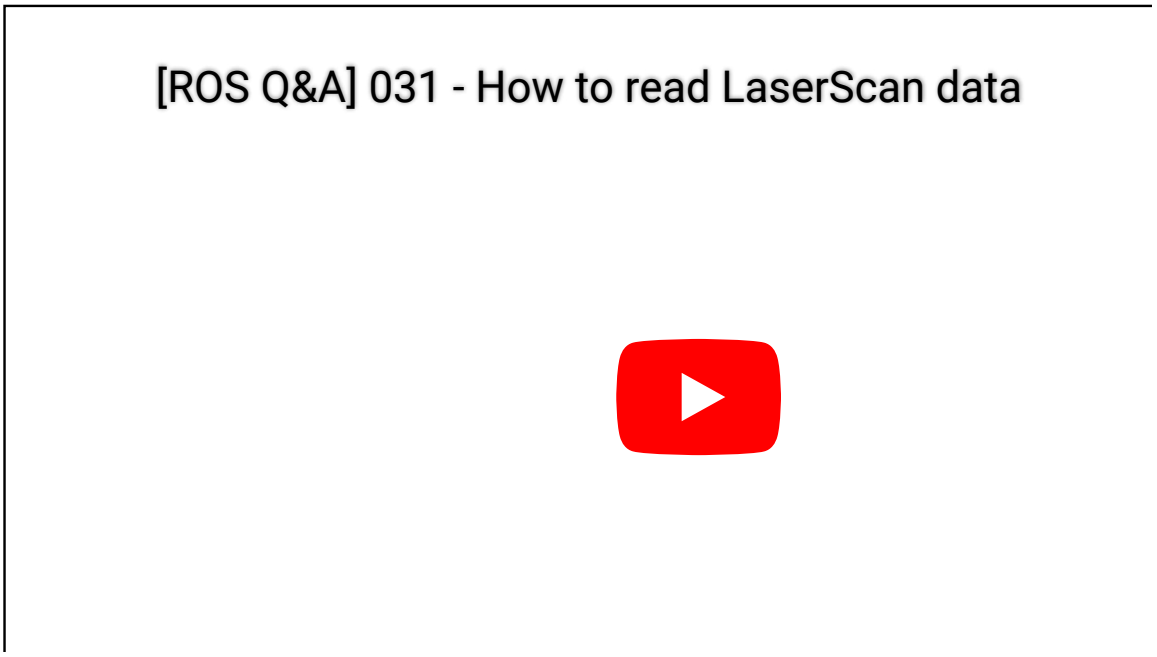
[ROS Q&A] 031 – How to read LaserScan data (ROS python)

ROS Q&A

26/09/2017

About

In this video, we are going to see this question on **ROS Answers** – **How to know the exact frame rate and angle of /scan on Turtlebot?**



Step 1. Open a project on ROS Development Studio(ROSDS)

We will reproduce the question by using ROSDS. You can create a free account **here**.

After registration, login and add a new ROSject. Now, run the project on ROSDS and launch the Turtlebot2 by clicking the Simulation button.

Step 2. Read LaserScan data

The simulation is up and running now. You can check all the topic available with the command:

```
$ rostopic list
```

The LaserScan topic is called /kobuki/laser/scan. You can type the following command into the terminal to check the topic.

```
$ rostopic echo /kobuki/lase/scan -n1
```

The -n1 flag prints the topic exactly once. You'll get something like this.

```
header:
  seq: 5
  stamp:
    secs: 2829
    nsecs:  69000000
  frame_id: "laser_sensor_link"
angle_min: -1.57079994678
angle_max: 1.57079994678
angle_increment: 0.00436940183863
time_increment: 0.0
scan_time: 0.0
range_min: 0.10000000149
range_max: 30.0
ranges: [inf, inf, inf, inf, inf, ....]
```

The following command will give you the information for the topic

```
rostopic info /kobuki/laser/scan
```

You'll see that the topic is using the message type called sensor_msgs/LaserScan. You can further check it with the command

```
rosmsg show sensor_msgs/LaserScan
```

The command gives you the message's structure.

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities
```

The **angle_min** and **angle_max** indicate the angle range(from -90 to 90 degree in this case) that the LaserScan is measuring and the **ranges** is an array which gives you the distance measured for each angle bin.

To explore the range value, let's create a package.

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg laser_values rospy
$ cd laser_values
$ mkdir launch
```

Add a python script under src with the following code

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

import rospy
from sensor_msgs.msg import LaserScan

def callback(msg):
    print len(msg.ranges)

rospy.init_node('scan_values')
sub = rospy.Subscriber('/kobuki/laser/scan', Lase
rospy.spin()
```

Normally, you'll need to give the script permission to execute with

```
$ chmod +x src/scan.py
```

Then we create a launch file under lunch directory to launch the script

```
<launch>
    <node pkg="laser_values" type="scan.py" name=

    </node>
</launch>
```

Now you can type

```
roslaunch laser_values laser.launch
```

to launch the script. You should also see that the length of the **ranges** array is 720 in the 180-degree range. So if we want to read the LaserScan data on the left, in front and on the right of the robot, we can change our script a bit.

```
...
def callback(msg):
    # values at 0 degree
    print msg.ranges[0]
    # values at 90 degree
    print msg.ranges[360]
    # values at 180 degree
    print msg.ranges[719]
...
```

That’s all now you get the value. You can play with it to navigate the robot.