



KUMO



PORCEL Koralie
VERMOT-DESROCHES Matthias

Table des matières

01 Introduction

02 Caméra et contrôleur

03 Déplacement

04 Vision

05 Démonstration

06 Conclusion

01

Introduction



02

• Choix de la caméra et du Microcontrôleur



Caméra



Cahier des charges :



- Champ de vision diagonale d'au moins 85°
- Caméra de profondeur pour détection d'obstacles
- Facile à implémenter sur le robot

Caméra



Choix : intel realsense d435 :

- Champ de vision diagonale $>90^\circ$
- Package ROS2



90 mm x 25 mm x 25 mm

Microcontrôleur



Jetson Nano



 ROS 2

Mise à jours :

- Système de la Carte
- Kernel
- ...

PB : incompatibilité avec la caméra

Microcontrôleur



Solutions : carte Xavier ?

Caractéristiques adaptées

Kernel compatible avec caméra

++ Cher : 600 - 1000€



 ROS 2

Microcontrôleur



Choix définitif : PC

 ROS 2

03

Déplacement



Déplacement : la simulation



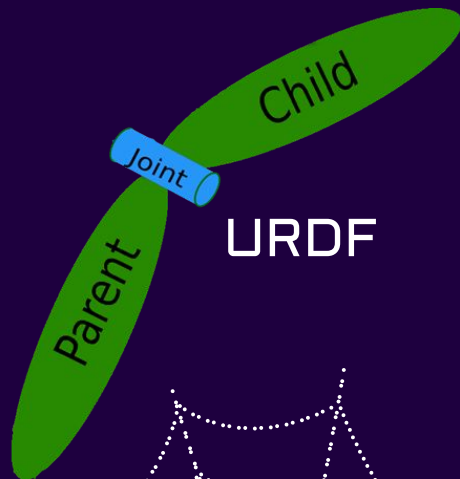
Créer un monde
sur Gazebo



Importer un
modèle du robot



Contrôler le robot
avec ROS2

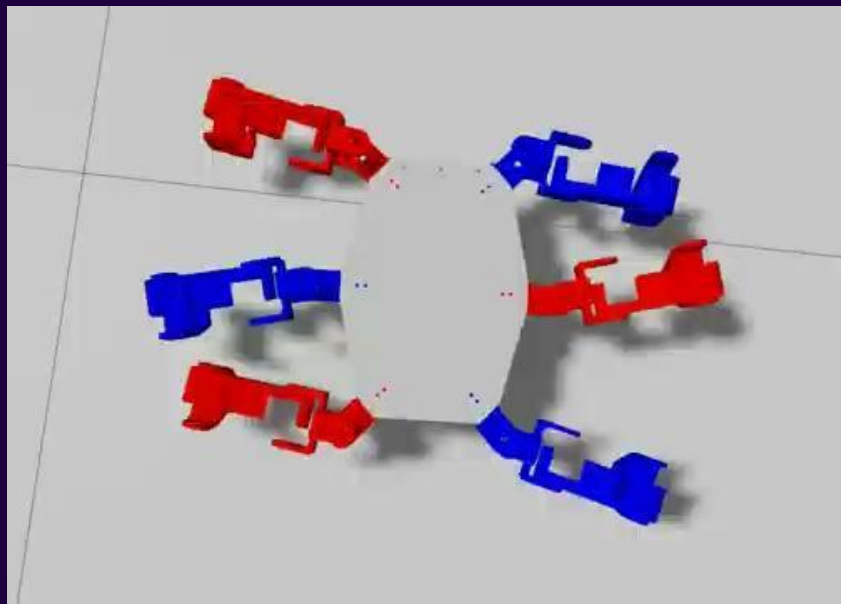


ROS

Déplacement : Tripode

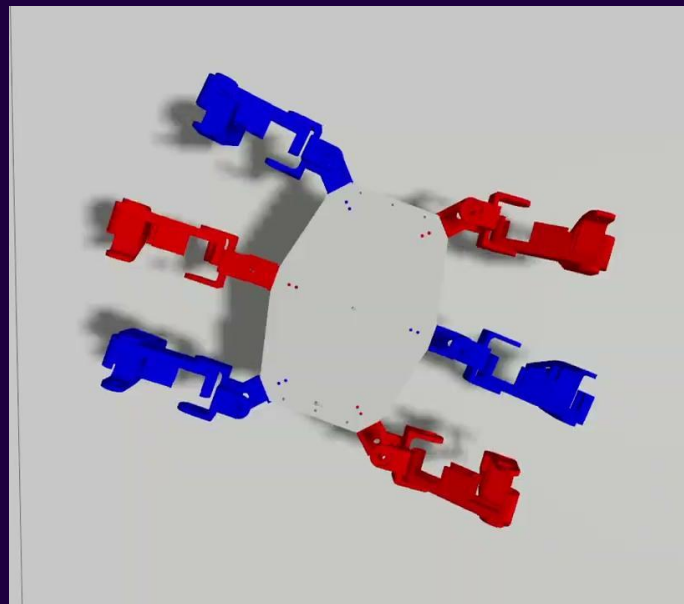
Avancer

Reculer



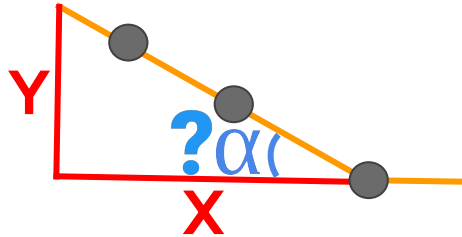
Tourner à gauche

Tourner à droite



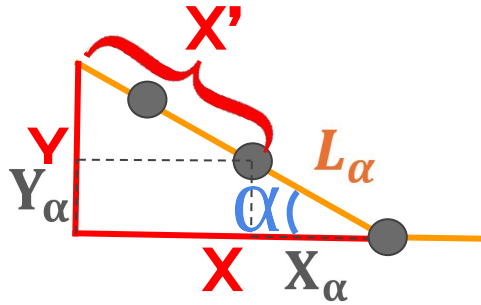
Déplacement : cinématique inverse

$$f(X, Y, Z) = [\alpha, \beta, \gamma]$$



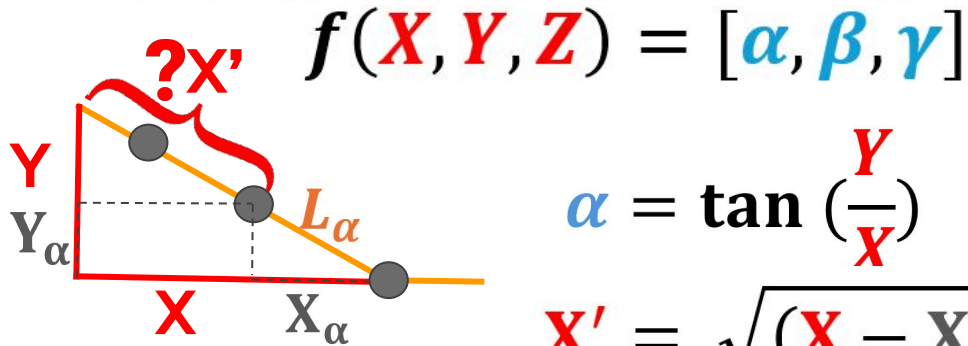
$$\alpha = \tan\left(\frac{Y}{X}\right)$$

Déplacement : cinématique inverse



$$\alpha = \tan \left(\frac{Y}{X} \right)$$

Déplacement : cinématique inverse



$$\alpha = \tan\left(\frac{Y}{X}\right)$$

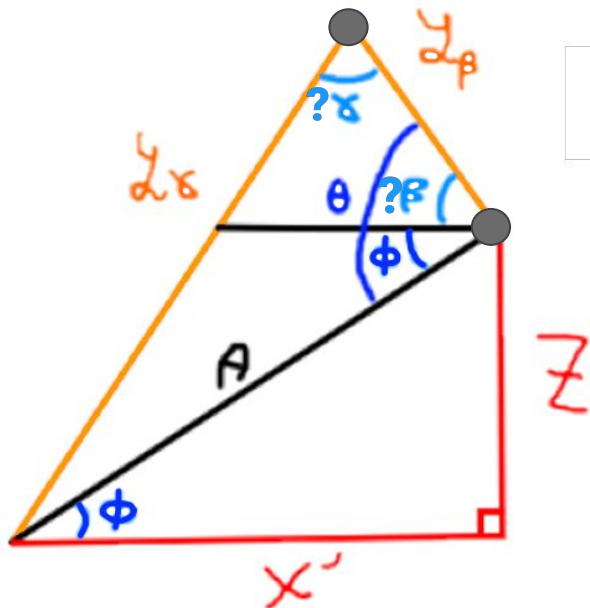
$$X' = \sqrt{(X - X_\alpha)^2 + (Y - Y_\alpha)^2}$$

$$X_\alpha = L_\alpha \cos(\alpha)$$

$$Y_\alpha = L_\alpha \sin(\alpha)$$

Déplacement : cinématique inverse

$$f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = [\alpha, \beta, \gamma]$$



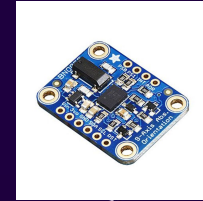
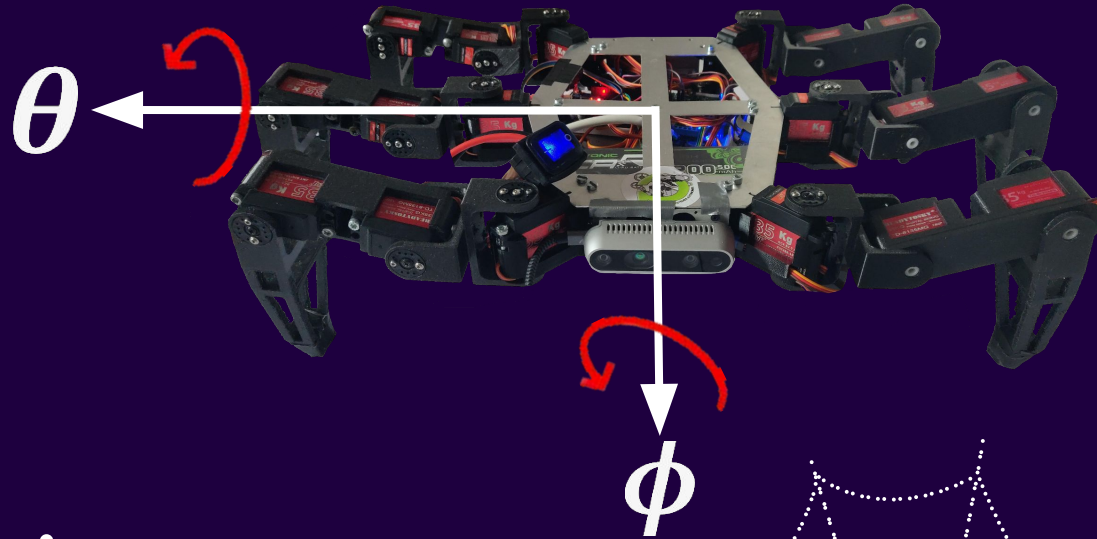
$$\gamma = \arccos\left(\frac{L_\beta^2 + L_\gamma^2 - A^2}{2L_\beta L_\gamma}\right)$$

$$\beta = \theta - \phi$$

$$\phi = \arcsin\left(\frac{Z}{X'}\right)$$

$$\theta = \arccos\left(\frac{L_\beta^2 + A^2 - L_\gamma^2}{2L_\beta A}\right)$$

Déplacement : filtre de Kalman



Accéléromètre

Gyromètre

a_x, a_y, a_z

p, q, r

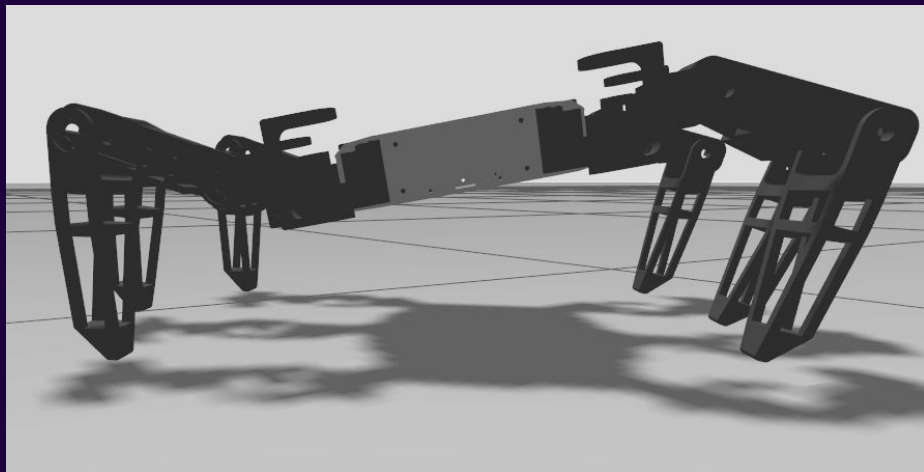
ϕ, θ

Modèle
mathématique

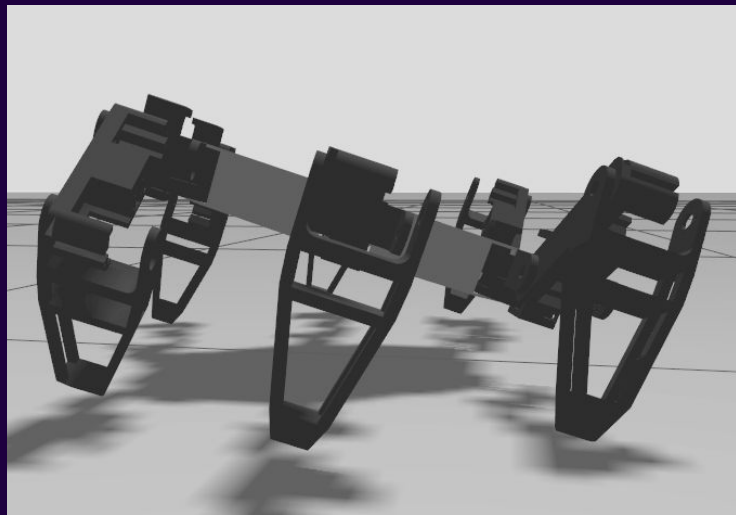
Déplacement : filtre de Kalman



Augmenter plus ou moins Z en fonction du sinus de l'angle



Roll



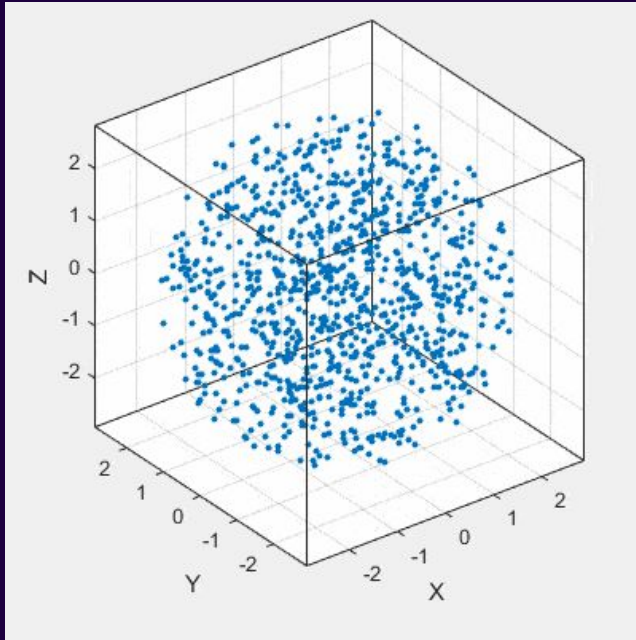
Pitch

04

Vision

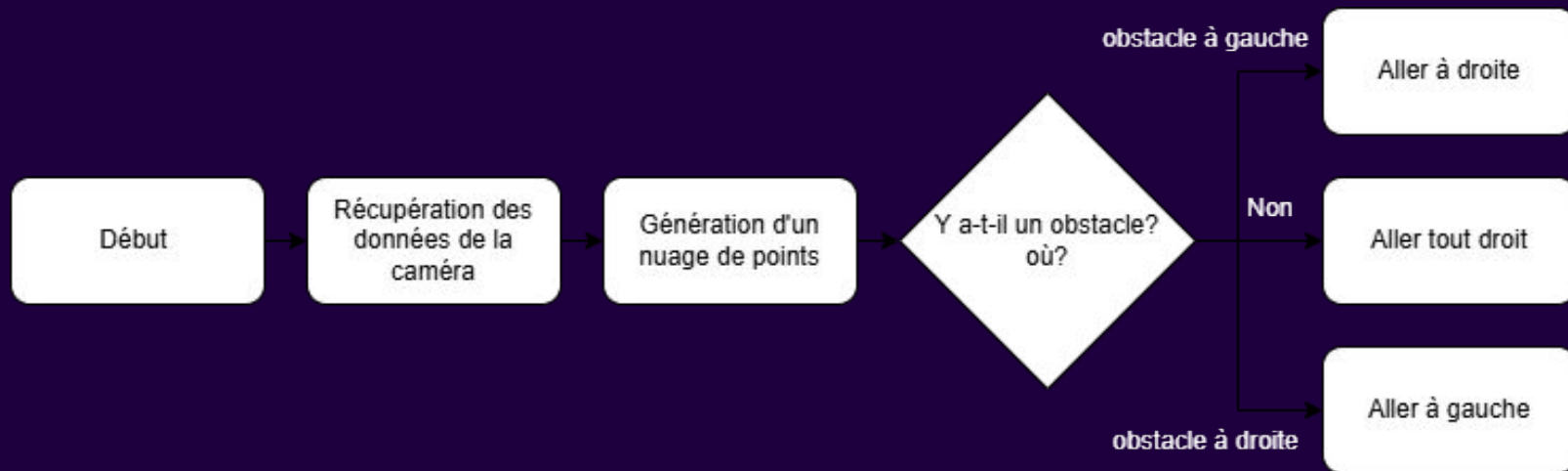


Base des programmes

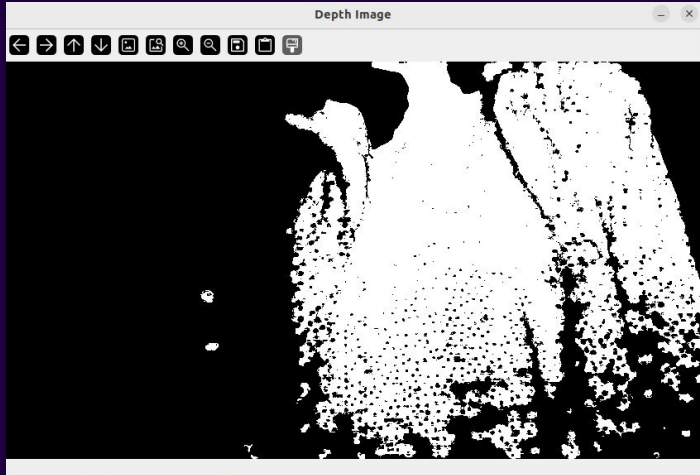


- Récupération des données de la caméra
- Génération d'un nuage de points
- Affichage du nuage

1er programme : éviter les obstacles



Améliorer la détection



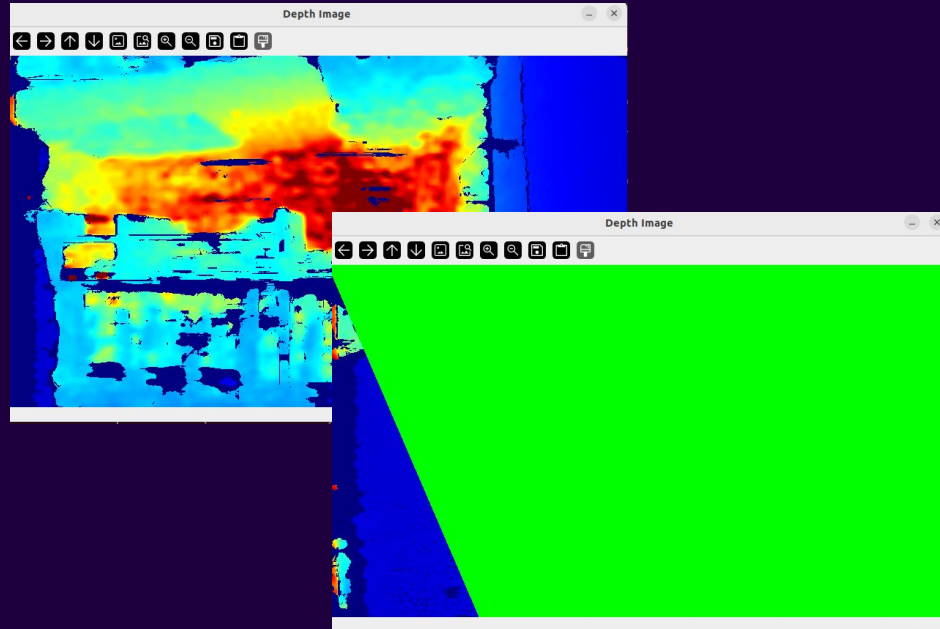
PB Nuage de points :

- coûteux en calcul
- Aucune compréhension environnement

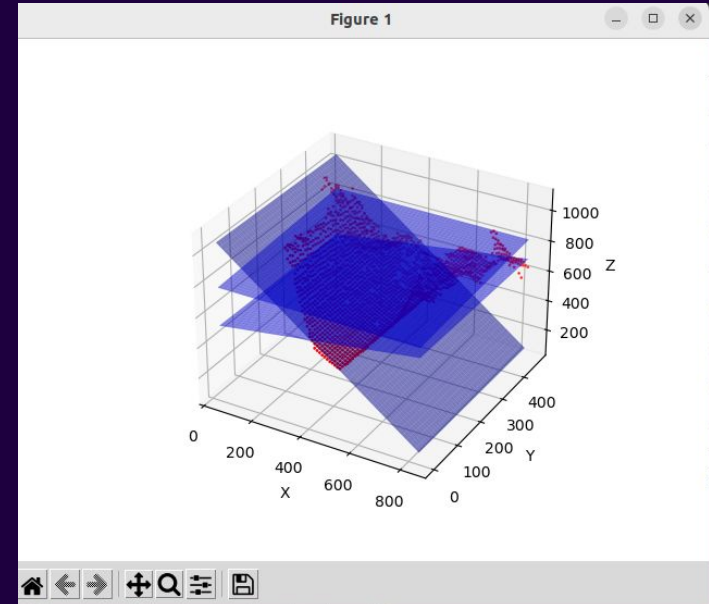
Solution : trouver des plans

Interpolation ou segmentation

Régression+interpolation



Segmentation

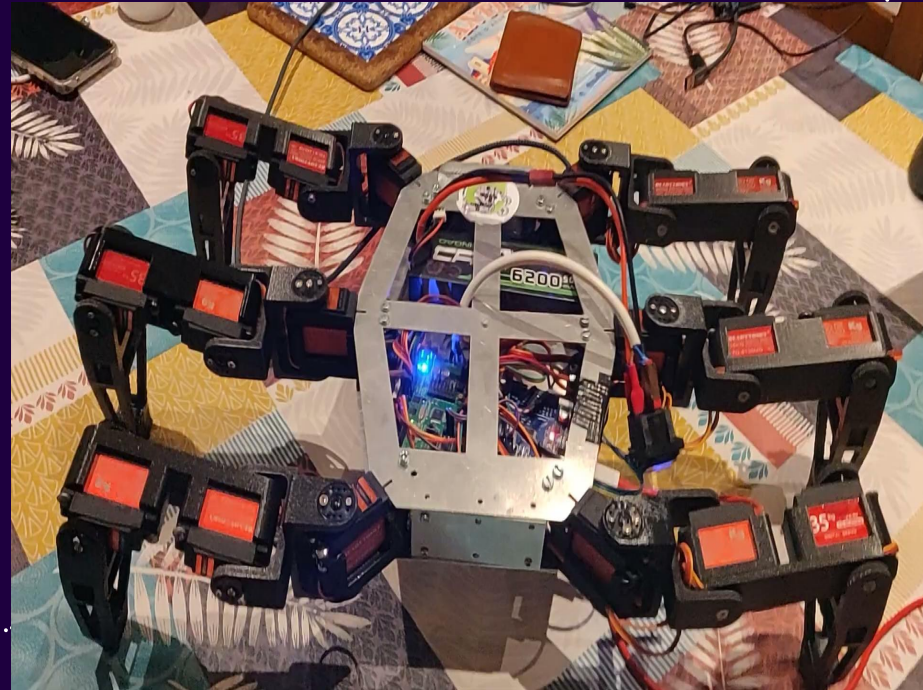
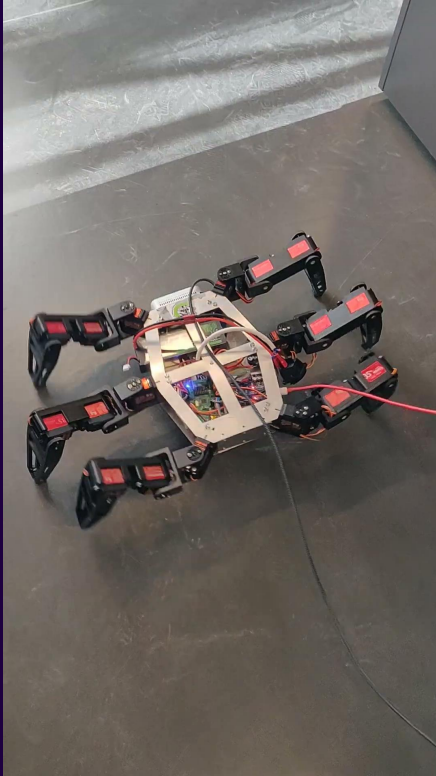


05

Démonstration



Démonstration



06

Conclusion : Merci

