

Intro to ROS on the youBot in Python

Ross A Knepper

3 October 2012

Using Messages

```
#!/usr/bin/env python

import roslib; roslib.load_manifest('pkgname')
import rospy

from msg_pkg_name.msg import DataType

if __name__ == "__main__":
    rospy.init_node('node_name')

    pub = rospy.Publisher('/topic', DataType)

    sub = rospy.Subscriber('/topic', DataType, callback)

last_msg = None

def callback(msg):
    last_msg = msg
```

Using Services

```
#!/usr/bin/env python

import roslib; roslib.load_manifest('pkgname')
import rospy

from srv_pkg_name.srv import DataType

if __name__ == "__main__":
    rospy.init_node('node_name')

    tc_srv = rospy.ServiceProxy('/service_name', DataType)

    s = rospy.Service('/service_name', DataType, callback)

last_request = None

def callback(srv):
    last_request = srv.fill_in_name
    return True

# [srv_pkg_name/DataType]:
# string fill_in_name
# ---
# bool complete
```

Commanding the youBot

Move base:

Publish to message /drcN/cmd_vel

```
$ rosmmsg show geometry_msgs/Twist
```

```
geometry_msgs/Vector3 linear
```

```
float64 x
```

```
float64 y
```

```
float64 z
```

```
geometry_msgs/Vector3 angular
```

```
float64 x
```

```
float64 y
```

```
float64 z
```

Ignored!



Commanding the youBot

Set position of arm:

```
Publish to /drcN/arm_1/arm_controller/position_command
```

```
$ rosmmsg show brics_actuator/JointPositions
```

```
brics_actuator/Poison poisonStamp
```

```
brics_actuator/JointValue[] positions
```

```
time timeStamp
```

```
string joint_uri ← "arm_joint_1"
```

```
string unit ← "rad"
```

```
float64 value
```

joint angle

Set velocity of arm:

```
Publish to /drcN/arm_1/arm_controller/velocity_command
```

```
$ rosmmsg show brics_actuator/JointVelocities
```

```
brics_actuator/Poison poisonStamp
```

```
brics_actuator/JointValue[] velocities
```

```
time timeStamp
```

```
string joint_uri ← "arm_joint_1"
```

```
string unit ← "s^-1 rad"
```

```
float64 value
```

Open or close the gripper:

```
Publish to /drcN/arm_1/gripper_controller/position_command
```

```
brics_actuator/JointPositions
```

```
brics_actuator/Poison poisonStamp
```

```
brics_actuator/JointValue[] velocities
```

```
time timeStamp
```

```
string joint_uri ← "gripper_finger_joint_r", "gripper_finger_joint_l"
```

```
string unit ← "m"
```

```
float64 value
```

0.0 – 0.011499

Getting positions from VICON

- Subscribe to /drc**N** for the base pose
- Subscribe to /drc**N**_arm for the wrist pose

```
[mit_msgs/MocapPosition]:  
string name  
float64 sample_count  
geometry_msgs/Vector3 translational  
  float64 x  
  float64 y  
  float64 z  
geometry_msgs/Vector3 axisangle  
  float64 x  
  float64 y  
  float64 z
```

Getting positions from TF

```
#!/usr/bin/env python

import roslib; roslib.load_manifest('robot')

import rospy
from tf import TransformListener
from geometry_msgs.msg import PoseStamped
```

```
if __name__ == "__main__":
    rospy.init_node('node_name')
```

```
    tfl = TransformListener()
```

```
    ps = PoseStamped()
    ps.header.frame_id = "/drc2/arm_link_4"
    ps.pose.orientation.w = 1
```

Defaults to all zeros

```
    success = False
    while not success:
```

```
        try:
            ps = tfl.transformPoint("/map", ps)
            success = True
        except Exception:
            time.sleep(0.01)
```

Need to make a valid quaternion