

# Manual brat robotic

Open Manipulator-X

## Cuprins

Cuprins.....	2
Instalarea software-ului necesar .....	3
Ubuntu linux .....	3
Creare sistem dual boot Ubuntu .....	3
Creare sistem dual boot Linux Mint .....	6
ROS .....	9
Kinetic Ubuntu 16.04 – Tested (working) .....	9
Kinetic Linux Mint 18.x (Manual installation) – Tested (working).....	9
Noetic – Tested (working on Ubuntu 20.04) .....	10
Instalare pachete ROS.....	11
Kinetic Ubuntu 16.04.x & Linux Mint 18.x – Tested (working).....	11
Noetic – Tested (not working, error installing moveit package) .....	11
Gazebo .....	11
Lansarea controller-ului pentru Gazebo.....	12
GUI.....	13
Interfețe de comunicare .....	13
MATLAB - ROS.....	15

## Instalarea software-ului necesar

### Ubuntu linux

Sistemul de operare Ubuntu este necesar pentru mediul de simulare Gazebo 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](#)

[Link instalare Linux Mint](#)

[Tutorial](#)

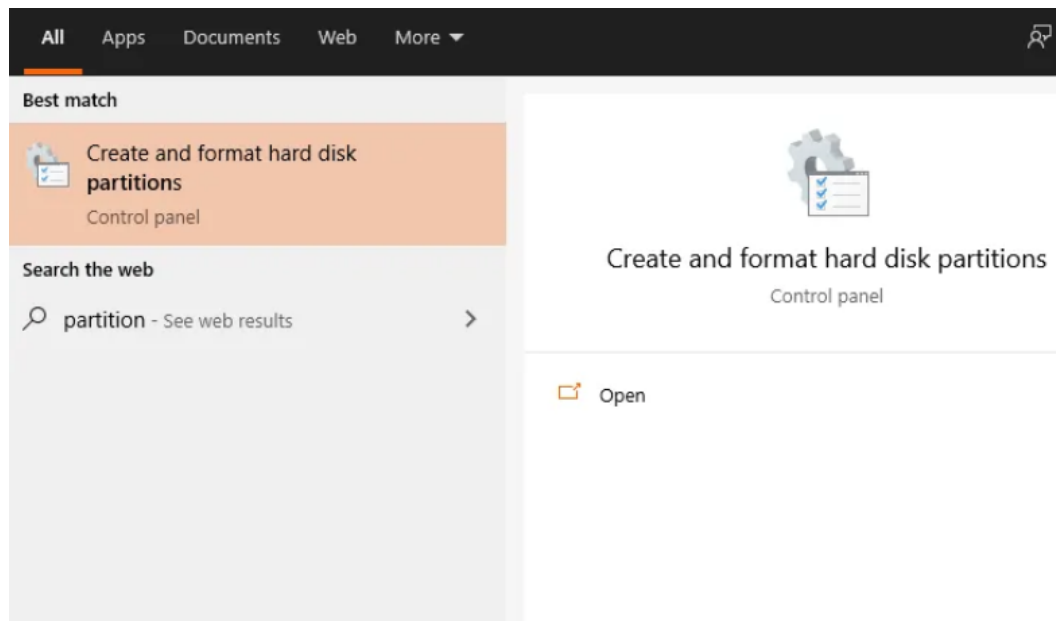
**! Atenție !** După instalarea versiunii de Ubuntu nu faceți update la o versiune mai nouă.

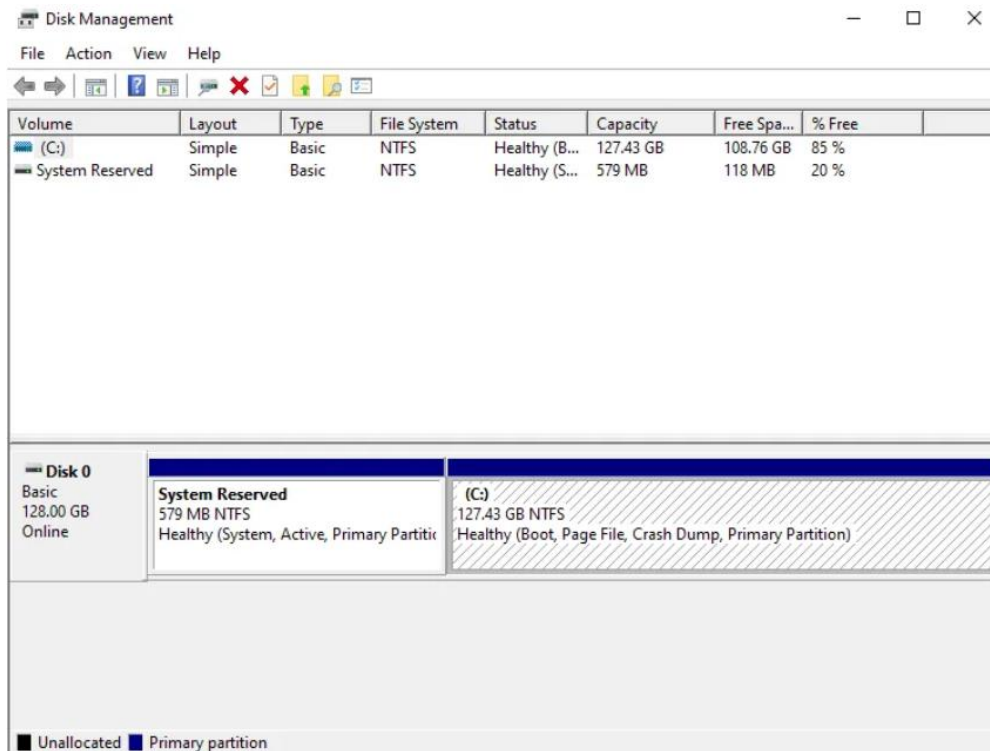
**! LIMITĂRI !** Dacă se alege instalarea Ubuntu 16.04.x sau Linux Mint 18.x s-ar putea să nu găsiți toate driver-ele pentru PC-ul vostru, de exemplu WI-FI sau TouchPad.

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

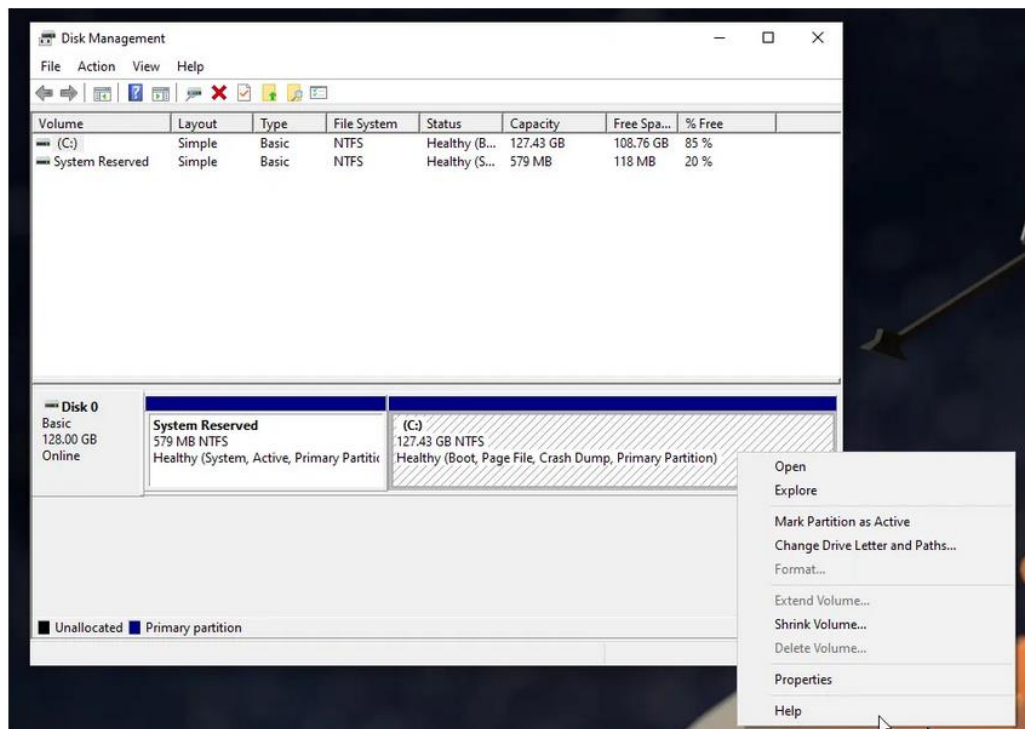
Pentru crearea unui sistem dual boot, primul pas este de a alocă o partiție pentru sistemul de operare Ubuntu. Pentru aceasta căutam în bara de cautare Windows, *Disk Management*.



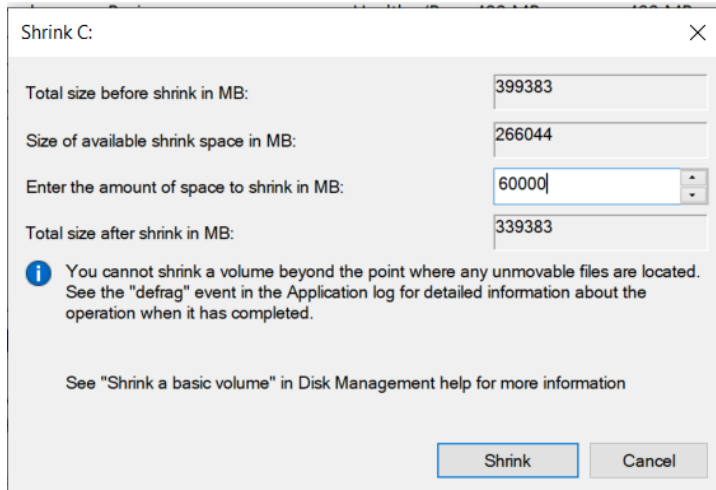


Următorul pas, este de a micșora dimensiunea partiției în care Ubuntu va fi instalat. Spațiul micșorat va fi folosit pentru instalare. Ținem cont de faptul că dacă se dorește instalarea MATLAB pe Ubuntu, va trebui să alocăm minim 60 Gb.

Alegem opțiunea de “Shrink Volume...”.



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

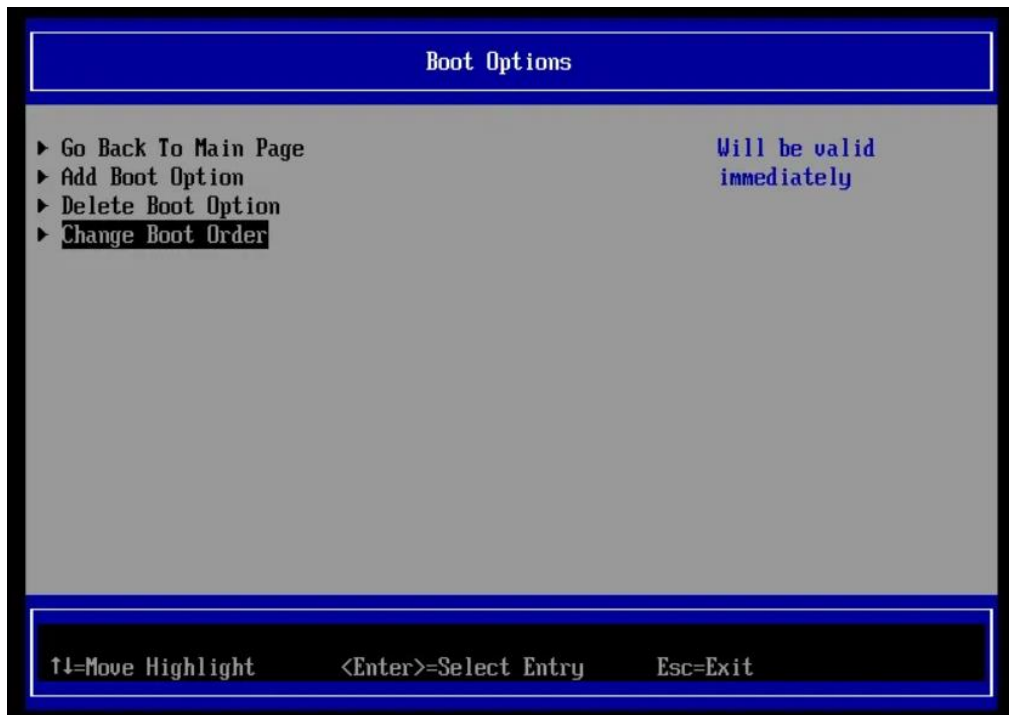


După ce a fost creată, zona de memorie va apărea nealocată.

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 încât opțiunea de USB să fie prima.

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



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 selectând 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

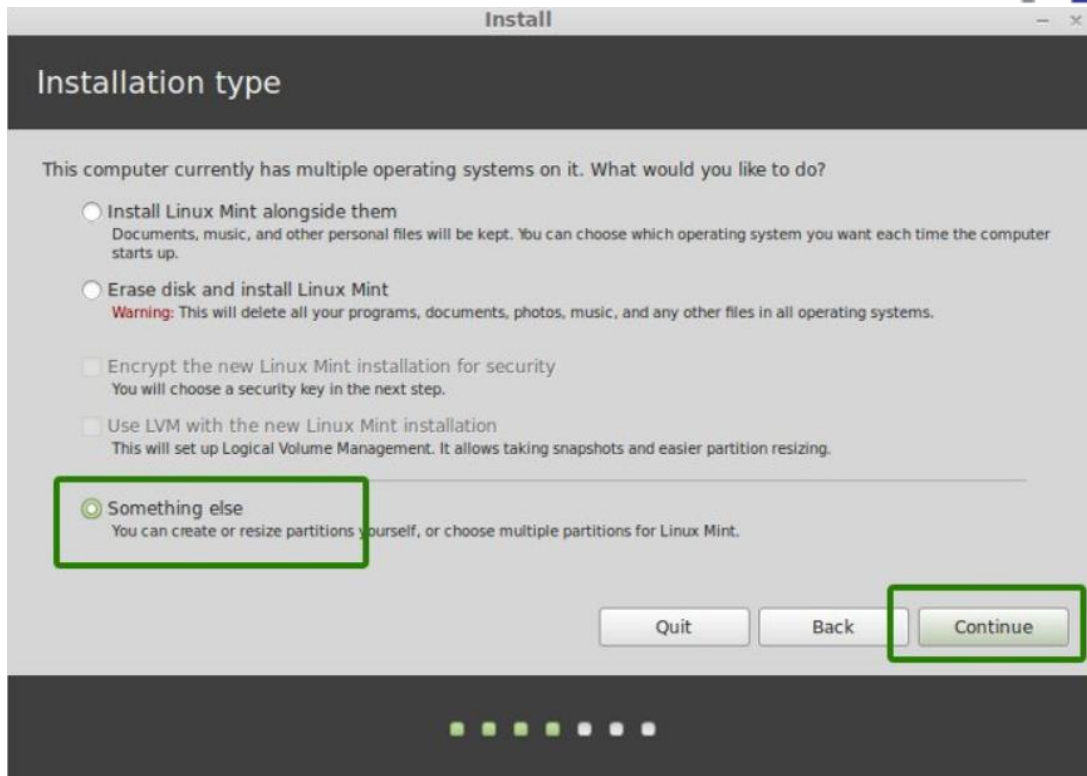
În cazul în care s-a dorit instalarea sistemului de operare Linux Mint (Asta dacă se dorește a lucra cu versiunea Kintetic de ROS) pașii pentru crearea unui sistem dual boot sunt următorii.

**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.

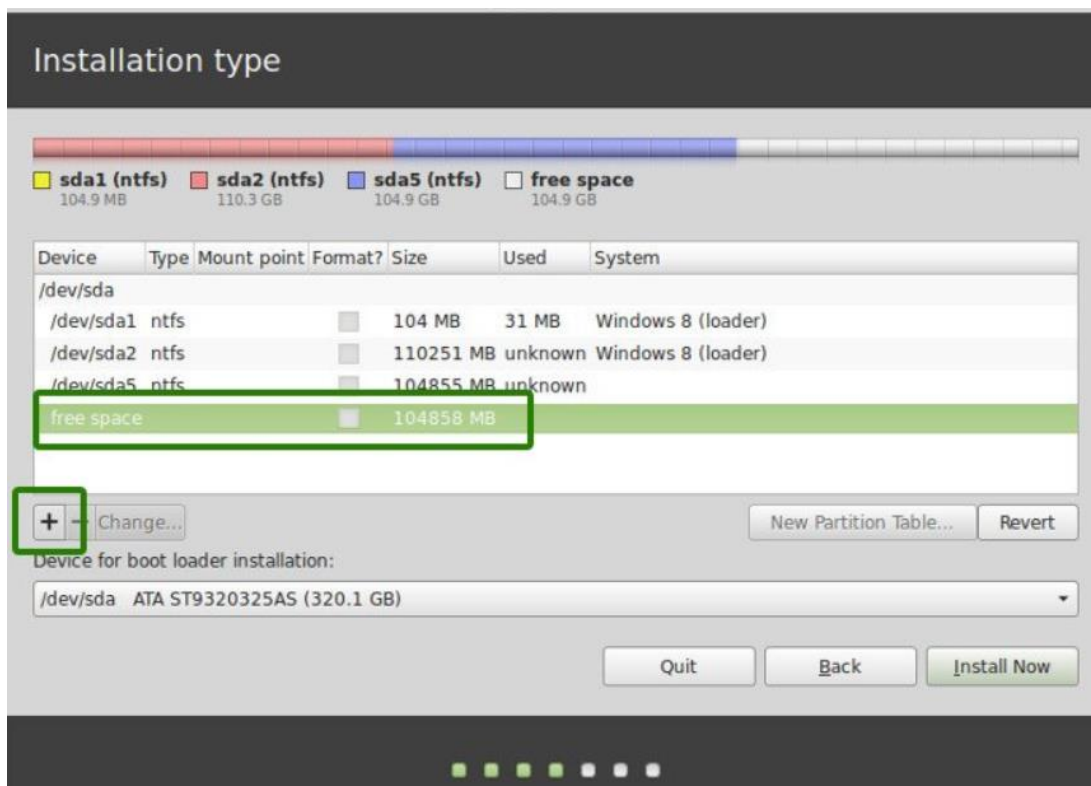


La fereastra “Installation type” vom selecta “Something else”.

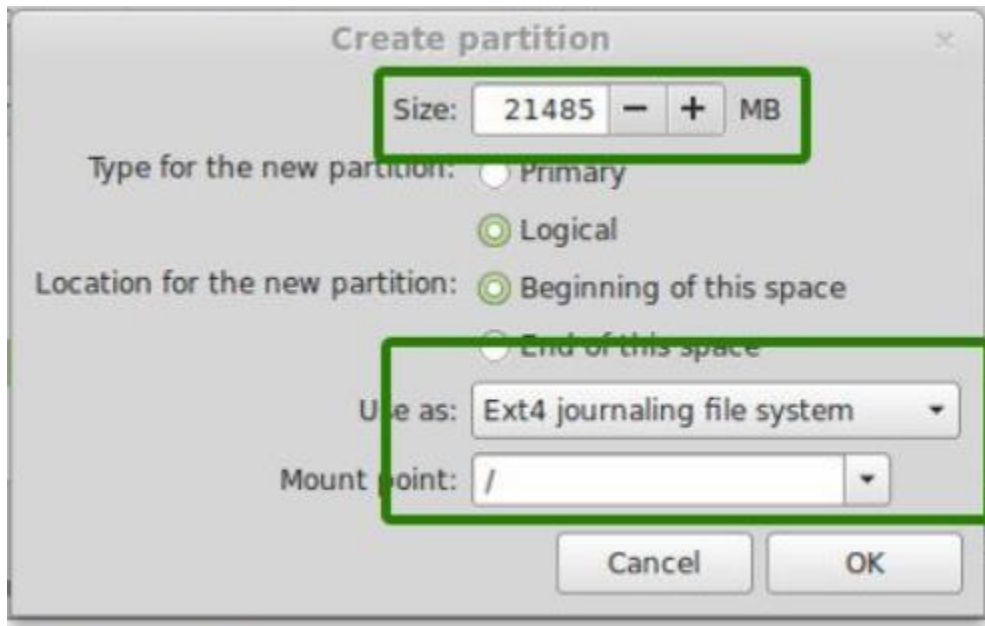


Mai departe va trebui să creăm 3 partiții root, swap și home.

Selectăm partiția nealocată creată in Windows.



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

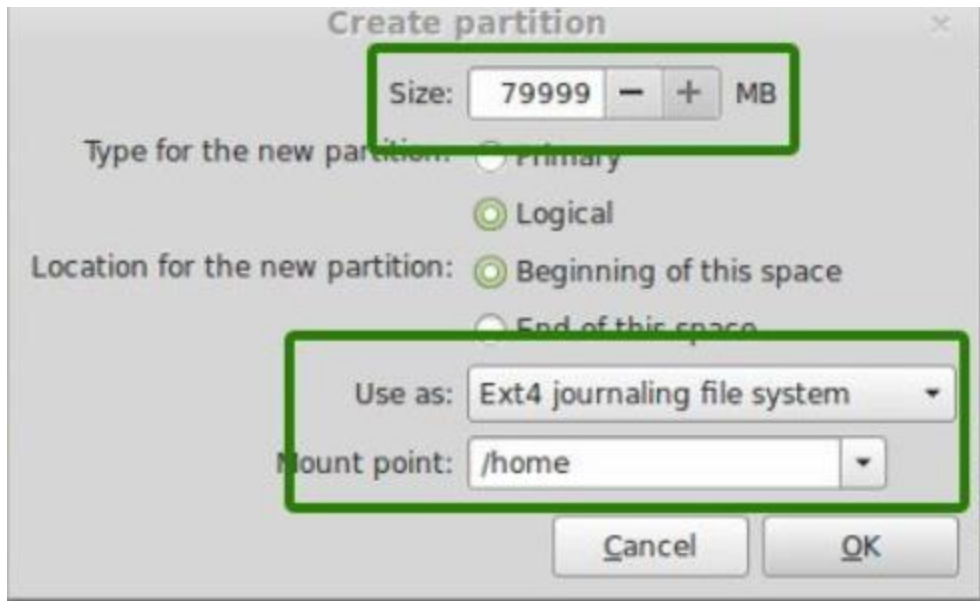


Partiția swap este următoarea, aceasta va trebui să fie semnificativă doar dacă memoria RAM de pe dispozitiv este limitată (ex: 2 – 4 Gb). Însă se pot aloca 4Gb indiferent de acest lucru.



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





După crearea partițiilor 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, această instalare este făcută cu ajutorul unui bash script (install\_ros\_kinetic/noetic.sh). Pentru o instalare manuală vizitați linkul: [Instalare Manuală](#)

## Instalare pachete ROS

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

### Descarcă dependențele

```
$ 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 dependențele

```
$ 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 în mod corect, următorul mesaj ar trebui să apară pe terminalul în care a fost lansat Gazeboo.

### *Kinetic*

```
SUMMARY
=====

PARAMETERS
* /open_manipulator/control_period: 0.01
* /open_manipulator/moveit_sample_duration: 0.05
* /open_manipulator/planning_group_name: arm
* /open_manipulator/using_moveit: False
* /open_manipulator/using_platform: False
* /roscdistro: kinetic
* /rosversion: 1.12.14

NODES
/
  open_manipulator (open_manipulator_controller/open_manipulator_controller)

ROS_MASTER_URI=http://localhost:11311

process[open_manipulator-1]: started with pid [9820]
[ INFO] [1544506914.862653563]: Ready to simulate /open_manipulator on Gazebo
```

## Noetic

```
SUMMARY
=====

PARAMETERS
* /open_manipulator_controller/control_period: 0.01
* /open_manipulator_controller/using_platform: True
* /roscdistro: noetic
* /rosversion: 1.15.9

NODES
/
  open_manipulator_controller (open_manipulator_controller/open_manipulator_controller)

auto-starting new master
process[master]: started with pid [5454]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to da579122-f0fa-11eb-9d7a-0790f3842b2b
process[rosout-1]: started with pid [5464]
started core service [/rosout]
process[open_manipulator_controller-2]: started with pid [5467]
port_name and baud_rate are set to /dev/ttyUSB0, 1000000
Joint Dynamixel ID : 11, Model Name : XM430-W350
Joint Dynamixel ID : 12, Model Name : XM430-W350
Joint Dynamixel ID : 13, Model Name : XM430-W350
Joint Dynamixel ID : 14, Model Name : XM430-W350
Gripper Dynamixel ID : 15, Model Name : XM430-W350
[INFO] Succeeded to init /open_manipulator_controller
```

## GUI

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

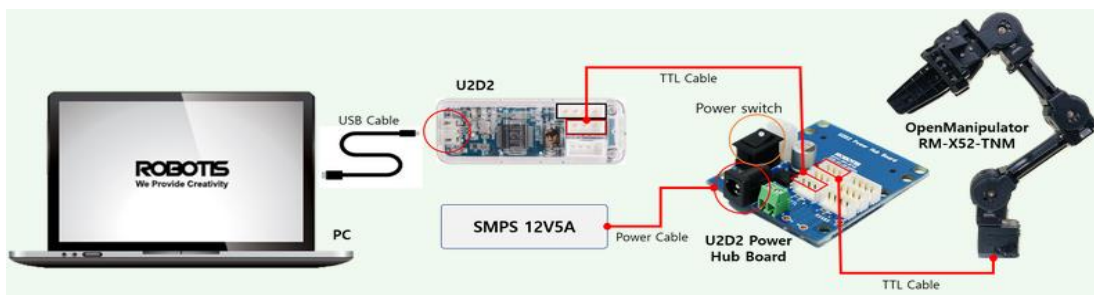
```
$ roslaunch open_manipulator_control_gui open_manipulator_control_gui.launch
```

Pentru mai multe informații sau în caz de nedumeriri apăsați [aici](#).

## Interfețe de comunicare

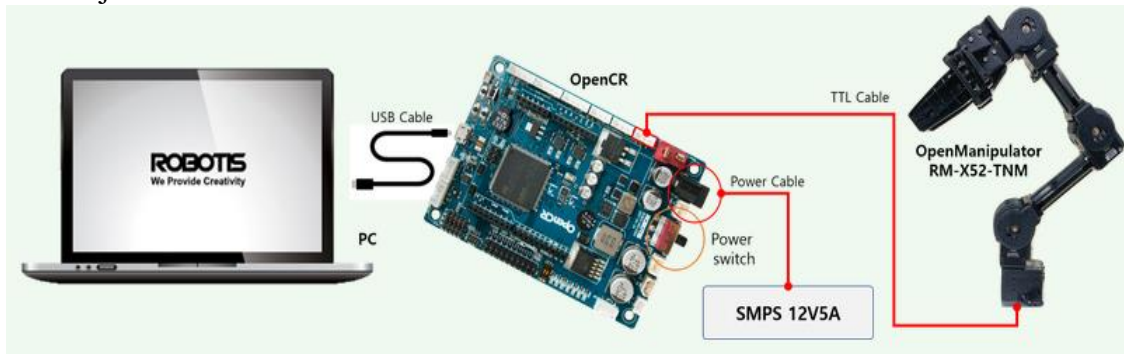
Pentru a comunica cu brațul sunt disponibile 2 interfețe:

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



Placa și cablul de alimentare pot fi achiziționate de [aici](#).

2. OpenCR ([disponibil](#)) – În cazul în care această modalitate este aleasă, OpenCR poate fi flash-uit cu ajutorul mediului Arduino IDE.



[Despre OpenCRv1.0.](#) (Accesează pentru mai multe informații cum ar fi memory mapping, update bootloader, diagrame de stări, protocoale de comunicații)

Instalează [ArduinoIDE pe Ubuntu.](#)

## 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 Multibody
- Robotics System Toolbox

[Link pentru schema Simulink a brațului robotic](#). - 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](#).

[Tutorial pentru MATLAB-ROS.](#)