

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 apparaît, désinstaller/réinstaller/upgrader dynamixelWizard via l'exécutable `~/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

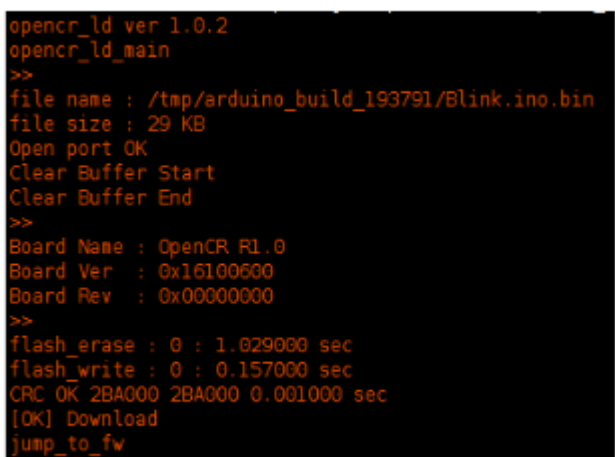
`src/open_manipulator/open_manipulator_x_controller/src/open_manipulator_x_controller.cpp`,
lignes 67-68, remplacer :

```
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



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 the first two lines)
- File and port open information (points to the file name, size, and port status)
- Bootloader version (points to the board information)
- Result of updating (points to the flash operations and CRC check)

- 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 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 ./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 $OPENCNCR_PORT $OPENCNCR_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)

Revision #20

Created 5 July 2023 09:25:07 by admin_idf

Updated 31 January 2024 13:07:43 by admin_idf