How to use ROS and Gazebo with the ROBOTIS OP2 - Summary

ROS Key Concepts

Nodes	Modular separated programs
Master	Main node, manages e.g. address spaces
Messages	Data structures to exchange information
Topics	Message “channels”, subscribe to topics to receive messages, publish on topics to send messages.

Example Publisher

```
//Initializing Publisher
ros::Publisher vel_pub_;
vel_pub_ =
nh_.advertise<geometry_msgs::Twist>("r
obotis_op/cmd_vel", 1);
//Sending message
geometry_msgs::Twist vel;
    vel.angular.z = 0.5;
    vel_pub_.publish(vel);
```

Example Subscriber

```
// Initializing Subscriber
ros::NodeHandle nh_;
ros::Subscriber image_sub_;
image_sub_ =
nh_.subscribe("/robotis_op/camera/imag
e_raw", 100,
&RobotisOPBallTrackingNode::imageCb,
this);
//Receiving Image Callback
void
RobotisOPBallTrackingNode::imageCb(con
st sensor_msgs::Image& msg)
{
   cv_bridge::CvImagePtr image_ptr;
   image_ptr =
cv_bridge::toCvCopy(msg,sensor_msgs::i
mage_encodings::RGB8);
   […]
}
```

Starting the simulator

Starting Gazebo

```
roslaunch robotis_op_gazebo
robotis_op_gazebo_position_control_soc
cer_field.launch
```

Wait until all controllers are loaded, then press the play button in Gazebo. During the first start it may take longer, since the models have to be downloaded from the database.

Starting the simulation walking

```
rosrun robotis_op_simulation_walking
walker.py
```

Starting the ball tracker

```
rosrun
robotis_op_ball_tracker_tutorial
robotis_op_ball_tracker_tutorial_node
```

Launching the real robot

```
ssh robotis@192.168.123.1
sudo killall demo
roslaunch robotis_op_onboard_launch
robotis_op_whole_robot.launch
export
ROS_MASTER_URI=http://192.168.123.1:11
311
```

Build source

```
cd ~/catkin_ws/
catkin_make
```

Operating the robot via Terminal

The robot can be operated by sending messages on topics. To see all available topics

```
rostopic list
```

To get more information about a topic **rostopic info TOPIC_NAME**

Example

```
rostopic info /robotis_op/cmd_vel
```

All topics starting with /robotis_op/ are related to the Robotis OP2.

/robotis_op/j_*/command command a joint position (radian) to the corresponding actuator

Example

```
rostopic pub
/robotis_op/j_pan_position_controller/
command std_msgs/Float64 "data: 0.5"
```

/robotis_op/enable_walking enables walking (real robot only)

Example

rostopic pub /robotis_op/enable_walking std_msgs/Bool "data: true"

/robotis_op/cmd_vel sets the walking velocity and direction

Example

```
rostopic pub /robotis_op/cmd_vel
geometry_msgs/Twist '{linear: {x: 0.5,
y: 0.0, z: 0.0}, angular: {x: 0.0, y:
0.0, z: 0.0}}'
```