3 - ROS2 - Manipulation Mobile

- Assemblage du Turtlebot et OpenManipulator-X et configuration initiale
- Installation et démarrage du Turtlebot & OpenManipulator-X

Assemblage du Turtlebot et OpenManipulator-X et configuration initiale

https://www.classes.cs.uchicago.edu/archive/2022/fall/20600-1/turtlebot_assembly_setup.html

A Word of Caution

This page is NOT intended for student use. This page is designed for course staff to build and maintaing the turtlebots. **If you are a student, please ensure you have permission from course staff before making any of the changes detailed on this page to any of the turtlebots**.

Setup Checklist

Turtlebot Assembly Tips

Turtlebot3

- DO NOT TIGHTEN the screws too much.
- Page 10: When connecting one pair of waffle-plates to another, make that the seams connecting each waffle-plate are parallel to one other.
- Page 17: When attaching the wheels to the dynamixels, do not tighten the screws too much.
- Page 28: Mount the LiDAR further up than the manual specifies so that the arm can fit on (see completed Turtlebots for reference).

OpenMANIPULATOR Arm

- OpenMANIPULATOR Dynamixel 12: when putting the back piece onto the servo, you will need to break off a plastic piece where the cable should be threaded through.
- Make sure the placement of the marking on the servo horns is identical to the diagram PRIOR to screwing anything down onto it.

OpenMANIPULATOR Arm First-Time Configuration

After assembling an OpenMANIPULATOR arm, its servos must first be configured properly before it can be used. By default, all the arm motors are set to the same ID (1) and the wrong baud rate, which causes collisions when trying to detect them.

- Setup OpenCR with Arduino IDE using OpenCR 1.0 manual, “Arduino IDE” section
- Download Dynamixel Wizard 2.0 if not already installed.

1. Connect a **SINGLE** motor (no daisy-chains in the arm) to the OpenCR module and **DISCONNECT ALL OTHER MOTORS** (the wheel motors!!)
2. Open up Dynamixel Wizard 2.0 and update the firmware for that motor by following this tutorial. The arm Dynamixel model is XM430-**W350**, and the wheel motors are XM430-**W210**.
3. Scan for connected Dynamixels using the “Scan” button on the top menu. If the scan does not turn up any results, you may need to change the scan options in the "Options" menu. By default, an unconfigured arm motor will have ID 1, be on Protocol 2.0, and have a baud rate of 57600 bps.
4. Change the ID for the detected motor from 1 to 11/12/13/14/15 (whichever you're doing the procedure for). Click on the “ID” item, and find the ID # you want in the lower right corner. Click it and press “Save”.
5. Change the baud rate to 1M (if not already 1M). Click on the “Baud Rate (Bus)” item, and find the 1 Mbps option. Click it and press “Save”.
6. Disconnect the motor (both in the wizard by clicking “Disconnect” up top and physically disconnecting from the board) and repeat the steps for the remaining ones.

Attaching/Detaching the OpenMANIPULATOR Arm

Physical Attachment/Detachment

Follow these steps below:

1. The arm can be attached and detached by first removing the top layer of the the Turtlebot. There are 10 screws around the perimeter of the top layer that you will need to remove with a Phillips head / cross screwdriver.
2. Once the top layer of the Turtlebot is loose, disconnect the cable that is attached to the LiDAR. This cable is multi-colored with several pins.

   > Note: ONLY remove this LiDAR cable in this step.

   Turtlebot3 with an arm
   Image not found or type unknown

   Once the top layer is fully disconnected, flip the top layer upside-down for easy access.

3. To connect the arm, there are **four screws** that you will need to insert through the **bottom side** of the top layer that connect to **Dynamixel #11**. The exact location of the screws are marked in the picture below. **Use an hex/allen key to secure them.**

Turtlebot3 with an arm

4. Once the arm is securely attached, thread the cable connected to DynaMIXEL #11 through the top layer (if not already threaded) and attach the other end to the remaining **3-pin** slot on the OpenCR board. It should be the open slot adjacent to the already occupied slots.
5. Reconnect the LiDAR cable and re-screw the top layer onto the robot.

To disconnect the arm, follow the same steps as above, but remove the four screws instead and disconnect the DynaMIXEL.

Updating Firmware and Files

Every time the arm is attached or detached, you will need to reconfigure environment variables and update the OpenCR firmware.

If you are *attaching* the arm, set the environment variable `OPENCR_MODEL=om_with_tb3` and ensure that `OPENCR_PORT=/dev/ttyACM0` in the `~/.bashrc` file.

> Note: You can also set those environment variables in the command line by running:

```
$ export OPENCR_PORT=/dev/ttyACM0
$ export OPENCR_MODEL=om_with_tb3
```

Source the file and update the firmware with:

```
$ source ~/.bashrc
```

```
$ cd ~/opencr_update && ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

If you are *detaching* the arm, set the environment variable `OPENCR_MODEL=waffle` and ensure that `OPENCR_PORT=/dev/ttyACM0` in the `~/.bashrc` file.

Enter the OpenCR board into firmware recovery mode by pressing and holding SW2, and simultaneously pressing RESET (you will need to remove the top layer of the turtlebot in order to access those OpenCR buttons). If you want to read more about the OpenCR bootloader, please refer to this page in the OpenCR 1.0 manual.

buttons on OpenCR

Image not found or type unknown

Source the file and update the firmware with:

```
$ source ~/.bashrc
```

```
$ cd ~/opencr_update && ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

Common Errors During Setup

Firmware Recovery Mode and Device Firmware Update for OpenCR Board

We found the **firmware recovery mode** not clearly detailed in the Turtlebot3 manual. Here are the steps for firmware recovery:

1. Push and hold SW2
2. Press RESET
3. Release SW2
4. Arduino code should upload successfully with `jump_with_fw`

Firmware recovery is different from **device firmware update (DFU)**, where the steps are:

1. Push and hold BOOT
2. Press RESET
3. Release BOOT

Troubleshooting `/dev/ttyACM0` Connection Issues

If the `/dev/ttyACM0` port isn't found or running `bringup` on the Pi is hanging:

- Make sure `$OPENCR_MODEL` is correct (`waffle` for no arm, `om_with_tb3` for arm)
- Disconnect and reconnect the USB cable connecting the Pi and OpenCR board
- Enter firmware recovery (SW2 + RESET)
- Reinstall the firmware with `./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr` (see more detailed steps above in the prior section on attaching/detaching the arm)
- [If the above doesn’t work] Restart `roscore`
- [If the above doesn’t work] Reupload the sketch using the Arduino IDE (install instructions) in `Examples > OpenCR > Etc > usb_to_dxl` to the OpenCR board (which you'll need to directly connect to your computer)

If the hydro rosserial / groovy Arduino error appears:

- Configure the arm motors as described in the previous section on attaching/detaching the arm
- Enter firmware recovery (SW2 + RESET)
- Reinstall the firmware with `./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr`
- Set up the wheel motors by uploading the sketch using the Arduino IDE (install instructions) in `Examples > TurtleBot3 > turtlebot3_setup > turtlebot3_setup_motor` to the OpenCR board (which you'll need to directly connect to your computer)

Installation et démarrage du Turtlebot & OpenManipulator-X

Installation Ubuntu Server

Version rapide

- Télécharger l'image compressée de la carte SD préinstallée
  https://seafile.unistra.fr/smart-link/dcc9e405-88a0-41d2-ab5c-3e796a6cebf3/
- Insérer la carte SD et démonter les partitions existantes

```
sudo umount /media/user/writable /media/user/system-boot
```

- Flasher la carte SD avec l'utilitaire `dd`

```
sudo gunzip -c ~/turtlebot3-manipulator-humble.img.gz | sudo of=/dev/sda status=progress
```

- Configurer la connexion automatique au réseau wifi et donner une IP fixe au robot (dans la plage DHCP autorisée par le routeur) :
  - Éjecter et réinsérer la carte SD pour qu'elle se monte
  - Modifier la `CONFIGURATION_RESEAU` (les éléments en majuscule) dans le fichier suivant :
    `sudo nano /media/user/writable/etc/netplan/99-hotspot-fablab.yaml`

```
            addresses: [IP_TURTLEBOT/24]
            gateway4: IP_BOX
            nameservers:
                addresses: [DNS_BOX_OPERATEUR, 9.9.9.9, 89.234.141.66]
            access-points:
                "SSID_WIFI":
                    password: PASSWORD_WIFI
```

Remarque : l'image a été créée après avoir suivi les instructions longues ci-dessous (et quelques workspace et package ros installés en plus) en lançant la commande suivante :

```
sudo dd if=/dev/sda status=progress | gzip -9 > ~/turtlebot3-manipulator-humble.img.gz
```

Reinitialiser le mot-de-passe

Voir la section 4 ici

- Connecter la carte SD sur un PC
- naviguer au point de montage, par exemple : `cd /media/user/writable`
- ouvrir le fichier `sudo nano /etc/passwd`
- modifier la ligne qui concerne votre user du système Ubuntu du turtlebot, par exemple ici `ubuntu`
  Avant : `ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash`
  Après (on supprime le `x`) : `ubuntu::1000:1000:Ubuntu:/home/ubuntu:/bin/bash`
- Redémarrer le système dans le TurtleBot
- Se connecter avec le login `ubuntu`
- Aucun MDP n'est demandé
- Lancer le programme de changement de MDP `sudo passwd`
- Rentrer le nouveau MDP 2 fois

Version longue

https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

Depuis un ordinateur sous Ubuntu 22.04

- Insérer la carte SD dans le navigateur
- Installer rpi-manager `sudo apt install rpi-imager`
- Sélectionner `CHOOSE OS`
  - autre système Linux (Other general-purpose OS)
  - Ubuntu Server 22.04 LTS (64bits)
- Sélectionner `CHOOSE STORAGE` la carte micro SD
- Cliquer sur `WRITE`

Depuis une VM WSL Ubuntu 22

Configurer la connexion automatique au réseau wifi et donner une IP fixe au robot (dans la plage DHCP autorisée par le routeur) :

- Éjecter et réinsérer la carte SD pour qu'elle se monte

- Créer le fichier suivant : `sudo nano /media/user/writable/etc/netplan/99-hotspot-fablab.yaml`

```
network:
    renderer: networkd
    ethernets:
        eth0:
            dhcp4: true
            dhcp6: true
            optional: true
    wifis:
        wlan0:
            dhcp4: false
            dhcp6: false
            addresses: [192.168.100.40/24]
            gateway4: 192.168.100.1
            nameservers:
                addresses: [192.168.100.1, 9.9.9.9, 89.234.141.66]
            access-points:
                fablab:
                    password: ...
    version: 2
```

- Désactiver la configuration du réseau par cloud-init en créant le fichier suivant : `sudo nano /media/user/writable/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`

```
network: {config: disabled}
```

Insérer la carte dans le robot, le démarrer assez proche du hotspot wifi configuré, se connecter en ssh depuis l'ordinateur :

- Utiliser l'adresse IP précédemment configurée `ssh ubuntu@192.168.100.40`
- mdp : `ubuntu`
- changer le mdp par un suffisamment sécurisé
- se connecter en ssh avec le nouveau mdp
- pour lancer des commandes en root utiliser `sudo` avec le même mdp
- Ne pas attendre la connexion réseau pour démarrer : `systemctl mask systemd-networkd-wait-online.service`
- Désactiver la veille et l'hibernation : `sudo systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target`

Installer **ROS 2 Humble** sans interfaces graphiques (ros-humble-desktop) qui seront lancées sur l'ordinateur externe :

```
sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo apt update && sudo apt install curl
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-
keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU_CODENAME)
main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
sudo apt update && sudo apt upgrade
sudo apt install ros-humble-ros-base ros-dev-tools
source /opt/ros/humble/setup.bash
echo 'source /opt/ros/humble/setup.bash' >> ~/.bashrc
```

Installer le workspace du turtlebot et les dépendances :

```
sudo apt install python3-argcomplete python3-colcon-common-extensions libboost-system-dev
build-essential
sudo apt install ros-humble-hls-lfcd-lds-driver
sudo apt install ros-humble-turtlebot3-msgs
sudo apt install ros-humble-dynamixel-sdk
sudo apt install libudev-dev
mkdir -p ~/turtlebot3_ws/src && cd ~/turtlebot3_ws/src
git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git
git clone -b ros2-devel https://github.com/ROBOTIS-GIT/ld08_driver.git
cd ~/turtlebot3_ws/src/turtlebot3
rm -r turtlebot3_cartographer turtlebot3_navigation2
cd ~/turtlebot3_ws/
echo 'source /opt/ros/humble/setup.bash' >> ~/.bashrc
source ~/.bashrc
colcon build --symlink-install --parallel-workers 1
echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc
source ~/.bashrc
```

Configurer le Port USB pour OpenCR :

```
sudo cp `ros2 pkg prefix turtlebot3_bringup`/share/turtlebot3_bringup/script/99-turtlebot3-cdc.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
```

ID de domain ROS pour la communication DDS :

```
echo 'export ROS_DOMAIN_ID=30 #TURTLEBOT3' >> ~/.bashrc
source ~/.bashrc
```

Configurer le modèle du LIDAR :

```
echo 'export LDS_MODEL=LDS-02' >> ~/.bashrc
source ~/.bashrc
```

Montage et Configuration des Dynamixels

https://www.classes.cs.uchicago.edu/archive/2022/fall/20600-1/turtlebot_assembly_setup.html

- Installer Arduino IDE : `sudo apt install arduino`
- https://emanual.robotis.com/docs/en/parts/controller/opencr10/#install-on-linux
- Connecter OpenCR
  https://emanual.robotis.com/docs/en/platform/turtlebot3/manipulation/#arduino-ide
- Installer Dynamixel Wizard
  https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/
- Lancer Dynamixel Wizard
  `cd ~/ROBOTIS/DynamixelWizard2`
  `bash DynamixelWizard2.sh`
- Si une erreur de dépendance apparait, désinstaller/réinstaller/upgrader dynamixelWizard via l'executable `~/ROBOTIS/DynamixelWizard2/maintenancetool`
  https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/#uninstall-linux

Pour l'Open-Manipulator

Configurer les dynamixel (baud et ID 11 à 15)

https://www.classes.cs.uchicago.edu/archive/2022/fall/20600-1/turtlebot_assembly_setup.html#arm-first-time :

- Connect a **SINGLE** motor (no daisy-chains in the arm) to the OpenCR module and **DISCONNECT ALL OTHER MOTORS** (the wheel motors!!)

- Open up Dynamixel Wizard 2.0 and update the firmware for that motor by following this tutorial. The arm Dynamixel model is XM430-**W350**, and the wheel motors are XM430-**W210**.
- Scan for connected Dynamixels using the “Scan” button on the top menu. If the scan does not turn up any results, you may need to change the scan options in the "Options" menu. By default, an unconfigured arm motor will have ID 1, be on Protocol 2.0, and have a baud rate of 57600 bps.
- Change the ID for the detected motor from 1 to 11/12/13/14/15 (whichever you're doing the procedure for). Click on the “ID” item, and find the ID # you want in the lower right corner. Click it and press “Save”.
- Change the baud rate to 1M (if not already 1M). Click on the “Baud Rate (Bus)” item, and find the 1 Mbps option. Click it and press “Save”.
- Disconnect the motor (both in the wizard by clicking “Disconnect” up top and physically disconnecting from the board) and repeat the steps for the remaining ones

Pour le Turtlebot

Via Arduino IDE ou DynamixelWizard en s'inspirant de :

https://emanual.robotis.com/docs/en/platform/turtlebot3/faq/#setup-dynamixels-for-turtlebot3

- Moteur gauche : ID=1
- Moteur droit : ID=2
- Baud rate : 1M

Test depuis un PC sans la raspberry

Téléopération du OpenManipulator-X seul

Suivre le tutoriel Foxy en remplaçant `foxy` par `humble` et `foxy-devel` par `ros2` en utilisant l'interface de communication OpenCR :

https://emanual.robotis.com/docs/en/platform/openmanipulator_x/quick_start_guide/

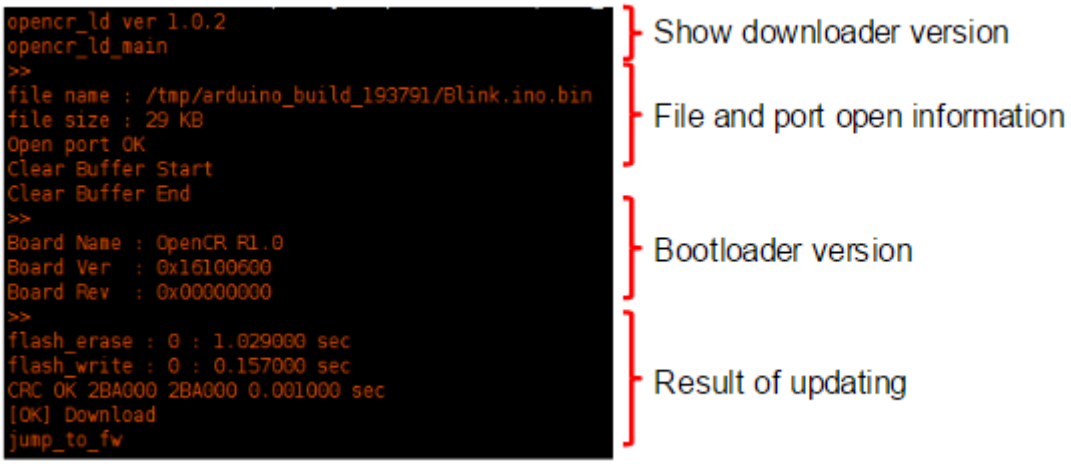
Pour tester le bon fonctionnement du bras et de sa pince, on connecte la carte OpenCR directement à un PC ayant ROS Humble préinstallé :

- Installer et compiler le workspace

```
sudo apt install ros-humble-rqt* ros-humble-joint-state-publisher
mkdir -p ~/openmanipulator_ws/src/
cd ~/openmanipulator_ws/src/
git clone -b ros2 https://github.com/ROBOTIS-GIT/DynamixelSDK.git
git clone -b ros2 https://github.com/ROBOTIS-GIT/dynamixel-workbench.git
```

```
git clone -b ros2 https://github.com/ROBOTIS-GIT/open_manipulator.git
git clone -b ros2 https://github.com/ROBOTIS-GIT/open_manipulator_msgs.git
git clone -b ros2 https://github.com/ROBOTIS-GIT/open_manipulator_dependencies.git
git clone -b ros2 https://github.com/ROBOTIS-GIT/robotis_manipulator.git
cd ~/openmanipulator_ws && colcon build --symlink-install
```

- Corriger le bug de compilation et re-compiler. Dans `src/open_manipulator/open_manipulator_x_controller/src/open_manipulator_x_controller.cpp`, lignes 67-68, remplacer :
  `this->declare_parameter("sim");`
  `this->declare_parameter("control_period");`
  par :
  `this->declare_parameter("sim", false);`
  `this->declare_parameter("control_period", 0.010);`
- Lancer `arduino`
- Uploader l'exemple `File > Examples > OpenCR > 10.Etc > usb_to_dxl` vers OpenCR



- Si cela réussit, jump_to_fw apparaît, sinon essayer d'uploader une seconde fois
- Lancer le contrôleur du robot. **Attention les moteurs vont bouger et se bloquer dans la position initiale**
  `ros2 launch open_manipulator_x_controller open_manipulator_x_controller.launch.py usb_port:=/dev/ttyACM0`
- Dans un second terminal, lancer le noeud de téléopération :
  `ros2 run open_manipulator_x_teleop teleop_keyboard`
- Piloter le robot dans l'espace Cartésien ou articulaire avec les touches indiquées

Programmation hors-ligne du OpenManipulator-X et TurtleBot3 depuis MoveIt

On suit le tutoriel https://emanual.robotis.com/docs/en/platform/turtlebot3/manipulation/#operate-the-actual-openmanipulator en installant tout ce qui est censé être installé sur le raspberry **[SBC]**/

[TurtleBot3] sur le PC **[Remote PC]**.

- Installer le workspace et compiler :

```
sudo apt install ros-humble-dynamixel-sdk ros-humble-ros2-control ros-humble-ros2-controllers
ros-humble-gripper-controllers ros-humble-moveit
cd ~/turtlebot3_ws/src/
git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3_manipulation.git
cd ~/turtlebot3_ws && colcon build --symlink-install
```

- Ajouter au `~/.bashrc` :

```
export ROS_DOMAIN_ID=30 #TURTLEBOT3
export LDS_MODEL=LDS-02
export TURTLEBOT3_MODEL=waffle_pi
export OPENCR_PORT=/dev/ttyACM0
export OPENCR_MODEL=turtlebot3_manipulation
```

- AVANT TOUT FLASHAGE DE OPENCR, se mettre en **mode debug** en
  - Rester appuyé sur le bouton SW2
  - Appuyer quelques secondes sur RESET
  - Relacher RESET
  - Relacher SW2
- ATTENTION SI LE MODE DEBUG n'est pas activé, il se peut `jump_fw` soit affiché mais que le flashage ait échoué.
- Configurer OpenCR pour turtlebot3_manipulation depuis Arduino `File > Examples > Turtlebot3 ROS2 > turtlebot3_manipulation` ou avec le prebuild :

```
rm -rf ./opencr_update.tar.bz2
wget https://github.com/ROBOTIS-GIT/OpenCR-
Binaries/raw/master/turtlebot3/ROS2/latest/opencr_update.tar.bz2
tar -xvf opencr_update.tar.bz2
cd ./opencr_update
./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

- Démarrer ROS Control :
  `ros2 launch turtlebot3_manipulation_bringup hardware.launch.py`
- **Le setup a fonctionné si le robot apparaît dans la bonne configuration dans RViz !**
- Dans un second terminal démarrer au choix :
  - MoveIt pour la programmation hors-ligne et planification de trajectoire :
    `ros2 launch turtlebot3_manipulation_moveit_config moveit_core.launch.py`

    - Piloter le robot en bougeant les flèches dans RViz et en cliquant sur "Plan and Execute"
  - MoveIt servo
    `ros2 launch turtlebot3_manipulation_moveit_config servo.launch.py`
    - et la téléopération avec le clavier (dans un 3ème terminal)
      `ros2 run turtlebot3_manipulation_teleop turtlebot3_manipulation_teleop`
    - Piloter le robot dans l'espace Cartésien ou articulaire avec les touches indiquées

Configuration OpenCR

Pour le Turtlebot : https://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup

Pour l'OpenManipulator-X : https://emanual.robotis.com/docs/en/platform/turtlebot3/manipulation/#opencr-setup

Dépendances manquantes :

```
sudo apt install ros-humble-hardware-interface
ros-humble-ros2-control ?
ros-humble-joint-state-publisher ?
```

Dépendances manquantes côté Raspberry :

```
sudo apt install rros-humble-gripper-controllers ros-humble-xacro
```

Dépendances manquantes côté PC :

```
sudo apt install ros-humble-moveit-servo
```

Issues :

https://forum.robotis.com/t/ros-2-foxy-openxmanuipaltor-bringup-issues/2142/9

https://github.com/ROBOTIS-GIT/open_manipulator/issues/212

https://github.com/ROBOTIS-GIT/open_manipulator/issues/209

https://www.classes.cs.uchicago.edu/archive/2022/fall/20600-1/turtlebot_assembly_setup.html#arm-first-time

---

Auteur: Gauthier Hentz, sur le wiki de l'innovation de l'IUT de Haguenau

Attribution-NonCommercial-PartageMemeConditions 4.0 International (CC BY-NC-SA 4.0)