

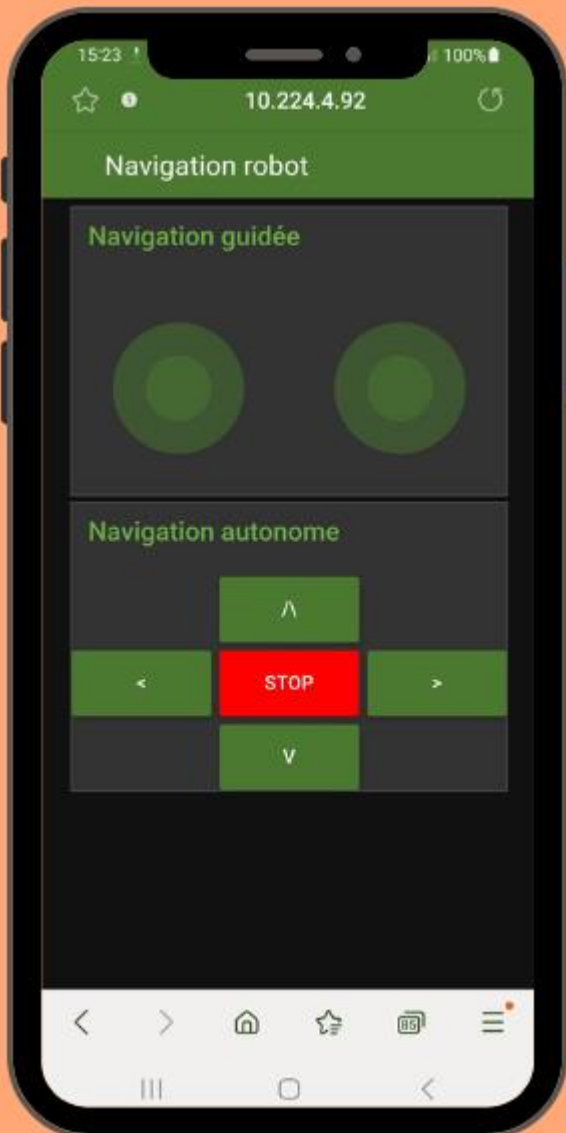
Projet M1 2022/2023

ROBOTIQUE ET DEVELOPPEMENT LOGICIEL

COMMENT CONTROLER LE ROBOT HOLONOME VIA SON TELEPHONE ?

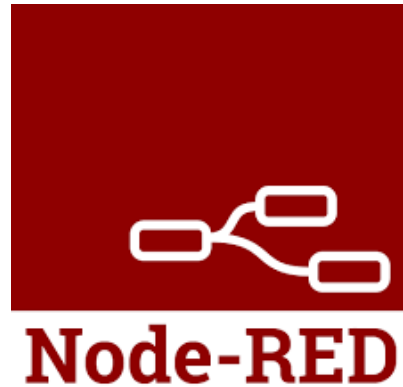
Un travail réalisé par :
Laurie AYEVEGHE BEKALE
Matthieu DESMYTTERE
Jean MROZIEWICZ

junia ISEN



COMMENT CONTROLER LE ROBOT HOLONOME VIA SON TELEPHONE

Présentation de Node RED



Node-RED est un environnement de développement basé sur un navigateur pour la programmation visuelle des nœuds. Il permet de créer facilement des flux de données en connectant des nœuds prédéfinis pour réaliser des tâches telles que l'IoT (Internet des objets), l'automatisation de tâches, la visualisation de données, etc.

Node-RED dispose d'une interface utilisateur graphique intuitive qui permet aux utilisateurs de créer et de gérer des flux de données sans écrire de code. Il est basé sur Node.js et utilise JavaScript pour créer des nœuds personnalisés. Il dispose également d'une grande bibliothèque de nœuds préconstruits pour faciliter le développement de flux de données.

Initialisation de Node RED

L'environnement est déjà installé sur l'ordinateur du robot holonome. Pour le lancer, il faut activer ROS sur le robot, grâce à la commande **roscore**. Ensuite, sur une autre fenêtre, il faudra lancer Node RED en saisissant **node-red** tout simplement. Cette commande rendra alors disponible notre interface web de travail, via l'adresse <http://127.0.0.1:1880/> sur le navigateur web. Le dernier flow en cours apparaîtra automatiquement, mais il est tout à fait possible d'en importer un autre depuis l'ordinateur. Si le dernier flow en cours ne correspond pas à l'interface de contrôle du robot, vous pouvez la retrouver sur l'ordinateur au chemin suivant : [Documents/application_node_red.json](#). Ainsi l'adresse <http://127.0.0.1:1880/ui>, permet d'accéder à l'interface utilisateur. Cette interface sera accessible depuis n'importe quel appareil connecté au même réseau que l'ordinateur qui a lancé Node RED. Il suffira de remplacer l'adresse IP locale par l'adresse IP de l'ordinateur : <http://10.224.4.92:1880/ui>.

Pour pouvoir tester l'interface ou la manette de Xbox sur le robot, lancez la commande **roslaunch holonomicrobot test_joy.launch**

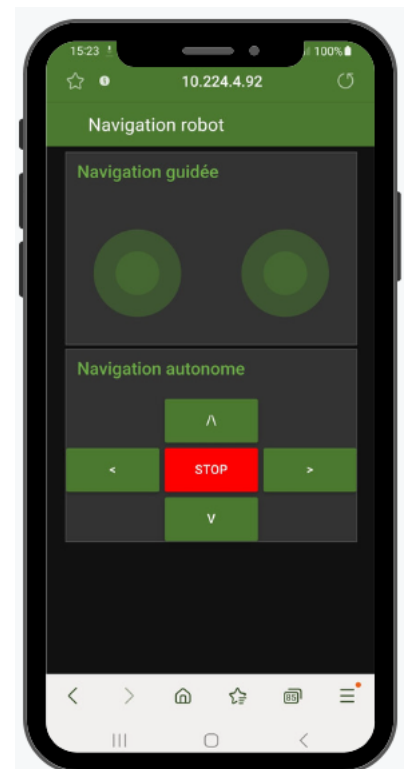
Explication de l'interface utilisateur

L'interface utilisateur est constituée de deux parties : une destinée à la navigation guidée et l'autre à la navigation autonome.

La navigation guidée comporte deux joysticks. Le premier (celui de gauche), permet de se déplacer dans toutes les directions ainsi qu'en diagonale. Le deuxième joystick (celui de droite) permet au robot de tourner dans un sens ou dans l'autre. Contrairement au premier joystick, il ne peut faire que des mouvements horizontaux (gauche-droite et vice versa). Ces deux joysticks sont totalement inspirés de ceux de la manette de Xbox, par conséquent ils ont les mêmes fonctions que sur cette dernière.

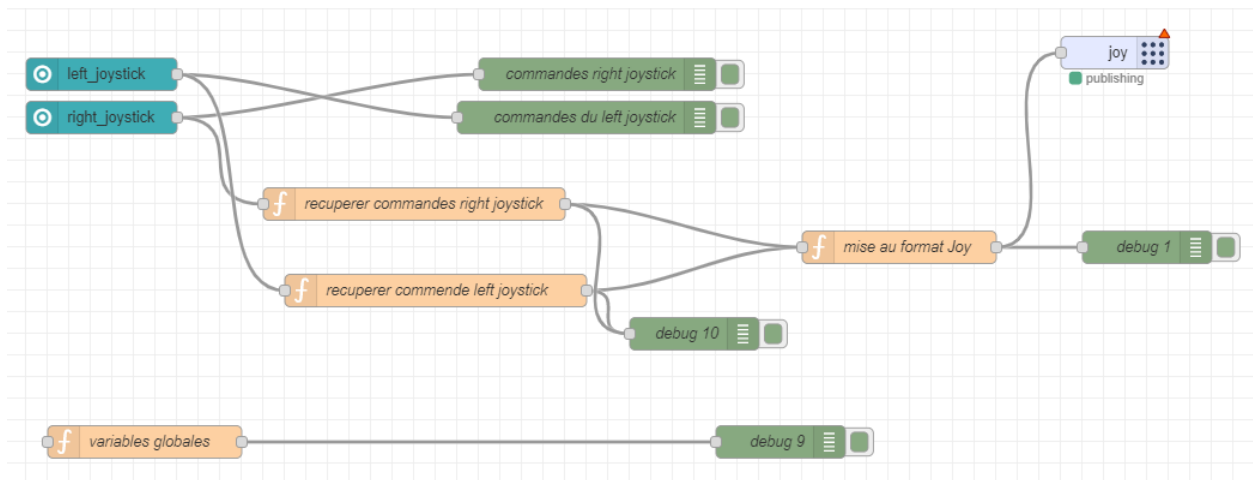
En ce qui concerne la navigation autonome, les flèches permettent d'indiquer au robot la direction à suivre. Ainsi tant qu'on ne lui aura pas dit "stop", celui-ci suivra la première instruction et n'en exécutera aucune autre pendant ce temps. En principe, il devrait pouvoir éviter les obstacles de lui-même pour suivre la direction de départ.

Pour cette interface, nous avons dû installer les modules suivants de Node RED : node-red-dashboard (pour la mise en page et les boutons), node-red-contrib-ui-joystick , node-red-contrib-looptimer (pour les boucles de répétition), snappy-ros (souscrire et publier sur un topic ros).



Explication de l'interface du workflow

Navigation guidée



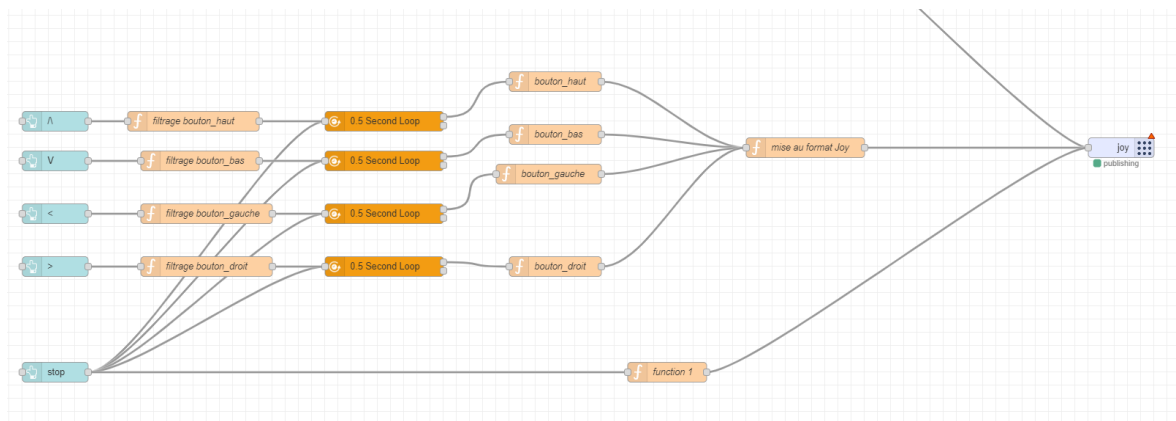
La photo ci-dessus présente la première partie du flow de l'application web concernant la navigation guidée. Les fonctions écrites sont présentées en beige, et en vert ce sont les nœuds de débogage.

La fonction **variables globales** définit est la première à se lancer, dès le lancement du flow. Elle permet d'initialiser à 0 les variables globales **x_right**, **y_right**, **y_left** et **x_left**, qui représentent les coordonnées envoyées par les joysticks right et left. Elle initialise également les variables **bouton_haut**, **bouton_bas**, **bouton_gauche**, **bouton_droit**, **boutons_x** et **boutons_y** qui représentent les différents boutons directionnels que nous avons.

Les messages envoyés par les joysticks sont filtrés par les fonctions **recupérer commandes right joystick** et **recupérer commandes left joystick** pour extraire les coordonnées x et y de chacun. Ces deux fonctions vont retourner chacune un objet sous forme de dictionnaire contenant les infos nécessaires pour faire bouger le robot.

Ensuite les données des deux joysticks vont être récupérées par la fonction **mise au format Joy** pour être regroupées au format Joy, avant d'être envoyé au topic « joy » pour contrôler le robot.

Navigation autonome



L'image ci-dessus représente la partie du flow concernant la navigation autonome. En bleu, on retrouve les boutons avec chacun une direction (**haut**, **bas**, **gauche**, **droite**) et le bouton **stop**. Chaque bouton est relié à une fonction de filtrage qui va permettre le lancement d'une commande uniquement si une autre n'est pas déjà en cours. La fonction **0.5 Second Loop** permet d'envoyer le message toutes les 0,5 secondes et interrompt sa transmission si elle reçoit un message dont le payload est "stop", message envoyé par le bouton **stop**. Pour cette partie, nous ne contrôlons pas la rotation du robot car celui-ci devrait tourner de lui-même pour éviter les obstacles. Lorsqu'on clique sur le bouton "stop", la **fonction 1** permet de remettre toutes les variables globales à zéro, pour éviter des erreurs lors d'une prochaine commande. Cette fonction envoie également un message sous format Joy au topic joy. Du côté de ROS, ce message permet d'arrêter les moteurs afin que le bouton B de la manette de Xbox soit considéré comme le bouton stop.

Initialisation de la partie hardware

Tout d'abord, veillez à ce que le bouton d'arrêt d'urgence ne soit pas enfoncé, en effet il est responsable de l'arrêt de l'alimentation des moteurs. Si ce bouton est enfoncé vous pouvez toujours travailler sur l'ordinateur embarqué du robot ainsi qu'avoir les retours des LIDAR et autres capteurs ou modules installés sur le robot mais vous ne pourrez pas le déplacer.

Vérifiez que l'interrupteur pour la recharge de la batterie soit sur O. Ensuite vous pouvez allumer le robot grâce à l'interrupteur général (voir figure suivante), veillez à ce que la LED rouge soit allumée.



Interrupteur général (interrupteur avec la LED rouge), interrupteur de recharge batterie (interrupteur O/I) et bouton d'arrêt d'urgence.

Initialisation du NUC

Connectez un clavier et une souris au PC embarqué (NUC) afin de pouvoir interagir avec l'écran. Ensuite, ouvrez la session **isen** avec le **mot de passe « isen »** puis lancez un terminal de commandes pour y taper les commandes nécessaires.

Utilisation de ROS (Robot operating system)

ROS est un système d'exploitation open-source permettant de faciliter la programmation des robots. En effet, il permet d'effectuer plusieurs tâches simultanément tout en facilitant la communication entre elles. Il fournit notamment une architecture flexible et un grand nombre de

bibliothèques. ROS dispose également d'une grande communauté, ce qui facilite la possibilité de trouver des sources. Afin d'initialiser ROS, taper **roscore**.

Initialisation du joystick

Si vous voulez utiliser le joystick, allumez la manette de xbox et patientez jusqu'à ce qu'elle se connecte au robot. La connexion est active lorsque le logo Xbox ne clignote plus (qu'il est allumé en continu). Si cela ne marche pas, allez dans les paramètres Bluetooth et oubliez la manette (Xbox Controller Wireless) puis reconnectez la.

Interprétation des commandes

Les commandes désirés seront envoyés sur un topic joy sous un format Joy que ce soit avec un joystick ou à partir de node-red. Afin de lire ces commandes et de les transformer en commandes moteurs, Il suffit de taper la commande **roslaunch holonomicrobot test_joy.launch** qui lancera un programme python (joystick.py dans le répertoire holonomicrobot) qui fera le lien entre commandes sous format Joy et commandes moteurs. Si vous désirez voir si les commandes sont bien reçues par ROS taper la commande **rostopic echo joy**.

Affichage des données lidars

Pour afficher les données des lidars sur RVIZ, tapez la commande **roslaunch holonomicrobot slam_with_controller_odom.launch**. Le lidar enverra les données sur le topic scan/ qui sera ensuite interprété par RVIZ qui les affichera. Si vous désirez afficher le gmapping,, tapez la commande **roslaunch holonomictobot gmapping.launch**.