Note: This tutorial assumes you know how to write well-formatted XML code.

🔆 Please ask about problems and questions regarding this tutorial on • answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Building a Visual Robot Model with URDF from Scratch

Description: Learn how to build a visual model of a robot that you can view in Rviz

Keywords: URDF

Tutorial Level: BEGINNER

Next Tutorial: Making the Model Move (/urdf/Tutorials/Building%20a%20Movable%20Robot%20Model%20with%20URDF)

Conten	ts
1. One	Shape
2. Multi	iple Shapes
3. Origi	ins
4. Mate	erial Girl
5. Finis	hing the Mode

In this tutorial, we're going to build a visual model of a robot that vaguely looks like R2D2. In later tutorials, you'll learn how to articulate the model (/urdf/Tutorials /Building%20a%20Movable%20Robot%20Model%20With%20URDF), add in some physical properties (/urdf/Tutorials

/Adding%20Physical%20and%20Collision%20Properties%20to%20a%20URDF%20Model), generate neater code with xacro (/urdf/Tutorials

/Using%20Xacro%20to%20Clean%20URD%20a%20URDF%20File) and make it move in Gazebo (/urdf/Tutorials/Using%20a%20URDF%20in%20Gazebo). But for now, we're going to focus on getting the visual geometry correct.

Before continuing, make sure you have the joint_state_publisher (/joint_state_publisher) package installed. If you installed urdf_tutorial (/urdf_tutorial) using apt-get, this should already be the case. If not, please update your installation to include that package (use rosdep to check).

All of the robot models mentioned in this tutorial (and the source files) can be found in the urdf_tutorial (/urdf_tutorial) package.

1. One Shape

First, we're just going to explore one simple shape. Here's about as simple as a urdf as you can make. Source (https://github.com/ros/urdf_tutorial/tree/master/urdf/01-myfirst.urdf)

```
Toggle line numbers
   1 <?xml version="1.0"?>
  2 <robot name="myfirst">
      <link name="base link">
  3
  4
         <visual>
   5
           <geometry>
   6
             <cylinder length="0.6" radius="0.2"/>
           </geometry>
   7
  8
        </visual>
       </link>
   9
 10 </robot>
```

To translate the XML into English, this is a robot with the name myfirst, that contains only one link (a.k.a. part), whose visual component is just a cylinder 0.6 meters long with a 0.2 meter radius. This may seem like a lot of enclosing tags for a simple "hello world" type example, but it will get more complicated, trust me.

To examine the model, launch the display.launch file:

\$ roslaunch urdf_tutorial display.launch model:=urdf/01-myfirst.urdf

This does three things. It

- · Loads the specified model into the parameter server
- Runs nodes to publish esensor_msgs/JointState (http://docs.ros.org/api/sensor_msgs/html/msg/JointState.html) and transforms (more on these later)
- Starts Rviz with a configuration file

Note that the roslaunch line above assumes that you are executing it from the urdf_tutorial (/urdf_tutorial) package directory (ie: the urdf directory is a direct child of the current working directory). If that is not the case, the relative path to 01-myfirst.urdf will not be valid, and you'll receive an error as soon as roslaunch tries to load the urdf to the parameter server.

A slightly modified argument allows this to work regardless of the current working directory:

\$ roslaunch urdf_tutorial display.launch model:='\$(find urdf_tutorial)/urdf/01-myfirst.urdf'

note the single quotes around the argument value.

You'll have to change all example roslaunch lines given in these tutorials if you are not running them from the urdf_tutorial package location.

After launching display.launch, you should end up with RViz showing you the following:

🔕 👁 👁 RViz
File View Plugins Help
Move Camera Select 2D Nav Goal 2D Pose Estimate
Gibbal Ontions
Background Color (23.0.119)
Fixed Frame / Dase link
Target Frame <fixed frame=""></fixed>
🗃 .Global Status: Warning
🗉 01. Grid (Grid) 🗸
🗄 Status: OK
Reference Frame < Fixed Frame >
Plane Cell Count 10
Normal Cell Cou 0
Cell Size 0.5
Line Style Lines
Color (228,228,228)
Alpha 0.5
Plane XY
Collision Enable
Update Interval 0
Alpha 1
Robot Descriptio robot description
TE Drafiv
.Global Status: OK
Add Remove 🧐 😭
Time
Wall Time: 1289424853.418646 Wall Elapsed: 3.456286 ROS Time: 1289424853.418643 ROS Elapsed: 3.456286 Reset

Things to note:

- The fixed frame is transform frame where the center of the grid is located. Here, it's a frame defined by our one link, base_link.
- The visual element (the cylinder) has its origin at the center of its geometry as a default. Hence, half the cylinder is below the grid.

2. Multiple Shapes

Now let's look at how to add multiple shapes/links. If we just add more link elements to the urdf, the parser won't know where to put them. So, we have to add joints. Joint elements can refer to both flexible and inflexible joints. We'll start with inflexible, or fixed joints. Source (https://github.com/ros/urdf_tutorial/tree/master/urdf/02-multipleshapes.urdf)

```
Toggle line numbers
  1 <?xml version="1.0"?>
  2 <robot name="multipleshapes">
     <link name="base_link">
  3
  4
        <visual>
  5
          <geometry>
  6
            <cylinder length="0.6" radius="0.2"/>
  7
         </geometry>
       </visual>
  8
  9
      </link>
 10
     <link name="right_leg">
 11
 12
        <visual>
 13
          <geometry>
 14
            <box size="0.6 0.1 0.2"/>
         </geometry>
 15
       </visual>
 16
 17
     </link>
 18
     <joint name="base_to_right_leg" type="fixed">
 19
       <parent link="base link"/>
 20
 21
        <child link="right_leg"/>
 22
      </joint>
 23
 24 </robot>
```

• Note how we defined a 0.6m x 0.1m x 0.2m box

• The joint is defined in terms of a parent and a child. URDF is ultimately a tree structure with one root link. This means that the leg's position is dependent on the base_link's position.

roslaunch urdf_tutorial display.launch model:=urdf/02-multipleshapes.urdf

🙆 💿 💿 RViz
File View Plugins Help
Marine Commany Solect 2D Nav Goal 2D Nave Setimate
Move camera Jelect 2D Nav Goal 2D Fose Estimate
Displays (g)
Background Color (23,0,119)
rixed rrame //Dase_link
anget name < næe frame >
Beference Frame < Fixed Frame>
Plane Cell Court 10
Normal Cell Coul 0
Cell Size 0.5
Line Style Lines
Color (228,228,228)
Alpha 0.5
Plane XY
□ Offset 0; 0; 0
🖬 02. Robot Model 🖌
🗄 Status: OK
Visual Enabled 📝
Collision Enable
Update Interval 0
Alpha 1
Robot Descriptio robot_description
TE Droffy
.Global Status: OK
Add Remove 🧶 😰
Time Example 1 Constraints and the constraints
Well Time, 1200424780 953104 Well Elevered, 2 105050 POC Time, 1390424780 953101 POC Elevered, 2 105050 Prest
Augu Hule: 17034741001037334 Mail Eidbadi: 131703400 LO2 Hille: 17034741001022131 LO2 Eidbadi: 31703400 Keset

Both of the shapes overlap with each other, because they share the same origin. If we want them not to overlap we must define more origins.

3. Origins

So R2D2's leg attaches to the top half of his torso, on the side. So that's where we specify the origin of the JOINT to be. Also, it doesn't attach to the middle of the leg, it attaches to the upper part, so we must offset the origin for the leg as well. We also rotate the leg so it is upright. Source (https://github.com/ros/urdf_tutorial/tree/master/urdf/03-origins.urdf)

Toggle line numbers

1	xml version="1.0"?
2	<robot name="origins"></robot>
3	<link name="base_link"/>
4	<visual></visual>
5	<geometry></geometry>
6	<cylinder length="0.6" radius="0.2"></cylinder>
7	
8	
9	
10	
11	<link name="right_leg"/>
12	<visual></visual>
13	<geometry></geometry>
14	<box size="0.6 0.1 0.2"></box>
15	
16	<pre><origin rpy="0 1.57075 0" xyz="0 0 -0.3"></origin></pre>
17	
18	
19	
20	<joint name="base_to_right_leg" type="fixed"></joint>
21	<parent link="base_link"></parent>
22	<child link="right_leg"></child>
23	<origin xyz="0 -0.22 0.25"></origin>
24	
25	
26	

- Let's start by examining the joint's origin. It is defined in terms of the parent's reference frame. So we are -0.22 meters in the y direction (to our left, but to the right relative to the axes) and 0.25 meters in the z direction (up). This means that the origin for the child link will be up and to the right, regardless of the child link's visual origin tag. Since we didn't specify a rpy (roll pitch yaw) attribute, the child frame will be default have the same orientation as the parent frame.
- Now, looking at the leg's visual origin, it has both a xyz and rpy offset. This defines where the center of the visual element should be, relative to its origin. Since we want the leg to attach at the top, we offset the origin down by setting the z offset to be -0.3 meters. And since we want the long part of the leg to be parallel to the z axis, we rotate the visual part PI/2 around the Y axis.

roslaunch urdf_tutorial display.launch model:=urdf/03-origins.urdf

🙆 💿 💿 RViz	
File View Plugins Help	
Maue Campan Solect 2D Nav Goal 2D Pare Estimate	
Muve caneral Sect 20 Nav Gail 20 Fise Estimate	
Displays (B)	
Background Color (23,0,119)	
Had Frame /base_link	
larget Frame <nxet frame=""></nxet>	
Reference Frame < Fixed Frame >	
Planeteel Count 10	
Normal Cell Coul 0	
Cell Size 0.5	
Line Style Lines	-
Color (228,228,228)	
Alpha 0.5	
Plane XY	\sim
⊞ Offset 0; 0; 0	
Die O2. Robot Model	
🗄 Status: OK	
Visual Enabled 🗹	\sim
Collision Enable	
Update Interval 0	
Alpha 1	
Robot Descriptio robot_description	
TE Dratty	
.Global Status: OK	
	\sim
Add Remove 😻 😭	
Time	×
Wall Line: L289424738.802296 ROS Elapsed: L199289 Rese	

- The launch file runs packages that will create TF frames for each link in your model based on your URDF. Rviz uses this information to figure out where to display each shape.
- If a TF frame does not exist for a given URDF link, then it will be placed at the origin in white (ref. related question (http://answers.ros.org/question/207947/how-do-youuse-externally-defined-materials-in-a-urdfxacro-file/)).

4. Material Girl

"Alright," I hear you say. "That's very cute, but not everyone owns a B21. My robot and R2D2 are not red!" That's a good point. Let's take a look at the material tag. Source (https://github.com/ros/urdf_tutorial/tree/master/urdf/04-materials.urdf)

Toggle line numbers

1	xml version="1.0"?
2	<robot name="materials"></robot>
3	
4	<material name="blue"></material>
5	<color rgba="0 0 0.8 1"></color>
6	
7	·/ material
, 8	<material name="white"></material>
9	<pre><color #="" 1="" rgba="#1"></color></pre>
10	<pre><colorigua= 1="" <="" material=""></colorigua=></pre>
11	
10	
12	diah menenukana limbus
13	<link name="base_link"/>
14	<visual></visual>
15	<geometry></geometry>
16	<cylinder length="0.6" radius="0.2"></cylinder>
17	
18	<material name="blue"></material>
19	
20	
21	
22	<link name="right_leg"/>
23	<visual></visual>
24	<geometry></geometry>
25	<box size="0.6 0.1 0.2"></box>
26	
27	<pre><origin rpy="0 1.57075 0" xyz="0 0 -0.3"></origin></pre>
28	<material name="white"></material>
29	
30	
31	
32	<joint name="base_to_right_leg" type="fixed"></joint>
33	<parent link="base_link"></parent>
34	<child link="right_leg"></child>
35	<origin xyz="0 -0.22 0.25"></origin>
36	
37	
38	<link name="left_leg"/>
39	<visual></visual>
40	<geometry></geometry>
41	<box size="0.6 0.1 0.2"></box>
42	
43	<pre><origin rpv="0 1.57075 0" xvz="0 0 -0.3"></origin></pre>
44	<pre><material name="white"></material></pre>
45	
46	
47	,
19	<pre><ioint name="base to left leg" type="fixed"></ioint></pre>
7 Q	<pre>sparse name base_co_rere_rey type= rixed**</pre>
4.) 5.0	cobild link="loft log"/>
50	$\sim 100 \text{ mm}^{-1} \text{ mm}^{-1$
52	<pre></pre>
52	-/ JOINT
22	
74	

• The body is now blue. We've defined a new material called "blue", with the red, green, blue and alpha channels defined as 0,0,0.8 and 1 respectively. All of the values can be in the range [0,1]. This material is then referenced by the base_link's visual element. The white material is defined similarly

• You could also define the material tag from within the visual element, and even reference it in other links. No one will even complain if you redefine it though.

 $\bullet\,$ You can also use a texture to specify an image file to be used for coloring the object

roslaunch urdf_tutorial display.launch model:=urdf/04-materials.urdf

File View Plugins Help
Maria Camera Select 2D Nav Goal 2D Pase Estimate
Background Color (23.0.119)
Fixed Frame (hase link
Target Frame < Fixed Frame>
.Global Status: OK
E Status: OK
Reference Frame < Fixed Frame >
Plane Cell Count 10
Normal Cell Cou 0
Cell Size 0.5
Line Style Lines
Color (228,228,228)
Alpha 0.5
Plane XY
☑ Offset 0; 0; 0
C2. Robot Model
E Status: OK
Visual Enabled
Collision Enable
Update interval 0
Alpha 1
Kobo Descriptio robot_description
.Global Status: OK
Add Remove 🧶 🏠
Time
weil IIIte: L209424301.07027 wail tabsed: 0.399383 KOS IIIte: L289424381.8/2023 KUS tabsed: 0.399385 Reset

5. Finishing the Model

Now we finish the model off with a few more shapes: feet, wheels, and head. Most notably, we add a sphere and a some meshes. We'll also add few other pieces that we'll use later. • Source (https://github.com/ros/urdf_tutorial/tree/master/urdf/05-visual.urdf)

Toggle line numbers

1 <?xml version="1.0"?> 2 <robot name="visual"> 3 <material name="blue"> 4 <color rgba="0 0 0.8 1"/> 5 6 </material> 7 <material name="black"> 8 <color rgba="0 0 0 1"/> </material> 9 10 <material name="white"> 11 <color rgba="1 1 1 1"/> 12 </material> 13 14 <link name="base link"> 15 <visual> 16 <geometry> <cylinder length="0.6" radius="0.2"/> 17 </geometry> 18 19 <material name="blue"/> </visual> 20 21 </link> 2.2 23 <link name="right_leg"> 24 <visual> 25 <geometry> 26 <box size="0.6 0.1 0.2"/> 27 </geometry> <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/> 28 29 <material name="white"/> </visual> 30 31 </link> 32 <joint name="base_to_right_leg" type="fixed"> 33 34 <parent link="base link"/> 35 <child link="right_leg"/> 36 <origin xyz="0 -0.22 0.25"/> 37 </joint> 38 39 <link name="right_base"> 40 <visual> 41 <geometry> 42 <box size="0.4 0.1 0.1"/> 43 </geometry> <material name="white"/> 44 </visual> 45 </link> 46 47 <joint name="right_base_joint" type="fixed"> 48 <parent link="right_leg"/> 49 50 <child link="right base"/> 51 <origin xyz="0 0 -0.6"/> 52 </joint> 53 <link name="right_front_wheel"> 54 55 <visual> <origin rpy="1.57075 0 0" xyz="0 0 0"/> 56 57 <geometry> <cvlinder length="0.1" radius="0.035"/> 58 59 </geometry> 60 <material name="black"/> 61 <origin rpy="0 0 0" xyz="0 0 0"/> </visual> 62 63 </link> <joint name="right_front_wheel_joint" type="fixed"> 64 <parent link="right_base"/> 65 <child link="right front wheel"/> 66 67 <origin rpy="0 0 0" xyz="0.133333333333 0 -0.085"/> 68 </joint> 69 <link name="right_back_wheel"> 70 <visual> 71 72 <origin rpy="1.57075 0 0" xyz="0 0 0"/> 73 <geometry> <cylinder length="0.1" radius="0.035"/> 74 </geometry> 75 76 <material name="black"/> </visual> 77

urdf/Tutorials/Building a Visual Robot Model ...

```
78
     </link>
     <joint name="right_back_wheel_joint" type="fixed">
 79
 80
       <parent link="right base"/>
 81
       <child link="right_back_wheel"/>
       <origin rpy="0 0 0" xyz="-0.133333333333 0 -0.085"/>
 82
     </joint>
 83
 84
 85
     <link name="left_leg">
 86
       <visual>
 87
         <geometry>
           <box size="0.6 0.1 0.2"/>
 88
 89
        </geometry>
         <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
 90
 91
         <material name="white"/>
       </visual>
 92
 93
     </link>
 94
     <joint name="base_to_left_leg" type="fixed">
 95
 96
       <parent link="base link"/>
 97
       <child link="left_leg"/>
 98
        <origin xyz="0 0.22 0.25"/>
     </joint>
99
100
101
     <link name="left_base">
       <visual>
102
103
        <geometry>
104
           <box size="0.4 0.1 0.1"/>
105
        </geometry>
         <material name="white"/>
106
107
       </visual>
     </link>
108
109
     <joint name="left_base_joint" type="fixed">
110
111
       <parent link="left_leg"/>
       <child link="left_base"/>
112
113
       <origin xyz="0 0 -0.6"/>
114
     </joint>
115
     <link name="left front wheel">
116
117
       <visual>
118
         <origin rpy="1.57075 0 0" xyz="0 0 0"/>
119
        <geometry>
           <cylinder length="0.1" radius="0.035"/>
120
121
        </geometry>
122
         <material name="black"/>
123
       </visual>
     </link>
124
125
     <joint name="left_front_wheel_joint" type="fixed">
       <parent link="left_base"/>
126
127
       <child link="left_front_wheel"/>
       <origin rpy="0 0 0" xyz="0.133333333333 0 -0.085"/>
128
129
     </joint>
130
131
     <link name="left_back_wheel">
132
       <visual>
133
         <origin rpy="1.57075 0 0" xyz="0 0 0"/>
       <geometry>
134
           <cylinder length="0.1" radius="0.035"/>
135
        </geometry>
136
137
         <material name="black"/>
138
       </visual>
139
     </link>
     <joint name="left_back_wheel_joint" type="fixed">
140
141
       <parent link="left_base"/>
142
       <child link="left_back_wheel"/>
143
       <origin rpy="0 0 0" xyz="-0.133333333333 0 -0.085"/>
     </ioint>
144
145
     <joint name="gripper_extension" type="fixed">
146
147
      <parent link="base_link"/>
       <child link="gripper_pole"/>
148
149
       <origin rpy="0 0 0" xyz="0.19 0 0.2"/>
150 </joint>
151
     <link name="gripper pole">
152
153
       <visual>
154
         <geometry>
155
           <cylinder length="0.2" radius="0.01"/>
```

urdf/Tutorials/Building a Visual Robot Model ...

```
156
         </geometry>
157
         <origin rpy="0 1.57075 0 " xyz="0.1 0 0"/>
158
       </visual>
159
     </link>
160
     <joint name="left_gripper_joint" type="fixed">
161
162
       <origin rpy="0 0 0" xyz="0.2 0.01 0"/>
163
       <parent link="gripper_pole"/>
       <child link="left_gripper"/>
164
165
     </joint>
166
167
     <link name="left_gripper">
168
       <visual>
         <origin rpy="0.0 0 0" xyz="0 0 0"/>
169
170
        <geometry>
171
           <mesh filename="package://urdf_tutorial/meshes/l_finger.dae"/>
172
         </geometry>
173
       </visual>
174
     </link>
175
176
     <joint name="left_tip_joint" type="fixed">
177
       <parent link="left_gripper"/>
       <child link="left_tip"/>
178
179
     </joint>
180
     <link name="left_tip">
181
182
       <visual>
183
         <origin rpy="0.0 0 0" xyz="0.09137 0.00495 0"/>
184
         <geometry>
185
           <mesh filename="package://urdf_tutorial/meshes/l_finger_tip.dae"/>
         </geometry>
186
187
       </visual>
     </link>
188
189
     <joint name="right_gripper_joint" type="fixed">
       <origin rpy="0 0 0" xyz="0.2 -0.01 0"/>
190
191
       <parent link="gripper_pole"/>
       <child link="right_gripper"/>
192
193
     </joint>
194
195
     <link name="right_gripper">
196
       <visual>
197
         <origin rpy="-3.1415 0 0" xyz="0 0 0"/>
198
         <geometry>
199
           <mesh filename="package://urdf_tutorial/meshes/l_finger.dae"/>
200
         </geometry>
201
       </visual>
     </link>
202
203
204
     <joint name="right_tip_joint" type="fixed">
205
       <parent link="right_gripper"/>
       <child link="right tip"/>
206
207
     </joint>
208
209
     <link name="right_tip">
210
       <visual>
211
         <origin rpy="-3.1415 0 0" xyz="0.09137 0.00495 0"/>
212
         <geometry>
213
           <mesh filename="package://urdf_tutorial/meshes/l_finger_tip.dae"/>
214
         </geometry>
       </visual>
215
216
     </link>
217
     ink name="head">
218
219
       <visual>
         <geometry>
220
221
           <sphere radius="0.2"/>
222
         </geometrv>
223
         <material name="white"/>
224
       </visual>
225
     </link>
226
     <joint name="head swivel" type="fixed">
227
       <parent link="base link"/>
228
       <child link="head"/>
229
       <origin xyz="0 0 0.3"/>
    </joint>
230
231
232
     <link name="box">
233
       <visual>
```

urdf/Tutorials/Building a Visual Robot Model ...



roslaunch urdf_tutorial display.launch model:=urdf/05-visual.urdf

I	8		(Viz								
	File	e View P	lugins He	elp							
		Move Cam	nera Sele	ct 2D Nav Goal	2D Pose Estir	nate					
ľ	Dis	plays			×						
		.Global (Options								
		Backgrou	nd Color	(23,0,119)							
		Fixed Fran	me	/base_link							
		Target Fra	me	<fixed frame=""></fixed>							
	Ð	.Global S	Status: O	к							
		01. Grid	(Grid)	\checkmark							
		🗄 Statu	s: OK					\sim			
		Refere	nce Fram	<pre><fixed frame=""></fixed></pre>					<u> </u>		
		Plane	Cell Count	10							
		Norma	al Cell Cou	0							
		Cell Si	ze	0.5	-						
		Line St	tyle	Lines							
		Color		(228,228,22	8)						
		Alpha		0.5							
		Plane		XY							
		Offset		0; 0; 0							
		02. Robo	ot Model	\checkmark							
		🗄 Statu	s: OK								
		Visual	Enabled	\checkmark							
		Collisi	on Enable								
		Update	e Interval	0							
		Alpha		1							
		Robot	Descriptio	robot_descriptio	n						
		TE Dro	fiv		- T						
1		Global S	tatus: O	к							
I											
			Dame								
L		Add	Remo	ove 🧇 😭		$\langle \rangle$					
	Tim	e									×
	Wa	II Time:	12894241	145.133794	Wall Elapsed	: 2.944775	ROS Time:	1289424145.133791	ROS Elapsed:	2.944775	Reset
н								· · · · · · · · · · · · · · · · · · ·			

How to add the sphere should be fairly self explanatory

Toggle lir	ne numbers
1	<link name="head"/>
2	<visual></visual>
3	<geometry></geometry>
4	<pre><sphere radius="0.2"></sphere></pre>
5	
6	<material name="white"></material>
7	
8	

The meshes here were borrowed from the PR2. They are separate files which you have to specify the path for. You should use the package://NAME_OF_PACKAGE/path notation. The meshes for this tutorial are located within the urdf_tutorial package, in a folder called meshes.

Toggle line numbers 1 <link name="left_gripper"> 2 <visual> 3 <origin rpy="0.0 0 0" xyz="0 0 0"/> 4 <geometry> <mesh filename="package://urdf_tutorial/meshes/l_finger.dae"/> 5 6 </geometry> 7 </visual> 8 </link>

• The meshes can be imported in a number of different formats. STL is fairly common, but the engine also supports DAE, which can have its own color data, meaning you don't have to specify the color/material. Often these are in separate files. These meshes reference the .tif files also in the meshes folder.

• Meshes can also be sized using relative scaling parameters or a bounding box size.

• We could have also referred to meshes in a completely different package, i.e. package://pr2_description/meshes/gripper_v0/l_finger.dae which will work if

the pr2_description package is installed.

There you have it. A R2D2-like URDF model. Now you can continue on to the next step, making it move (/urdf/Tutorials /Building%20a%20Movable%20Robot%20Model%20with%20URDF).

Except where otherwise noted, the ROS wiki is licensed under the

Creative Commons Attribution 3.0 (http://creativecommons.org/licenses /by/3.0/) | Find us on Google+ (https://plus.google.com /113789706402978299308)

Brought to you by: 🏠 Open Source Robotics Foundation

(http://www.osrfoundation.org)