



# Documentation pour la prise en main du RB-Vogui-XL



Juillet 2025

# Table des matières

<b>1</b>	<b>Démarrage du robot</b>	<b>3</b>
1.1	Allumer le robot . . . . .	3
1.2	Arrêt d'urgence . . . . .	5
1.3	Charge du robot . . . . .	6
1.4	Éteindre le robot . . . . .	6
<b>2</b>	<b>IHM du robot</b>	<b>7</b>
2.1	Partie Monitor . . . . .	7
2.1.1	Localization panel . . . . .	7
2.1.2	Status Summary . . . . .	10
2.1.3	System Monitor . . . . .	11
2.2	Partie Sequencer . . . . .	12
<b>3</b>	<b>Manipulation du Vogui</b>	<b>13</b>
3.1	Contrôle du robot . . . . .	13
3.2	Connexion/Reconnexion de la manette . . . . .	14
3.3	Contrôle du bras . . . . .	15
3.3.1	Allumer le bras . . . . .	15
3.3.2	Contrôler avec l'interface Remmina . . . . .	16
<b>4</b>	<b>Fonctionnalités du robot</b>	<b>21</b>
4.1	Principe de ROS . . . . .	21
4.2	Qu'est-ce que catkin_ws ? . . . . .	21
4.3	Qu'est-ce qu'un package ROS ? . . . . .	21
4.4	Création d'un package . . . . .	21
4.5	Compilation . . . . .	22
4.6	Exécution d'un nœud . . . . .	23
4.7	Communication entre packages : les topics et les nœuds . . . . .	23
<b>5</b>	<b>Démonstration</b>	<b>27</b>
5.1	Contrôle du bras avec la manette . . . . .	27
5.1.1	Etape 1 : Démarrage du robot et du bras . . . . .	27
5.1.2	Etape 2 : Vérification de l'interface Remmina . . . . .	27
5.1.3	Etape 3 : Connexion au robot en SSH . . . . .	27
5.1.4	Etape 4 : Lancement du programme . . . . .	28
5.1.5	Etape 4 bis : Lancement de la démo . . . . .	29
5.2	Les différentes commandes . . . . .	30
5.2.1	Déplacements "manuels" . . . . .	36
5.2.2	Arrêt du programme . . . . .	37
5.3	Solutions en cas de problèmes rencontrés . . . . .	38
<b>6</b>	<b>Utilisation de Rviz</b>	<b>39</b>
6.1	Lancement de RViz . . . . .	39
6.2	Présentation de l'interface . . . . .	39
6.3	Estimation de la position initiale : 2D Pose Estimate . . . . .	41
6.4	Définir un objectif de navigation : 2D Nav Goal . . . . .	41

<b>7</b>	<b>Interface réseau Teltonika</b>	<b>42</b>
7.1	Connecter le robot à internet . . . . .	43
7.2	Interface routeur Vogui_5GHz . . . . .	44
<b>8</b>	<b>Réinitialisation du robot</b>	<b>47</b>

# 1 Démarrage du robot

## 1.1 Allumer le robot

1) Il faut d'abord commencer par tourner la clé (ON/OFF) : les LEDs présentes à l'avant du robot sont blanches et clignotent. Attendre que les LEDs soient vertes et fixes.



FIGURE 1 : Interface du robot



FIGURE 2 : Les LED sont vertes et fixes : le robot a démarré avec succès

2) Les boutons "CPU" et "Restart" sont allumés. Pour lancer le système, il faut d'abord allumer le bouton "coup de poing" en tirant le bouton rouge vers le haut. Puis il faut faire un appui long sur le bouton "Restart" jusqu'à entendre le bruit d'un contacteur (la lumière du bouton doit s'éteindre). Dans le cas contraire, éteindre et rallumer le bouton coup de poing sans fil et réessayer.



FIGURE 3 : Bouton CPU et Restart allumés



FIGURE 4 : Bouton coup de poing sans fil

## 1.2 Arrêt d'urgence

Aux 4 coins du robot, on retrouve les boutons d'arrêt d'urgence (il y a un cinquième bouton, le bouton coup de poing sans fil). L'appui de n'importe quel bouton stoppe immédiatement le système. Pour le redémarrage du système, il faut d'abord désenclencher le bouton d'arrêt d'urgence actif. Il suffit ensuite de faire un appui long sur le bouton "Restart"



FIGURE 5 : Bouton d'arrêt d'urgence



FIGURE 6 : Les LEDs sont rouges et clignotent : l'arrêt d'urgence est actif

### 1.3 Charge du robot



FIGURE 7 : Témoin lumineux (LEDs vertes et jaunes) qui indique que le robot est en charge

### 1.4 Éteindre le robot

Pour éteindre le robot, il faut d'abord s'assurer qu'il ne soit pas en plein processus de démarrage et qu'il ne soit pas en mouvement ou en mode arrêt d'urgence. Dans le cas où le robot est prêt à être arrêté, on commence par faire un appui long sur le bouton "CPU" jusqu'à l'extinction de la lumière verte du bouton. Les LED à l'avant du robot devraient être bleues et fixes, ce qui signifie que nous pouvons tourner la clé (ON/OFF) pour éteindre le robot.



FIGURE 8 : Témoin lumineux (LEDs bleues) qui indique que le robot est en charge

## 2 IHM du robot

Comme pour la plupart des robots, le RB-Vogui-XL est équipé d'une interface homme-machine (IHM) qui permet à l'utilisateur d'accéder facilement à des informations comme l'état du robot, des capteurs ou encore le niveau de batterie. Pour y accéder, il suffit de connecter un écran via un câble HDMI et de brancher un clavier et une souris sur le hub USB du robot. Cliquer sur le navigateur Mozilla et entrer dans la barre de recherche :

**192.168.0.200/robotnik\_hmi/index.php**

### 2.1 Partie Monitor

#### 2.1.1 Localization panel

Cette section permet d'utiliser différentes fonctions liées à l'environnement. Elle permet de visualiser une représentation en nuage de points de l'environnement du robot. Ce nuage de points est déterminé à partir des données obtenues en temps réel par les 2 capteurs lidars du robot. Vous pouvez observer le menu déroulant :

-la fonction "localization\_2D" permet de choisir un environnement (map 2D) dans lequel on veut placer le robot. On trouve les différentes cartes enregistrées dans le menu déroulant "Environments". Pour en sélectionner une, il faut presser sur le bouton play. La carte choisie sera aussi visualisable sur Rviz (cf 6. Utilisation de Rviz).

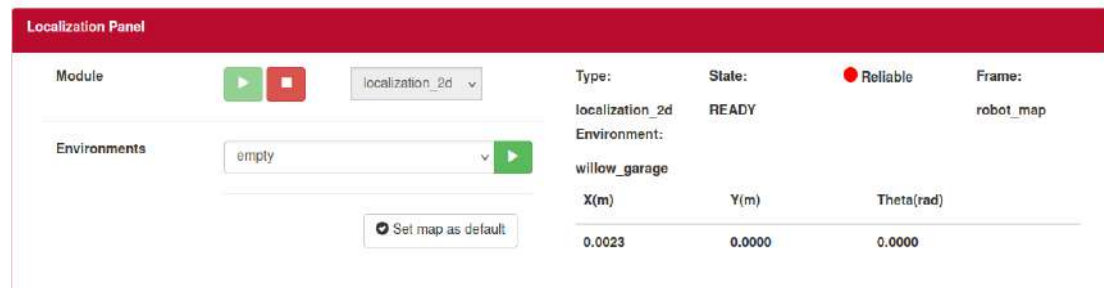


FIGURE 9 : Localization\_2D



Pour modifier la position et l'orientation du robot sur la carte, il faut utiliser le curseur de pose directionnelle.

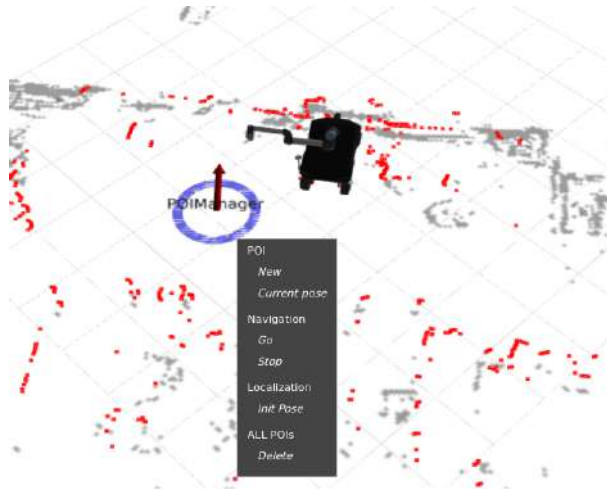


FIGURE 10 : Curseur de pose directionnelle

Pour modifier la position, faire un clic gauche et maintenir la flèche rouge. Pour modifier l'orientation, faire un clic gauche et maintenir le cercle bleu. Une fois que la position vous convient, faire un clic droit sur le curseur et sélectionner "Init Pose". Le robot se déplacera à la position définie.

-La fonction "mapping\_2D" permet de réaliser un enregistrement d'un environnement en 2D (une pièce par exemple). Pour commencer, il suffit de cliquer sur le bouton vert "play" et de déplacer le robot avec la manette de PS4 dans toute la pièce. Lorsque le mapping est terminé, on peut l'enregistrer en écrivant le nom du fichier puis en cliquant sur le bouton orange "save"

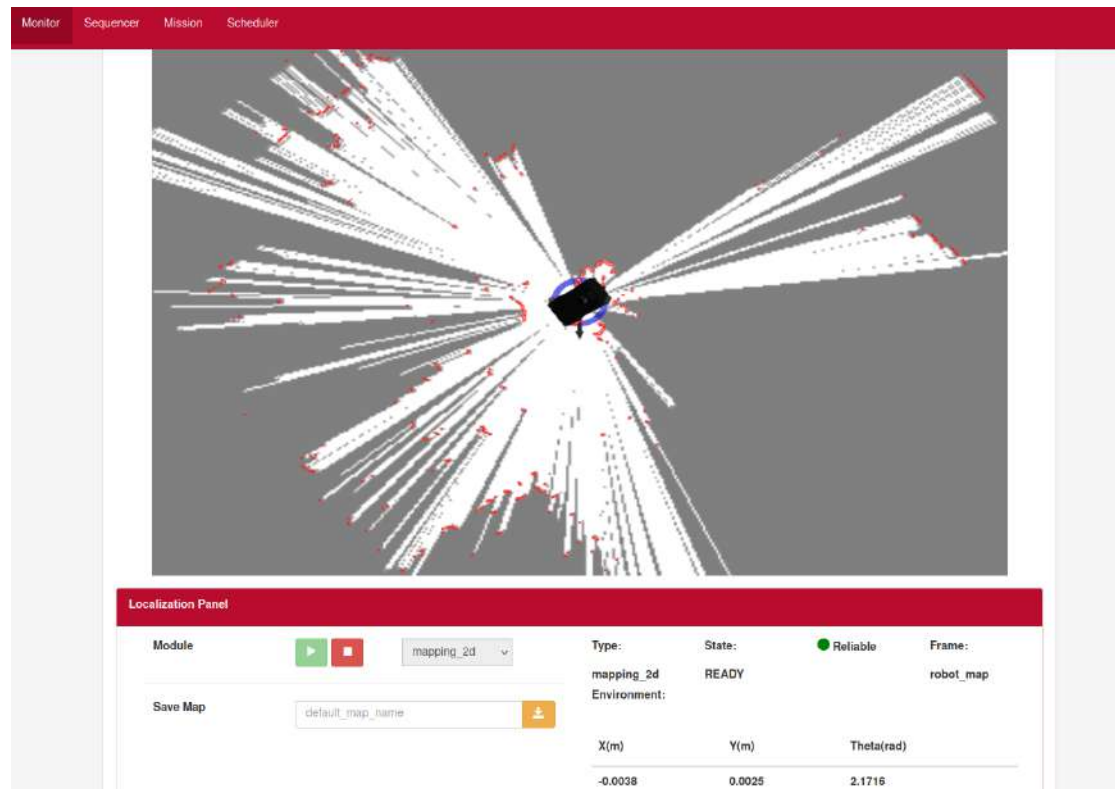


FIGURE 11 : Réalisation d'un mapping 2D

### 2.1.2 Status Summary

La section « Status summary » de l'onglet Monitor nous donne l'état de la batterie, du statut des capteurs et des moteurs, le statut des entrées sorties du contrôleur, l'état de la mémoire, des CPU, la vitesse/position/orientation en temps réel

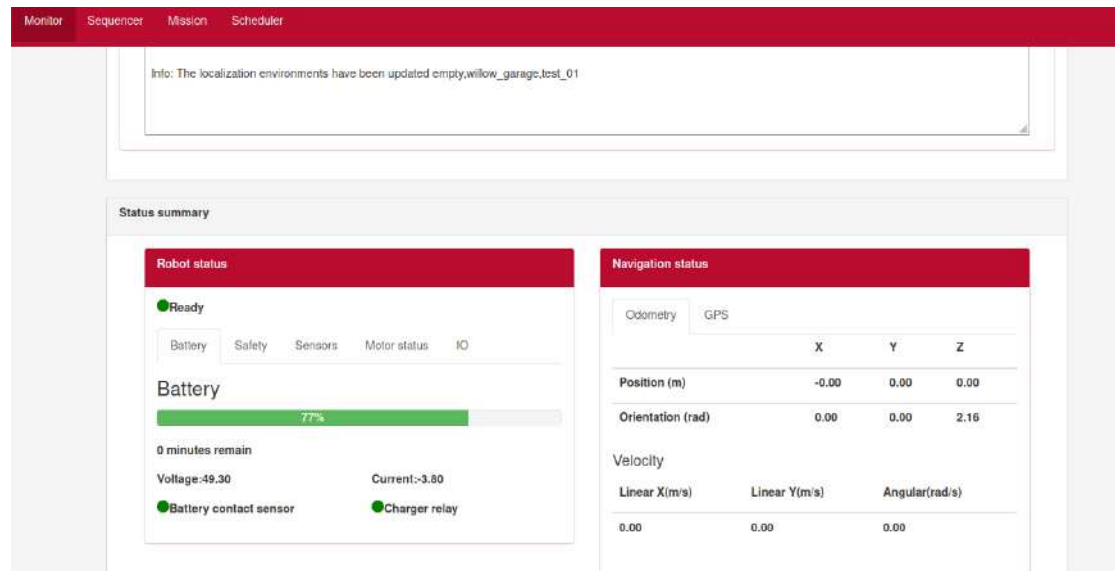


FIGURE 12 : Status summary

### 2.1.3 System Monitor

La section « System Monitor » de l'onglet Monitor nous donne le pourcentage d'utilisation des 16 cœurs du CPU, la température des 4 clusters de cœur ainsi que le pourcentage d'utilisation de la mémoire.

System Monitor

CPU (Usage)

Name	Status	Usage
CPU-1	OK	36.0%
CPU-2	OK	44.0%
CPU-3	OK	35.0%
CPU-4	OK	32.0%
CPU-5	OK	48.0%
CPU-6	OK	36.0%
CPU-7	OK	33.0%
CPU-8	OK	46.0%
CPU-9	OK	49.0%
CPU-10	OK	42.0%
CPU-11	OK	38.0%
CPU-12	OK	44.0%
CPU-13	OK	37.0%
CPU-14	OK	35.0%
CPU-15	OK	83.0%
CPU-16	OK	39.0%

CPU (Temperature)

Name	Temperature(°C)
Core-1	56

Memory (Usage)

Name	Free	Total	Used
Physical	22864MB	31923MB	6197MB
Swap	2047MB	2047MB	0MB
Physical w/o buffers	24651MB	0MB	2861MB

HDD (Usage)

Name	Status	Size	Use	Available
/dev/nvme0n1p2	OK	457GB	7GB	405GB

FIGURE 13 : System Monitor

## 2.2 Partie Sequencer

L'onglet « Sequencer » nous permet d'enregistrer des séquences pour automatiser le déplacement du robot. C'est un système d'assemblage de fonctions qui prend la forme de blocs (à la manière de Scratch).

-Le bloc "GOTO" permet d'enregistrer une position à partir d'un référentiel de base. Il suffit de déplacer le robot à la position voulue et de l'enregistrer. Le robot trouvera de lui-même le moyen de rejoindre cette position.

-Le bloc "DOCK" permet de demander au robot de rejoindre automatiquement la station de charge.

-Le bloc "WAIT" permet de stopper le robot pendant une période demandée.

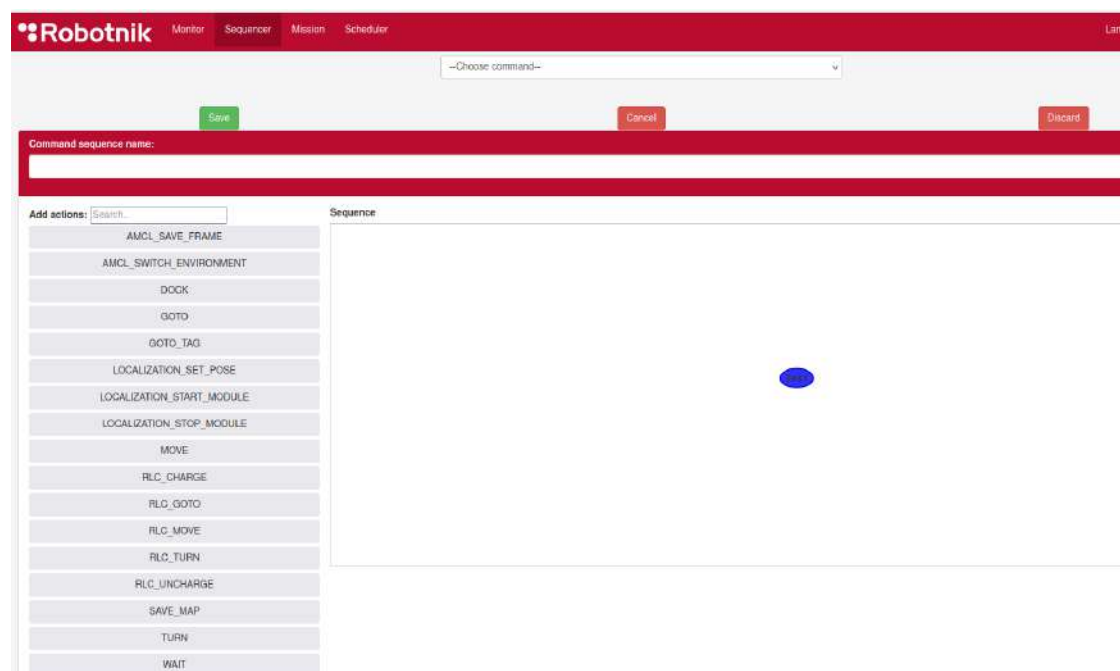


FIGURE 14 : Création d'une séquence

## 3 Manipulation du Vogui

### 3.1 Contrôle du robot

Le robot RB-Vogui-XL possède 4 roues omnidirectionnelles, ce qui lui permet de se déplacer dans toutes les directions. On déplace le robot à l'aide d'une manette de PS4 connectée en Bluetooth via un dongle Bluetooth USB.



FIGURE 15 : Dongle bluetooth usb

Vous pouvez voir ci-dessous à quoi correspondent les différentes touches de la manette. Si vous voulez piloter le robot, il faut savoir qu'il existe plusieurs sécurités :

- le bouton deadman : il n'est possible de déplacer le robot que lorsque l'on maintient le bouton R1 de la manette enfoncé.
- l'arrêt d'urgence : tant que le robot est en arrêt d'urgence, on ne peut pas le déplacer (pour désactiver l'arrêt d'urgence : cf 1.1 Allumer le robot)



FIGURE 16 : Touches de la manette

### 3.2 Connexion/Reconnexion de la manette

Une fois que le robot est démarré, on a juste à presser sur le bouton "Start" de la manette.

- Si la barre lumineuse s'allume en bleu, cela veut dire que la manette s'est bien connectée, vous pouvez déplacer le robot
- Si la barre lumineuse clignote en blanc, cela veut dire que la manette ne s'est pas bien appairée. Réessayez une fois pour en être sûr. Si ça ne fonctionne toujours pas, suivre les étapes suivantes :

Sur l'ordi du robot, aller dans "Settings" et cliquez sur l'onglet "Bluetooth". Le nom de la manette est "Wireless Controller" et son état est probablement "Disconnected". Cliquer sur cette ligne puis sur le bouton "Remove Device".

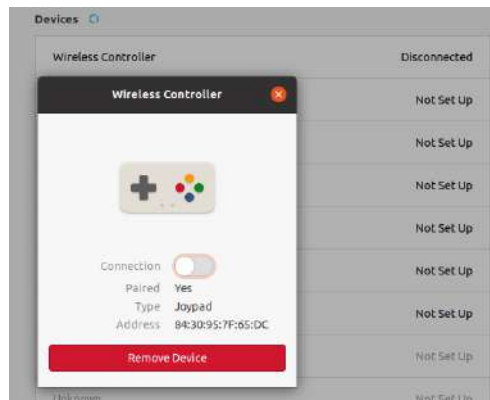


FIGURE 17 : Remove Device de la manette

Ensuite, dans les paramètres, désactiver puis réactiver le Bluetooth. Dans la foulée, maintenir simultanément les boutons "Start" et "Share" sur la manette. La barre lumineuse clignote rapidement en blanc. Vous devriez voir apparaître de nouveau "Wireless Controller" à l'état "Not Set Up". Il suffit de presser sur cette ligne et la manette devrait se connecter (il faut que la manette soit en train de clignoter pour réussir l'appairage).

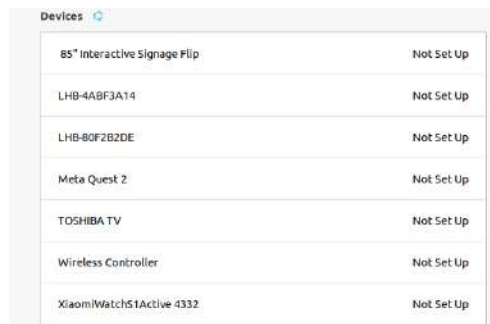


FIGURE 18 : Setup de la manette

### 3.3 Contrôle du bras

Le robot RB-Vogui-XL est équipé d'un bras robotique de type 'cobot'. À l'heure de l'écriture de ce document, il y a deux possibilités pour le contrôler : via l'application intégrée (Remmina) ou via le programme réalisé par les étudiants permettant de le contrôler directement via la manette de PS4 du robot.

#### 3.3.1 Allumer le bras

Avant de manipuler le bras, il faut d'abord l'allumer. Il faut simplement presser sur le bouton "ON ARM".



FIGURE 19 : bouton "ON ARM"

Il va s'allumer en vert. Le processus de démarrage est assez long, attendez 5 minutes avant d'aller plus loin. Une fois cette période passée, nous allons pouvoir avoir accès à l'IMH (Interface Homme Machine) du bras.



### 3.3.2 Contrôler avec l'interface Remmina

La manière la plus simple et intuitive de contrôler le bras est d'utiliser le logiciel dédié. Pour cela, vous pouvez ouvrir l'application *Remmina* (l'application est installée dans l'ordinateur du robot).



FIGURE 20 : icône de Remmina

Lorsque vous allez ouvrir l'application, il va vous être demandé d'entrer un mot de passe :

R0b0tn1K

Vous pourrez par la suite entrer l'IP du robot : *192.168.0.210*  
Il faudra probablement entrer à nouveau le mot de passe.

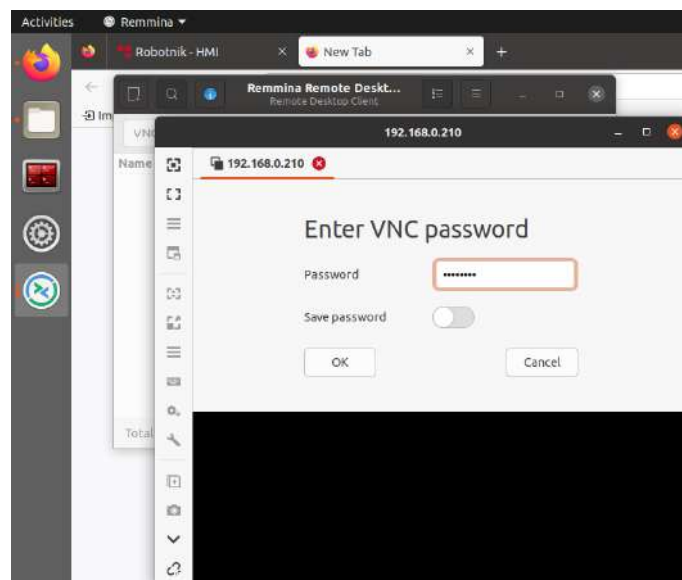


FIGURE 21 : Connexion au bras

Une fois connecté, vous arriverez sur la page principale de l'application, sur laquelle vous pourrez choisir le mode de contrôle en haut à droite (ici, veuillez à être en 'automatic'), les différents mouvements possibles (flèches), le choix du repère de déplacements (outils, base, monde).

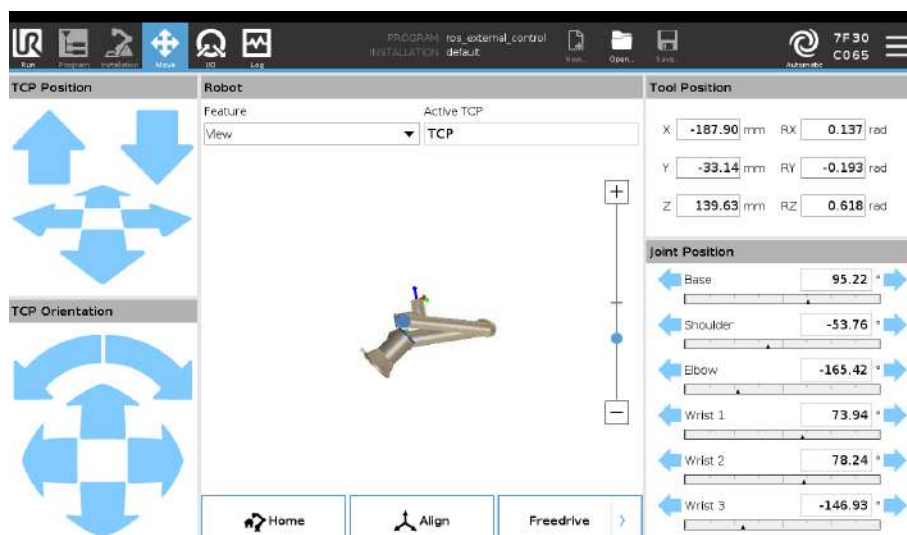


FIGURE 22 : interface de Remmina

Avant de pouvoir le déplacer, si lors du dernier arrêt le robot était en mode automatique, il va falloir l'activer manuellement. Pour cela, vous verrez un bouton coloré en bas à gauche. Cliquez dessus, et une fenêtre comme ci-dessous devrait apparaître. Suivez les instructions pour activer le bras. Dans le cas où l'étape "Robot Operational" est déjà active, cela veut dire que le bras est en marche et que l'on peut l'utiliser.

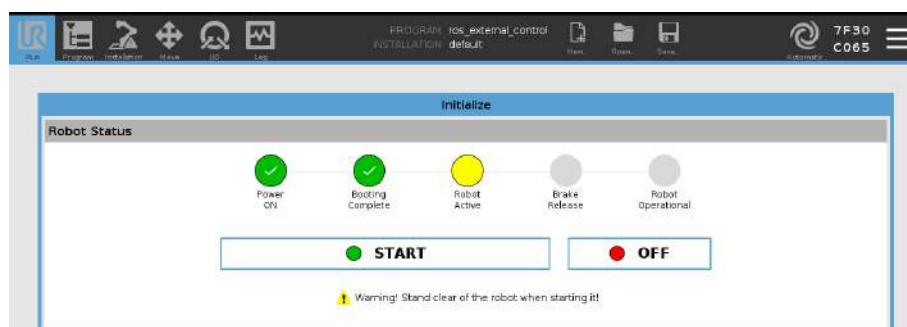


FIGURE 23 : fenetre de démarrage du bras

### Mode Freedrive :

Il est possible de déplacer "à la main" le bras, littéralement. Pour cela, il faut appuyer sur la touche 'Freedrive' sur l'interface de Remmina, et un menu apparaîtra. Ensuite, si le bouton 'Freedrive' est maintenu, il est possible de bouger le bras à votre convenance. **Attention**, veuillez le déplacer lentement, être au moins 2 lors de cette manipulation, et vérifier les valeurs des articulations avant de valider une position (valeurs lisibles à droite).

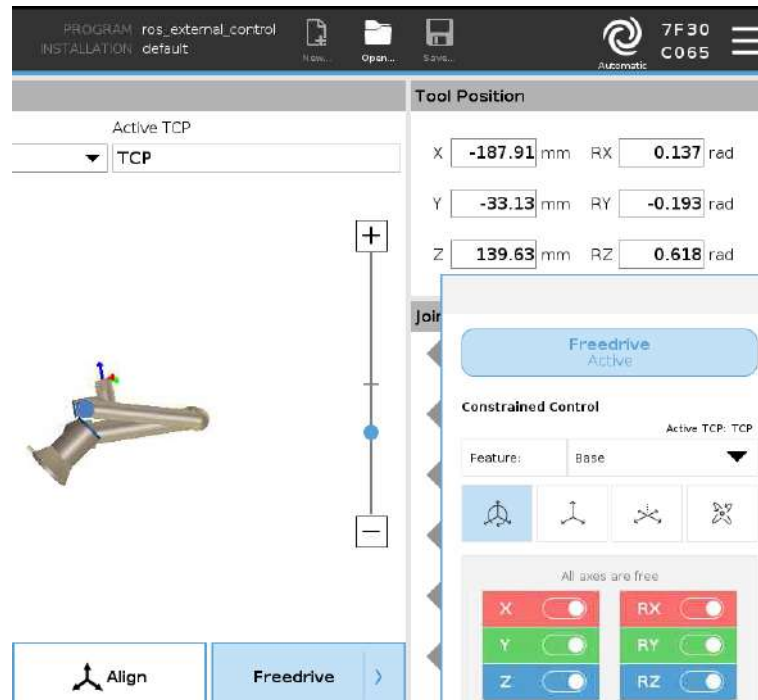


FIGURE 24 : Mode freedrive

### Contrôle de la pince avec Remmina :

Il est possible d'ouvrir et de fermer la pince Schunk installée sur le bras UR10e via le contrôle des entrées/sorties digitales. Pour ouvrir l'interface de celles-ci, cliquez sur l'icône I/O :

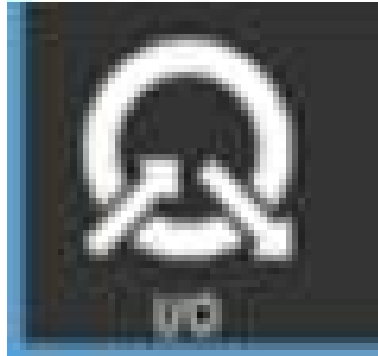


FIGURE 25 : Icône pour accéder aux entrées/sorties

Ensuite, vous pourrez contrôler chaque entrée ou sortie en cliquant sur le carré correspondant à côté. Celles qui nous intéressent sont les 'Digital Output'.

Si vous cliquez sur *0*, la pince s'ouvrira.

Si vous cliquez sur *1*, elle se fermera.

**Attention,** Veuillez décocher chaque sortie avant d'en sélectionner une autre, sinon les deux actions seront effectuées en simultané, et cela forcera sur les moteurs dans la pince.

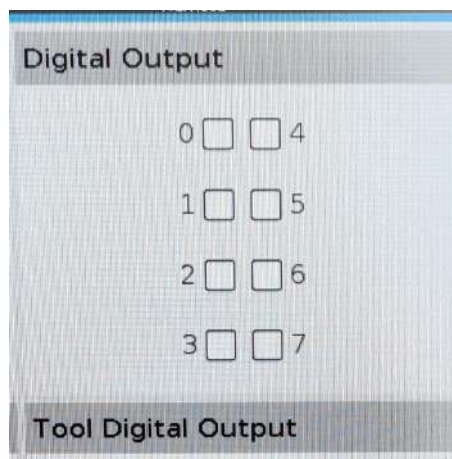


FIGURE 26 : Sorties pour contrôler la pince

### Mode Manual :

Le mode "Manual" est un mode qui permet de faire des modifications sur le bras. Pour le sélectionner, cliquer sur le logo 'Automatic' (en haut à droite dans l'interface) puis cliquer sur "Manual". Ce mode est protégé par un mot de passe. Taper "easybot" pour accéder à la page.



FIGURE 27 : Passage en mode "Remote Control"

Dans l'onglet "Safety", il est possible de faire des modifications, notamment au niveau des limites de contraintes du bras. Le bras est de type "cobo" donc il existe une sécurité dans le cas d'un contact avec un obstacle. Mais si on décide de tolérer une certaine résistance (par exemple dans le cas d'un mouvement de poussée), on peut modifier les limites dans l'onglet "Robot Limits".

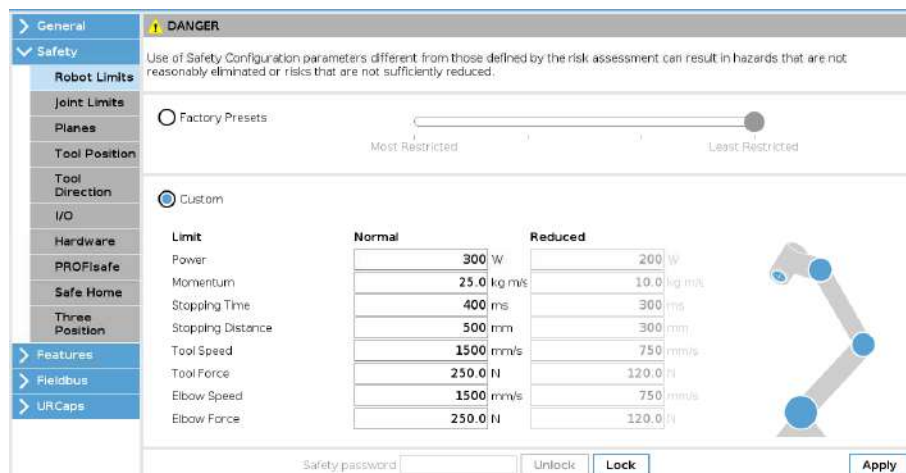


FIGURE 28 : Modifications dans le mode Manual

Note : Pour sortir du mode "Manual", cliquer sur le logo 'Manual' (en haut à droite dans l'interface) puis cliquer sur "Automatic". Il faudra de nouveau taper le mot de passe "easybot".

## 4 Fonctionnalités du robot

### 4.1 Principe de ROS

Le robot fonctionne avec le système ROS (Robot Operating System) : il permet de développer, gérer et faire communiquer les différents composants du robot de manière modulaire.

### 4.2 Qu'est-ce que catkin\_ws ?

Dans ROS, le workspace est un environnement de développement où l'on crée, compile et gère les différents composants logiciels du robot. Le workspace par défaut s'appelle généralement `catkin_ws`

Dans l'environnement de travail, on retrouve :

- `src/` Contient les packages ROS
- `build/` Dossier généré après compilation
- `devel/` Contient les exécutable et les configurations

Avant d'utiliser les fichiers compilés, il est important "sourcer" le dossier `devel` :

```
source devel/setup.bash
```

### 4.3 Qu'est-ce qu'un package ROS ?

Un package ROS (pkg) est une unité qui contient tout le nécessaire pour une fonctionnalité : scripts, fichiers de configuration, dépendances, etc. On peut en créer autant que l'on veut,

Chaque package peut contenir :

- Des nœuds (nodes) : les programmes exécutés.
- Des fichiers launch : pour lancer plusieurs nœuds en même temps.
- Des fichiers de configuration : YAML, URDF, etc.
- Des dépendances à d'autres packages.

Chaque package contient au minimum deux fichiers essentiels :

- `package.xml` : décrit le package (nom, version, dépendances).
- `CMakeLists.txt` : configuration de la compilation.

### 4.4 Création d'un package

Depuis le dossier `src/` :

```
cd /catkin_ws/src
catkin_create_pkg nom_du_package std_msgs rospy roscpp
```

Ajoutez ensuite vos scripts (par ex. dans un dossier `scripts/`) et rendez-les exécutable :

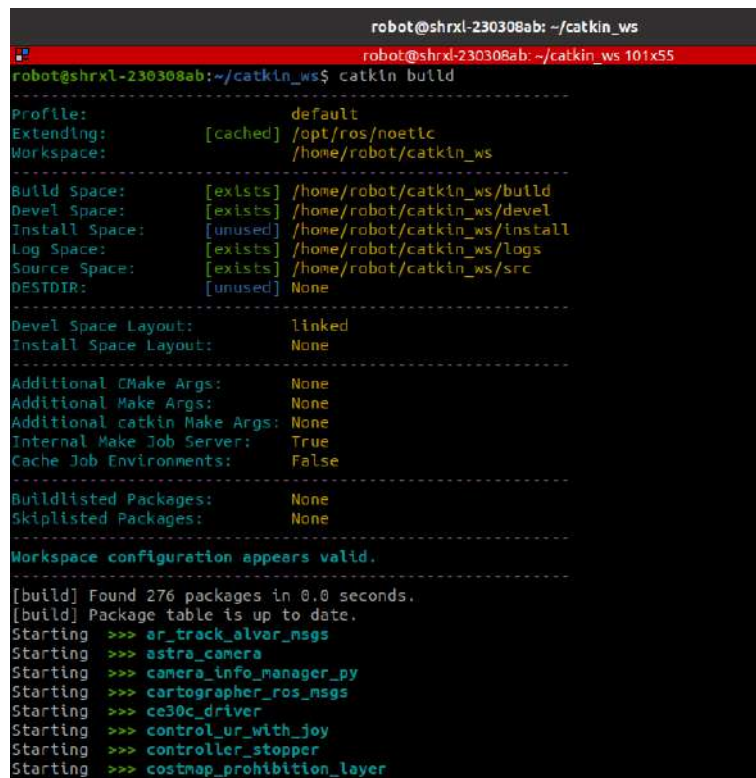
```
chmod +x mon_script.py
```

## 4.5 Compilation

Une fois les packages ajoutés ou modifiés, on compile l'ensemble. ATTENTION : il faut lancer cette commande uniquement dans le répertoire racine "catkin\_ws". Pour cela, il suffit de taper cette commande :

```
cd catkin_ws
```

Cela permet de mettre à jour les dossiers build/ et devel/.



```
robot@shrxl-230308ab: ~/catkin_ws
robot@shrxl-230308ab: ~/catkin_ws 101x55
robot@shrxl-230308ab:~/catkin_ws$ catkin build
-----
Profile:                        default
Extending:                      [cached] /opt/ros/noetic
Workspace:                      /home/robot/catkin_ws
-----
Build Space:                    [exists] /home/robot/catkin_ws/build
Devel Space:                    [exists] /home/robot/catkin_ws/devel
Install Space:                  [unused] /home/robot/catkin_ws/install
Log Space:                      [exists] /home/robot/catkin_ws/logs
Source Space:                   [exists] /home/robot/catkin_ws/src
DESTDIR:                        [unused] None
-----
Devel Space Layout:             linked
Install Space Layout:           None
-----
Additional CMake Args:          None
Additional Make Args:           None
Additional catkin Make Args:     None
Internal Make Job Server:       True
Cache Job Environments:         False
-----
Buildlisted Packages:           None
Skiplist Packages:              None
-----
Workspace configuration appears valid.
-----
[build] Found 276 packages in 0.0 seconds.
[build] Package table is up to date.
Starting >>> ar_track_alvar_msgs
Starting >>> astro_camera
Starting >>> camera_info_manager_py
Starting >>> cartographer_ros_msgs
Starting >>> ce30c_driver
Starting >>> control_ur_with_joy
Starting >>> controller_stopper
Starting >>> costmap_prohibition_layer
```

FIGURE 29 : Lancement de la commande catkin build

Après cette étape, on doit "sourcer" notre environnement, c'est-à-dire exécuter un fichier de configuration. Si on oublie de sourcer, les modifications que l'on a apportées à un fichier ne sont pas prises en compte, un package créé peut ne pas être visible... cela doit devenir un réflexe quand on modifie quelque chose dans l'environnement ROS. Quand on ouvre un nouveau terminal, il est vierge donc on le "source" à nouveau.

Pour "sourcer", il suffit simplement de taper dans le terminal (toujours dans le répertoire racine "catkin\_ws") :

```

Starting >>> ur_calibration
Finished <<< ur_gripper_controller [ 0.2 seconds ]
Starting >>> robot_bringup
Finished <<< rbkairos_sim [ 0.2 seconds ]
Finished <<< rb_theron_sim_bringup [ 0.2 seconds ]
Finished <<< universal_robots [ 0.2 seconds ]
Finished <<< robot_bringup [ 0.3 seconds ]
Finished <<< ur_calibration [ 0.6 seconds ]
[build] Summary: All 276 packages succeeded!
[build] Ignored: None.
[build] Warnings: 2 packages succeeded with warnings.
[build] Abandoned: None.
[build] Failed: None.
[build] Runtime: 43.3 seconds total.
robot@shrxl-230308ab:~/catkin_ws$ source devel/setup.bash

```

FIGURE 30 : Lancement de la commande source

## 4.6 Exécution d'un nœud

Après avoir compilé et sourcé, on est prêt à lancer des nœuds :

```
roslaunch nom_du_package nom_du_script.py
```

Ou via un fichier de lancement (fichier qui permet de lancer plusieurs nœuds en même temps) :

```
roslaunch nom_du_package fichier.launch
```

## 4.7 Communication entre packages : les topics et les nœuds

Les packages ROS communiquent entre eux via un système de nœuds et de topics : pour voir cela de manière plus schématique, on peut visualiser leurs relations à l'intérieur du robot avec l'outil "rqt\_graph". Pour cela, il suffit simplement de taper "rqt\_graph" dans le terminal.

Pour observer le schéma complet, il faut sélectionner l'onglet "Nodes/topics (all)" et cliquer sur la flèche bleue :

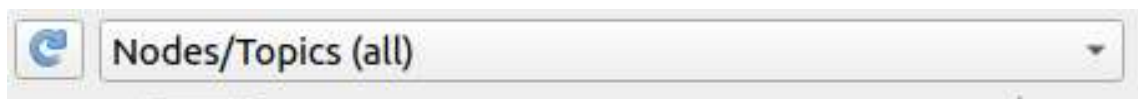


FIGURE 31 : Actualisation de rqt\_graph





```

robot@shrxl-230308ab:~/catkin_ws$ rostopic echo /robot/joy
header:
  seq: 77754
  stamp:
    secs: 1749821738
    nsecs: 124282486
  frame_id: "/dev/input/js_base"
axes: [-0.7086175680160522, 0.8252121806144714, -0.0, 0.0, 0.0, -0.0, -0.0, -0.0, -0.41720650573349, 0.0, 0.0, -0.0, -0.0, -0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 77755
  stamp:
    secs: 1749821738
    nsecs: 126057478
  frame_id: "/dev/input/js_base"
axes: [-0.7086175680160522, 0.8155017495155334, -0.0, 0.0, 0.0, -0.0, -0.0, -0.0, -0.4185183644294739, 0.0, 0.0, -0.0, -0.0, -0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 77756
  stamp:
    secs: 1749821738
    nsecs: 131754318
  frame_id: "/dev/input/js_base"
axes: [-0.7086175680160522, 0.8155017495155334, -0.0, 0.0, 0.0, -0.0, -0.0, -0.0, -0.4209459722042084, 0.0, 0.0, -0.0, -0.0, -0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---

```

FIGURE 33 : Commande rostopic echo /robot/joy

La commande "rostopic info /robot/joy" nous montre comment ce topic interagit avec d'autres noeuds.

On voit que :

- le noeud "/robot/pad\_teleop" publie des données sur le topic /robot/joy
- le noeud "/robot/choice\_joy" lit les données du topic /robot/joy, on dit qu'il est abonné au topic

```

^Crobot@shrxl-230308ab:~/catkin_ws$ rostopic info /robot/joy
Type: sensor_msgs/Joy

Publishers:
 * /robot/pad_teleop (http://shrxl-230308ab:32985/)

Subscribers:
 * /robot/choice_joy (http://shrxl-230308ab:36769/)

```

FIGURE 34 : Commande rostopic info /robot/joy

Cette relation entre ces deux noeuds et ce topic peut être visualisée directement avec "rqt\_graph" :

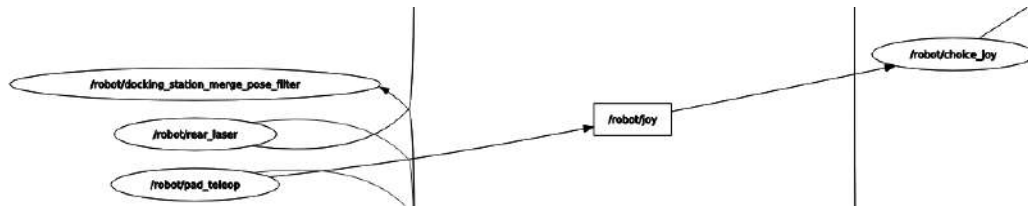


FIGURE 35 : Visualisation de `/robot/joy` dans `rqt_graph`

Pour plus d'infos vous pouvez aller sur ce site : <https://wiki.ros.org/catkin>

## 5 Démonstration

### 5.1 Contrôle du bras avec la manette

Il est possible de déplacer le robot au démarrage avec la manette, mais nous avons ajouté une fonctionnalité qui permet à la fois de piloter la base mobile et le bras via la manette de PS4.

#### 5.1.1 Etape 1 : Démarrage du robot et du bras

Pour commencer, il faut s'assurer que le robot et le bras sont bien allumés. Pour ce faire, il faut simplement suivre les étapes expliquées dans la partie "1. Démarrage du robot". Une fois que le robot est prêt à l'emploi, il faut regarder si vous pouvez le contrôler avec la manette. Dans le cas où la manette n'est pas connectée, regardez la partie "3.2 Connexion/Reconnexion de la manette". Si vous pouvez contrôler le robot, vous pouvez démarrer le bras (suivre les étapes de la partie "3.3 Contrôle du bras").

#### 5.1.2 Etape 2 : Vérification de l'interface Remmina

Avant de lancer le programme, il faut s'assurer que le bras soit bien en mode 'Remote control'. Dans l'interface, cliquez sur le logo 'automatic' (en haut à droite) puis cliquez sur "Remote control".



FIGURE 36 : Passage en mode "Remote Control"

#### 5.1.3 Etape 3 : Connexion au robot en SSH

Pour plus de praticité, il est préférable de lancer des commandes à distance. Le SSH permet de se connecter à un serveur ou à un autre ordinateur distant pour y exécuter des commandes. Pour pouvoir se connecter, il faut ouvrir un terminal Linux. Si vous avez un PC qui est sous Windows, il faut installer un WSL comme Ubuntu (nous avons utilisé la version 20.04.6 LTS).

Voici les étapes à suivre :

- Se connecter en wifi au réseau « SHXRL-230308AB\_2G » avec le mdp « R0b0tn1K »
- Ouvrir un terminal Linux et taper la commande : `ssh -X -C robot@192.168.0.200`

- Tapez le mdp « R0b0tn1K » (c'est normal que vous ne voyez pas ce que vous écrivez, le mot de passe reste invisible). Tapez "Entrée" et vous devriez voir apparaître ceci :

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Jun 16 14:33:51 2025 from 192.168.0.133
->Loading robot_params from /home/robot/robot_params
All the env variables configuration are located at /home/robot/robot_params
-> Read /home/robot/robot_params/bringup.env
-> Read /home/robot/robot_params/robot_params.env
-> Read /home/robot/robot_params/sensors_params.env
-> Read /home/robot/robot_params/battery_params.env
-> Read /home/robot/robot_params/localization_params.env
-> Read /home/robot/robot_params/manipulation_params.env
-> Read /home/robot/robot_params/navigation_params.env
-> Read /home/robot/robot_params/perception_params.env
-> Read /home/robot/robot_params/rlc_params.env
-> Read /home/robot/robot_params/simulation_params.env
-> Read /home/robot/robot_params/hmi_params.env
-> Read /home/robot/robot_params/interfaces_params.env
-> Read /home/robot/robot_params/safety_params.env
-> Read /home/robot/robot_params/external_machine_params.env
-> Read /opt/ros/noetic/setup.bash
-> Robot workspace is /home/robot/catkin_ws
-> Read /home/robot/catkin_ws/devel/setup.bash

ROS_MASTER_URI   = http://shrxl-230308ab:11311
ROS_HOSTNAME     = shrxl-230308ab
WORKSPACE        = /home/robot/catkin_ws/devel/setup.bash

robot@shrxl-230308ab:~$ |
```

FIGURE 37 : Interface ssh

#### 5.1.4 Etape 4 : Lancement du programme

Arrivé à cette étape, tout est prêt ! Il vous suffit simplement d'ouvrir un terminal et de taper cette commande :

```
roslaunch control_ur_with_joy ur_joy_control_origine.py
```

Normalement, vous devriez avoir un message de confirmation de connexion au robot :

```
robot@shrxl-230308ab:~$ roslaunch control_ur_with_joy ur_joy_control_origine.py
[INFO] [2025-03-11 10:49:30]: Connexion réussie au robot.
```

FIGURE 38 : Message de validation de la connexion au bras

A cet instant, il est toujours possible de déplacer le robot. Si vous appuyez sur le bouton 'option' de la manette (voir détails dans la prochaine section), vous passerez en mode 'contrôle du bras'. Les messages d'envoi de commande vont alors être affichés :

```
robot@shrd-230308ab:~ 80x24
[INFO] [2025-03-11 10:49:58]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42399781942367554, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:49:59]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:49:59]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42358165979385376, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:49:59]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:49:59]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42413654923439826, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:49:59]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:49:59]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
4233042001724243, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:49:59]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:50:00]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
4238590896129608, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:50:00]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:50:00]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42337357997894287, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:50:00]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:50:00]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42337357997894287, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-03-11 10:50:00]: 🚀 Réactivation de l'écoute du joystick...
[INFO] [2025-03-11 10:50:00]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.0, -0.0)
42392846941947937, 0.0, 0.0, -0.0, -0.0, -0.0)
```

FIGURE 39 : Écoute d'envoi des commandes du bras

#### 5.1.5 Etape 4 bis : Lancement de la démo

Dans le cas où vous voulez faire une démo à des visiteurs, passez l'étape 4. Veillez à ce que les étapes 1 à 3 soient bien respectées. Ouvrez un terminal et tapez cette commande :

`./demo_vogui.sh`

Cela lancera à la fois la commande pour bouger le bras de l'étape 4 et le logiciel RViz avec une config .rviz personnalisée qui affiche d'un côté la webcam positionnée sur le bras et de l'autre côté une vue 3D interactive du robot simulé.

## 5.2 Les différentes commandes

Ci-dessous vous trouverez les différents mouvements que vous pouvez faire avec le bras et les boutons qui leur sont associés.

**Attention**, veuillez à avoir assez de place autour du robot. Ne manipulez pas le bras seul. À ce jour, aucune zone morte n'a été définie au bras, il n'est donc pas conscient de la présence de la base du Vogui, il est possible de taper dedans ou dans d'autres objets.

Touches attribuées à la manette :



FIGURE 40 : Touches de la manette pour le contrôle du bras

Positions pré-enregistrées :

**Position de base :**

Pour amener le bras en position de base, vous pouvez appuyer sur le bouton *carré*.



FIGURE 41 : Position de base



**Position haute :** ATTENTION! Prévoir un espace au-dessus du robot (minimum un mètre)

Pour amener le bras en position haute, vous pouvez appuyer sur le bouton *croix*.



FIGURE 42 : Position haute

**Position d'appui gauche :** ATTENTION! Prévoir un espace à gauche du robot (minimum un mètre)

Pour amener le bras en position appui gauche, vous pouvez appuyer sur le bouton *stick gauche*.



FIGURE 43 : Position appui gauche

**Position d'appui droit :** ATTENTION! Prévoir un espace à droite du robot (minimum un mètre)

Pour amener le bras en position appui droit, vous pouvez appuyer sur le bouton *stick droit*.



FIGURE 44 : Position appui droit

#### Contrôle de la pince avec la manette :

Il est aussi possible de contrôler l'ouverture/fermeture de la pince avec la manette. Pour cela, il vous suffit d'appuyer sur le bouton *triangle*. Il est déconseillé d'appuyer trop rapidement sur le bouton, même si théoriquement cela ne devrait pas être problématique, puisqu'un délai a été intégré dans le code.

**Attention**, ne touchez pas la pince lors de son fonctionnement. Ne mettez pas vos doigts ou d'objets fragiles entre les mors lors de la fermeture.



FIGURE 45 : Image de la pince (fermée)

### 5.2.1 Déplacements "manuels"

En plus des positions pré-enregistrées, il est possible de déplacer la bride du robot comme bon vous semble. Pour cela, vous devrez utiliser les boutons et joysticks de la manette. Les déplacements se font à l'aide d'une translation effectuée avec une commande de type *movej*, ce qui permet d'éviter certaines singularités. Cependant, si le bras se bloque, heurte un obstacle ou atteint une singularité, il faut le réinitialiser via l'interface Remmina. Pour cela, il suffit de suivre les instructions qui apparaîtront automatiquement sur celle-ci.

Ces déplacements s'effectuent dans le repère cartésien de la bride, représentés ci-dessous :

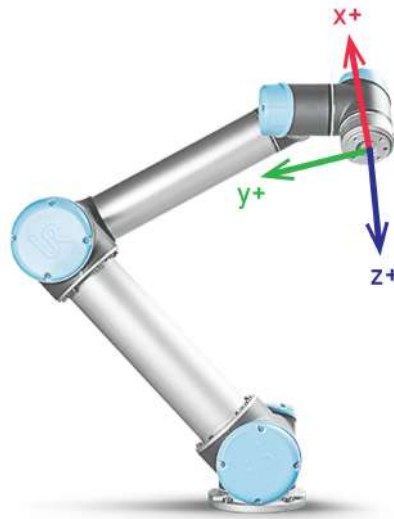


FIGURE 46 : Repère Tool utilisé pour les déplacements

Pour vous aider à déterminer l'axe X, il est approximativement perpendiculaire avec le dessus de la pince. Si la configuration d'installation de la caméra utilisée pour le projet CoHoMa III de 2025 est inchangée, cet axe est orienté vers le haut lorsque la caméra est parallèle au sol et à l'endroit (caméra au-dessus de la bride, comme visible sur l'image de la pince précédente).

Les déplacements ont été définis de telle sorte à ce qu'ils soient au maximum intuitifs lors de la téléopération du bras.

Chaque mouvement de translation linéaire entraînera un déplacement de 2 cm dans la direction voulue, et chaque rotation 0.10 radians selon l'angle voulu. Si une nouvelle commande est envoyée avant que le bras ait eu le temps de terminer le mouvement précédent, celui-ci s'arrêtera et entamera le nouveau mouvement demandé. Ceci est aussi valable pour les mouvements pré-enregistrés.

#### Déplacement selon Z :

Pour **avancer** ou **reculer** la bride, il faut respectivement mettre le *stick gauche vers l'avant* ou le *stick gauche vers l'arrière*.

#### Déplacement selon Y :

Pour décaler la bride sur la **gauche** ou sur la **droite**, il faut respectivement mettre le *stick gauche sur la gauche* ou le *stick gauche sur la droite* .

#### Déplacement selon X :

Pour **monter** ou **descendre** la bride (si l'on se place du point de vue de la caméra, cela correspond à +X et -X), il faut respectivement appuyer sur le *bouton croix haut* ou *bouton croix bas*

#### Rotation selon Z :

Pour effectuer une rotation selon Z, ce qui correspond à faire **incliner la bride à gauche** ou **incliner la bride à droite** du point de vue de la caméra, il faut appuyer respectivement sur le *bouton croix gauche* ou le *bouton croix droite*.

#### Rotation selon Y :

Pour effectuer une rotation selon Y, ce qui correspond à faire **tourner la bride vers le haut** ou **tourner la bride vers le bas** du point de vue de la caméra, il faut respectivement mettre le *joystick droit en vers le haut* ou le *joystick droit vers le bas*.

#### Rotation selon X :

Pour effectuer une rotation selon Y, ce qui correspond à faire **tourner la bride à gauche** ou **tourner la bride à droite** du point de vue de la caméra, il faut respectivement mettre le *joystick droit en vers la gauche* ou le *joystick droit vers la droite*.

### **5.2.2 Arrêt du programme**

Pour arrêter le programme qui contrôle le bras, il vous suffit d'appuyer sur *rond*. Ce bouton agit comme un arrêt d'urgence sur la manette et stoppe tout envoi de nouvelle commande.



### 5.3 Solutions en cas de problèmes rencontrés

Il est possible que certains problèmes interviennent voici une liste non exhaustive :

- Cas où vous avez activé l'arrêt d'urgence alors que le programme était lancé : le bras se met automatiquement en arrêt d'urgence. Il faut d'abord commencer par appuyer sur le bouton "OFF ARM" sur le robot. Quand la lumière du bouton "ON ARM" est éteinte, appuyer dessus pour relancer le bras. Ensuite, se rendre sur l'interface Remmina pour remettre en route le bras (cf 3.3.2 Contrôler avec l'interface Remmina).
- Cas où le bras n'est pas en arrêt d'urgence mais que le bras ne bouge pas : il est possible que le programme lancé s'arrête brutalement.

```
[INFO] [2025-06-20 14:28:44]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.4411991536617279, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-06-20 14:28:44]: Réactivation de l'écoute du joystick...
[INFO] [2025-06-20 14:28:44]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.408599853515625, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-06-20 14:28:45]: Réactivation de l'écoute du joystick...
[INFO] [2025-06-20 14:28:45]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, -0.4257318079471588, 0.0, 0.0, -0.0, -0.0, -0.0)
[INFO] [2025-06-20 14:28:45]: Réactivation de l'écoute du joystick...
[INFO] [2025-06-20 14:28:45]: (-0.0, -0.0, -0.0, 1.0, 1.0, -0.0, -0.0, -0.0, 0.0009002990555018187, -0.34695092041949463, 0.0, 0.0, -0.0, -0.0, -0.0)
[ERROR] [2025-06-20 14:28:45]: KeyboardInterrupt: Function joy_callback at 0x7f9c0c000000
Traceback (most recent call last):
  File "/opt/ros/noetic/lib/python3/dist-packages/rmpg/topics.py", line 799, in _invoke_callback
    cb(msg)
  File "/home/robot/catkin_ws/src/robot/packages/control_robot_with_joy/src/joy_control_origine.py", line 120, in joy_callback
    if statePin[0].state == False: #verifier le stat du pin 0
NameError: name 'statePin' is not defined
^[[INFO] [2025-06-20 14:28:07]: Connexion fermée.
robot@shrs1-230308ab: $ Connection to 192.168.8.200 closed by remote host.
Connection to 192.168.8.200 closed.
robot@shrs1-230308ab: $
```

FIGURE 47 : Exemple de crash possible

Dans ce cas, tenter d'arrêter le programme en faisant la commande CTRL + C puis relancer. Il est aussi possible que votre PC ait été déconnecté du wifi du routeur du robot. Dans ce cas, se reconnecter comme expliqué (cf 5.1.3 Connexion au robot en SSH).

## 6 Utilisation de Rviz

RViz (ROS Visualization) est un outil graphique qui vous permet de visualiser et d'interagir avec les différentes données du robot en temps réel. RViz permet de visualiser la représentation 3D du robot, sa position sur une carte, de visualiser certaines données provenant de capteurs (comme le LiDAR), et de lui donner des ordres de navigation.

### 6.1 Lancement de RViz

Pour lancer RViz depuis un terminal connecté au robot (ou via SSH), on utilise la commande suivante :

```
rviz
```

Une fois RViz ouvert, une interface graphique s'affiche avec :

- Une carte générée
- La position actuelle estimée du robot
- Les obstacles détectés par les LiDAR.

### 6.2 Présentation de l'interface

L'interface de Rviz est divisée en plusieurs parties :

- Une partie "Display". Ici, vous pouvez cocher/décocher des paramètres pour observer les modifications sur la modélisation 3D du robot.

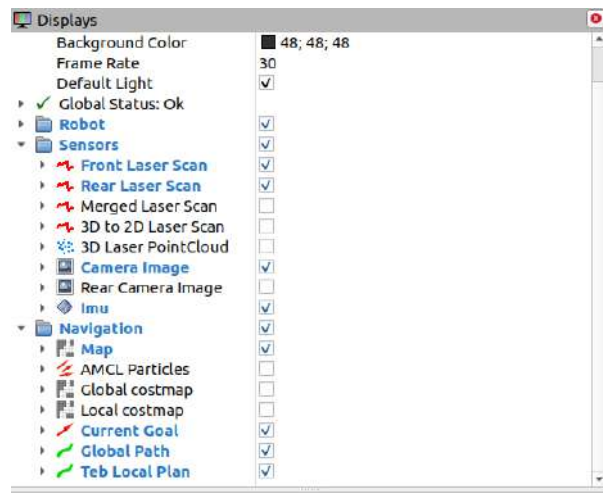


FIGURE 48 : Partie "Display"



- Une partie "Visualisation". Ici vous avez une vue 3D interactive du monde simulé. Lorsque l'on déplace le robot réel, le robot simulé se déplace également dans l'environnement.

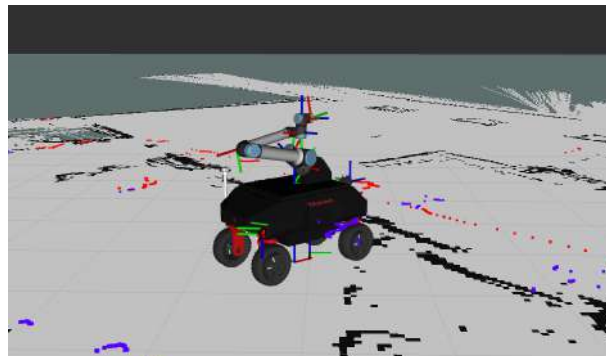


FIGURE 49 : Partie "Visualisation"

Pour que le robot soit dans la map de votre choix, il faut la sélectionner dans l'IHM du robot. De plus, par défaut, le robot simulé se place aléatoirement sur la carte virtuelle. Pour modifier manuellement sa position, il faut déplacer le curseur de pose directionnelle depuis l'IHM du robot. Tout cela est expliqué dans la partie 2.1.1 Localization panel).

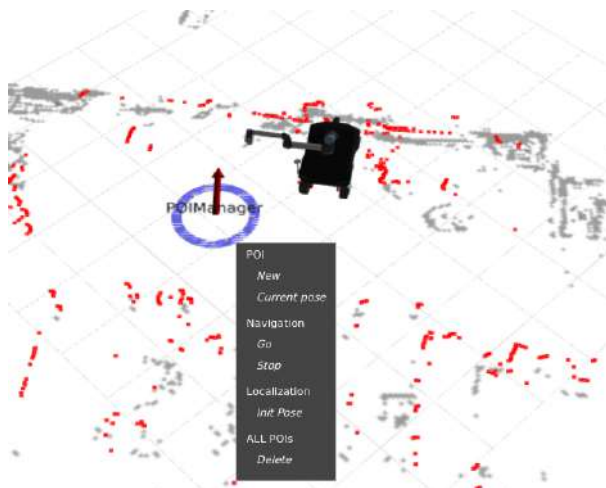
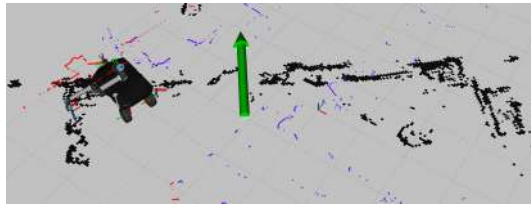


FIGURE 50 : Curseur de pose directionnelle

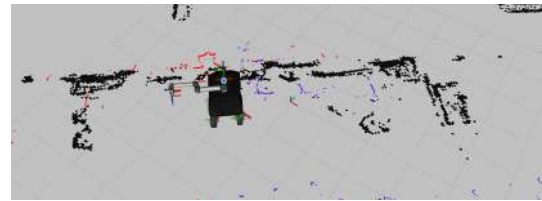
### 6.3 Estimation de la position initiale : 2D Pose Estimate

Avant de pouvoir naviguer, le robot doit connaître sa position approximative sur la carte.

1. Dans RViz, sélectionner l'outil **2D Pose Estimate** (en haut de l'interface, icône en forme de flèche bleue).
2. Cliquer sur la carte à l'endroit approximatif où se trouve physiquement le robot.
3. Maintenir le clic et orienter la flèche dans la direction dans laquelle le robot est tourné.
4. Relâcher pour envoyer l'estimation de position.



Position du robot avant



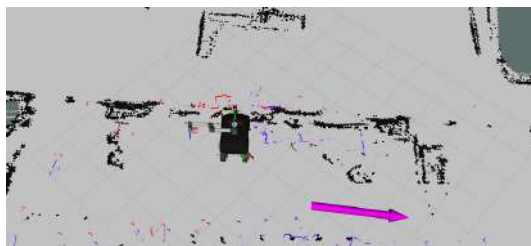
Position du robot après

### 6.4 Définir un objectif de navigation : 2D Nav Goal

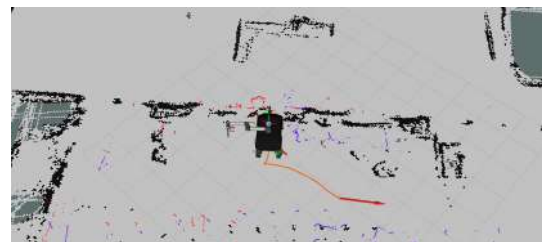
Une fois la position du robot connue, on peut lui demander de se déplacer à un point précis de la carte.

1. Sélectionner l'outil **2D Nav Goal** (icône avec une flèche verte).
2. Cliquer sur la carte à l'endroit souhaité pour que le robot s'y rende.
3. Maintenir le clic et orienter la flèche dans la direction finale désirée.
4. Relâcher pour envoyer la commande.

Le robot calculera alors un chemin en évitant les obstacles détectés et se déplacera vers la position demandée.



Position du robot avant

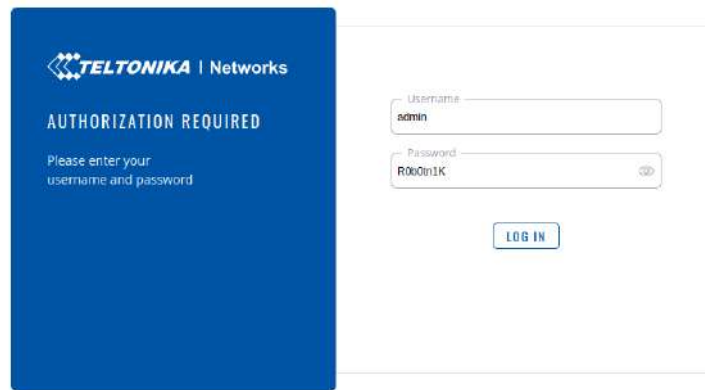


Position du robot après

## 7 Interface réseau Teltonika

Le robot la communication réseau robot embarque un routeur de la marque **Teltonika**, le modèle **RUTX11**. Ce petit boîtier permet notamment de créer un réseau local qui permet au robot et à un ou plusieurs ordinateurs de pouvoir communiquer. Une fois qu'un ordinateur est connecté, on peut ensuite accéder à l'ordinateur du robot via une connexion **SSH**, ce qui nous permet d'exécuter des commandes et des programmes à distance. Le routeur n'a pas forcément besoin d'une connexion Internet pour fonctionner.

L'interface d'administration du routeur est accessible en tapant l'adresse **192.168.0.1** dans un navigateur. Pour se connecter, il faut taper "admin" avec le mot de passe "R0b0tn1K".

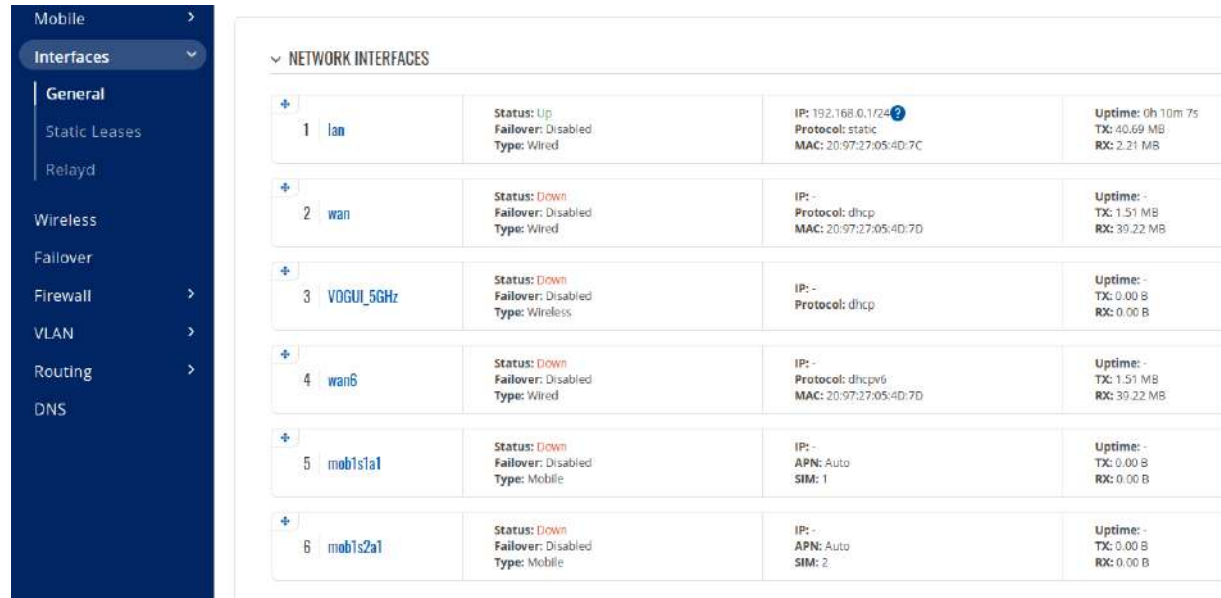


The image shows the login interface of a Teltonika RUTX11 router. On the left, a blue sidebar displays the Teltonika logo and the text "AUTHORIZATION REQUIRED" followed by "Please enter your username and password". The main content area is white and features a login form with two input fields: "Username" with the value "admin" and "Password" with the value "R0b0tn1K". A "LOG IN" button is positioned below the password field.

FIGURE 53 : Page de connexion Teltonika

## 7.1 Connecter le robot à internet

Dans l'onglet **Interfaces** puis **Général**, on peut visualiser les différentes interfaces réseaux du Vogui. Elles peuvent être locales (LAN), étendues (WAN), wifi (VOGUI\_5GHz)...



NETWORK INTERFACES			
1	lan	Status: Up Failover: Disabled Type: Wired	IP: 192.168.0.1/24 Protocol: static MAC: 20:97:27:05:4D:7C Uptime: 0h 10m 7s TX: 40.69 MB RX: 2.21 MB
2	wan	Status: Down Failover: Disabled Type: Wired	IP: - Protocol: dhcp MAC: 20:97:27:05:4D:7D Uptime: - TX: 1.51 MB RX: 39.22 MB
3	VOGUI_5GHz	Status: Down Failover: Disabled Type: Wireless	IP: - Protocol: dhcp Uptime: - TX: 0.00 B RX: 0.00 B
4	wan6	Status: Down Failover: Disabled Type: Wired	IP: - Protocol: dhcpv6 MAC: 20:97:27:05:4D:7D Uptime: - TX: 1.51 MB RX: 39.22 MB
5	mob1s1a1	Status: Down Failover: Disabled Type: Mobile	IP: - APN: Auto SIM: 1 Uptime: - TX: 0.00 B RX: 0.00 B
6	mob1s2a1	Status: Down Failover: Disabled Type: Mobile	IP: - APN: Auto SIM: 2 Uptime: - TX: 0.00 B RX: 0.00 B

FIGURE 54

Si vous voulez avoir accès à internet avec le VOGUI, branchez un câble de type RJ45 entre le routeur "VOGUI\_5GHz" (port LAN2) et le robot (port WAN).



FIGURE 55

Rafraîchir la page : le status de l'interface wan devrait être "up", avec une adresse IP attribuée dynamiquement (type 192.168.1.X)

1	lan	Status: Up Failover: Disabled Type: Wired	IP: 192.168.0.1/24 Protocol: static MAC: 20:97:27:05:4D:7C
2	wan	Status: Up Failover: Disabled Type: Wired	IP: 192.168.1.31/24 Protocol: dhcp MAC: 20:97:27:05:4D:7D
3	VOGUI_5GHz	Status: Down Failover: Disabled Type: Wireless	IP: - Protocol: dhcp

FIGURE 56

## 7.2 Interface routeur Vogui\_5GHz

On peut vérifier que le robot a accès à internet en accédant à l'interface web du routeur présent dans la pièce (Vogui\_5GHz). Pour y accéder, utiliser l'ordinateur CO-ROBOT12 présent à côté du routeur. Ouvrir le navigateur Mozilla et cliquer sur le raccourci (voir ci-dessous) :

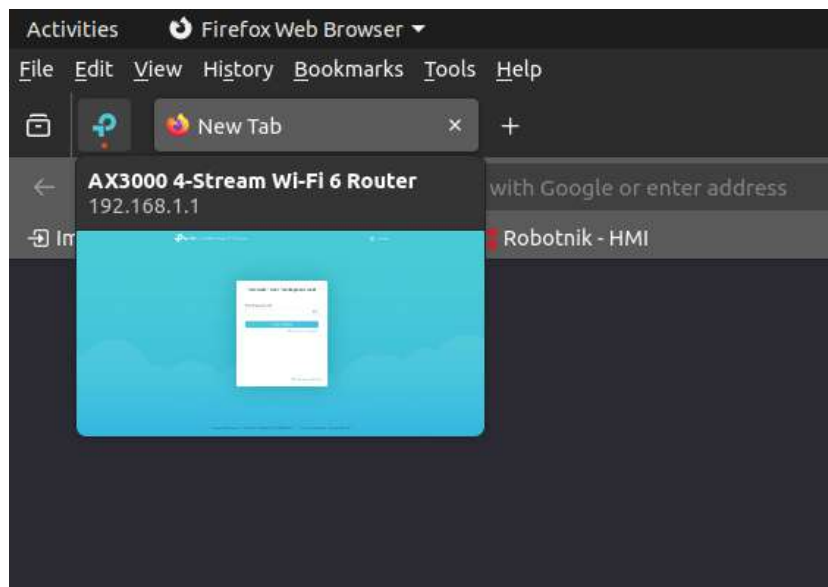


FIGURE 57

Vous arriverez sur cette page qui indique une erreur. Dans la barre de recherche, modifiez l'url en enlevant le 's' de 'https' (cela est dû à une modification de l'adresse IP du routeur).



FIGURE 58

Rentrer le mot de passe suivant pour accéder à la page web : raoul2019

**Connexion avec mot de passe local**

Mot de passe local:

  
  
  
[Mot de passe oublié ?](#)

FIGURE 59

Cliquer sur l'onglet "Cartographie du réseau" puis "Clients". On accède à une page qui montre tous les appareils connectés au routeur. Si le routeur échange bien des données avec le routeur du robot (Teltonika-RUTX11), le statut de l'interface est "wired".

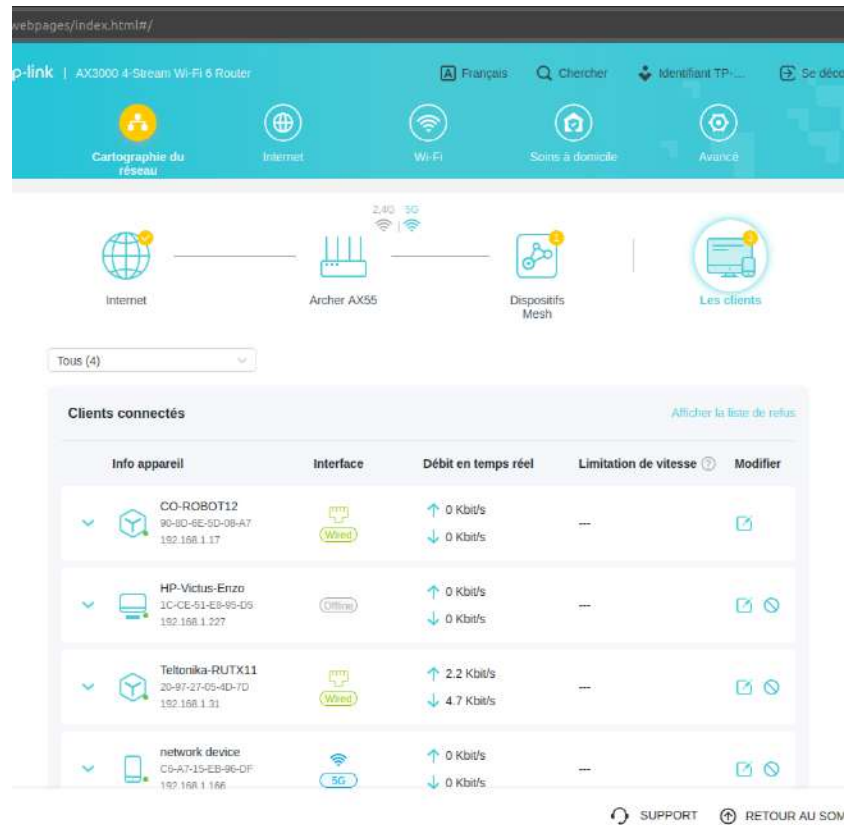


FIGURE 60

## 8 Réinitialisation du robot

Le robot fait tourner un grand nombre d'éléments (noeuds/topics/nuage de points...) et il arrive que le robot plante. Il n'est pas toujours simple de trouver des solutions pour réparer les problèmes.

Dans quels cas il faut réinitialiser le robot :

- Au démarrage, les barres lumineuses du robot clignotent en blanc indéfiniment et il n'est pas possible de déplacer le robot. Un logo rouge avec un point d'exclamation apparaît sur l'ordinateur du robot et annonce crash report « joint\_state\_publisher ». Il se peut aussi qu'un autre noeud crash, les plantages peuvent survenir de différentes manières.
- Après le démarrage, les barres lumineuses du robot clignotent en orange indéfiniment. Il se peut que l'un des capteurs du robot soit dysfonctionnel ou bien qu'il y a un problème de configuration réseau. Il est possible que le robot puisse se déplacer mais vous ne pourrez pas utiliser certaines fonctionnalités (comme le déplacement autonome).

Ce sont des problèmes que nous n'avons jamais réussi à résoudre manuellement, dans ce cas on est obligé de réinitialiser le robot. En fait, on a enregistré une image du robot avec une configuration fonctionnelle et on va en quelque sorte « écraser » la configuration actuelle pour avoir un robot de nouveau fonctionnel. Cette image est enregistrée dans un disque dur. On va utiliser une clé USB qui contient Clonezilla un logiciel de clonage de disque. Il permet de créer une image exacte d'un disque dur et de la restaurer ensuite à l'identique sur le même disque ou sur un autre.



### Protocole pour réinitialiser le robot :

Tout d'abord, il faut savoir que tous les scripts et packages qui ont été créés après la dernière réinitialisation seront effacés. Il faut donc s'assurer de pouvoir sauvegarder tous ces documents sur une clé USB. Tous les documents que nous avons faits sont disponibles directement sur un Drive (flashez le QR Code ci-dessous).



FIGURE 61 : <https://drive.google.com/drive/u/2/folders/1zofopYfFQtkzpK1xR6Z6Ghe90-izmYcq>

Pour commencer, il faut :

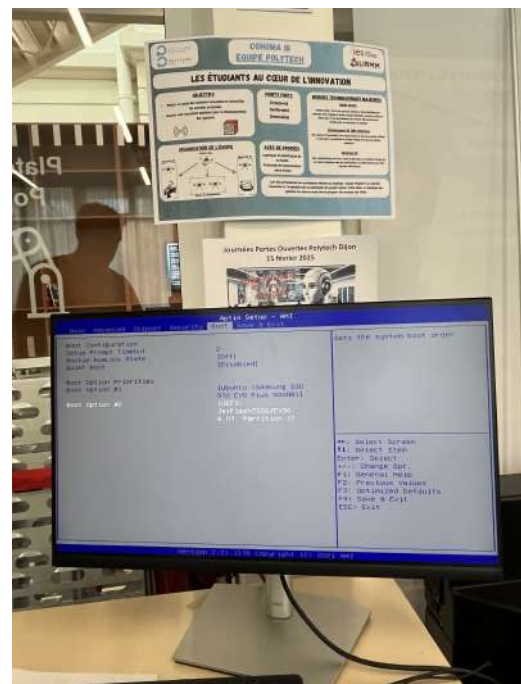
- Éteindre le robot
- S'assurer que l'écran et le clavier sont bien branché au robot
- Brancher la clé USB "Clonezilla" et le disque dur "CO-ROB-421-DDR" sur le hub USB (voir photo Étape 1)
- Tourner la clé sur "ON"
- Quand vous voyez ceci affiché à l'écran, appuyer sur la touche <Delete> ou <Esc> du clavier



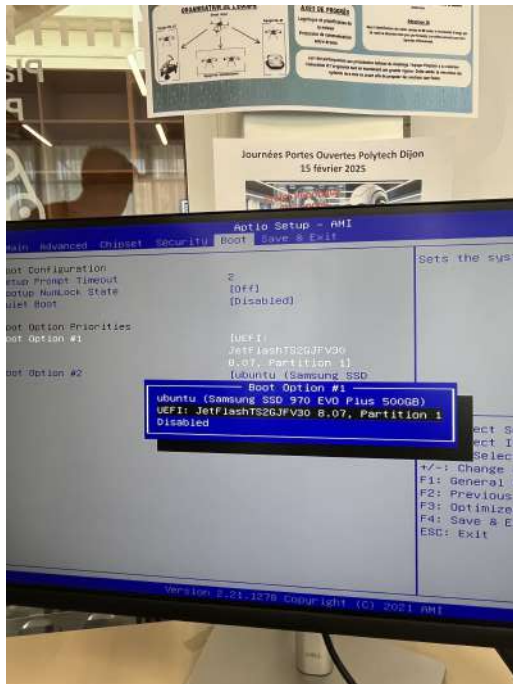
FIGURE 62 : Basculer en mode <Setup>



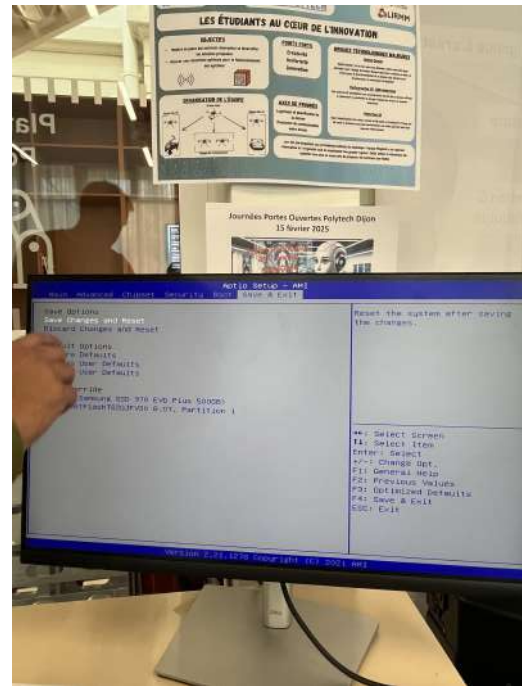
Étape 1 : Brancher la clé USB et le disque dur



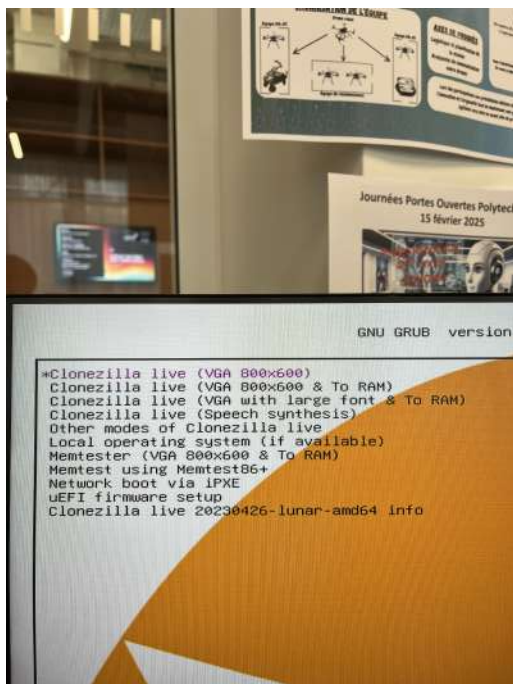
Étape 2 : Sélectionner Boot Option 1



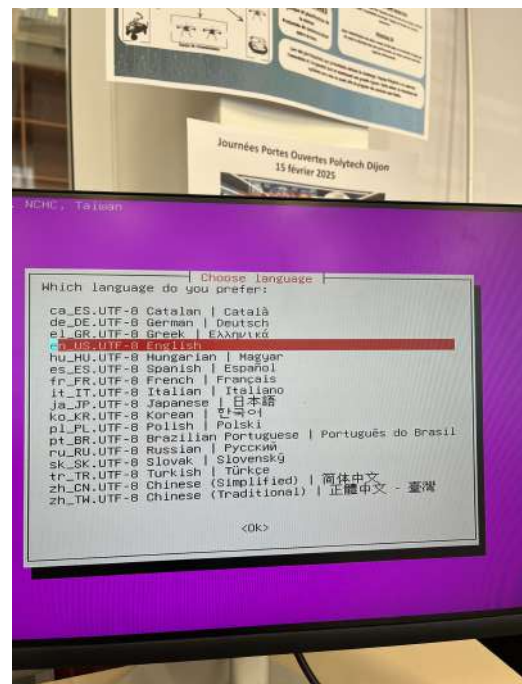
Étape 3 : Sélectionner UEFI JetFlashTS2GJFV30 en tant qu'Option 1



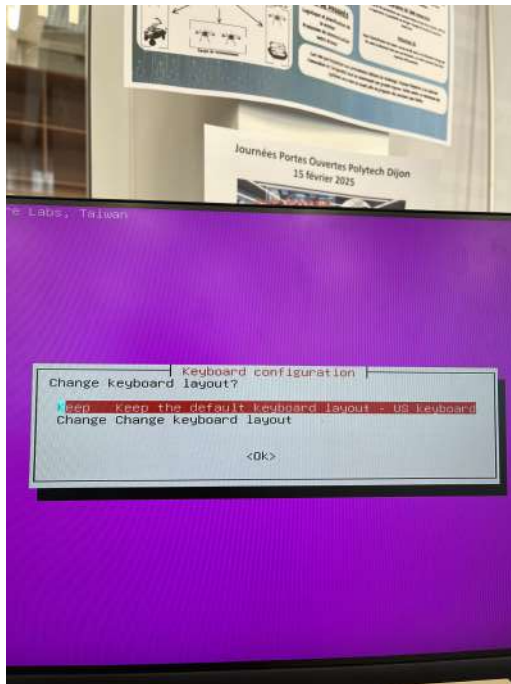
Étape 4 : Sélectionner Save Changes and Reset



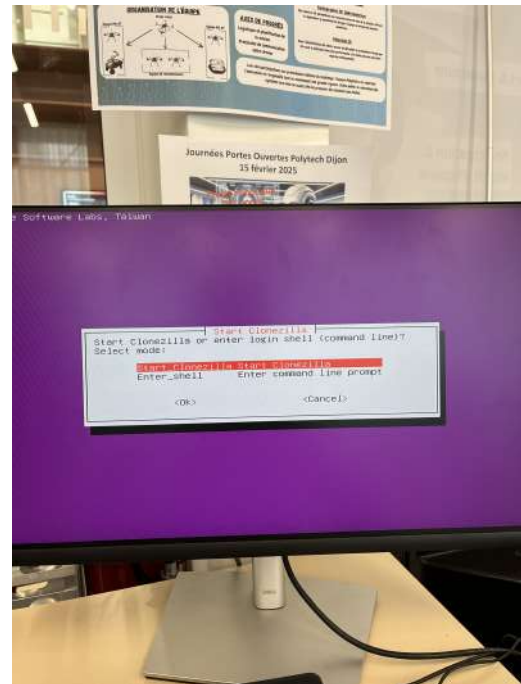
Étape 5 : Sélectionner la première ligne



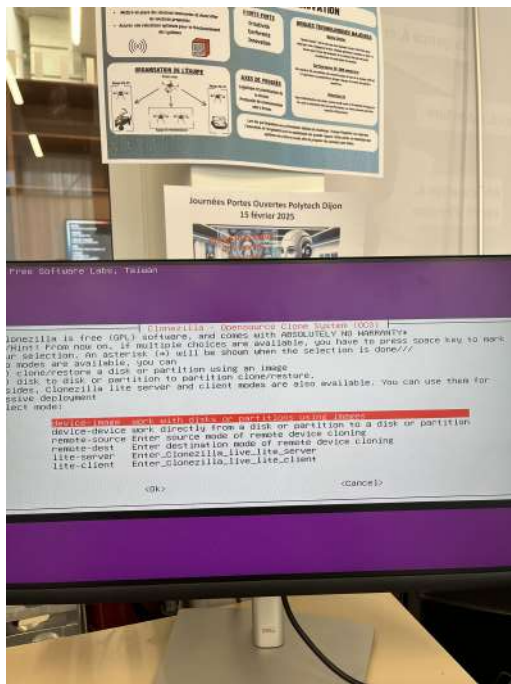
Étape 6 : Choisir la langue



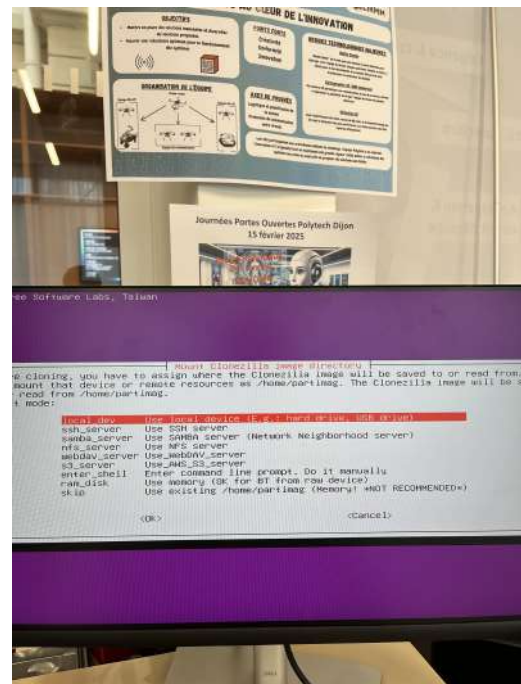
Étape 7 : Sélectionner Keep



Étape 8 : Sélectionner Start\_Clonezilla



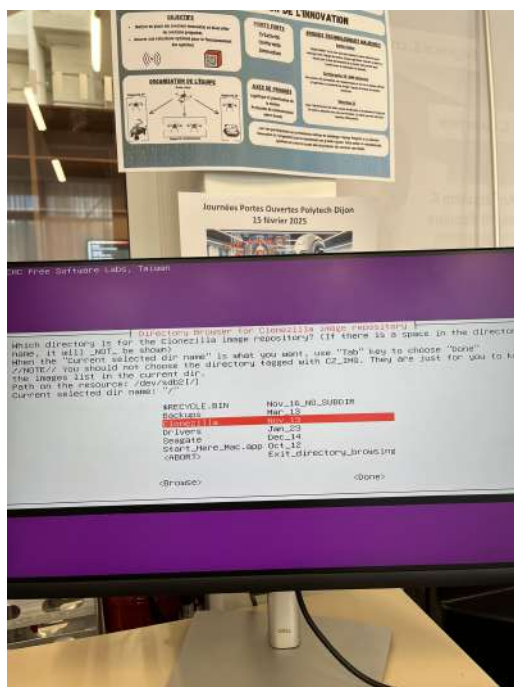
Étape 9 : Sélectionner device-image



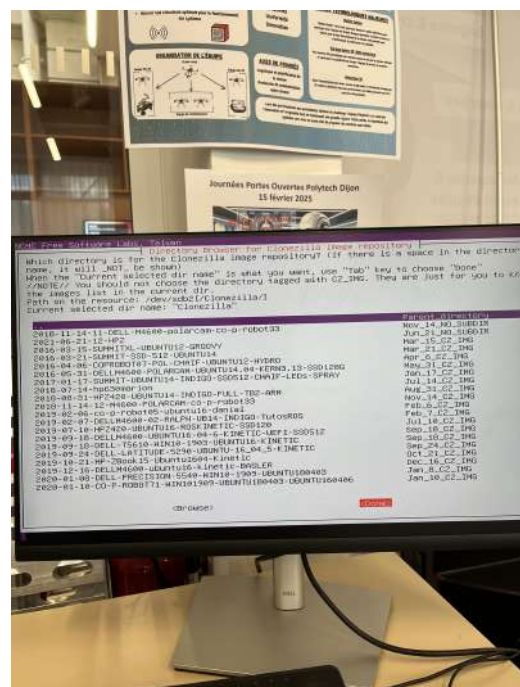
Étape 10 : Sélectionner local\_dev



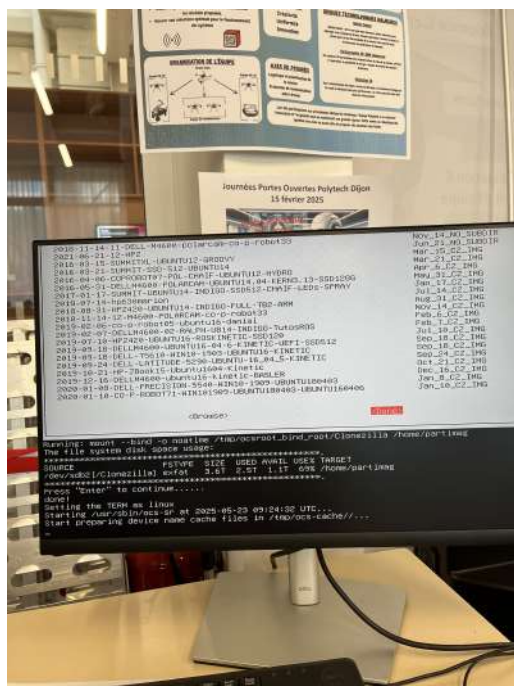




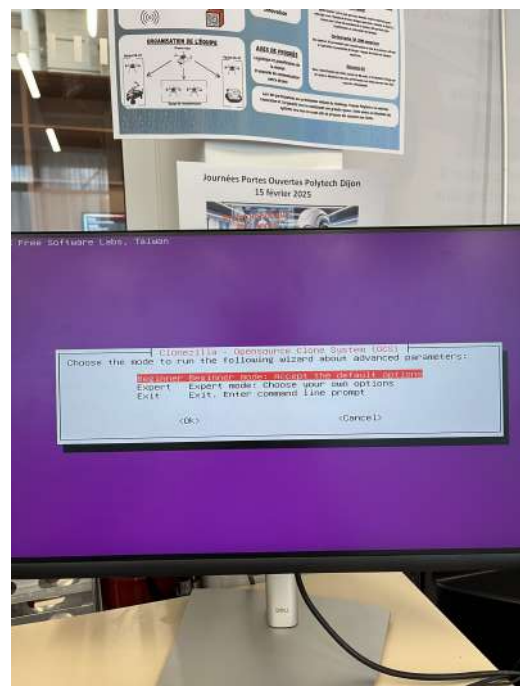
## Étape 15 : Sélectionner Clonezilla



Étape 16 : Sélectionner <Done>

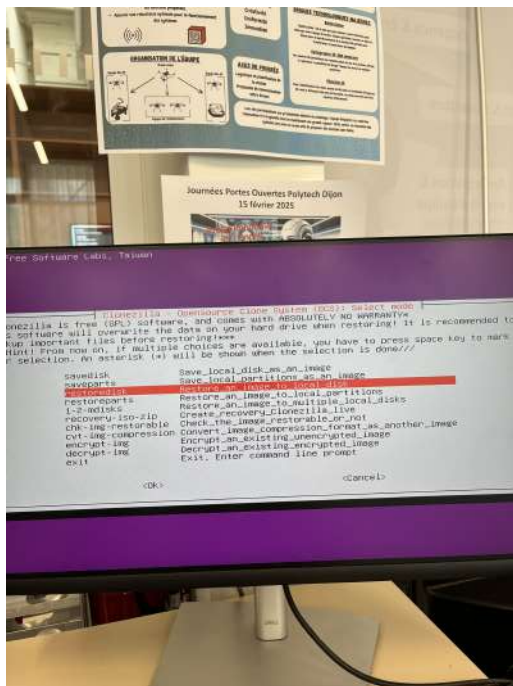


Étape 17 : Presser sur Entrée

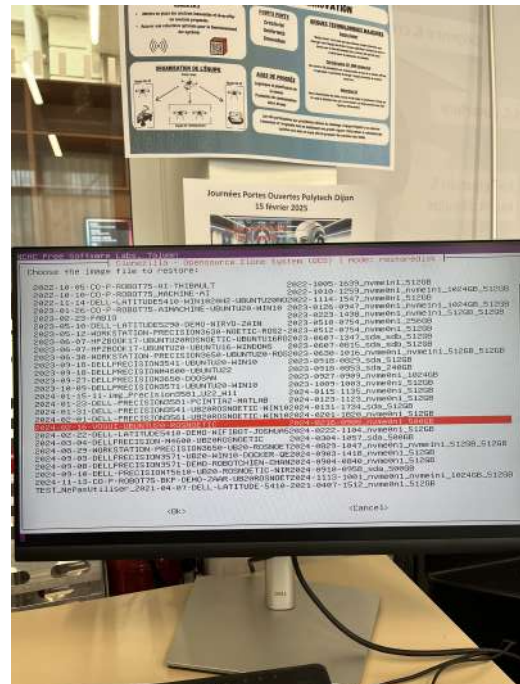


## Étape 18 : Sélectionner Beginner

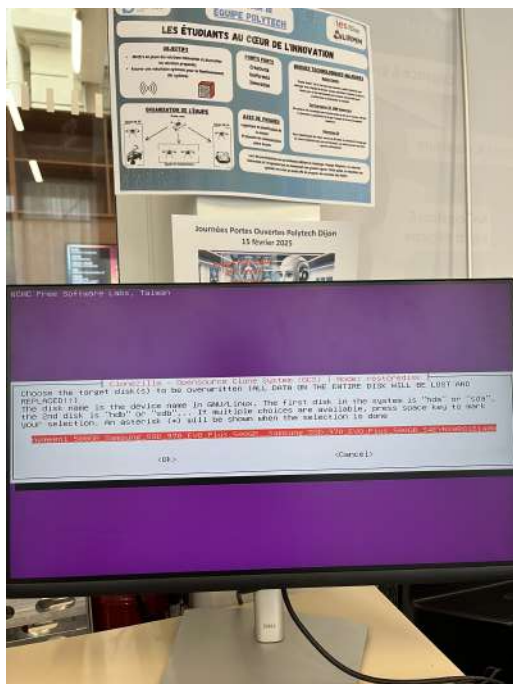




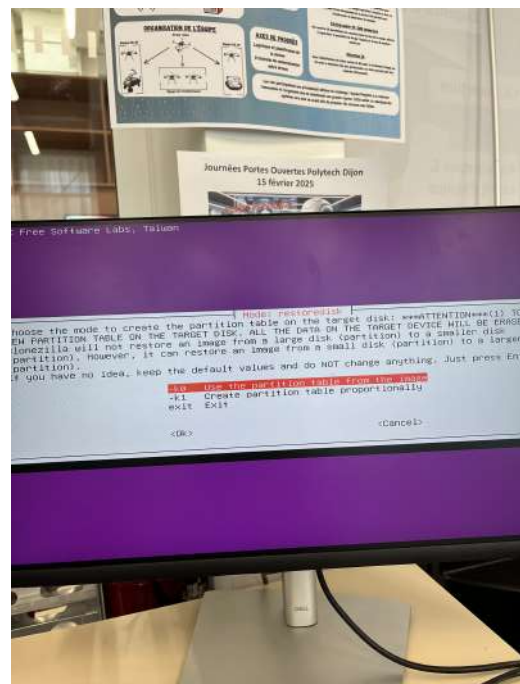
Étape 19 : Sélectionner restoredisk



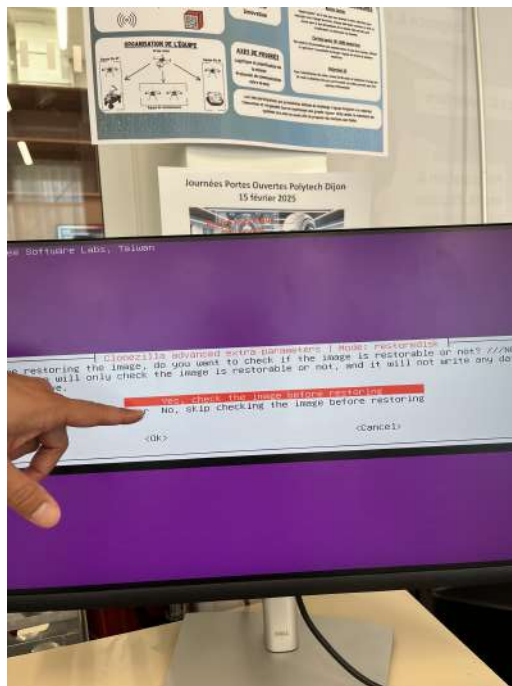
Étape 20 : Sélectionner VOGUE-UBUNTU20-ROSNOETIC



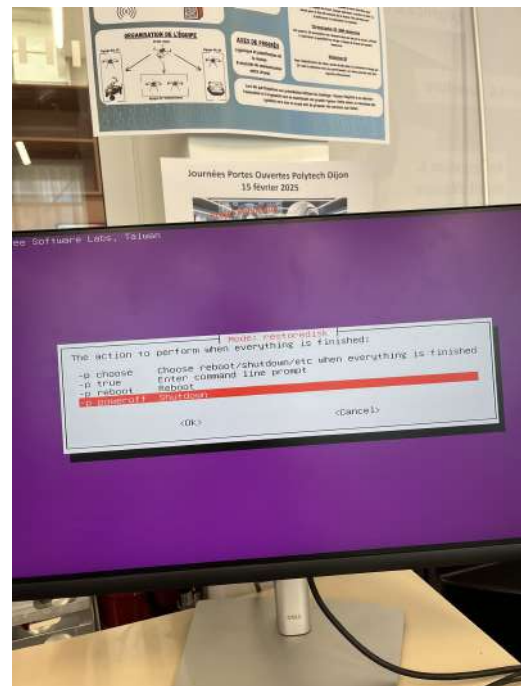
Étape 21 : Sélectionner nvme0n1



Étape 22 : Sélectionner -k0



Étape 23 : Sélectionner Yes



Étape 24 : -p poweroff



FIGURE 75 : Étape 25 : Appuyer sur "y" à deux reprises pour confirmer la réinitialisation



Vous n'avez maintenant plus qu'à tourner la clé sur "OFF" pour éteindre le robot et la réinitialisation sera terminée.

Protocole pour le premier redémarrage du robot :

- La première chose à faire est de compiler une fois tous les fichiers (suivre les étapes de la partie "4.5 Compilation")
  - Ajouter les packages "**pkg\_trans\_cohoma**" et "**control\_ur\_with\_joy**" dans le répertoire **catkin\_ws/src/robot\_packages**. Vous les trouverez dans le dossier "Packages\_Vogui" dans le Drive.
  - Compiler indépendamment les deux nouveaux packages : dans le répertoire **catkin\_ws** lancer dans un premier temps la commande **catkin build pkg\_trans\_cohoma** puis ensuite **catkin build control\_ur\_with\_joy**. Il faut faire cela pour éviter les problèmes de compilation.
  - Donner les droits d'exécution à tous les fichiers des packages en faisant **chmod -R 777 control\_ur\_with\_joy** (faire de même pour le package **pkg\_trans\_cohoma**).
  - Modifications à apporter dans le package "robotnik\_pad" : dans le chemin **catkin\_ws/src/robot\_package/others/robotnik\_pad/robotnik\_pad/config** fichier "robotnik\_pad\_plugins\_askermann\_ps4.yaml" changer le nom du topic à la ligne **joy\_topic : "joy"** -> "**joyMoteur**"
- catkin\_ws/src/robot\_package/others/robotnik\_pad/robotnik\_pad/src**, fichier "robotnik\_pad.cpp" : modifier le nom du topic à la ligne **joy\_topic : "joy"** -> "**joyMoteur**".