Manual brat robotic

Open Manipulator-X

Cuprins

[Cuprins 2](#_Toc132027805)

[Instalarea software-ului necesar 3](#_Toc132027806)

[Ubuntu linux 3](#_Toc132027807)

[Creare sistem dual boot Ubuntu 3](#_Toc132027808)

[Creare sistem dual boot Linux Mint 6](#_Toc132027809)

[ROS 9](#_Toc132027810)

[Kinetic Ubuntu 16.04 – Tested (working) 9](#_Toc132027811)

[Kinetic Linux Mint 18.x (Manual installation) – Tested (working) 9](#_Toc132027812)

[Noetic – Tested (working on Ubuntu 20.04) 10](#_Toc132027813)

[Instalare pachete ROS 11](#_Toc132027814)

[Kinetic Ubuntu 16.04.x & Linux Mint 18.x – Tested (working) 11](#_Toc132027815)

[Noetic – Tested (not working, error installing moveit package) 11](#_Toc132027816)

[Gazeboo 11](#_Toc132027817)

[Lansarea controller-ului pentru Gazeboo 12](#_Toc132027818)

[GUI 13](#_Toc132027819)

[Interfețe de comunicare 13](#_Toc132027820)

[MATLAB - ROS 15](#_Toc132027821)

# Instalarea software-ului necesar

## Ubuntu linux

Sistemul de operare Ubuntu este necesar pentru mediul de simulare Gazeboo care va fi instalat ulterior.

Versiunea de Ubuntu care trebuie instalată depinde de versiunea de ROS (Robot Operating System) care se dorește a fi instalată.

ROS Kinetic -> Ubuntu v16.04.x sau Linux Mint 18.x

ROS Noetic -> Ubuntu v20.04.x

[Link instalare Ubuntu](https://releases.ubuntu.com/?_ga=2.219480409.1908348701.1627362888-1311983284.1627362888)

[Link instalare Linux Mint](https://www.linuxmint.com/edition.php?id=217)

[Tutorial](https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview)

**! Atenție ! Dupa instalarea versiunii de Ubuntu nu faceți update la o versiune mai noua.**

**! LIMITĂRI !** **Dacă se alege instalarea Ubuntu 16.04.x s-ar putea să nu gasiți toate driver-ele pentru PC-ul vostru, de exemplu WI-FI sau TouchPad.**

Se recomandă a NU se folosi o mașină virtuală (VM) pentru instalarea sistemului de operare Ubuntu, ci să se creeze un sistem dual boot.

## Creare sistem dual boot Ubuntu

Pentru crearea unui sistem dual boot, primul pas este de a aloca o partiție pentru sistemul de operare Ubuntu. Pentru aceasta căutam în bara de cautare Windows, *Disk Management.*

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Următorul pas, este de a micșora dimensiunea partiției în care Ubuntu va fi instalat. Spațiul micșorat va fi folosit pentru instalare. Ținem cont de faptul că dacă se dorește instalarea MATLAB pe Ubuntu, va trebui să alocăm minim 60 Gb.

Alegem opțiunea de “Shrink Volume...”.

Graphical user interface, application

Description automatically generated

Ca și exemplu în imaginea de mai jos partiția “C:” este micșorată cu 60Gb pentru a face loc sistemului de operare Ubuntu.

Graphical user interface, text, application, email

Description automatically generated

După ce a fost creată, zona de memorie va aparea nealocată.

Se va introduce stick-ul USB (Care are OS-ul de Ubuntu) în PC și se va instala în cadrul partiției create, păstrând totodată și sistemul de operare original.

Pentru ca PC-ul să pornească de pe USB, trebuie să modificăm ordinea de “bootare” din BIOS astfel încăt opțiunea de USB să fie prima.

Pentru a intra în BIOS restartăm PC-ul iar în timp ce porneste apasăm tasta F1, F2, F12, ESC sau DEL (Depinde de producător).

Graphical user interface, text, application

Description automatically generated

Accesăm „Change Boot Order” și punem pe primul loc USB-ul, salvăm și ieșim.

În fereastra de instalare Ubuntu selectăm partiția creată anterior și o formatăm selectănd la “mount point” opțiunea “/”. După care se apasă pe „install now”.

Pentru a verifica versiunea de Ubuntu putem deschide un terminal și introduce comanda:

$ lsb\_release -a

## Creare sistem dual boot Linux Mint

În cazul în care s-a dorit instalarea sistemului de operare Linux Mint (Asta dacă se dorește a lucra cu versiunea Kinetic de ROS) pașii pentru crearea unui sistem dual boot sunt următorii.

**Crearea partiției este la fel ca la Ubuntu.**

După ce am pornit SO de pe stick-ul USB vom observa că nu avem opțiunea de install. Ea va fi posibilă abia după ce pornim SO.

Text, whiteboard

Description automatically generated

La fereastra “Installation type” vom selecta “Something else”.

Graphical user interface, text, application

Description automatically generated

Mai departe va trebui să creăm 3 partiții root, swap și home.

Selectăm partiția nealocată creată in Windows.

Graphical user interface, text, application

Description automatically generated

Vom crea partiția root de 20Gb, ext4 format și mount point “/”.

Graphical user interface, text, application

Description automatically generated

Partiția swap este următoarea, aceasta va trebui să fie semnificativă doar dacă memoria RAM de pe dispozitiv este limitată (ex: 2 – 4 Gb). Însă se pot aloca 4Gb indiferent de acest lucru.

Graphical user interface, text, application, chat or text message

Description automatically generated

Cea mai mare partiție ar trebui să fie cea de home. Aici va fi stocat workspace-ul.

Graphical user interface, text, application, chat or text message

Description automatically generated

După crearea partiților se apasă pe “install now”. După restart sistemul dual boot este gata.

În cazul în care simțiți că Linux Mint merge greu introduceți următoarele comenzi:

$ sudo apt-get remove irqbalance

$ sudo reboot

## ROS

După ce sistemul de operare a fost instalat cu succes, următorul pas este instalarea sistemului de operare pentru roboți (ROS).

Următoarele comenzi trebuie executate.

Aveți grija ca fiecare comanda sa fie pe o singura linie!

### Kinetic Ubuntu 16.04 – Tested (working)

$ sudo apt-get update

$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_kinetic.sh

$ chmod 755 ./install\_ros\_kinetic.sh

$ bash ./install\_ros\_kinetic.sh

### Kinetic Linux Mint 18.x (Manual installation) – Tested (working)

**Setup source:**

$ sudo sh -c 'echo "deb <http://packages.ros.org/ros/ubuntu> xenial main" > /etc/apt/sources.list.d/ros-latest.list'

**Setup keys:**

$ sudo apt-get install curl

$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -

**Install ROS:**

$ sudo apt-get update

$ sudo apt-get install ros-kinetic-desktop-full

**Install dependencies:**

$ sudo apt-get install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential

**Initialize rosdep:**

$ sudo apt-get install python-rosdep

$ sudo rosdep init

$ rosdep update

**ROS environment setup:**

$ sudo apt-get install git

$ mkdir -p ~/catkin\_ws/src

$ source /opt/ros/kinetic/setup.bash

$ catkin\_init\_workspace

$ cd ~/catkin\_ws && catkin\_make

$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

$ echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc

$ echo "export ROS\_MASTER\_URI=http://localhost:11311" >> ~/.bashrc

$ echo "export ROS\_HOSTNAME=localhost" >> ~/.bashrc"

$ source ~/.bashrc

### Noetic – Tested (working on Ubuntu 20.04)

$ sudo apt update

$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_noetic.sh

$ chmod 755 ./install\_ros\_noetic.sh

$ bash ./install\_ros\_noetic.sh

În cazul sistemului de operare Ubuntu, acestă instalare este facută cu ajutorul unui bash script (install\_ros\_kinetic/noetic.sh). Pentru o instalare manuală vizitați linkul: [Instalare Manuală](https://emanual.robotis.com/docs/en/platform/openmanipulator_x/quick_start_guide/)

## Instalare pachete ROS

### Kinetic Ubuntu 16.04.x & Linux Mint 18.x – Tested (working)

Descarcă dependințele

$ source ~/.bashrc

$ sudo apt-get install ros-kinetic-ros-controllers ros-kinetic-gazebo\* ros-kinetic-moveit\* ros-kinetic-industrial-core

$ sudo apt-get install ros-kinetic-dynamixel-sdk ros-kinetic-dynamixel-workbench\*

$ sudo apt-get install ros-kinetic-robotis-manipulator

Descarcă si compilează pachetele Open Manipulator-X

$ cd ~/catkin\_ws/src/

$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator.git

$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator\_msgs.git

$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator\_simulations.git

$ git clone https://github.com/ROBOTIS-GIT/open\_manipulator\_dependencies.git

$ cd ~/catkin\_ws && catkin\_make

### Noetic – Tested (not working, error installing moveit package)

Descarcă dependințele

$ source ~/.bashrc

$ sudo apt-get install ros-noetic-ros-controllers ros-noetic-gazebo\* ros-noetic-moveit\* ros-noetic-industrial-core

$ sudo apt install ros-noetic-dynamixel-sdk ros-noetic-dynamixel-workbench\*

$ sudo apt install ros-noetic-robotis-manipulator

Descarcă si compilează pachetele Open Manipulator-X

$ cd ~/catkin\_ws/src/

$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator.git

$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator\_msgs.git

$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/open\_manipulator\_simulations.git

$ git clone https://github.com/ROBOTIS-GIT/open\_manipulator\_dependencies.git

$ cd ~/catkin\_ws && catkin\_make

## Gazeboo

Dacă am urmat pașii anteriori Gazeboo ar trebui să fie instalat deja pe sistem (A fost instalat odată cu pachetele de ROS). Comenzile ce urmează nu depind de versiunea de ROS instalată.

Se poate executa scriptul LaunchSim.sh, pentru a porni simularea in Gazebo.

$ bash LaunchSim.sh

Dacă dorim să nu utilizăm scriptul putem executa comenzile următoare manual.

**NU executați comenzile de mai jos dacă ați pornit scriptul LaunchSim.sh**

Pentru a porni Gazeboo executăm comanda:

$ roslaunch open\_manipulator\_gazebo open\_manipulator\_gazebo.launch

### Lansarea controller-ului pentru Gazeboo

Pentru a controla robotul din simulare trebuie deschis un controller cu comanda:

$ roslaunch open\_manipulator\_controller open\_manipulator\_controller.launch use\_platform:=false

Dacă controller-ul a fost deschis in mod corect, următorul mesaj ar trebui să apară pe terminalul în care a fost lansat Gazeboo.

#### Kinetic

Text

Description automatically generated

#### Noetic

Text

Description automatically generated

### GUI

Pentru a opera brațul, un GUI trebuie deschis cu comanda:

$ roslaunch open\_manipulator\_control\_gui open\_manipulator\_control\_gui.launch

Pentru mai multe informații sau în caz de nedumeriri apăsați [aici](https://emanual.robotis.com/docs/en/platform/openmanipulator_x/ros_simulation/#launch-gazebo).

# Interfețe de comunicare

Pentru a comunica cu brațul sunt disponibile 2 interfețe:

1. U2D2 (indisponibil) – Brațul folosește TTL pentru a comunica, însă nu este disponibil modulul U2D2 Power Hub Board pentru a alimenta brațul.

Diagram

Description automatically generated

Placa și cablul de alimentare pot fi achiziționate de [aici](http://en.robotis.com/shop_en/list.php?ca_id=3020a0&sort=&sortodr=&page=1).

1. OpenCR (disponibil) – În cazul în care această modalitate este aleasă, OpenCR poate fi flash-uit cu ajutorul mediului Arduino IDE.A picture containing text, electronics

   Description automatically generated

[Despre OpenCRv1.0](https://emanual.robotis.com/docs/en/parts/controller/opencr10/#bootloader). (Accesează pentru mai multe informații cum ar fi memory mapping, update bootloader, diagrame de stari, protocoale de comunicații)

Instaleză [ArduinoIDE pe Ubuntu.](https://emanual.robotis.com/docs/en/parts/controller/opencr10/#install-on-linux)

# MATLAB - ROS

Putem comanda brațul robotic cu ajutorul pachetului ROS Toolbox din MATLAB.

Se recomandă să nu instalăm MATLAB pe Ubuntu, ci să-l instalăm pe Windows.

Următoarele Toolbox-uri trebuie instalate în MATLAB:

* Simscape
* Simscape Multybody
* Robotics System Toolbox

[Link pentru schema Simulink a brațului robotic](https://github.com/IsacRiccardo/OpenManipulatorX). - user-ul poate modifica codul pentru a mișca cuplele ducându-se la: *Modelling-> Model Settings-> Model Properties -> Callbacks -> InitFcn*

Modele simulink aditionale se pot găsi la acest [link](https://github.com/mathworks-robotics/designing-robot-manipulator-algorithms).

[Tutorial pentru MATLAB-ROS.](https://forum.robotis.com/t/matlab-robot-manipulator-walkthrough-algorithms-with-openmanipulator-x/401)