If you are unfamiliar with Linux and ROS, we suggest starting with the desktop versions of the SWs and later migrate to the lite ones.

1. Download the Raspbery Pi OS (either Desktop or Lite version) from the following link: <https://www.raspberrypi.org/software/operating-systems/>
2. Mount the image on the SD with the help of balenaetcher software:

<https://www.balena.io/etcher/>

1. Setup the environment. Here you can choose the method of development:
   1. Peripherals approach (connect keyboard, monitor and mouse attached)
   2. Remote approach (VNC or SSH connection). The following lines are explaining the ssh aproach
      1. You can set up the network by creating a **wpa\_supplicant.conf** file under boot (SD card).

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=RO

network={

        ssid="SSID"

        psk="Passwd"

}

* + 1. You can enable the ssh and the i2c connection by creating a file **ssh.** another file **i2c.** and a file **camera.** under boot (SD card).
    2. Scan the network for your new IP when you power up te RPi
    3. You can connect to the RPi IP from any terminal with the command ssh [pi@192.168.\*.\*](mailto:pi@192.168.*.*)

1. Add the ROS Debian repo to the OS

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu buster main" > /etc/apt/sources.list.d/ros-noetic.list'

1. Add official ROS key

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

1. Pull all meta info from ROS noetic packages

sudo apt-get update && sudo apt-get upgrade

1. Install build dependencies

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

1. Install pip3

sudo apt install python3-pip

1. Install opencv:

sudo apt install libopencv-dev python3-opencv

1. Setup ROS dependency sources/repos

sudo rosdep init

rosdep update

1. Fetch and install ROS dependencies

mkdir -p ~/ros\_catkin\_ws

cd ~/ros\_catkin\_ws

Lite version:

rosinstall\_generator ros\_comm sensor\_msgs cv\_bridge --rosdistro noetic --deps --wet-only --tar > noetic-ros\_comm-wet.rosinstall

wstool init src noetic-ros\_comm-wet.rosinstall

Desktop version

rosinstall\_generator desktop --rosdistro noetic --deps --wet-only --tar > noetic-desktop-wet.rosinstall

wstool init src noetic-desktop-wet.rosinstall

rosdep install -y --from-paths src --ignore-src --rosdistro noetic -r --os=debian:buster

1. Compile ROS packages
   1. Since the ROS project is resource consuming, it is also recommended, but not mandatory, to increase the swap memory to 1 GB. You can decrease it afterwards. By following the same steps and setting it to 100

sudo dphys-swapfile swapoff

sudoedit /etc/dphys-swapfile

CONF\_SWAPSIZE=1024

sudo dphys-swapfile setup

sudo dphys-swapfile swapon

sudo src/catkin/bin/catkin\_make\_isolated --install -DCMAKE\_BUILD\_TYPE=Release --install-space /opt/ros/noetic -j1 -DPYTHON\_EXECUTABLE=/usr/bin/python3

1. Verify installation

source /opt/ros/noetic/setup.bash

roscore

1. You can also set the sourcing at the startup of each new terminal.

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

1. Fix missing package

sudo apt install libatlas-base-dev

1. Copy the project locally and move inside it
2. Install python dependencies

pip3 install -r requirements\_rpi.txt

pip3 install numpy --upgrade

1. Set up the i2c communication for the IMU by following the Setting up the Raspberry Pi side from this tutorial:

<https://github.com/RPi-Distro/RTIMULib/tree/master/Linux>

1. For additional topics, check the official ROS documentation for Raspbery Pi:

<http://wiki.ros.org/ROSberryPi/>

1. Build and run the prepared project

catkin\_make

source devel/setup.bash

roslaunch utils run\_automobile\_remote.launch

It will run the same brain project only in ROS variant. If you want to test it remotely you can run the remotecontroltransmitter and camerareceiver from the startup project (don’t forget to edit the IP’s from CameraTransmitterProcess on rpi and the remotecontroltransmitterProcess from the remote).

If you wish to install additional ROS packages after the installation, you will have to:

cd ~/ros\_catkin\_ws

sudo rm -rf build\_isolated/ devel\_isolated/ and src

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

rosinstall\_generator name\_of\_new\_pkg --deps --exclude RPP > new\_pkg.rosisntall

wstool init src new\_pkg.rosisntall

sudo -s

nano /root/.bashrc

add source /opt/ros/noetic/setup.bash

source /root/.bashrc

catkin\_make\_isolated --install -DCMAKE\_BUILD\_TYPE=Release --install-space /opt/ros/noetic

nano /root/.bashrc

remove source /opt/ros/noetic/setup.bash

exit