GUARDIAN-WAM MOBILE ROBOT

System Start-up Manual
Version 4.1

Robotnik Automation, S.L.L.

29/02/2012

RBTNK-DOC-120229E

INDEX

WARNING!

The Guardian robot is a powerful platform able to move up to speeds of 2m/s. Special care must be taken while operating the robot. Please follow these safety recommendations:

- Do not allow anybody near the robot while the motor drives are enabled.

- Keep the emergency stop button within reach of an operator while running if there is potential danger to persons or equipment.

- Enable the emergency stop while servicing the tracks or the wheels.

- Check regularly that the emergency stop works properly.

- Do not sprinkle water or oil on the robot or power charging cord. Contact with water or oil can cause electric shock or malfunction of the unit.

- Charge the robot only with the charger provided.

- Do not attempt to disassemble or modify the robot.

- Do not use the unit where flammable or corrosive gas is present. Leaked gas accumulated around the unit can cause fire or explosion.

- Plug the power cord into the wall outlet firmly. Incomplete insertion into the wall outlet makes the plug hot and can cause a fire.

- Check that the plug is not covered with dust.

1. Hardware

1.1. Electrical System

1.1.1. Main power supply

The robot receives its power from a 48V battery. With this battery, the robot is able to operate up to 4 hours depending on the robot movements.

The battery is connected directly to the robot through the security contactor (K1) and the control contactor (K2). There is a connector in the back of the robot where the charger can be connected. It is possible to charge the robot with the control connector open, but it is recommended to cut the power to the motor by pushing the emergency stop button.

Charging time is around 2 hours.

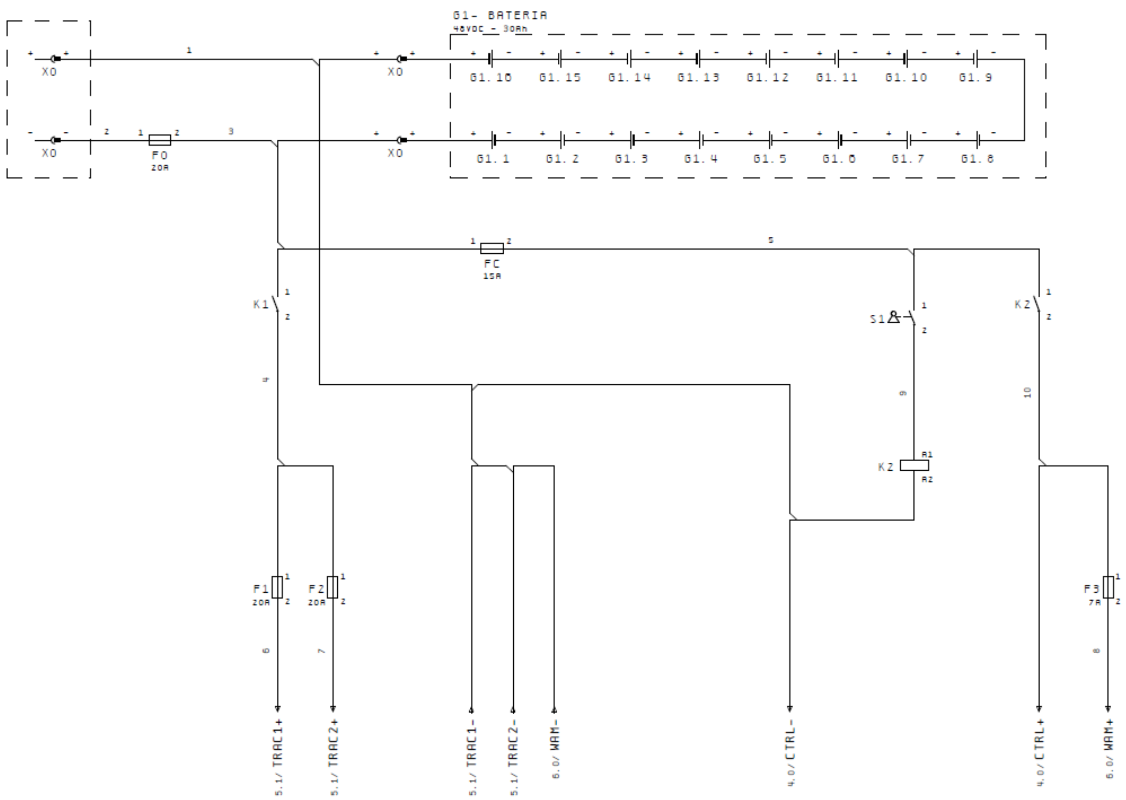

Figure 1 - Power supply drawing

1.1.2. Control Panel

The back cover of the robot has several buttons and indicators with the following functions:

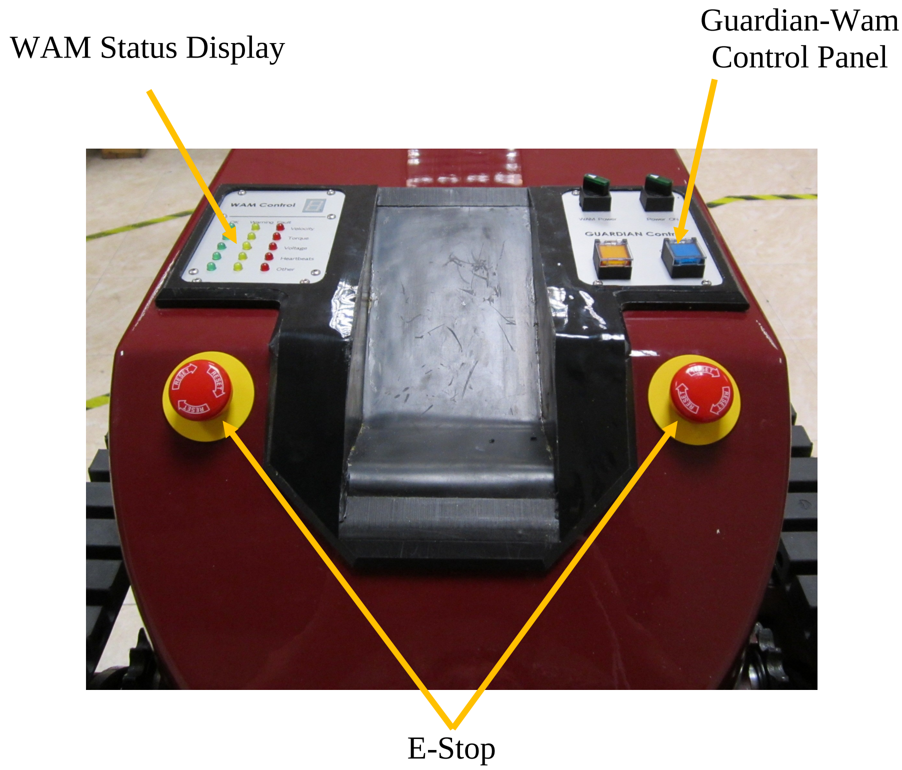

Figure 2 - Control panel

- The general ON/OFF indicator/switch cuts the power of the whole robot and indicates that the power supply has been activated.
- The WAM ON/OFF indicator/switch turns on/off the power of the WAM arm and indicates that the power supply has been activated.
- The restart indicator/button indicates that the security system has been activated, so motors cannot be moved

again until it is restarted. Pressing this button restarts/resets the security system.
- Each emergency stop button cuts power to both motors.
- The robot computer indicator/button indicates that the Guardian computer is on. Pressing this button enables power to the Guardian PC.

1.1.3. Start-up sequence

Please follow these steps:

1) Turn on the “power” switch to supply the whole system with electricity. It will turn on the “restart indicator” light.
2) Press the “Restart” button to enable motor power.
3) Press the “CPU Start” button to start the Guardian’s robot control computer.
4) After the computer boots, it is possible to connect to the system remotely via WiFi or directly via wired Ethernet.

1.1.4. Emergency Stop

The Guardian robot has an emergency stop system that cuts power to any mobile element that might injure a person or cause damage to the robot’s environment. In order to activate the security system, push either one of the emergency buttons located on the back cover.

Figure 3 - Emergency stop

It is strongly recommended to keep the emergency stop pressed whenever any maintenance operation is being done to the robot such as adjusting the camera, installing new components, or servicing the drive train. This will prevent injury or damage due to unexpected movement of the robot.

To recover the system from the emergency stop mode, please follow these steps:

- Ensure that it is safe to enable the robot at this time.
- Turn the emergency stop buttons clockwise until they release (about 1/8 of a turn).
- Push the “Restart” button located on the back panel.

1.2. Game-Pad System

1.2.1. Connection

This Bluetooth Game-pad is used for performing manual movements of the robot. The Bluetooth receiver is connected to a USB port of the computer inside the Guardian.

Figure 6 - Game-pad circuit

1.2.2. Game-pad Control Sequence

It is possible to use the Game-pad for operating the Guardian and its peripheral devices. Please follow these instructions:

1. Turn on the Robot
2. Turn on the Guardian computer by pressing the “CPU Start” button on the Guardian’s back panel. The computer indicator should illuminate.
3. Wait until the computer is started.
4. Release the emergency stop buttons.
5. Press the “Restart” button to reset the security system
6. When the system is ready, enable the Game-pad by pressing its “PAIRING” button.
7. Depending on the button configuration, you will now be able to move the robot, and activate its peripheral devices.

By default, the configuration of the buttons is the following:

Figure 8 - Gamepad front view

Figure 9 - Gamepad top view

1.3. RC Take-Over (optional)

It is possible to take the control of the Guardian using the RC emitter. The PC is not needed for this function; only the general ON/OFF switch must be activated.

1. Switch ON the RC emitter.
2. Switch CH5 of the RC emitter to the ON position. (Switch Up is RC OFF, and switch Down is RC ON)
3. Now you can move the robot with the joysticks of the RC emitter. Move left joystick up and down for traction, and the right one for direction. The normal use is left and right but up and down has special use if the optional back steering kit is installed.

Figure 5 - RC Emitter

2. Software

This section describes how to test the different components and how to operate the rover from an external PC using the ROS framework. A set of simulation experiments in Gazebo/Stage are also presented. The simulated Summit rover matches the kinematics and dynamics of the real rover. Nearly all the experiments performed in simulation should work with minor adjustments on the real robot.

2.1. Testing components in ROS with rviz

Refer to manual “System installation and configuration” in order to configure the network and the connection to the robot.

2.1.1. Testing the webcam sphere

In a linux console, launch the following launch configurations:

1) Launches the video driver.

```
$> roslaunch logitech_usb_webcam low_res.launch
```

2) Launches the pan-tilt driver.

```
$> roslaunch sphereptz test_sphereptz.launch
```

3) Visualization of the camera images in your local computer, you can run either rviz or the image_view node.

```
$> export OGRE_RTT_MODE=Copy (depending on the configuration)
$> rosrun rviz rviz
```

Remember to select /base_footprint as *Fixed Frame* in rviz *.Global Options.*

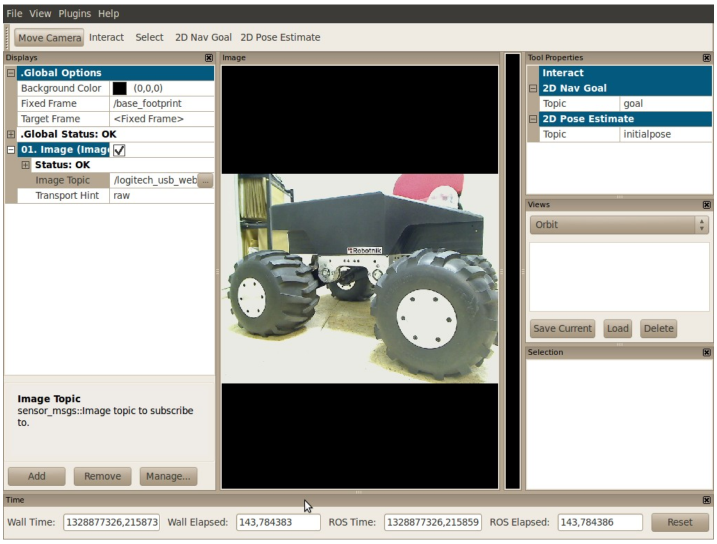

Figure 7. sphere image in rviz

```
$> rosrun image_view image_view:=<image topic>
```

4) To test the pan-tilt motion:

```
>roslaunch guardian_pad guardian_pad.launch
```

**Refer to section “1.2 Game-pad system” to see the button configuration.

2.1.2. Testing the Hokuyo laser ranger (optional component)

Run the following launch files in a linux terminal:

1) Launches the Hokuyo node

```
$> roslaunch hokuyo_node hokuyo_test.launch
```

2) Visualization of the range values.

```
$>export OGRE_RTT_MODE=Copy (depending on the configuration)
$>rosrun rviz rviz
```

After launching rviz, it is possible to load the hokuyo configuration file located in the hokuyo_node directory (roscd hokuyo_node).

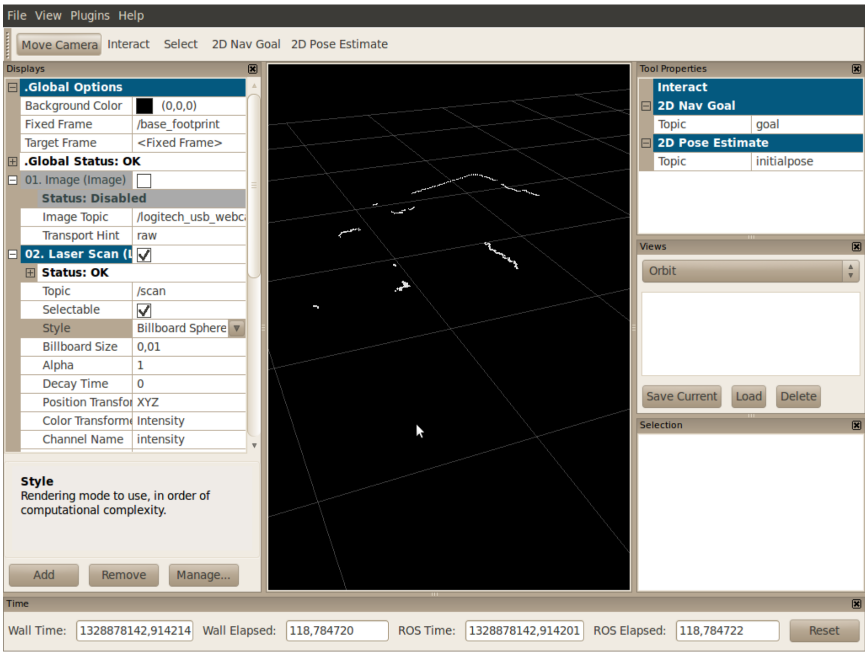

Figure 8. visualization of laser scans in rviz

2.1.3. Testing the inertial measurement unit, IMU (optional component)

```
>roslauch microstrain_3dmgx2_imu microstrain_3dmgx2.launch
```

After launching rviz, it is possible to load the IMU configuration file located in the microstrain_node directory (roscd microstrain_3dmgx2_imu). It will show the world frame and the robot IMU frame. Move the robot manually or with the joystick to check how the robot frame moves.

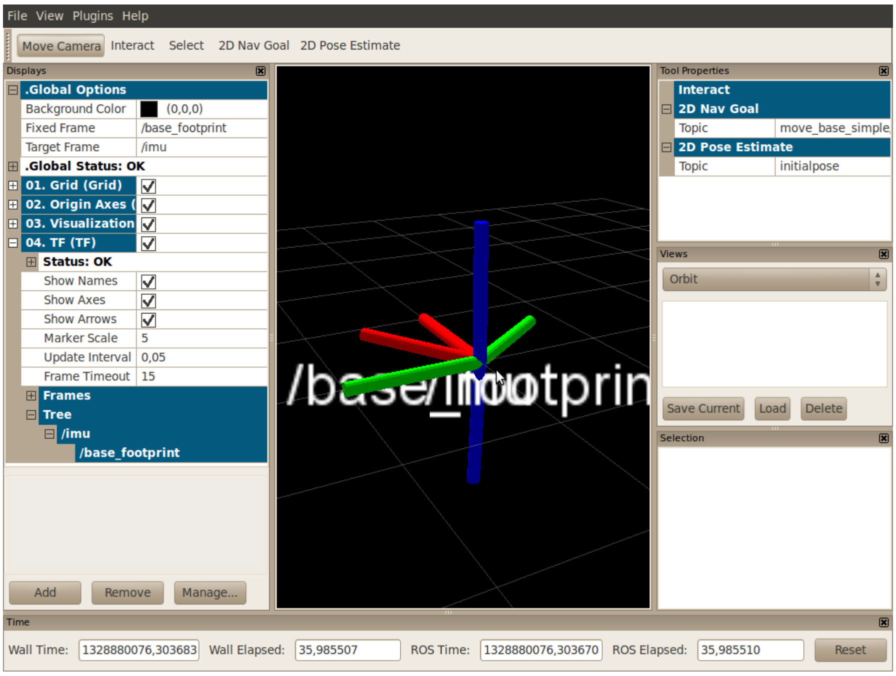

Figure 9.visualization of IMU frame in rviz

To see the complete IMU data:

```
>rostopic echo /imu/data
---
header:
  seq: 6711
  stamp:
    secs: 1328880205
    nsecs: 539570303
  frame_id: imu
orientation:
  x: 0.976004692987
  y: 0.21651466317
  z: 0.0121415921153
  w: 0.0197181787683
orientation_covariance: [0.0012250000000000002, 0.0, 0.0, 0.0,
0.0012250000000000002, 0.0, 0.0, 0.0, 0.0012250000000000002]
angular_velocity:
  x: 0.00280542019755
  y: -0.00460833963007
  z: 0.000865295878612
angular_velocity_covariance: [2.8900000000000004e-08, 0.0, 0.0, 0.0,
2.8900000000000004e-08, 0.0, 0.0, 0.0, 2.8900000000000004e-08]
linear_acceleration:
  x: 0.16600713928
  y: 0.426372697365
  z: -9.80492916756
linear_acceleration_covariance: [0.009604000000000014, 0.0, 0.0, 0.0,
0.009604000000000014, 0.0, 0.0, 0.0, 0.009604000000000014]
2. …
```

2.1.4. Testing the Microsoft Kinect 3d sensor (optional component)

```
>roslaunch openni_launch openni.launch
>rosrun rviz rviz
```

In rviz, select as fixed frame /camera_link, add point_cloud2 object, and configure topic camera/depth_registered/points.

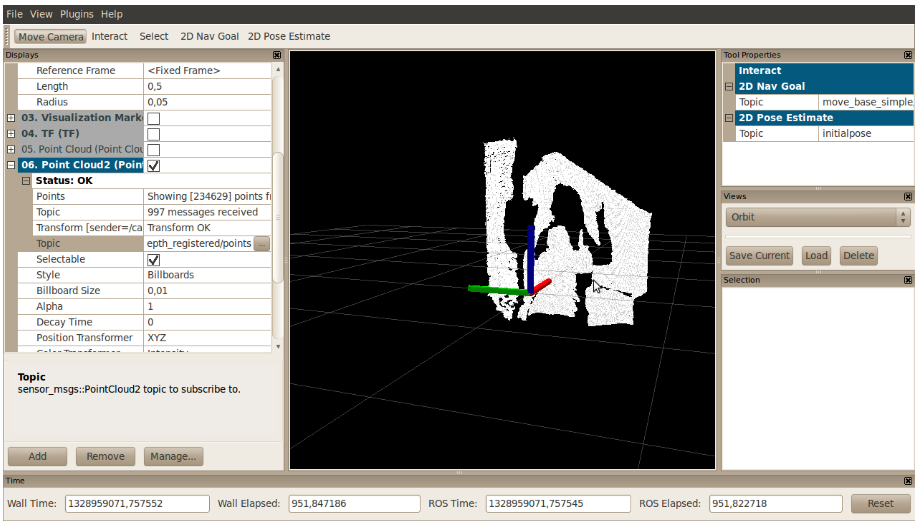

Figure 10.visualization of kinect point clouds in rviz

To see the infrared image, add image object and configure topic camera/image_ir/image_raw.

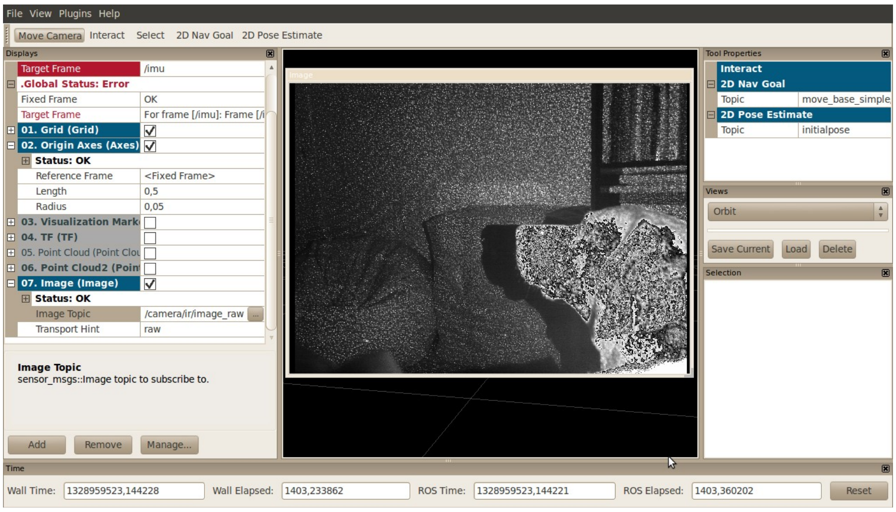

Figure 11.visualization of infrared images in rviz

2.1.5. Testing the modbus_io board (optional component)

Follow these steps to test the Analog and Digital Input/Output board:

1) Check the IP address (refer to the manual “System installation and configuration”)

2) Launch the node:

```
$> roslaunch modbus_io_node test_io.launch
```

3) View the present digital and analog values:

```
$> rostopic echo /modbus_io_node/input_output
```

4) Modify the digital and analog outputs:

```
$> rosservice call /modbus_io/write_digital_output 3 false
$> rosservice call /modbus_io/write_digital_output 0 true
$> rosservice call /modbus_io/write_analog_output 0 5.0
$> rosservice call /modbus_io/write_analog_output 1 6.0
```

2.1.6. Testing the Basler camera (optional component)

To test the Basler camera you should have installed and configured it properly (refer to the manual “System Installation and Configuration”). Once the camera is configured, you will able to launch and visualize the camera images.

1) Manual configuration and image_view

```
$> export ROS_NAMESPACE=basler_camera; rosrun camera_aravis camnode
$> export ROS_NAMESPACE=basler_camera; rosrun image_proc image_proc
$> rosrun image_view image_view image:=/basler_camera/image_color
```

2) Dynamic configuration and image_view

```
$> roslaunch camera_aravis test_basler_robotnik.lanch
```

It is also possible to view the camera image in rviz. However, not all image formats are supported. The list of supported formats can be found here:

http://www.ros.org/wiki/rviz/DisplayTypes/Camera

RVIZ currently supports RGB8, RGBA8, BGR8, BGRA8, MONO8, MONO16, bayer encoded (treated as MONO8), 8UC4 and 8SC4 (treated as BGRA8), 8UC3 and 8SC3 (treated as BGR8), 8UC1 and 8SC1 (treated as MONO8), 16UC1 and 16SC1 (treated as MONO16).

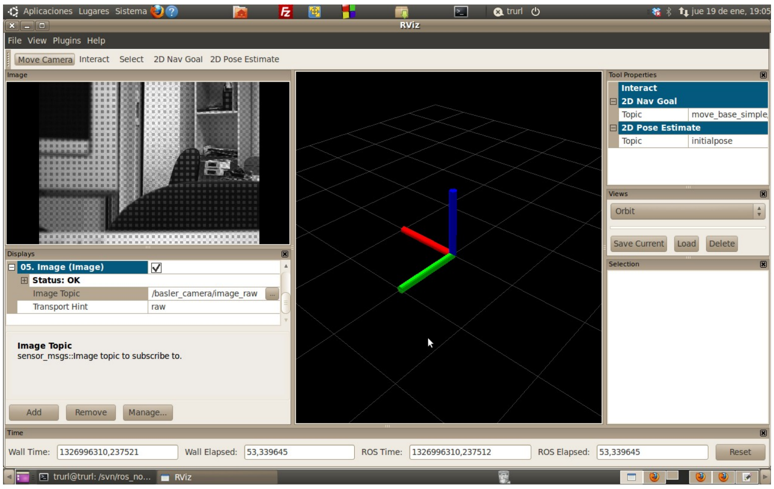

Figure 12.visualization of basler camera video stream in rviz

2.2. Running the complete robot

By default, all the ROS nodes are launched automatically when the robot is initialized.

If you want to restart the nodes manually, you can stop the processes by following these instructions:

1) Use the ps command in order to find all the processes related to ROS.

```
$> ps –ef | grep guardian
```

2) If there are processes in the list, you can stop them using this type of command sequence:

```
$> kill –s SIGTERM <pid-of-first-process>
$> kill –s SIGTERM <pid-of-Nth-process>
```

3) Start a new terminal (CTRL+ALT+T) and run

```
$>roscore
```

4) Once roscore is running you can run each component individually in different terminals or you can launch the configuration file available in guardian_complete directory which will launch all the nodes at the same time.

```
$>roslaunch guardian_complete guardian_complete.launch
```

5) If you want to control the robot by the joystick/game-pad, you will have to run the guardian_pad node:

```
$>roslaunch guardian_pad guardian_pad.launch
```

2.3. Simulating the Guardian robot in ROS

The guardian simulation stack is in the guardian_sim folder. Before starting the simulation, check that the whole stack is correctly compiled. Follow the steps:

```
>roscd guardian_sim
>cd guardian_description
>rm CMakeCache.txt
>cmake .
>rosmake
```

Repeat the same procedure for all the included packages of the stack, and finally repeat rosmake in the stack directory:

```
>roscd guardian_sim
>rosmake
```

To start the simulation:

Teleoperation

Launching the simulated environment and the robot:

```
$> roslaunch guardian_description guardian_robotnik.launch
```

Launching the keyboard teleoperation node:

```
$> roslaunch guardian_controller teleop_keyboard.launch
```

Launching the joystick teleoperation node:

```
$> roslaunch guardian_joystick_teleop guardian_joy.launch
```

It is possible to use *the guardian_node* that is used to control the real robot, changing the launch file configuration. The name of some topics has to be changed.

```
$> roslaunch guardian_node guardian_node.launch
```

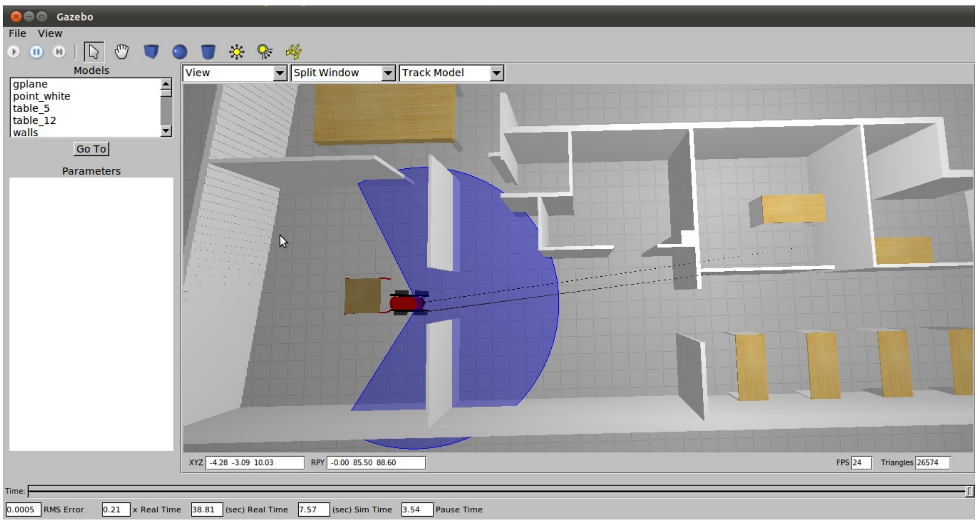

Figure 13. Guardian robot in Gazebo

This will start the guardian robot in the *robotnik.world* environment.

Autonomous navigation

Launching the simulated environment and the robot:

```
$> roslaunch guardian_description guardian_robotnik.launch
```

Launching the AMCL navigation:

```
$> roslaunch guardian_odometry guardian_sim_odometry.launch
$> roslaunch guardian_2dnav move_base_amcl.launch
```

Launching the SLAM navigation:

```
$> roslaunch guardian_odometry guardian_sim_odometry.launch
$> roslaunch guardian_2dnav move_base_slam.launch

$> rosrun rviz rviz
```

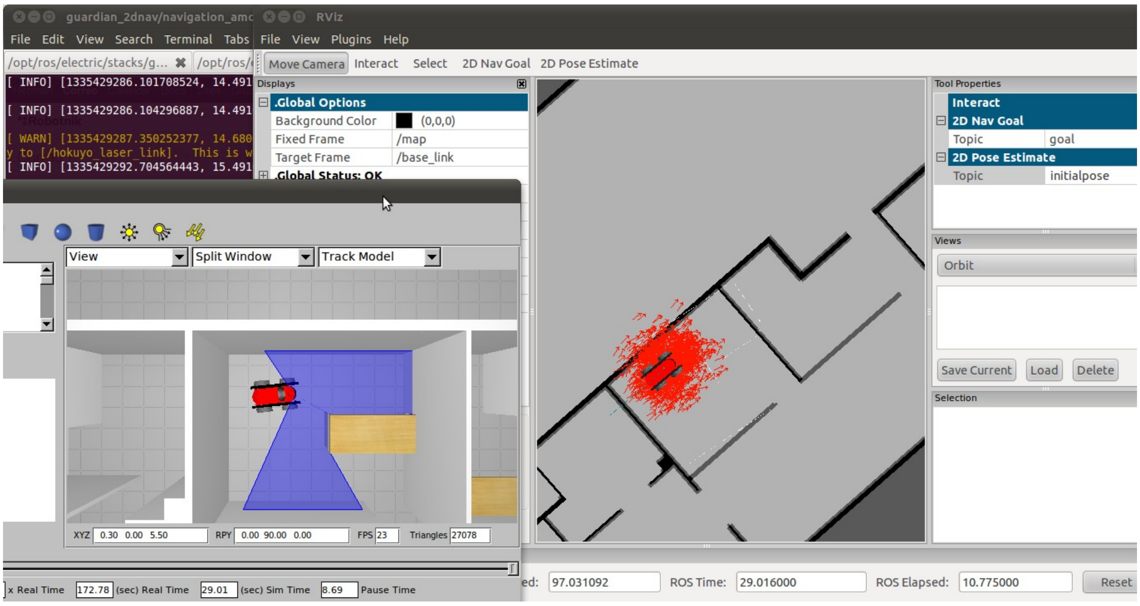

Figure 14 Guardian navigation and RVIZ

Refer to www.ros.org for further information about ROS navigation.