

# Installation et démarrage du OpenManipulator-X & Turtlebot 3

Suivre d'abord <https://innovation.iha.unistra.fr/books/robotique-open-source/page/installation-et-demarrage-du-turtlebot-3>

## Montage et Configuration des Dynamixel

[https://www.classes.cs.uchicago.edu/archive/2022/fall/20600-1/turtlebot\\_assembly\\_setup.html](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/](https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/)
- Lancer Dynamixel Wizard  
`cd ~/ROBOTIS/DynamixelWizard2`  
`bash DynamixelWizard2.sh`
- Si une erreur de dépendance apparaît, désinstaller/réinstaller/upgrader dynamixelWizard via l'exécutable `~/ROBOTIS/DynamixelWizard2/maintenancetool`  
[https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel\\_wizard2/#uninstall-linux](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](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/](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,
```

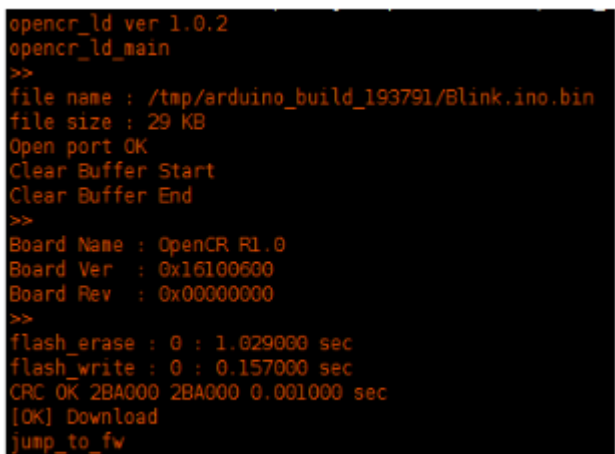
lignes 67-68, remplacer :

```
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



The screenshot shows the OpenCR bootloader interface with the following text:

```
opencr_ld ver 1.0.2
opencr_ld_main
>>
file name : /tmp/arduino_build_193791/Blink.ino.bin
file size : 29 KB
Open port OK
Clear Buffer Start
Clear Buffer End
>>
Board Name : OpenCR R1.0
Board Ver : 0x16100600
Board Rev : 0x00000000
>>
flash_erase : 0 : 1.029000 sec
flash_write : 0 : 0.157000 sec
CRC OK 2BA000 2BA000 0.001000 sec
[OK] Download
jump_to_fw
```

Annotations on the right side of the screenshot:

- Show downloader version (points to 'opencr\_ld ver 1.0.2')
- File and port open information (points to 'file name', 'file size', and 'Open port OK')
- Bootloader version (points to 'Board Name', 'Board Ver', and 'Board Rev')
- Result of updating (points to 'flash\_erase', 'flash\_write', 'CRC OK', and '[OK] Download')

- 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 :
- 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 OPENCNCR_PORT=/dev/ttyACM0
export OPENCNCR_MODEL=turtlebot3_manipulation
```

- AVANT TOUT FLASHAGE DE OPENCNCR, 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 ./opencnrc_update.tar.bz2
wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS2/latest/opencnrc_update.tar.bz2
tar -xvf opencnrc_update.tar.bz2
cd ./opencnrc_update
./update.sh $OPENCNCR_PORT $OPENCNCR_MODEL opencnrc
```

- 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](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/212)

[https://github.com/ROBOTIS-GIT/open\\_manipulator/issues/209](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](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)

---

Revision #27

Created 5 July 2023 09:25:07 by admin\_idf

Updated 15 January 2025 14:53:45 by admin\_idf