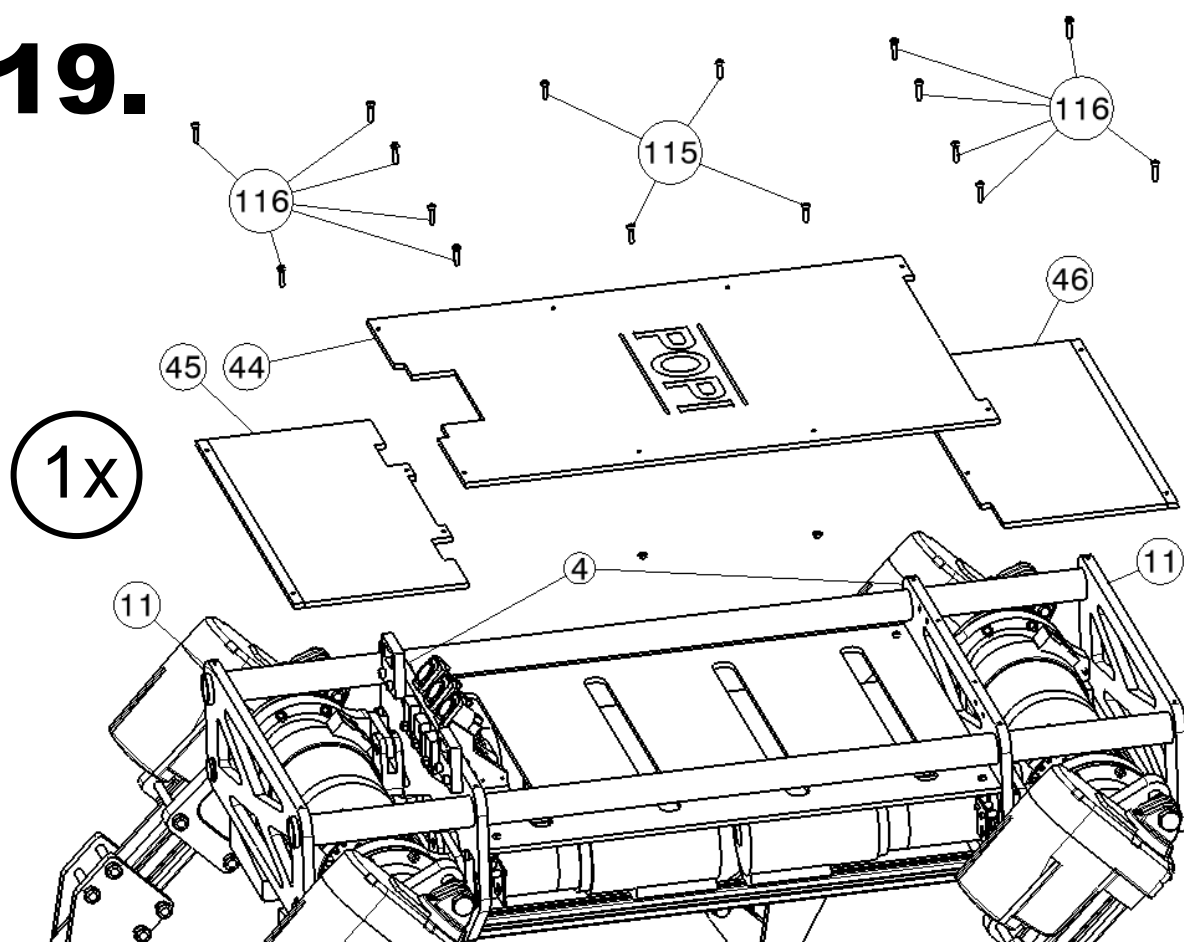
Fixer les accroches **43** par les vis **113** au travers de la plaque et **43**.

Ajouter des rondelles aux têtes de vis ainsi que du côté des écrous.

Le robot doit maintenant être câblé électriquement avant la prochaine étape.

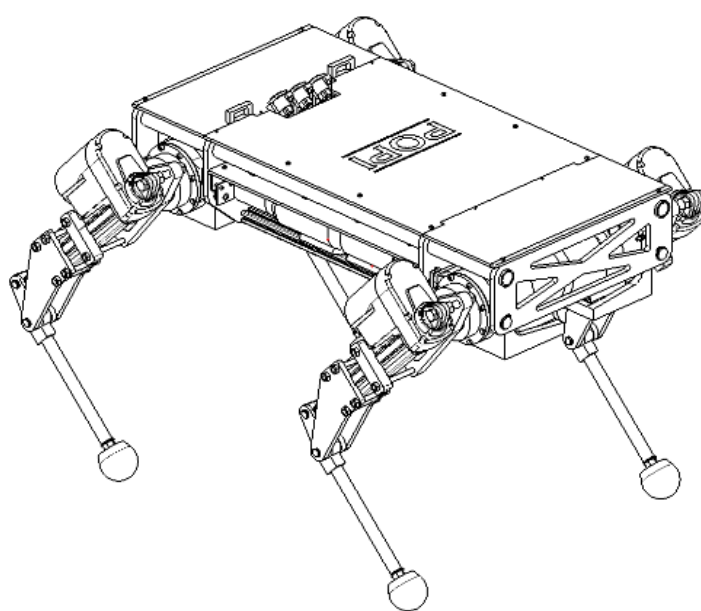


19.



Fixer les vis **116** à la plaque **44** par des écrous. Ces vis serviront simplement à reposer la plaque sur les tubes. Fixer les plaques plexiglass **45** et **46** aux plaques **4** et **11** par les vis **115**. Serrer faiblement.

20.



Enjoy.