2015 ROS Yaz Okulu
Uygulamalar - II

Revizyon : 1.00
Tarih : 13.08.2015

DOKÜMAN REVİZYON SAYFASI

REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	-	Tümü	İlk versiyon

İÇİNDEKİLER

1. Uygulama-1: Gezgin Robot Gazebo Modeli Oluşturma

Gazebo'nun çalışırlığını kontrol edelim.

```
$ gazebo
```

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg myrobot roscpp rospy urdf
$ cd ~/catkin_ws/src/myrobot
$ mkdir launch
$ mkdir urdf
$ mkdir worlds
```

Boş dünyayı oluşturan launch dosyasını oluşturalım

```
$ cd ~/catkin_ws/src/myrobot/launch
$ gedit empty_world.launch
```

```
<launch>
  <!-- these are the arguments you can pass this launch file, for example paused:=true -->
  <arg name="paused" default="false"/>
  <arg name="use_sim_time" default="true"/>
  <arg name="gui" default="true"/>
  <arg name="headless" default="false"/>
  <arg name="debug" default="false"/>
  <arg name="world_name" default="worlds/empty.world"/> <!-- Note: the world_name is with
respect to GAZEBO_RESOURCE_PATH environmental variable -->


  <!-- set use_sim_time flag -->
  <group if="$(arg use_sim_time)">
    <param name="/use_sim_time" value="true" />
  </group>


  <!-- set command arguments -->
  <arg unless="$(arg paused)" name="command_arg1" value=""/>
  <arg     if="$(arg paused)" name="command_arg1" value="-u"/>
```

```
 <arg unless="$(arg headless)" name="command_arg2" value=""/>
 <arg     if="$(arg headless)" name="command_arg2" value="-r"/>
 <arg unless="$(arg debug)" name="script_type" value="gzserver"/>
 <arg     if="$(arg debug)" name="script_type" value="debug"/>

 <!-- start gazebo server-->
 <node name="gazebo" pkg="gazebo_ros" type="$(arg script_type)" respawn="false"
output="screen"
            args="$(arg command_arg1) $(arg command_arg2) $(arg world_name)" />

 <!-- start gazebo client -->
 <group if="$(arg gui)">
  <node name="gazebo_gui" pkg="gazebo_ros" type="gzclient" respawn="false" output="screen"/>
 </group>

</launch>
```

Robotu boş harita üzerine türeten launch dosyasını oluşturalım.

```
$ gedit ~/catkin_ws/src/myrobot/launch/myrobot.launch
```

```
<launch>

 <include file="$(find myrobot)/launch/empty_world.launch">
   <arg name="paused" value="false"/>
   <arg name="use_sim_time" value="true"/>
   <arg name="gui" value="true"/>
   <arg name="headless" value="false"/>
   <arg name="debug" value="false"/>
 </include>
 <arg name="name" default="myrobot"/>
 <arg name="x" default="0.0"/>
 <arg name="y" default="0.0"/>
 <arg name="z" default="0.0"/>
 <arg name="roll" default="0"/>
 <arg name="pitch" default="0"/>
```

```
 <arg name="yaw" default="0"/>



 <!-- Convert an xacro and put on parameter server -->
 <param name="robot_description" command="$(find xacro)/xacro.py $(find
myrobot)/urdf/myrobot.urdf.xacro" />


 <!-- Spawn evarobot into Gazebo -->
 <node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model"
   args="-param robot_description
        -urdf
        -x $(arg x)
        -y $(arg y)
        -z $(arg z)
        -R $(arg roll)
        -P $(arg pitch)
        -Y $(arg yaw)
        -model $(arg name)"
 respawn="false" output="screen"/>

 <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
  respawn="false" output="screen">
  <remap from="/joint_states" to="/joint_states" />
 </node>

 <node name="joint_state_publisher" pkg="joint_state_publisher"
              type="joint_state_publisher" />


</launch>
```

Robotun ana gövdesini oluşturalım.

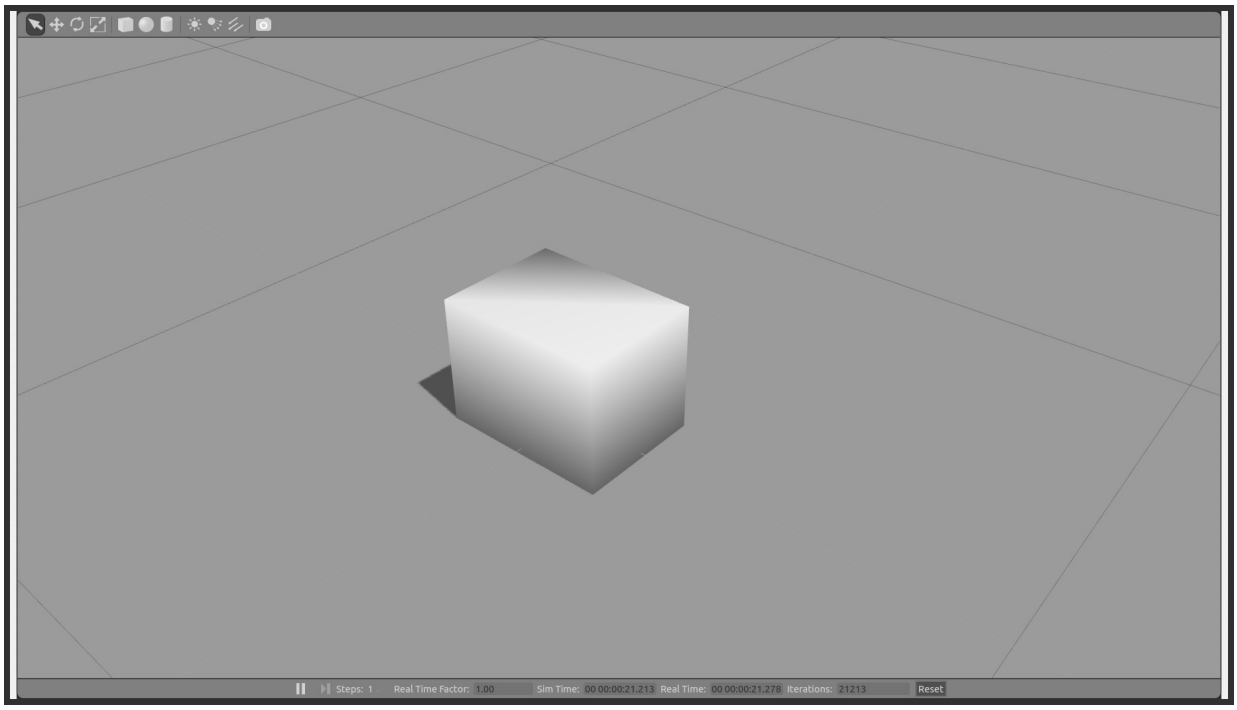
```
$ cd ~/catkin_ws/src/myrobot/urdf
$ gedit myrobot.urdf.xacro
```

```
<?xml version="1.0"?>
<robot name="myrobot" xmlns:xacro="http://ros.org/wiki/xacro">
	<link name="base_link">
		<collision>
			<origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
			<geometry>
				<box size="0.40 0.30 0.30"/>
			</geometry>
		</collision>
		<visual>
			<origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
			<geometry>
				<box size="0.40 0.30 0.30"/>
			</geometry>
			<material name="blue">
				<color rgba="0 0 .8 1"/>
			</material>
		</visual>
		<inertial>
			<origin xyz="0 0 0" rpy="0 0 0"/>
			<mass value="5"/>
			<inertia
			ixx="5.0" ixy="0.0" ixz="0.0"
			iyy="5.0" iyz="0.0"
			izz="5.0"/>
		</inertial>
	</link>
</robot>
```

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

Oluşturulan modelin çalışırlığını kontrol edelim.

```
$ roslaunch myrobot myrobot.launch
```



Modeli tekrar açıp sürüş tekelerini ekleyelim.

```
$ gedit ~/catkin_ws/src/myrobot/urdf/myrobot.urdf.xacro
```

```
<?xml version="1.0"?>
<robot name="myrobot" xmlns:xacro="http://ros.org/wiki/xacro">



	<!-- base_link -->
	<link name="base_link">
		<collision>
			<origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
			<geometry>
				<box size="0.40 0.30 0.30"/>
			</geometry>
		</collision>
		<visual>
			<origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
			<geometry>
				<box size="0.40 0.30 0.30"/>
			</geometry>
			<material name="blue">
```

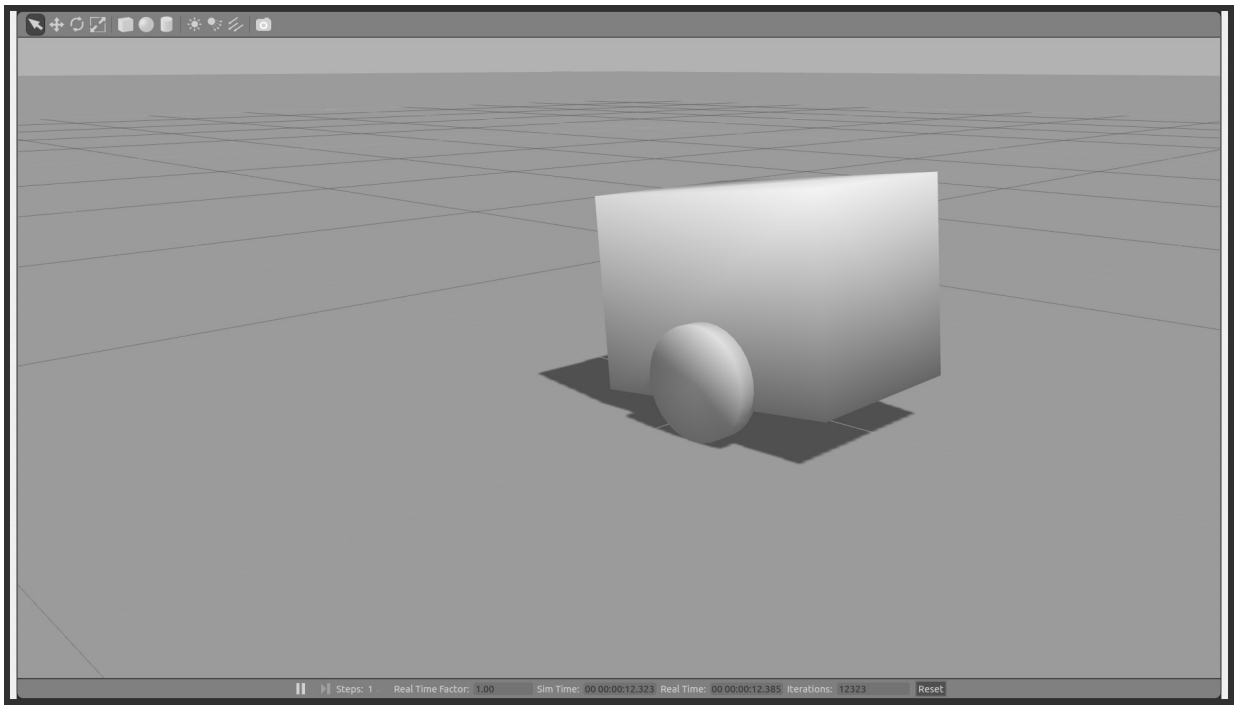
```
                        <color rgba="0 0 .8 1"/>
                    </material>
                </visual>
                <inertial>
                    <origin xyz="0 0 0" rpy="0 0 0"/>
                    <mass value="5"/>
                    <inertia
                    ixx="5.0" ixy="0.0" ixz="0.0"
                    iyy="5.0" iyz="0.0"
                    izz="5.0"/>
                </inertial>
        </link>

        <!-- sol teker link -->
        <link name="left_wheel_link">
                <collision>
                    <origin rpy="1.57075 0 0" xyz="0 0 0"/>
                    <geometry>
                        <cylinder length="0.035" radius="0.085"/>
                    </geometry>
                </collision>
                <visual>
                    <origin rpy="1.57075 0 0" xyz="0 0 0.0"/>
                    <geometry>
                        <cylinder length="0.035" radius="0.085"/>
                    </geometry>
                    <material name="black"/>
                </visual>
                <inertial>
                    <origin xyz="0 0 0" rpy="0 0 0"/>
                    <mass value="0.75"/>
                    <inertia ixx="1e-2" ixy="0" ixz="0" iyy="1e-2" iyz="0" izz="1e-2" />
                </inertial>
        </link>

        <!-- sol teker joint -->
        <joint name="left_wheel_joint" type="continuous">
```

```
		<axis xyz="0 1 0"/>
		<parent link="base_link"/>
		<child link="left_wheel_link"/>
		<origin rpy="0.0 0.0 0.0" xyz="0.0 0.16 -0.115"/>
		<limit effort="100" velocity="100"/>
		<joint_properties damping="0.0" friction="0.0"/>
	</joint>

	<!-- sag teker link -->
	<link name="right_wheel_link">
		<collision>
			<origin rpy="1.57075 0 0" xyz="0 0 0"/>
			<geometry>
				<cylinder length="0.035" radius="0.085"/>
			</geometry>
		</collision>
		<visual>
			<origin rpy="1.57075 0 0" xyz="0 0 0.0"/>
			<geometry>
				<cylinder length="0.035" radius="0.085"/>
			</geometry>
			<material name="black"/>
		</visual>
		<inertial>
			<origin xyz="0 0 0" rpy="0 0 0"/>
			<mass value="0.75"/>
			<inertia ixx="1e-2" ixy="0" ixz="0" iyy="1e-2" iyz="0" izz="1e-2" />
		</inertial>
	</link>

	<!-- sag teker joint -->
	<joint name="right_wheel_joint" type="continuous">
		<axis xyz="0 1 0"/>
		<parent link="base_link"/>
		<child link="right_wheel_link"/>
		<origin rpy="0.0 0.0 0.0" xyz="0.0 -0.16 -0.115"/>
		<limit effort="100" velocity="100"/>
```

```
                <joint_properties damping="0.0" friction="0.0"/>
        </joint>
</robot>
```

Yeni modelin çalışırlığını kontrol edelim.

```
$ roslaunch myrobot myrobot.launch
```



Caster tekerleri modele ekleyelim.

```
$ gedit ~/catkin_ws/src/myrobot/urdf/myrobot.urdf.xacro
```

```
<?xml version="1.0"?>
<robot name="myrobot" xmlns:xacro="http://ros.org/wiki/xacro">


        <!-- base_link -->
        <link name="base_link">
                <collision>
                        <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
```

```
                <geometry>
                        <box size="0.40 0.30 0.30"/>
                </geometry>
        </collision>
        <visual>
                <origin xyz="0.0 0.0 0.0" rpy="0.0 0.0 0.0" />
                <geometry>
                        <box size="0.40 0.30 0.30"/>
                </geometry>
                <material name="blue">
                        <color rgba="0 0 .8 1"/>
                </material>
        </visual>
        <inertial>
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <mass value="5"/>
                <inertia
                ixx="5.0" ixy="0.0" ixz="0.0"
                iyy="5.0" iyz="0.0"
                izz="5.0"/>
        </inertial>
</link>

<!-- sol teker link -->
<link name="left_wheel_link">
        <collision>
                <origin rpy="1.57075 0 0" xyz="0 0 0"/>
                <geometry>
                        <cylinder length="0.035" radius="0.085"/>
                </geometry>
        </collision>
        <visual>
                <origin rpy="1.57075 0 0" xyz="0 0 0.0"/>
                <geometry>
                        <cylinder length="0.035" radius="0.085"/>
                </geometry>
                <material name="black"/>
```

```
		</visual>
		<inertial>
			<origin xyz="0 0 0" rpy="0 0 0"/>
			<mass value="0.75"/>
			<inertia ixx="1e-2" ixy="0" ixz="0" iyy="1e-2" iyz="0" izz="1e-2" />
		</inertial>
</link>

<!-- sol teker joint -->
<joint name="left_wheel_joint" type="continuous">
		<axis xyz="0 1 0"/>
		<parent link="base_link"/>
		<child link="left_wheel_link"/>
		<origin rpy="0.0 0.0 0.0" xyz="0.0 0.16 -0.115"/>
		<limit effort="100" velocity="100"/>
		<joint_properties damping="0.0" friction="0.0"/>
</joint>

<!-- sag teker link -->
<link name="right_wheel_link">
		<collision>
			<origin rpy="1.57075 0 0" xyz="0 0 0"/>
			<geometry>
				<cylinder length="0.035" radius="0.085"/>
			</geometry>
		</collision>
		<visual>
			<origin rpy="1.57075 0 0" xyz="0 0 0.0"/>
			<geometry>
				<cylinder length="0.035" radius="0.085"/>
			</geometry>
			<material name="black"/>
		</visual>
		<inertial>
			<origin xyz="0 0 0" rpy="0 0 0"/>
			<mass value="0.75"/>
			<inertia ixx="1e-2" ixy="0" ixz="0" iyy="1e-2" iyz="0" izz="1e-2" />
```

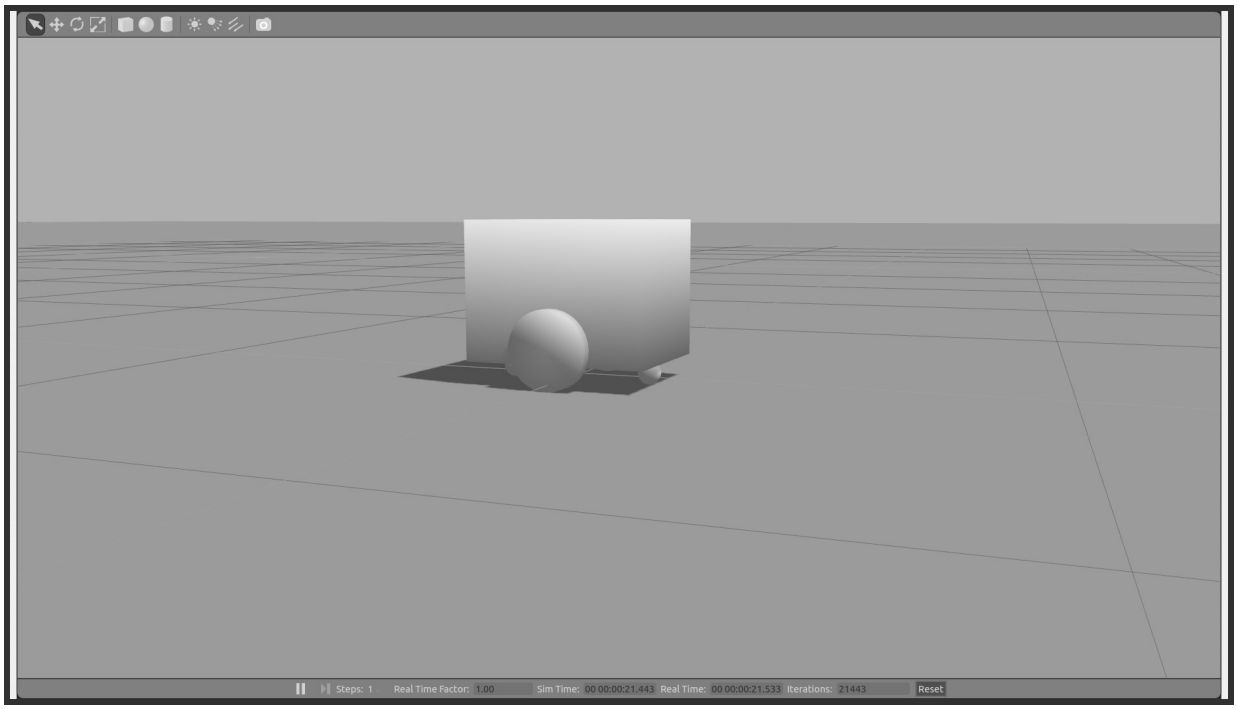
```
		</inertial>
	</link>

	<!-- sag teker joint -->
	<joint name="right_wheel_joint" type="continuous">
		<axis xyz="0 1 0"/>
		<parent link="base_link"/>
		<child link="right_wheel_link"/>
		<origin rpy="0.0 0.0 0.0" xyz="0.0 -0.16 -0.115"/>
		<limit effort="100" velocity="100"/>
		<joint_properties damping="0.0" friction="0.0"/>
	</joint>

	<!-- On caster link-->
	<link name="front_caster_link">
			<collision>
				<origin rpy="0 0 0" xyz="0 0 0"/>
				<geometry>
					<sphere radius="0.025"/>
				</geometry>
			</collision>

			<visual>
				<geometry>
					<sphere radius="0.025"/>
				</geometry>
				<material name="black">
					<color rgba="0 0 0 1"/>
				</material>
			</visual>

			<inertial>
				<origin xyz="0 0 0" rpy="0 0 0"/>
				<mass value="0.5"/>
				<inertia ixx="1e-1" ixy="0" ixz="0" iyy="1e-1" iyz="0" izz="1e-
1" />
			</inertial>
```

```
        </link>

        <gazebo reference="front_caster_link">
                <mu1>0.2</mu1>
                <mu2>0.2</mu2>
        </gazebo>

<!-- On caster joint-->
        <joint name="front_caster_joint" type="continuous">
                <parent link="base_link"/>
                <child link="front_caster_link"/>
                <origin rpy="-1.57075 0.0 0.0" xyz="0.1604 0.0 -0.175"/>
                <axis xyz="0.577350269 0.577350269 0.577350269"/>
                <limit effort="10" velocity="10"/>
                <joint_properties damping="0.0" friction="0.3"/>
        </joint>

<!-- Arka caster link -->
        <link name="rear_caster_link">
                <collision>
                        <origin rpy="0 0 0" xyz="0 0 0"/>
                        <geometry>
                                <sphere radius="0.025"/>
                        </geometry>
                </collision>

                <visual>
                        <geometry>
                                <sphere radius="0.025"/>
                        </geometry>
                        <material name="black">
                                <color rgba="0 0 0 1"/>
                        </material>
                </visual>

                <inertial>
                        <origin xyz="0 0 0" rpy="0 0 0"/>
```

```
                                        <mass value="0.5"/>
                                        <inertia ixx="1e-1" ixy="0" ixz="0" iyy="1e-1" iyz="0" izz="1e-
1" />
                                </inertial>
                        </link>

                        <gazebo reference="rear_caster_link">
                                <mu1>0.2</mu1>
                                <mu2>0.2</mu2>
                        </gazebo>

                <!-- Arka caster joint -->
                        <joint name="rear_caster_joint" type="continuous">
                                <parent link="base_link"/>
                                <child link="rear_caster_link"/>
                                <origin rpy="-1.57075 0.0 0.0" xyz="-0.1604 0.0 -0.175"/>
                                <axis xyz="0.577350269 0.577350269 0.577350269"/>
                                <limit effort="10" velocity="10"/>
                                <joint_properties damping="0.0" friction="0.3"/>
                        </joint>
</robot>
```

Caster'lı modelin çalışırlığını kontrol edelim.

```
$ roslaunch myrobot myrobot.launch
```


Steps: 1
Real Time Factor: 1.00
Sim Time: 00 00:00:21.443
Real Time: 00 00:00:21.533
Iterations: 21443
Reset

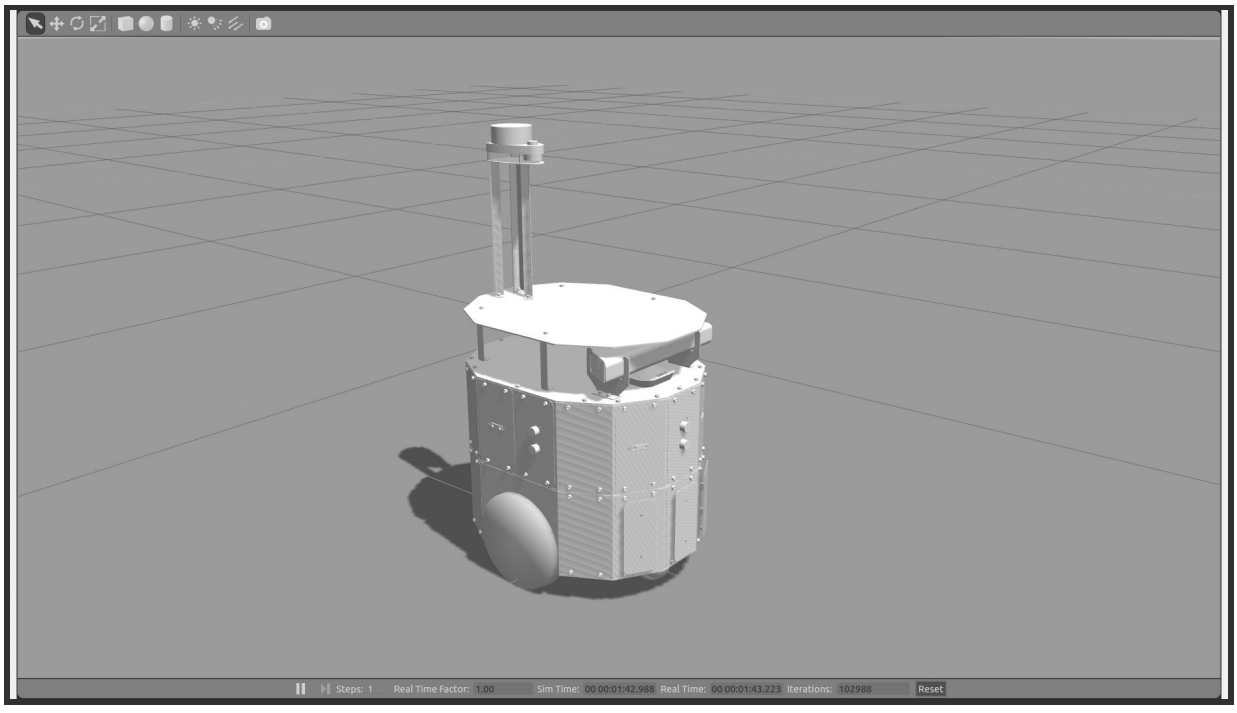
2. Uygulama-2: Modele Görsellik Giydirme (Mesh)

```
$ mkdir ~/catkin_ws/src/myrobot/meshes
$ cp eva50.stl ~/catkin_ws/src/myrobot/meshes
```

```
<visual>
        <origin xyz="0.065 -0.11 -0.15" rpy="0.0 0.0 0.0" />
        <geometry>
                <mesh filename="package://myrobot/meshes/eva50.stl" scale="0.01 0.01 0.01"/>
        </geometry>
        <material name="blue">
                <color rgba="0 0 .8 1"/>
        </material>
</visual>
```

Yeni Modeli test edelim. Model açılırken terminalde hata oluşacaktır.

```
$ roslaunch myrobot myrobot.launch
```



3. Uygulama-3: Modele Sensör Ekleme

Daha önceki çalışmalarda oluşturulan robot modeli açılır.

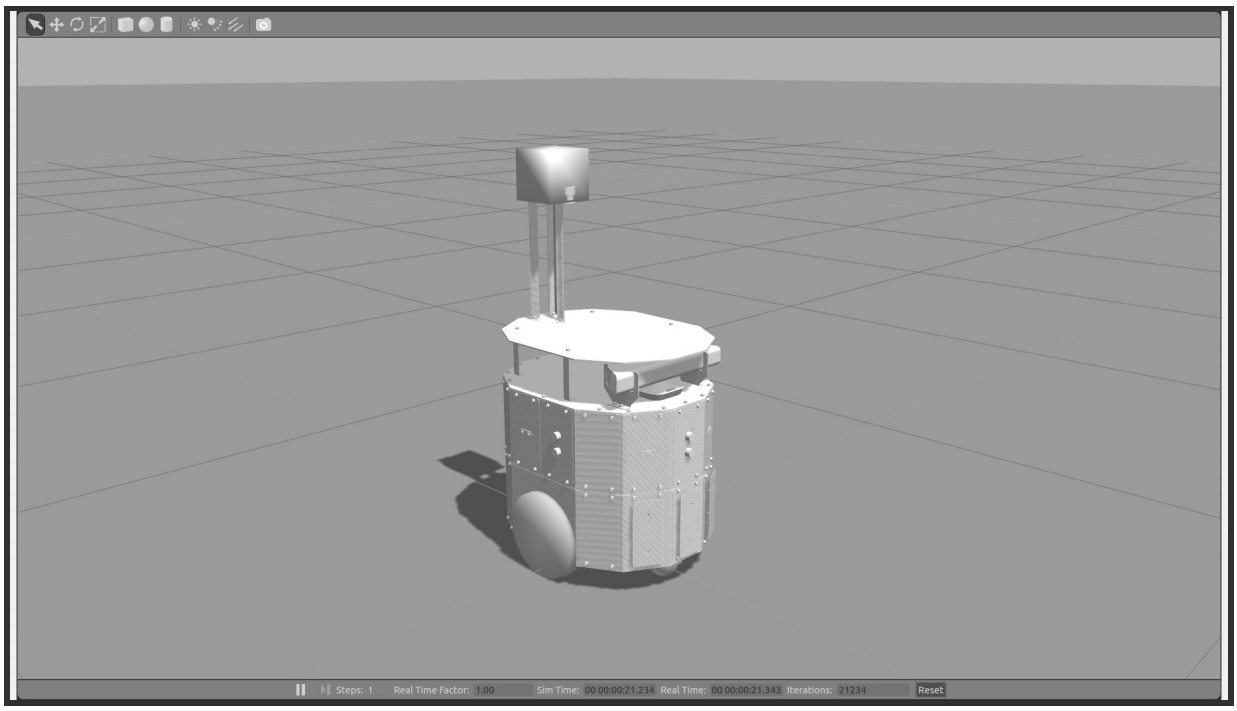
```
$ gedit ~/catkin_ws/src/myrobot/urdf/myrobot.urdf.xacro
```

Modelin sonuna </robot> tag'nin hemen başına aşağıdaki kodlar eklenir.

```
<link name="lidar_link">
        <collision>
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <geometry>
                        <box size="0.1 0.1 0.1"/>
                </geometry>
        </collision>
        <visual>
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <geometry>
                        <box size="0.1 0.1 0.1"/>
                </geometry>
        </visual>
        <inertial>
                <mass value="1e-5" />
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <inertia ixx="1e-6" ixy="0" ixz="0" iyy="1e-6" iyz="0" izz="1e-6" />
        </inertial>
</link>

<joint name="lidar_joint" type="fixed">
        <axis xyz="0 1 0" />
        <origin xyz="-0.15 0.0 0.5415" rpy="0.0 0.0 -1.5708"/>
        <parent link="base_link"/>
        <child link="lidar_link"/>
</joint>
```

Yeni Modeli test edelim. Model açılırken terminalde hata oluşacaktır.

```
$ roslaunch myrobot myrobot.launch
```



4. Uygulama-4: Modele hazır pluginlerin dahil edilmesi

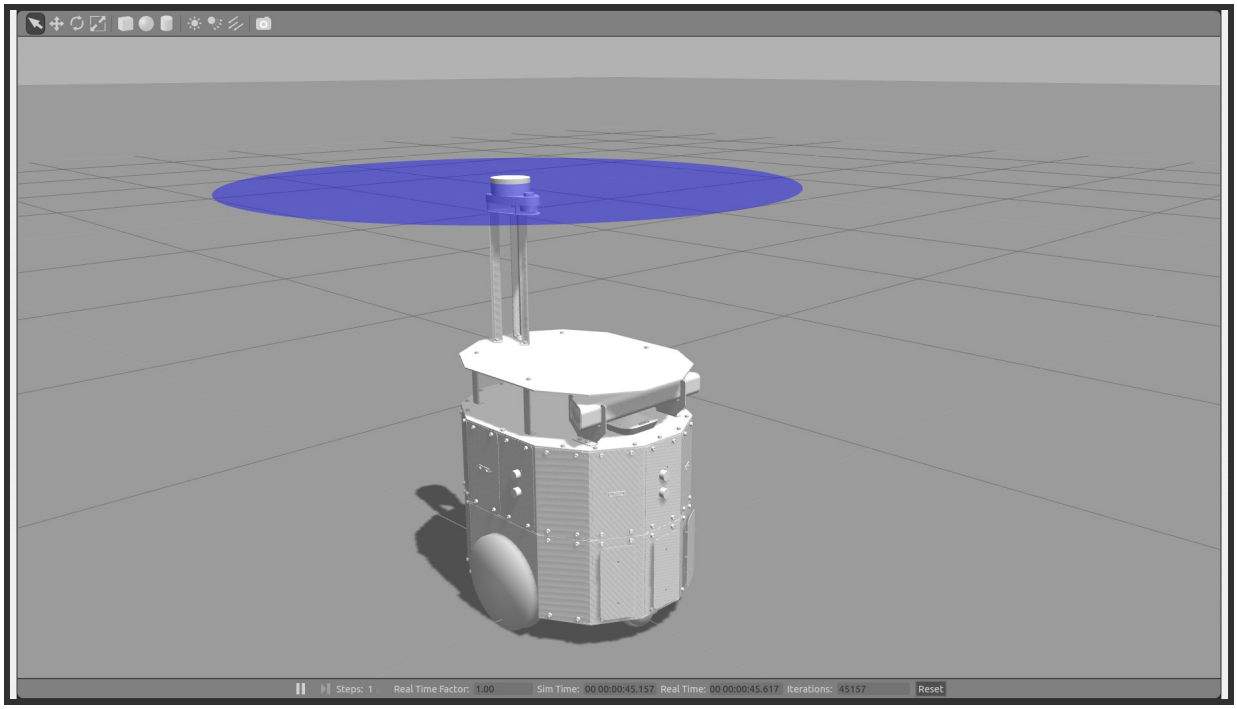
Bir önceki çalışmada oluşturulan robot modeli açılır.

```
$ gedit ~/catkin_ws/src/myrobot/urdf/myrobot.urdf.xacro
```

Modelin sonuna </robot> tag'nin hemen başına aşağıdaki kodlar eklenir.

```
<gazebo reference="lidar_link">
        <sensor type="gpu_ray" name="head_hokuyo_sensor">
        <pose>0 0 0 0 0 0</pose>
        <visualize>true</visualize>
        <update_rate>40</update_rate>
        <ray>
                <scan>
                        <horizontal>
                                <samples>720</samples>
                                <resolution>1</resolution>
                                <min_angle>-3.1416</min_angle>
                                <max_angle>3.1416</max_angle>
                        </horizontal>
                </scan>
                <range>
                        <min>0.5</min>
                        <max>5.0</max>
                        <resolution>0.1</resolution>
                </range>
                <noise>
                <type>gaussian</type>
<!--        Noise parameters based on published spec for Hokuyo laser
        achieving "+-30mm" accuracy at range < 10m.  A mean of 0.0m and
        stddev of 0.01m will put 99.7% of samples within 0.03m of the true
        reading.  -->
                        <mean>0.0</mean>
                        <stddev>0.01</stddev>
                </noise>
        </ray>
```

```
                <plugin name="lidar_controller" filename="libgazebo_ros_gpu_laser.so">
                        <robotNamespace>sensor</robotNamespace>
                        <topicName>lidar</topicName>
                        <frameName>lidar_link</frameName>
                </plugin>
        </sensor>
</gazebo>
```



5. Uygulama-5: Sensörlerden veri okuma

5.1 Simülasyon (Gazebo)

Evarobot modeli kurulumu yapılır.

```
$ cd ~/catkin_ws/src/
# rplidar paketi indirilir.
$ git clone https://github.com/robopeak/rplidar_ros.git -b slam
# teleop paketi indirilir.
$ sudo apt-get install ros-indigo-teleop-twist-keyboard
# gazebo ros paketleri indirilir.
$ git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b indigo-devel
$ git clone https://github.com/ros-controls/ros_control.git -b indigo-devel
$ git clone https://github.com/ros-controls/control_toolbox.git -b indigo-devel
$ git clone https://github.com/ros-controls/realtime_tools.git -b indigo-devel
# hector gazebo modeli indirilir.
$ git clone https://github.com/tu-darmstadt-ros-pkg/hector_gazebo.git -b indigo-devel
$ cd ~/catkin_ws
# indirilen paketler derlenir.
$ catkin_make
$ cd ~/catkin_ws/src
# evarobot modeli indirilir.
$ git clone https://github.com/inomuh/evapc_ros.git -b eva50
$ cd ~/catkin_ws
$ catkin_make
```

Evarobot gazebo modelinde sensörleri okuma

```
$ roslaunch evarobot_description evarobot.launch
```

rostopic echo komutları kullanarak ve daha önceki uygulamada yazılan subscriber değiştirilek sonar sensor okunacaktır.

Rviz'de görselleştirme

```
$ rosrun rviz rviz
```

```
File → Open Config → <kullanıcı adı>/catkin_ws/src/evapc_ros/evarobot_description/media/gazebo_demo.rviz
```

5.2 Gerçek Ortam (evarobot)

5.2.1 Kurulum

Bilgisayar; bağlantı ayarlarının yapılması

```
$ sudo gedit /etc/hosts
```

```
192.168.3.10    evarobotDSK
192.168.3.90    <bilgisayar adı>
```

```
$  gedit ~/.bashrc
```

```
export ROS_HOSTNAME=<bilgisayar adı>
export ROS_MASTER_URI=http://<bigisayar adı>:11311
export LC_NUMERIC=C
```

bashrc'de değişik yapıldıktan sonra terminal kapatıp açılmalı ya da terminalde $ bash komutu çalıştırılmalıdır.

Çoklu ros master ile çalışmak için multimaster_fkie isimli paket indirilir ve derlenir.

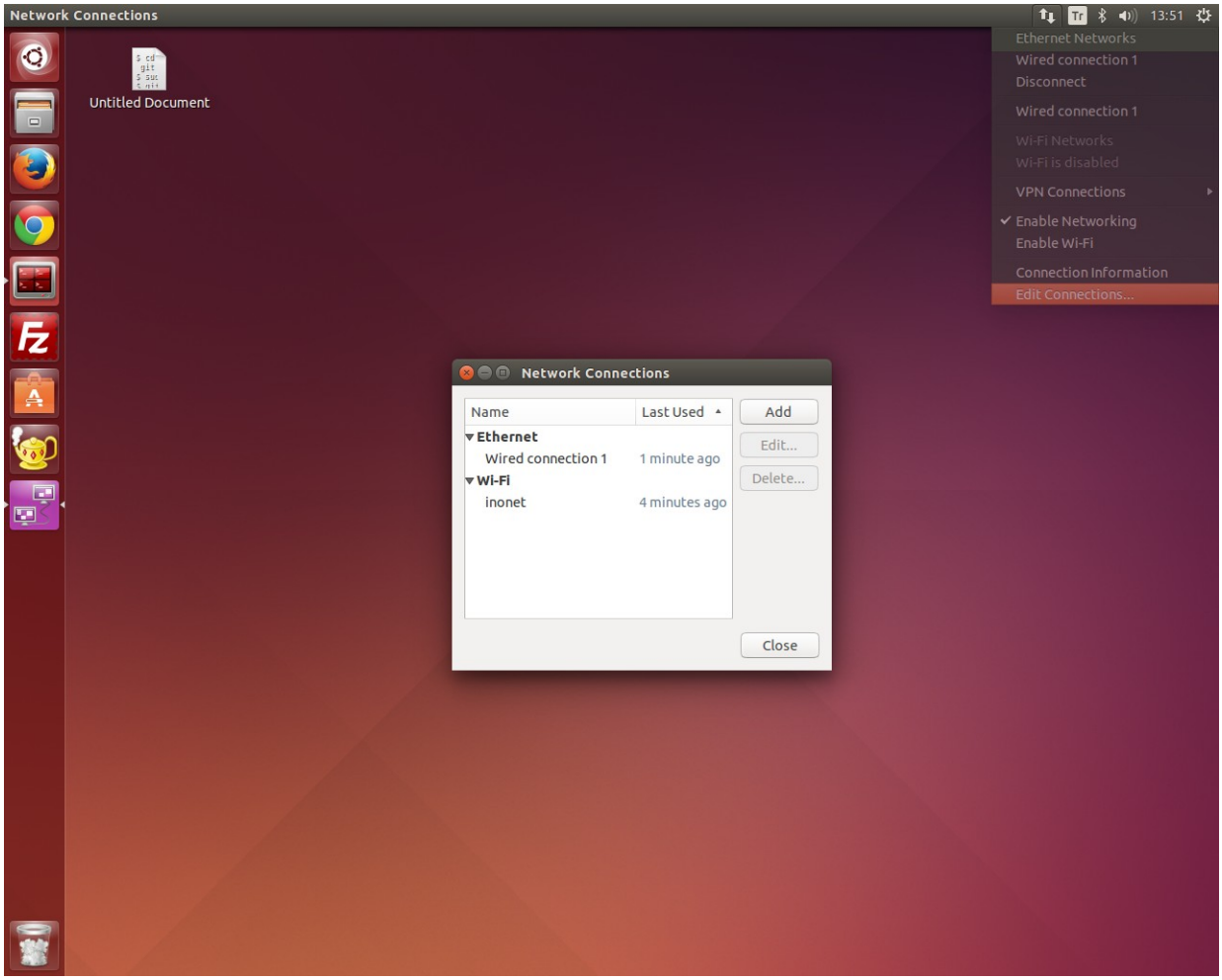
```
$ cd ~/catkin_ws/src/
$ git clone https://github.com/fkie/multimaster_fkie.git -b indigo-devel
$ cd ~/catkin_ws
$ catkin_make
```

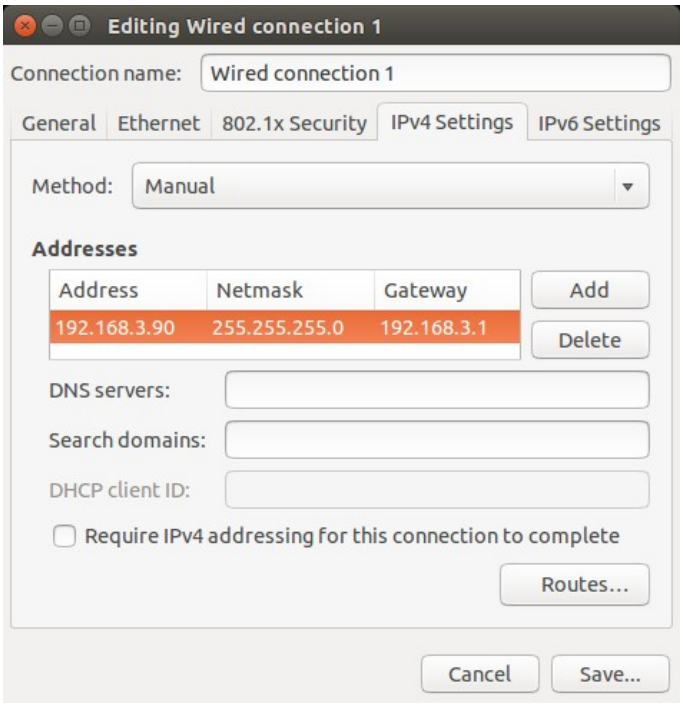
Ağdaki ros master'ları bulmak için launch dosyasında ufak bir değişiklik yapılır.

```
$ gedit catkin_ws/src/multimaster_fkie/master_discovery_fkie/launch/master_discovery.launch
```

```
<param name="mcast_group" value="224.0.0.0" />
```

Ethernet ayarlarının yapılması





Evarobot; bağlantı ayarları yapılmalıdır. (şifre: 12345)

```
$ ssh pi@192.168.3.10
```

```
$ sudo nano /etc/hosts
```

```
192.168.3.90    <bilgisayar adı>
```

5.2.2 Çalıştırma

Evarobot

evarobot için en az 6 terminal açılır. Her bir terminale super kullanıcı izni verilmelidir.

```
$ sudo -s
```

Terminal 1

```
$ roslaunch evarobot_driver driver.launch
```

Terminal 2

```
$ roslaunch evarobot_controller evarobot_controller.launch
```

Terminal 3

```
$ roslaunch evarobot_odometry evarobot_odometry.launch
```

Terminal 4

```
$ roslaunch evarobot_sonar evarobot_sonar.launch
```

Terminal 5

```
$ roslaunch master_discovery_fkie master_discovery.launch
```

Terminal 6

```
$ roslaunch master_sync_fkie master_sync.launch
```

Bilgisayar

```
$ roslaunch master_discovery_fkie master_discovery.launch
$ roslaunch master_sync_fkie master_sync.launch
$ rostopic list
```

rostopic echo komutları kullanarak ve daha önceki uygulamada yazılan subscriber değiştirilek sonar sensor okunacaktır.