

---

# **robdos\_sim Documentation**

*Release 0.1*

**Francisco J. Garcia R.**

**Apr 04, 2017**



---

## Contents

---

<b>1</b>	<b>Contents:</b>	<b>3</b>
1.1	How to install suricate robot . . . . .	3
1.2	Suricate robot tutorials . . . . .	4
1.3	Contact us . . . . .	5



Simulations for robdos project, more info: <http://www.robdosteam.com>



## How to install suricate robot

In order to use this project, we must install first ROS (tested on Jade and kinectic), then compile the project and finally run some simulations and tests.

### Install ROS and dependencies

Follow ros installation procedure:

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

we can summarize the steps:

Open a command windows on ubuntu and run the following commands:

- Prepare ubuntu for installation:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /  
↳etc/apt/sources.list.d/ros-latest.list'  
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 0xB01FA116  
sudo apt-get update
```

- Install ROS

For PC/Laptop we should install full desktop version:

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

For Raspberry PI 2 and Odroid we can install ROS-Base:

```
sudo apt-get install ros-kinetic-ros-base
```

Install rosdep:

```
sudo rosdep init
rosdep update
```

And finally prepare environment:

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

## Compile project

First we need to install some dependencies and ROS packages:

```
sudo apt-get install libqwt-dev ros-kinetic-teleop-twist-joy ros-kinetic-rviz-imu-
↳plugin python-smbus ros-kinetic-rqt-multiplot ros-kinetic-mavros*
```

Finally we create a workspace for project, clone github repository, install dependencies and compile it:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
git clone https://gitlab.com/SIMULACION_SUBMARINA/robdos_sim -b kinetic
cd ..
source devel/setup.bash
rosdep install robdos_sim
catkin_make
```

## Test project

Once the project has been compiled successfully, we can run a simulation that includes robot using a simple scenario.

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch robdos_sim robdos_sim.launch
```

## Suricate robot tutorials

This section explain some examples of how to use the gazebo simulation and ROS in order to play with the robdos robot and develop some extensions.

this section assume you already have installed robdos project.

### Play bag files

In order to understand how variables changes inside the robot, let's play with a recorded bag files.

Remember download first the bag files in your computer and change the path into launch file (robdos\_sim\_play\_bags.launch)

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch robdos_sim robdos_sim_play_bags.launch
```

Once the launch file have started, a RVIZ window will open and show the movement of robot.



## Waypoints using gazebo and RVIZ

You can execute a simple demo using waypoints (mavros\_msgs/WaypointList), a simulated environment in gazebo and RVIZ.

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch robdos_sim robdos_waypoints.launch
```

This command will open 2 main windows: a RVIZ viewer for waypoint visualization and a gazebo window with a simulated robot that execute the proper velocity commands.

The previous demo can be summarized in the following processes:

- Waypoint publisher

A python script (waypoint\_publisher.py), it publishes a topic /mavros/mission/waypoints (type mavros\_msgs/WaypointList) with a set of points to be reached by the robot.

it generates also a set of markers using topic /robdos/makers\_waypoints (type visualization\_msgs/Marker) for visualization using RVIZ.

- Waypoint controller

A python script (waypoint\_controller.py), it publishes a velocity command for mavlink using topic /mavros/rc/out (type mavros\_msgs/RCOut).

It subscribes for odometry information (position/orientation) to /robdos/odom (type nav\_msgs/Odometry) (from gazebo simulation but also possible to subscribe for real odometry topic)

It also subscribes for waypoint list /mavros/mission/waypoints in order to perform a simple controller.

This demo only considers the first waypoint, but it will include the other coming soon.

- Gazebo plugin (gazebo\_sub\_drive)

In order to perform a realistic simulation of the robot, we have develop gazebo plugin that publishes simulated odometry and subscribes for desired velocity commands (mavros\_msgs/RCOut).

This node convert RCOut values into valid angular velocity for thrusters, then, the Lift-Drag plugin and gazebo computes the resulting force over the robot.

## Contact us

- If you want more info about the project:

Francisco J. Garcia R. - [garciar@rhrk.uni-kl.de](mailto:garciar@rhrk.uni-kl.de)

- robdos simulation license:

Copyright (C) 2016 Francisco Garcia

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

This project is released under GPLv2 license, please consider that external plugins may have a different type of license.

Developed by:

- Francisco J. Garcia R. - [garcia@rhrk.uni-kl.de](mailto:garcia@rhrk.uni-kl.de)