|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task. No.: | 4 | Points: | 5 | Turtlesim subscriber |

|  |
| --- |
| Objectives:  1. Using ROS topics. 2. See messages in topic. 3. Display messages. |

|  |
| --- |
| Description: We saw how to move the turtlesim with sending velocity command through publisher. The goal of this exercise is to get robot position in a Python node with a subscriber. |

|  |  |
| --- | --- |
| Step | Action |
| 1 | Restart *turtlesim\_node* and close *move\_turtle.py*. |
| 2 | Show the turtle’s current position with the following command.  $ rostopic echo /turtle1/pose |
| 3 | See the position message type.  $ rostopic type /turtle1/pose turtlsim/Pose  $ rosmsg show turtlesim/Pose  There are five terms inside the message: x, y, theta, linear velocity, and angular velocity. |
| 4 | Modify the existing *move\_turtle.py* and add the option to subscribe the */turtle1/pose* topic. It is publishing velocity and subscribing the position from the *turtlesim* node at the same time. Save the following code as *move\_turtle\_get\_pose.py*.  #!/usr/bin/env python import rospy from geometry_msgs.msg import Twist from turtlesim.msg import Pose import sys #/turtle1/Pose topic callback def pose_callback(pose): rospy.loginfo("Robot X = %f : Y=%f Z=%f\n",pose.x,pose.y,pose.theta) def move_turtle(lin_vel,ang_vel):  rospy.init_node('move_turtle', anonymous=True) pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10) #Creating new subscriber: Topic name= /turtle1/pose: Callback name: pose_callback rospy.Subscribe('/turtle1/pose',Pose, pose_callback) rate = rospy.Rate(10) # 10hz vel = Twist() while not rospy.is_shutdown(): vel.linear.x = lin_vel vel.linear.y = 0 vel.linear.z = 0 vel.angular.x = 0 vel.angular.y = 0 vel.angular.z = ang_vel rospy.loginfo("Linear Vel = %f: Angular Vel = %f",lin_ vel,ang_vel)  pub.publish(vel) rate.sleep() if_name_=='_main_': try move_turtle(float(sys.argv[1]),float(sys.argv[2])) except rospy.ROSInterruptException: pass |
| 5 | Run the code by using the following commands.  $ roscore  $ rosrun turtlesim turtlesim_node |
| 6 | Move the robot and see its position.  $ rosrun hello_world move_turtle_get_pose.py 0.2 0.1 |

 This is shared under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License