

All steps needed to run my visual controller on Ubuntu 20.04:

Table of contents

- [Installing ROS](#)
 - [Ardupilot with MAVProxy](#)
 - [Gazebo](#)
 - [Mission Planner](#)
 - [Yolo libraries](#)
 - [More informations](#)
 - [Other usefull commands](#)
 - [Contact](#)
 - [Direct Sources](#)
 - [Based upon Intelligent Quads series](#)
-

Installing ROS:

Lets start with getting packages.

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

Instead of focal (Ubuntu 20) get your version of OS.

```
sudo apt install curl # if you haven't already installed curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc |
sudo apt-key add -
```

```
sudo apt update
```

```
sudo apt install ros-noetic-desktop-full
```

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

If you want your ros in every terminal then type:

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

Very important - dependencies

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-
generator python3-wstool build-essential
```

```
sudo rosdep init
rosdep update
```

Catkin:

First get python 2 instead of python 3.

```
sudo apt install python2
```

Switch it to default version.

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python2
1
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3
2
```

Now we good to go:

```
sudo apt-get install python3-wstool python3-rosinstall-generator python3-
catkin-tools
```

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws
catkin init
```

From catkin_ws directory run:

```
wstool init ~/catkin_ws/src
rosinstall_generator --upstream mavros | tee /tmp/mavros.rosinstall
rosinstall_generator mavlink | tee -a /tmp/mavros.rosinstall
```

```
wstool merge -t src /tmp/mavros.rosinstall
wstool update -t src
rosdep install --from-paths src --ignore-src --rosdistro `echo $ROS_DISTRO`
-y --os=ubuntu:focal
catkin build
```

Resorce again

```
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
source ~/.bashrc
sudo
~/catkin_try/src/mavros/mavros/scripts/install_geographiclib_datasets.sh
```

Get the example file to check if it works!

```
cd ~/catkin_ws/src
git clone https://github.com/Intelligent-Quads/iq_sim.git
echo
"GAZEBO_MODEL_PATH=${GAZEBO_MODEL_PATH}:$HOME/catkin_ws/src/iq_sim/models"
>> ~/.bashrc
cd ~/catkin_ws
catkin build
source ~/.bashrc
```

We are done with ros ...at lest for now

Ardupilot with MAVProxy

```
cd ~
sudo apt install git
git clone https://github.com/ArduPilot/ardupilot.git
cd ardupilot
```

Dependencies in ardupilot's directory:

```
Tools/environment_install/install-prereqs-ubuntu.sh -y
. ~/.profile
```

Get the newest version of the Copter in my case 4.2 Check it on github.

```
sudo apt-get install python3-pip python3-matplotlib python3-serial python3-opencv
```

```
sudo pip install pexpect
```

Checkout version which you are interested in.

```
git checkout Copter-4.0
git submodule update --init --recursive
```

```
cd ~/ardupilot/ArduCopter
sim_vehicle.py -w
```

Gazebo:

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-
stable focal main" > /etc/apt/sources.list.d/gazebo-stable.list'
```

Instead of focal (Ubuntu 20) get your version of OS.

```
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

```
sudo apt update
```

```
sudo apt-get install gazebo11 libgazebo11-dev
```

```
cd ~
git clone https://github.com/khancyr/ardupilot_gazebo.git
cd ardupilot_gazebo
```

```
mkdir build
cd build
cmake ..
make -j4
sudo make install
```

Source it:

```
echo 'source /usr/share/gazebo/setup.sh' >> ~/.bashrc
echo 'export GAZEBO_MODEL_PATH=~/.ardupilot_gazebo/models' >> ~/.bashrc
. ~/.bashrc
```

Run to check if it is okay:

Terminal 1:

```
gazebo --verbose ~/.ardupilot_gazebo/worlds/iris_arducopter_runway.world
```

Terminal 2:

```
cd ~/.ardupilot/ArduCopter/
sim_vehicle.py -v ArduCopter -f gazebo-iris --console
```

Mission Planner:

Instal mono:

```
sudo apt install gnupg ca-certificates
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
echo "deb https://download.mono-project.com/repo/ubuntu stable-focal main"
| sudo tee /etc/apt/sources.list.d/mono-official-stable.list
sudo apt update
sudo apt install mono-complete
```

Yolo libraries:

```
sudo apt install nvidia-cuda-toolkit
```

```
cd ~/catkin_ws/src
git clone https://github.com/kunaltyagi/darknet_ros.git
cd darknet_ros
git checkout opencv4
git submodule update --init --recursive
```

Building:

```
catkin build -DCMAKE_BUILD_TYPE=Release -DCMAKE_C_COMPILER=/usr/bin/gcc-8
```

Configure YOLO/Darknet

in the file `ros.yaml` specifies ros parameters. You can find this file under `darknet_ros/darknet_ros/config`. You will need to change the image topic from `/camera/rgb/image_raw` to

```
/webcam/image_raw
```

The file `darknet_ros.launch` will launch the darknet/yolo ros node. You can find this file under `darknet_ros/darknet_ros/launch` in this file you can choose which version of yolo you would like to run by changing

```
<arg name="network_param_file"          default="$(find
darknet_ros)/config/yolov2-tiny.yaml"/>
```

the options are as follows

- yolov1: Not recommended. this model is old
- yolov2: more accurate, and faster.
- yolov3: about as fast as v2, but more accurate. Yolo v3 has a high GPU ram requirement to train and run. If your graphics card does not have enough ram, use yolo v2
- tiny-yolo: Very fast yolo model. Would recommend for application where speed is most important. Works very well on Nvidia Jetson

Quote from [iqs tutorial no yolo](#)

Lastly lets add

```
echo 'export
ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/catkin_ws/src/darknet_ros/darknet_ro
s' >> ~/.bashrc
source ~/.bashrc
```

More informations:

Other usefull commands:

low framerate:

```
sudo dpkg --configure -a
```

Contact:

- Made by Krzysztof B as a part of BEng thesis.
- You can msg me via [github](#).

Direct Sources:

- <http://wiki.ros.org/noetic/Installation/Ubuntu>
- <https://pjreddie.com/darknet/yolo/>
- <https://ardupilot.org/copter/index.html>
- <https://www.python.org/downloads/>
- <https://classic.gazebosim.org/download>
- <https://ardupilot.org/mavproxy/>

Based upon [Intelligent Quads](#) series:

- https://github.com/Intelligent-Quads/iq_tutorials/blob/master/docs/installing_gazebo_arduplugin.md
- https://github.com/Intelligent-Quads/iq_tutorials/blob/master/docs/Installing_Ardupilot_20_04.md
- https://github.com/Intelligent-Quads/iq_tutorials/blob/master/docs/installing_ros_20_04.md
- https://github.com/Intelligent-Quads/iq_tutorials/blob/master/docs/intro_to_yolo.md