# Sistemas de Tempo Real (STR)
# - Práticas lab. -

## Robot Operating System (ROS)

- ROS Installation
- Setup, Configuration, Packges
- Nodes
- Visualization (Rviz)
- LiDAR node

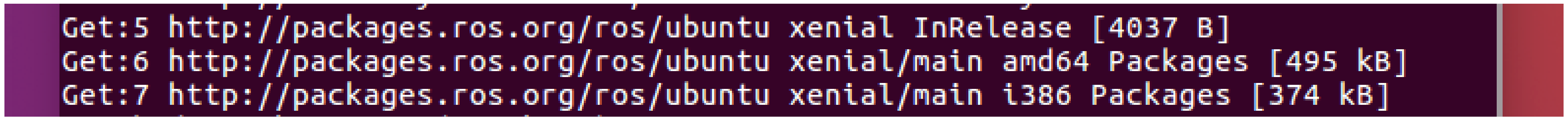
Cristiano Premebida
*cpremebida@isr.uc.pt*

Created on Nov/2017
Updated on Nov/2017

# ROS SETUP

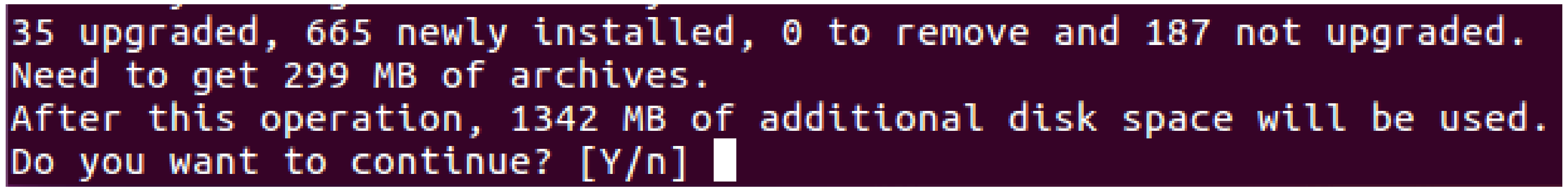
## ROS installation

- `sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'`
- `sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116`
- `sudo apt-get update`



```
Get:5 http://packages.ros.org/ros/ubuntu xenial InRelease [4037 B]
Get:6 http://packages.ros.org/ros/ubuntu xenial/main amd64 Packages [495 kB]
Get:7 http://packages.ros.org/ros/ubuntu xenial/main i386 Packages [374 kB]
```

```
sudo apt-get install ros-kinetic-desktop
```



```
35 upgraded, 665 newly installed, 0 to remove and 187 not upgraded.
Need to get 299 MB of archives.
After this operation, 1342 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
sudo rosdep init
rosdep update
```

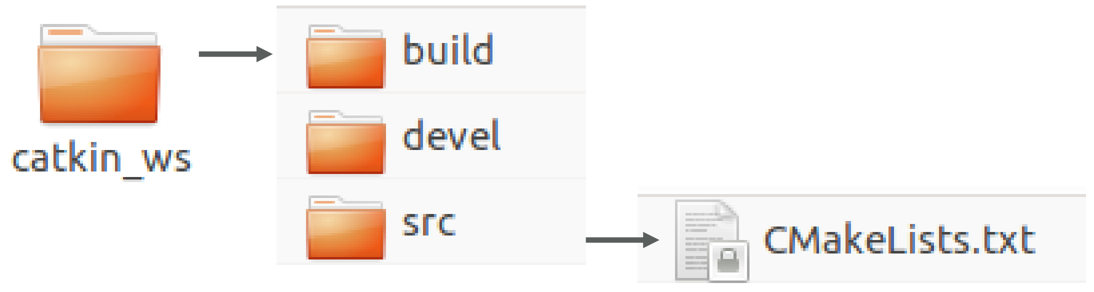
```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

source ~/.bashrc
```

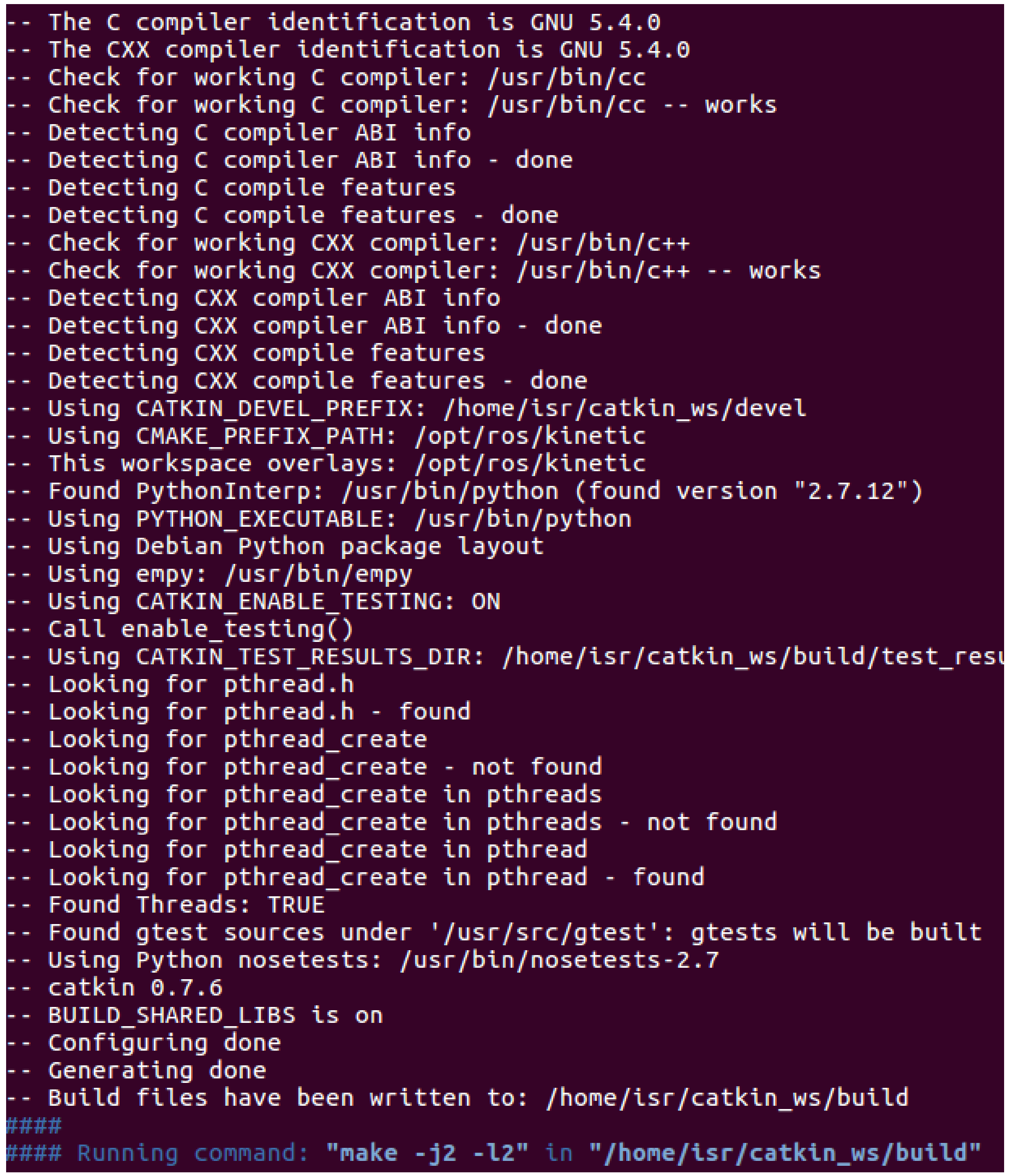
# ROS WORKSPACE HIERARCHY

## ROS Workspace Configuration

```
$ mkdir -p ~/catkin_ws/src

$ cd ~/catkin_ws/

$ catkin_make
```

```
echo "source ~/catkin_ws/devel/setup.bash” >> ~/.bashrc

source ~/.bashrc
```





```
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/isr/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python (found version "2.7.12")
-- Using PYTHON_EXECUTABLE: /usr/bin/python
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/isr/catkin_ws/build/test_resu
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.6
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/isr/catkin_ws/build
####
#### Running command: "make -j2 -l2" in "/home/isr/catkin_ws/build"
```

# ROS PACKAGE

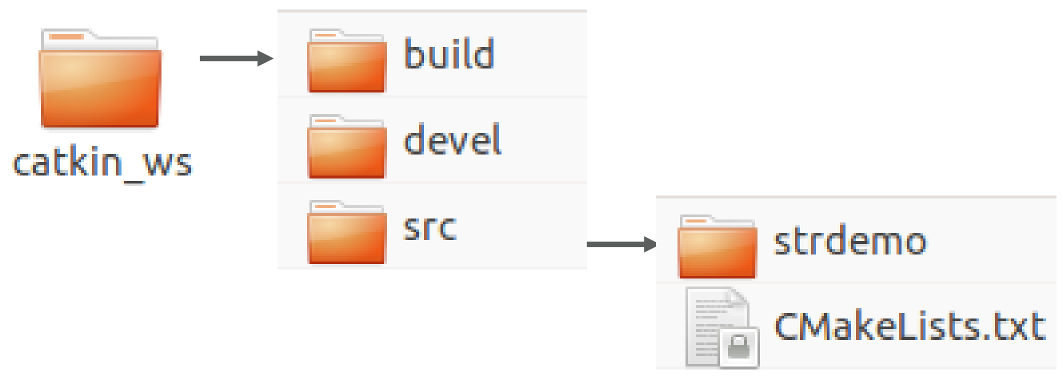
## Create a new Package

```
cd ~/catkin_ws/src
catkin_create_pkg strdemo std_msgs sensor_msgs rospy roscpp
```

ROS workspace (see Tutorial 1.1)

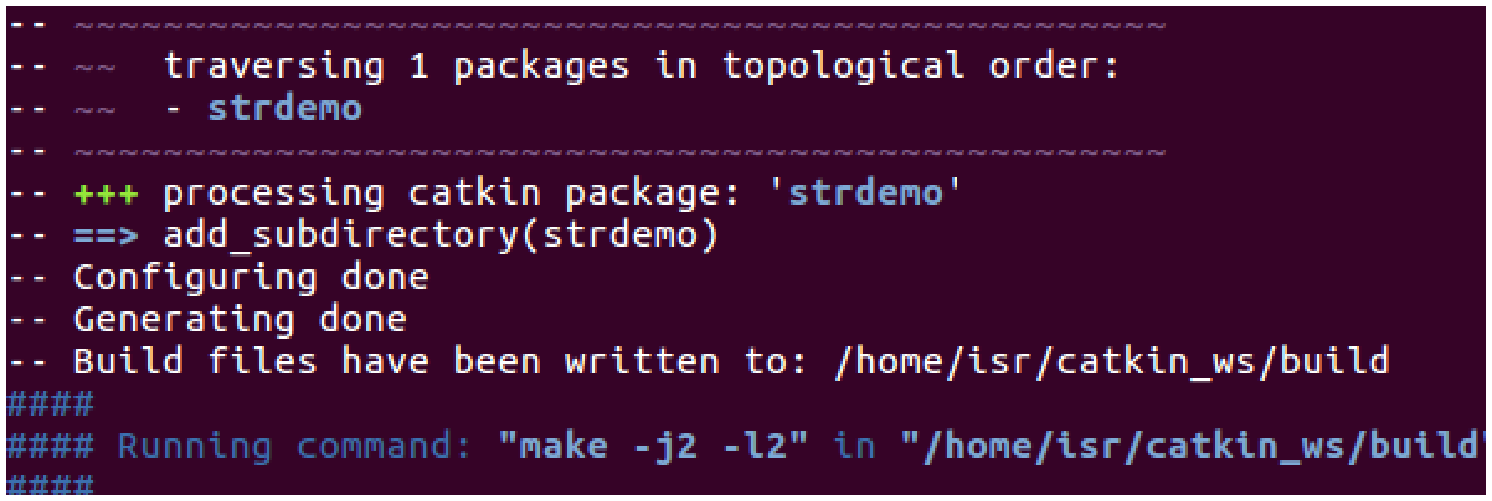
<package_name>

[depend1] [depend2] [depend3] [depend4]



## Build the new Package

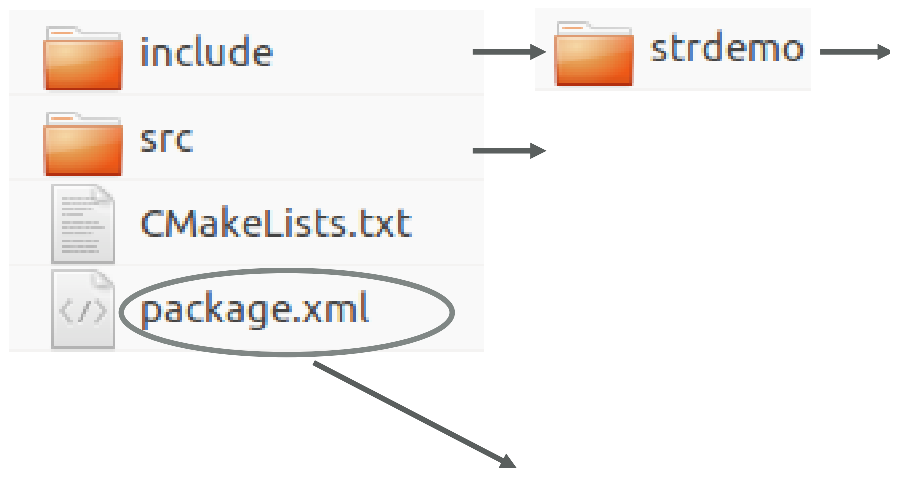
```
cd ~/catkin_ws
catkin_make
catkin_make -DCMAKE_BUILD_TYPE=Release
```



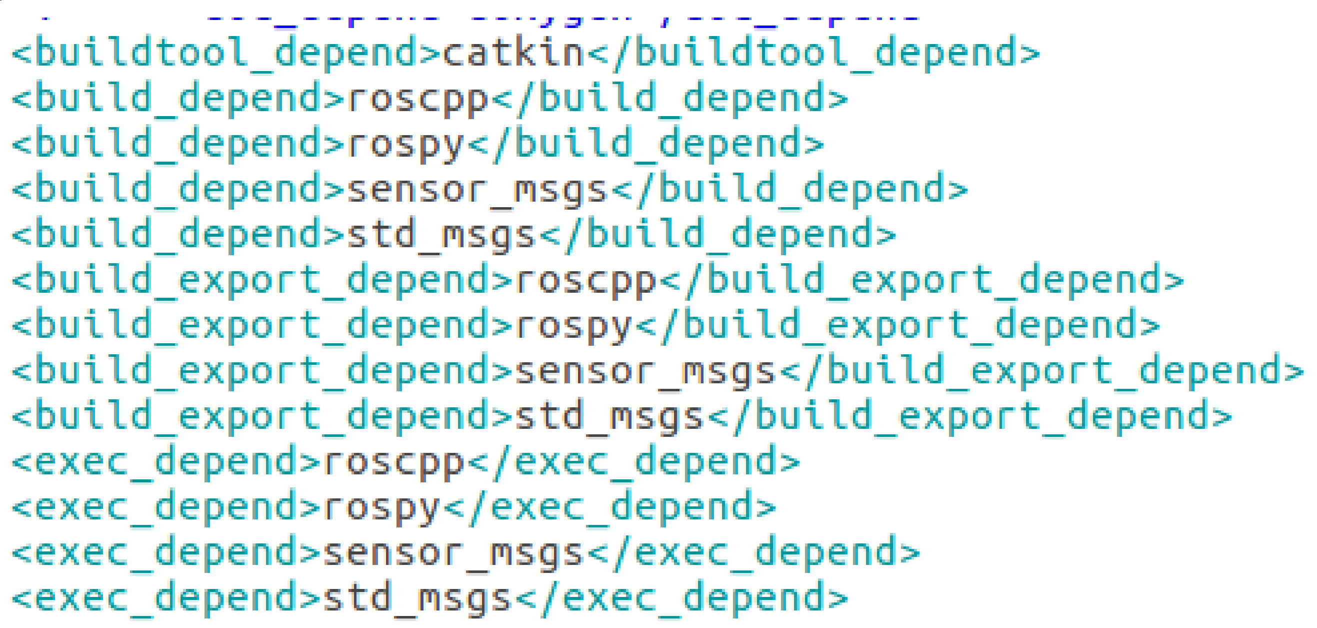
```
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- ~~  traversing 1 packages in topological order:
-- ~~  - strdemo
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- +++ processing catkin package: 'strdemo'
-- ==> add_subdirectory(strdemo)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/isr/catkin_ws/build
####
#### Running command: "make -j2 -l2" in "/home/isr/catkin_ws/build
####
```

# ROS PACKAGE HIERARCHY

## Package Structure



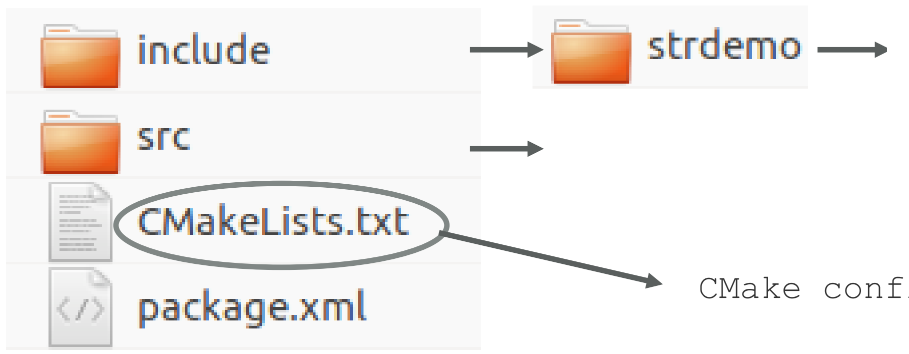
Package Dependencies



```
<buildtool_depend>catkin</buildtool_depend>
<build_depend>roscpp</build_depend>
<build_depend>rospy</build_depend>
<build_depend>sensor_msgs</build_depend>
<build_depend>std_msgs</build_depend>
<build_export_depend>roscpp</build_export_depend>
<build_export_depend>rospy</build_export_depend>
<build_export_depend>sensor_msgs</build_export_depend>
<build_export_depend>std_msgs</build_export_depend>
<exec_depend>roscpp</exec_depend>
<exec_depend>rospy</exec_depend>
<exec_depend>sensor_msgs</exec_depend>
<exec_depend>std_msgs</exec_depend>
```

# ROS PACKAGE HIERARCHY

## Package Structure



CMake configuration file

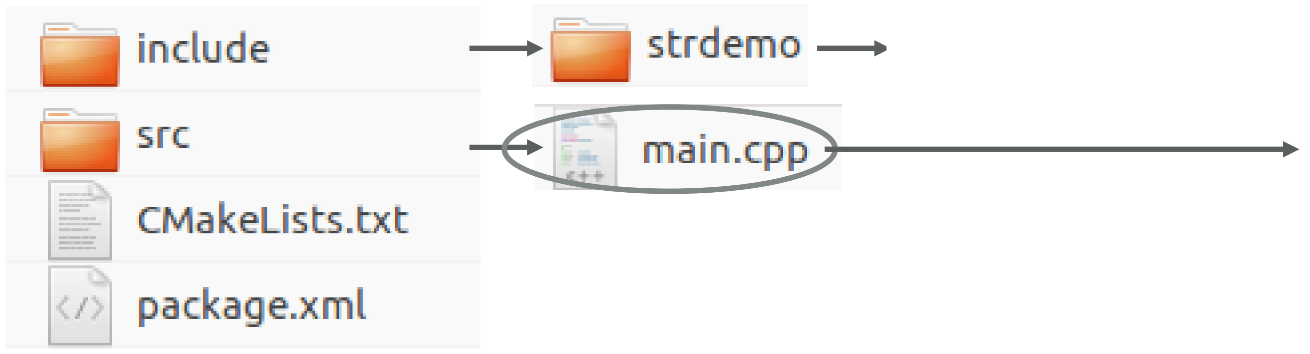
```
## Compile as C++11, supported in ROS Kinetic and newer
# add_compile_options(-std=c++11)

find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  sensor_msgs
  std_msgs
)

## Generate messages in the 'msg' folder
# add_message_files(
#   FILES
#   Message1.msg
#   Message2.msg
# )
```

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
# add_executable(${PROJECT_NAME}_node src/strdemo_node.cpp)
```

## Adding a source file

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
 add_executable(${PROJECT_NAME}_node src/main.cpp)
```

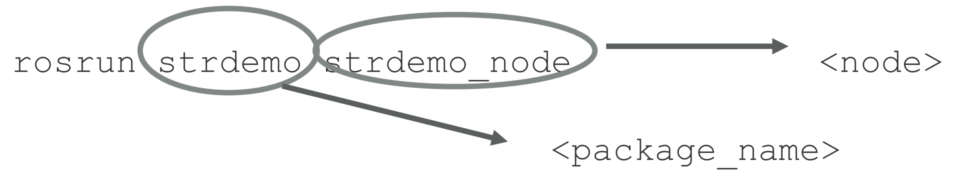
```
## Specify libraries to link a library or executable target against
 target_link_libraries(${PROJECT_NAME}_node
   ${catkin_LIBRARIES}
 )
```



```
#include <iostream>

int main(){
    std::cout<<"Hello World"<<std::endl;
}
```

```
catkin_make
-- +++ processing catkin package: 'strdemo'
-- ==> add_subdirectory(strdemo)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/isr/catkin_ws/build
####
#### Running command: "make -j2 -l2" in "/home/isr/catkin_ws/build"
####
Scanning dependencies of target strdemo_node
[ 50%] Building CXX object strdemo/CMakeFiles/strdemo_node.dir/src/main.cpp.o
[100%] Linking CXX executable /home/isr/catkin_ws/devel/lib/strdemo/strdemo_node
[100%] Built target strdemo_node
```

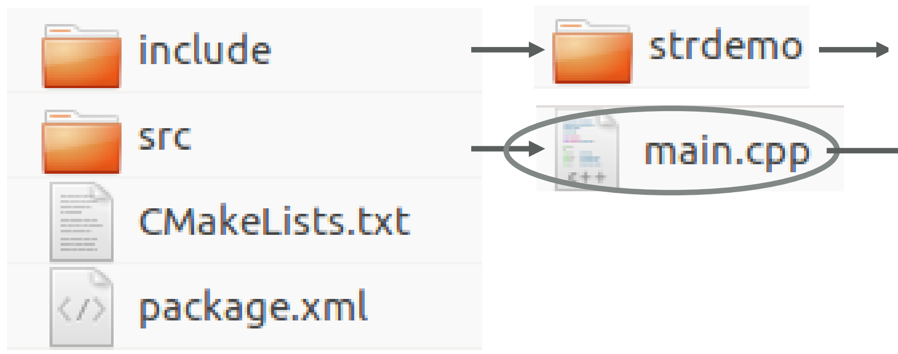
## Running the new node



```
isr@pc:~/catkin_ws$ rosrun strdemo strdemo_node
Hello World
```

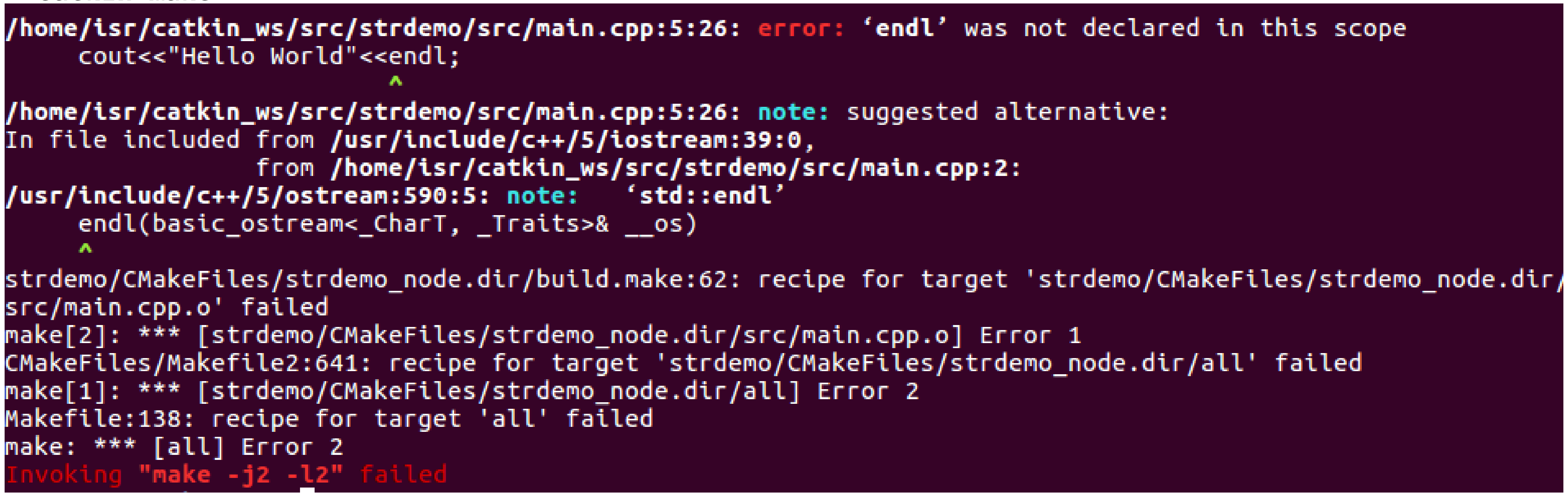
# ROS FIRST NODE

## Adding a source file with errors



```cpp
#include <iostream>

int main(){
     cout<<"Hello World"<<endl;
}
```

`catkin_make`



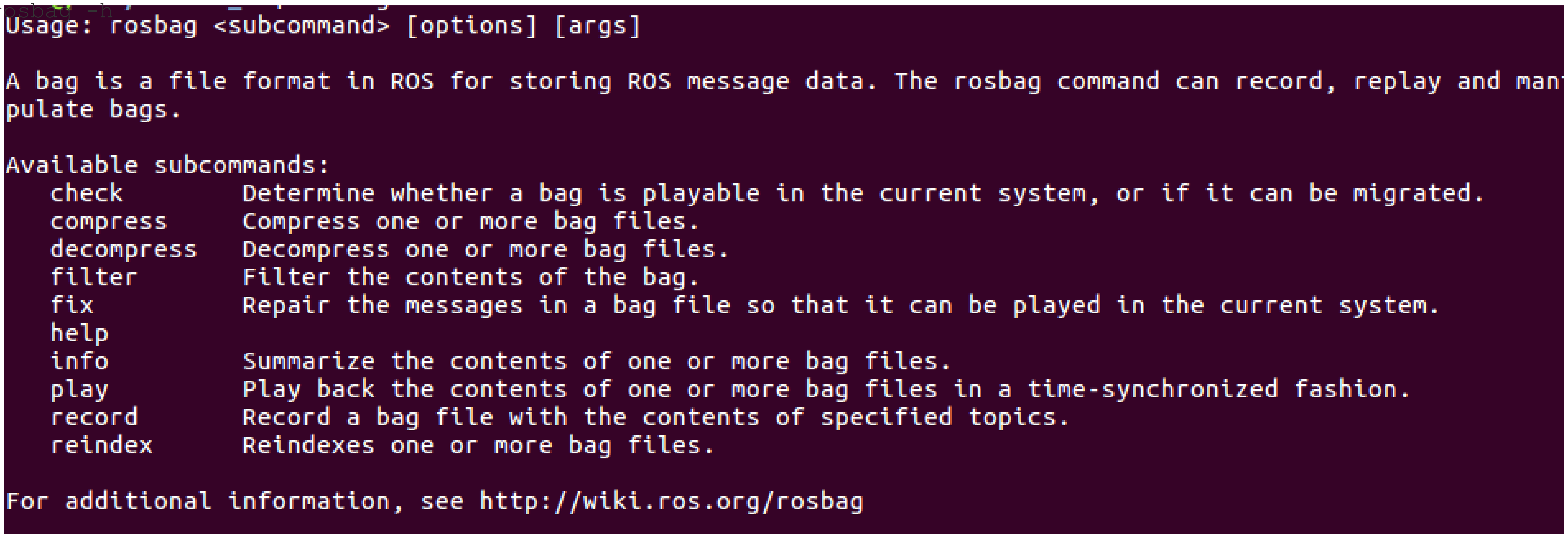
```
/home/isr/catkin_ws/src/strdemo/src/main.cpp:5:26: error: ‘endl’ was not declared in this scope
     cout<<"Hello World"<<endl;
                          ^
/home/isr/catkin_ws/src/strdemo/src/main.cpp:5:26: note: suggested alternative:
In file included from /usr/include/c++/5/iostream:39:0,
                 from /home/isr/catkin_ws/src/strdemo/src/main.cpp:2:
/usr/include/c++/5/ostream:590:5: note:   ‘std::endl’
     endl(basic_ostream<_CharT, _Traits>& __os)
     ^
strdemo/CMakeFiles/strdemo_node.dir/build.make:62: recipe for target 'strdemo/CMakeFiles/strdemo_node.dir/
src/main.cpp.o' failed
make[2]: *** [strdemo/CMakeFiles/strdemo_node.dir/src/main.cpp.o] Error 1
CMakeFiles/Makefile2:641: recipe for target 'strdemo/CMakeFiles/strdemo_node.dir/all' failed
make[1]: *** [strdemo/CMakeFiles/strdemo_node.dir/all] Error 2
Makefile:138: recipe for target 'all' failed
make: *** [all] Error 2
Invoking "make -j2 -l2" failed
```

## Rosbag files - SAVE

```
rosbag record -a

rosbag record /velodyne /odom
```

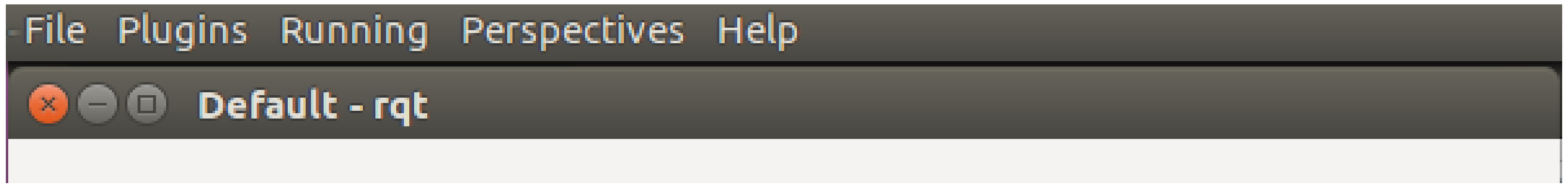
## Rosbag files - PLAY

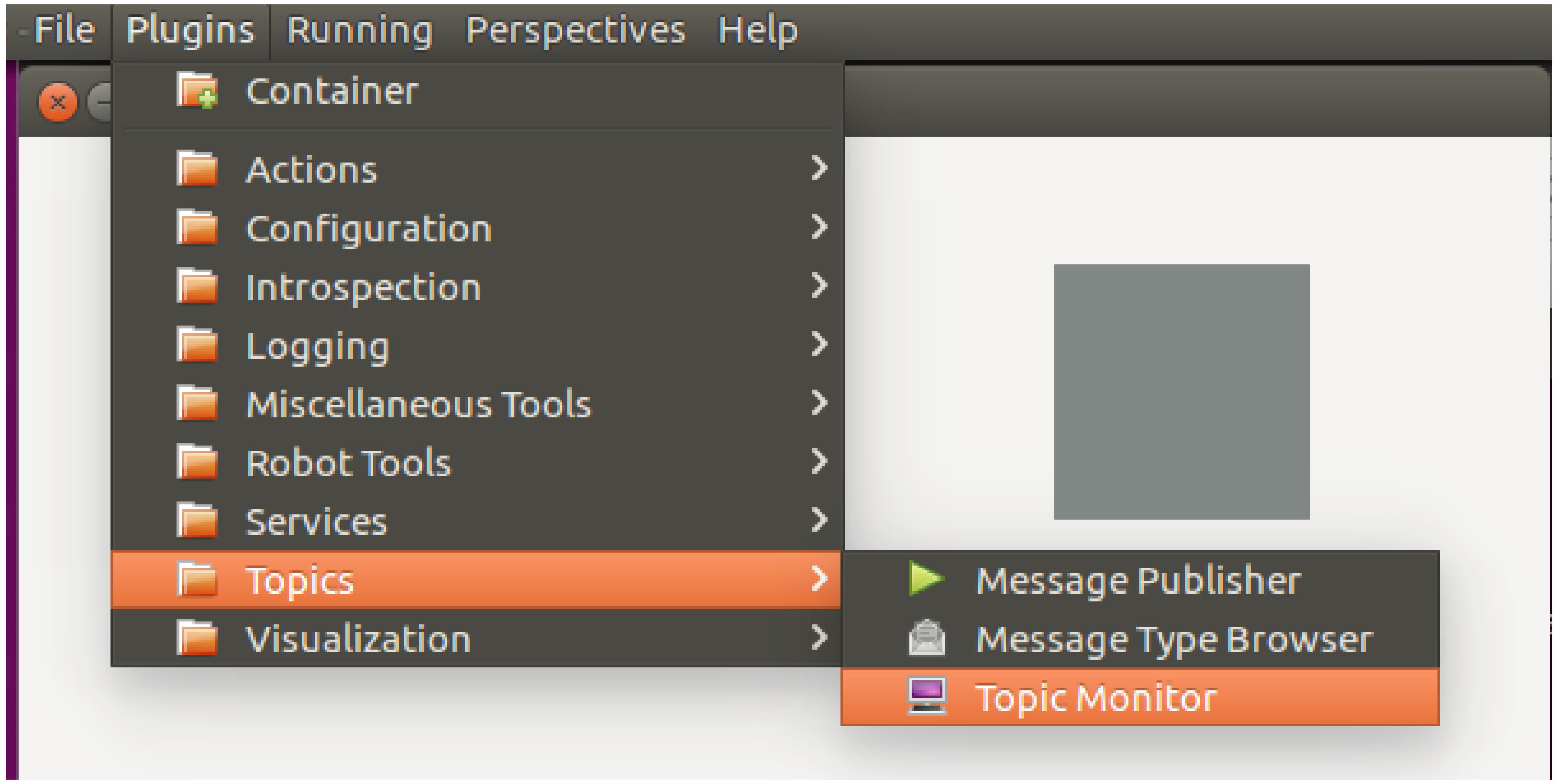
```
rosbag play name.bag

rosbag play name.bag -rate 0.1

rosbag -h
```

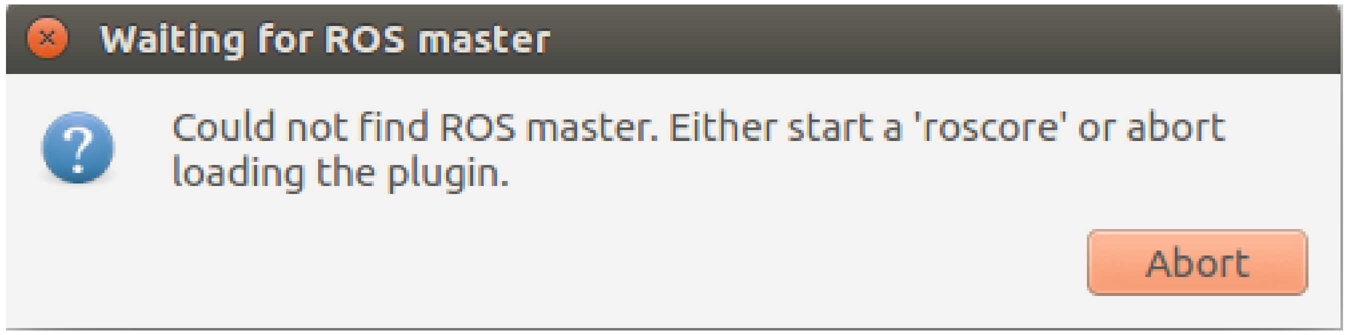


```
Usage: rosbag <subcommand> [options] [args]

A bag is a file format in ROS for storing ROS message data. The rosbag command can record, replay and man
pulate bags.

Available subcommands:
   check        Determine whether a bag is playable in the current system, or if it can be migrated.
   compress     Compress one or more bag files.
   decompress   Decompress one or more bag files.
   filter       Filter the contents of the bag.
   fix          Repair the messages in a bag file so that it can be played in the current system.
   help
   info         Summarize the contents of one or more bag files.
   play         Play back the contents of one or more bag files in a time-synchronized fashion.
   record       Record a bag file with the contents of specified topics.
   reindex      Reindexes one or more bag files.

For additional information, see http://wiki.ros.org/rosbag
```

# DATASETS - RQT

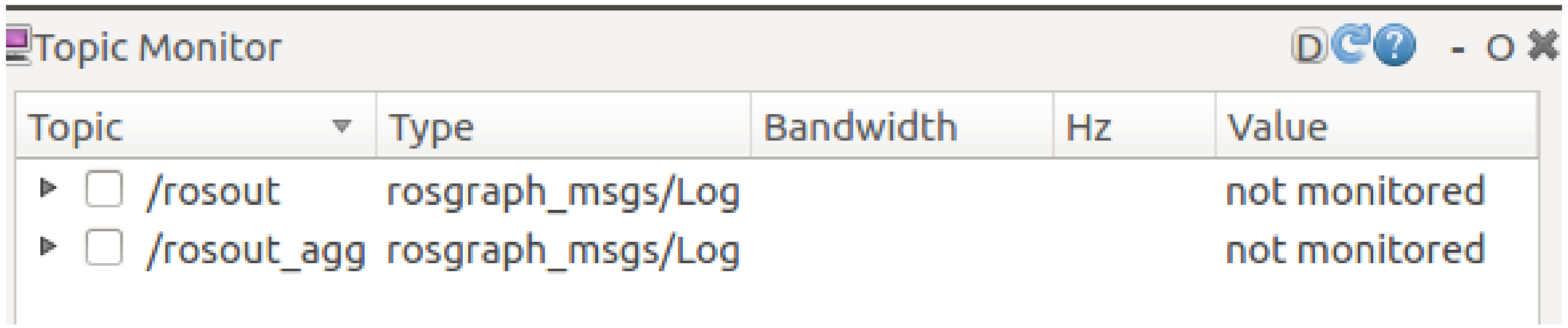





Waiting for ROS master

Could not find ROS master. Either start a 'roscore' or abort loading the plugin.

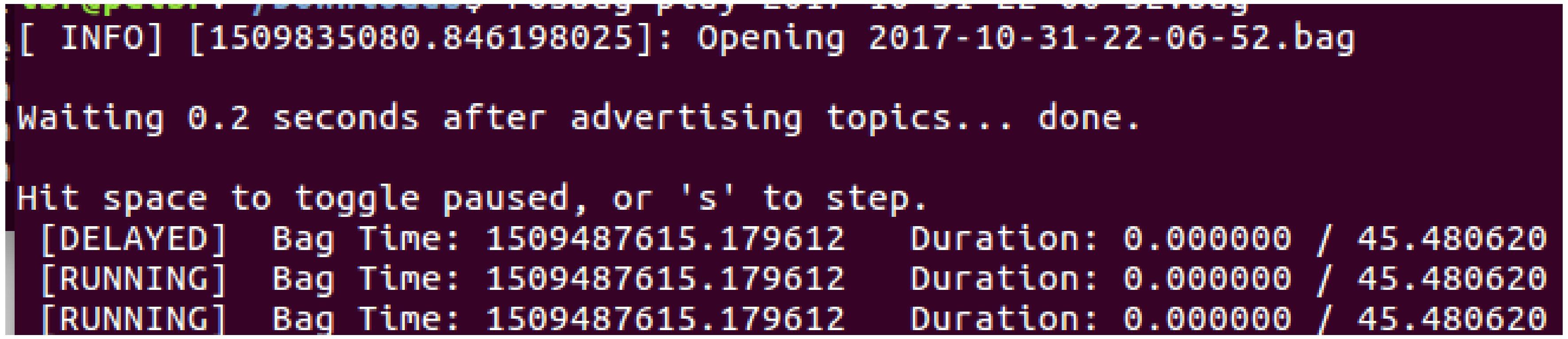
# DATASETS - RQT



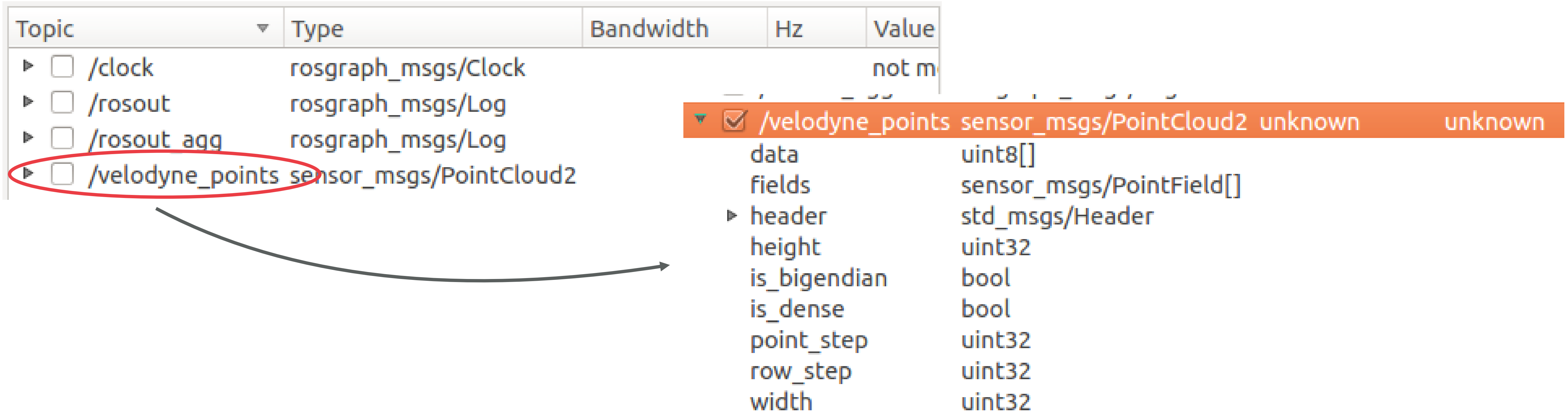
Topic Monitor

| Topic | Type | Bandwidth | Hz | Value |
|---|---|---|---|---|
| /rosout | rosgraph_msgs/Log | | | not monitored |
| /rosout_agg | rosgraph_msgs/Log | | | not monitored |

```
rosbag play 2017-10-31-22-06-52.bag
```



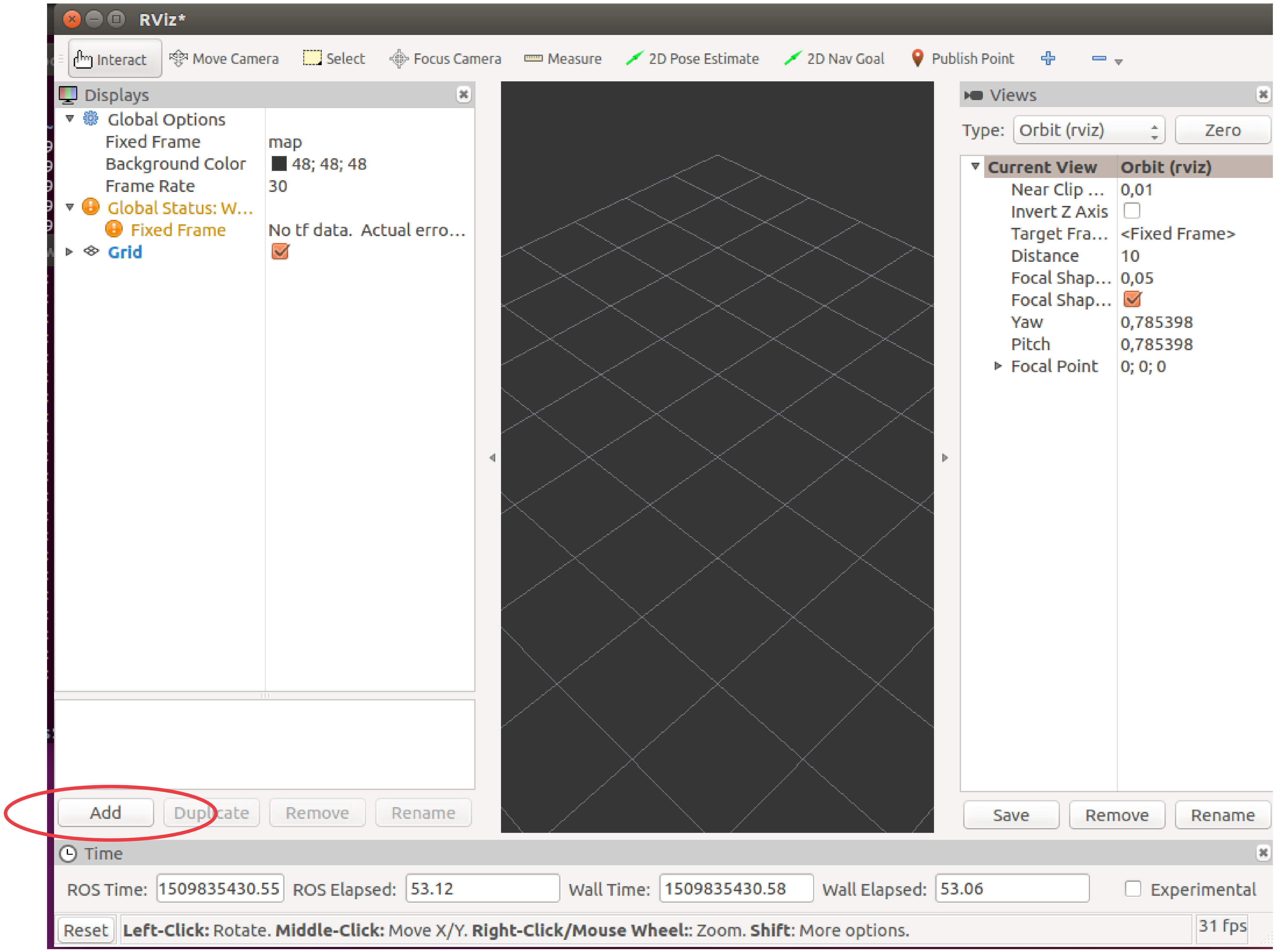
```
[ INFO] [1509835080.846198025]: Opening 2017-10-31-22-06-52.bag

Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
 [DELAYED]  Bag Time: 1509487615.179612   Duration: 0.000000 / 45.480620
 [RUNNING]  Bag Time: 1509487615.179612   Duration: 0.000000 / 45.480620
 [RUNNING]  Bag Time: 1509487615.179612   Duration: 0.000000 / 45.480620
```



| Topic | Type | Bandwidth | Hz | Value |
|---|---|---|---|---|
| /clock | rosgraph_msgs/Clock | | | not m |
| /rosout | rosgraph_msgs/Log | | | |
| /rosout_agg | rosgraph_msgs/Log | | | |
| /velodyne_points | sensor_msgs/PointCloud2 | | | |

| | | | |
|---|---|---|---|
| /velodyne_points | sensor_msgs/PointCloud2 | unknown | unknown |
| data | uint8[] | | |
| fields | sensor_msgs/PointField[] | | |
| header | std_msgs/Header | | |
| height | uint32 | | |
| is_bigendian | bool | | |
| is_dense | bool | | |
| point_step | uint32 | | |
| row_step | uint32 | | |
| width | uint32 | | |

# DATASETS - RVIZ



RViz*

Interact · Move Camera · Select · Focus Camera · Measure · 2D Pose Estimate · 2D Nav Goal · Publish Point

Displays

| | |
|---|---|
| Global Options | |
| Fixed Frame | map |
| Background Color | ■ 48; 48; 48 |
| Frame Rate | 30 |
| Global Status: W… | |
| Fixed Frame | No tf data. Actual erro… |
| Grid | ✓ |

Add · Duplicate · Remove · Rename

Views

Type: Orbit (rviz) · Zero

| | |
|---|---|
| Current View | Orbit (rviz) |
| Near Clip … | 0,01 |
| Invert Z Axis | ☐ |
| Target Fra… | <Fixed Frame> |
| Distance | 10 |
| Focal Shap… | 0,05 |
| Focal Shap… | ✓ |
| Yaw | 0,785398 |
| Pitch | 0,785398 |
| Focal Point | 0; 0; 0 |

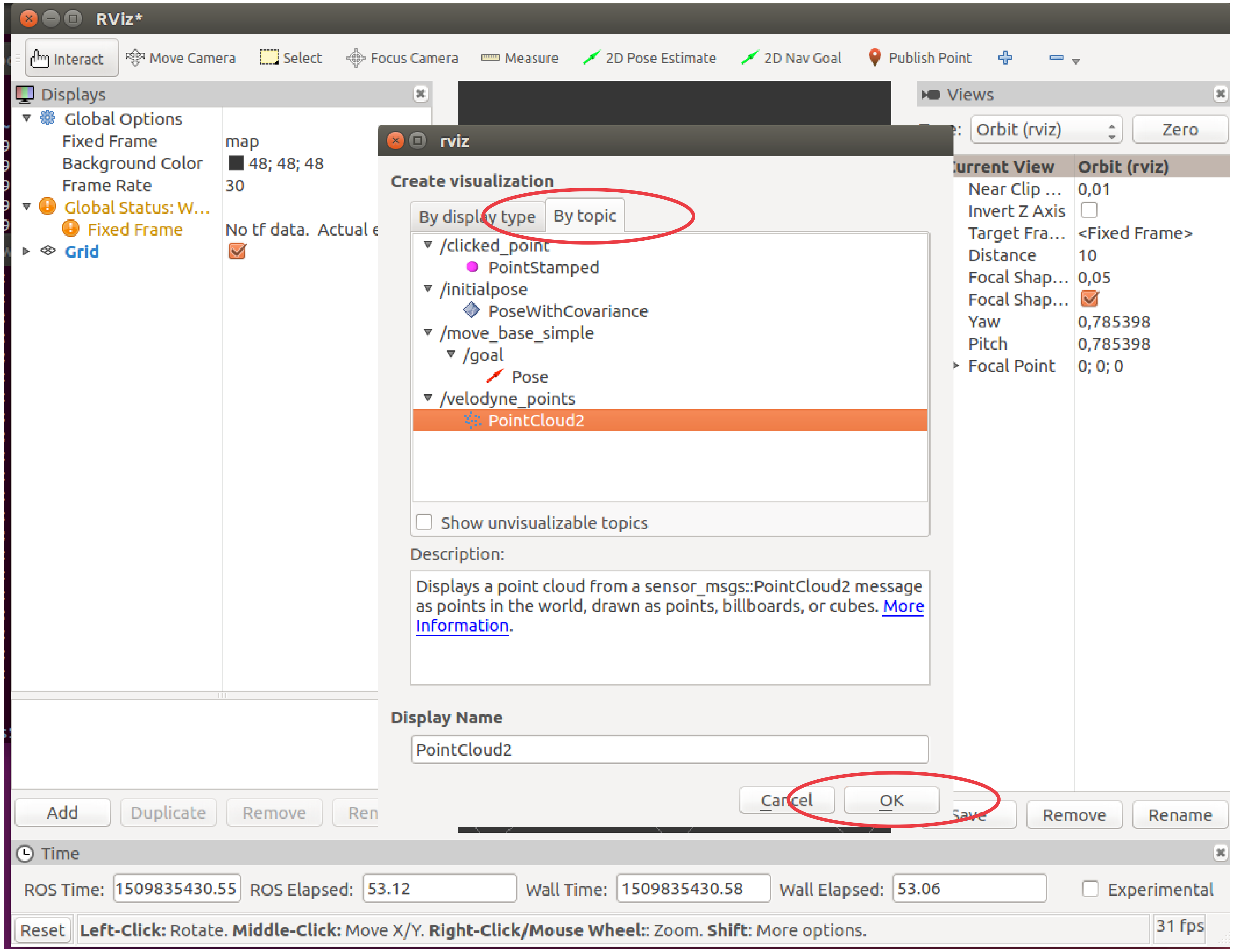
Save · Remove · Rename

Time

ROS Time: 1509835430.55 ROS Elapsed: 53.12 Wall Time: 1509835430.58 Wall Elapsed: 53.06 ☐ Experimental

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options. 31 fps

# DATASETS - RVIZ



| | |
|---|---|
| Global Options | |
| Fixed Frame | map |
| Background Color | ■ 48; 48; 48 |
| Frame Rate | 30 |
| Global Status: W… | |
| Fixed Frame | No tf data. Actual e |
| Grid | ✓ |

rviz

Create visualization

```
▾ /clicked_point
    PointStamped
▾ /initialpose
    PoseWithCovariance
▾ /move_base_simple
  ▾ /goal
      Pose
▾ /velodyne_points
    PointCloud2
```

Show unvisualizable topics

Description:

Displays a point cloud from a sensor_msgs::PointCloud2 message as points in the world, drawn as points, billboards, or cubes. More Information.

Display Name

PointCloud2

Orbit (rviz)

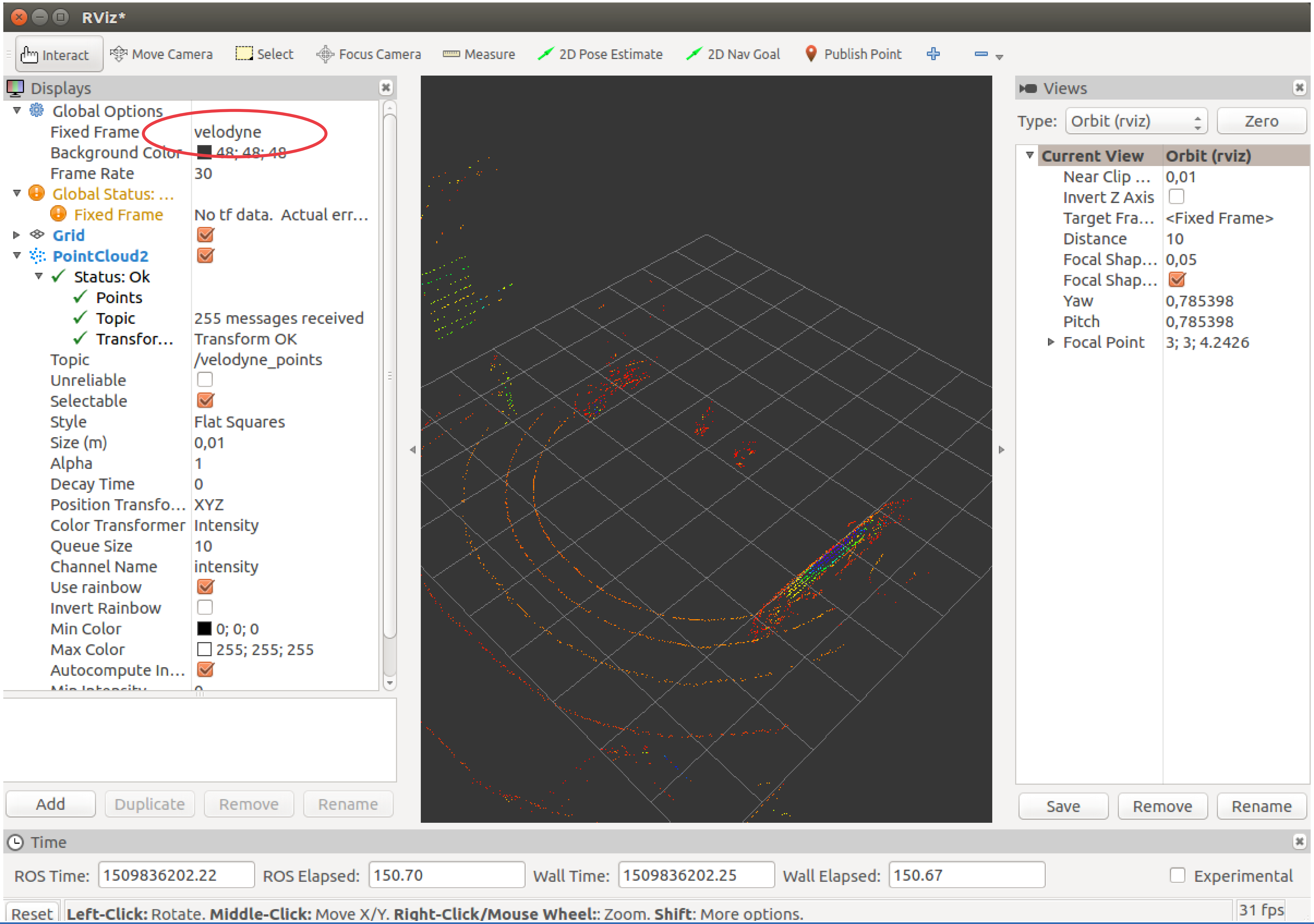
| Current View | Orbit (rviz) |
|---|---|
| Near Clip … | 0,01 |
| Invert Z Axis | ☐ |
| Target Fra… | <Fixed Frame> |
| Distance | 10 |
| Focal Shap… | 0,05 |
| Focal Shap… | ✓ |
| Yaw | 0,785398 |
| Pitch | 0,785398 |
| Focal Point | 0; 0; 0 |

ROS Time: 1509835430.55 ROS Elapsed: 53.12 Wall Time: 1509835430.58 Wall Elapsed: 53.06 Experimental

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 31 fps

# DATASETS - RVIZ

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



| | |
|---|---|
| Global Options | |
| Fixed Frame | velodyne |
| Background Color | ■ 48; 48; 48 |
| Frame Rate | 30 |
| Global Status: ... | |
| Fixed Frame | No tf data. Actual err... |
| Grid | ✓ |
| PointCloud2 | ✓ |
| Status: Ok | |
| Points | |
| Topic | 255 messages received |
| Transfor... | Transform OK |
| Topic | /velodyne_points |
| Unreliable | ☐ |
| Selectable | ✓ |
| Style | Flat Squares |
| Size (m) | 0,01 |
| Alpha | 1 |
| Decay Time | 0 |
| Position Transfo... | XYZ |
| Color Transformer | Intensity |
| Queue Size | 10 |
| Channel Name | intensity |
| Use rainbow | ✓ |
| Invert Rainbow | ☐ |
| Min Color | ■ 0; 0; 0 |
| Max Color | □ 255; 255; 255 |
| Autocompute In... | ✓ |

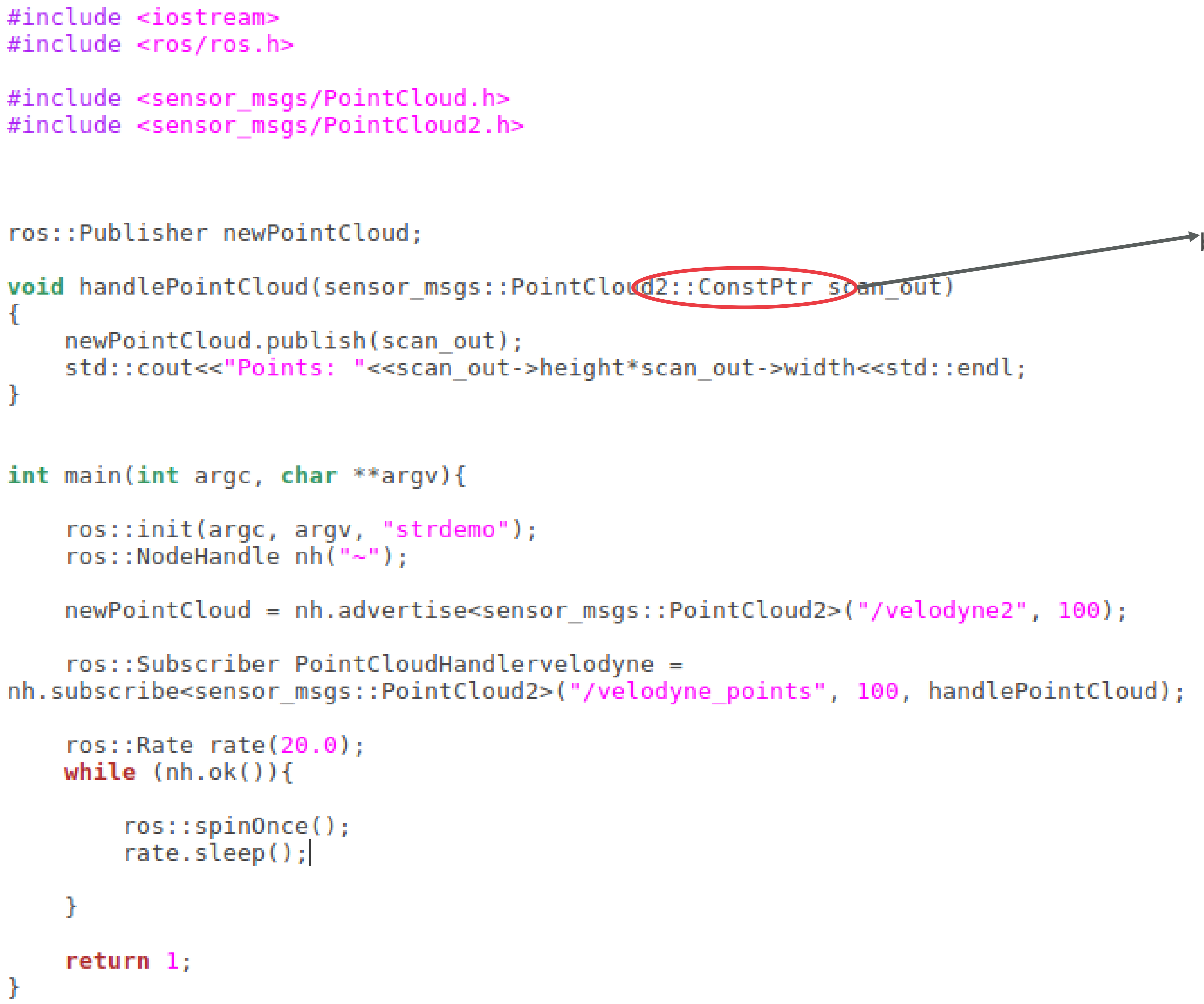
Type: Orbit (rviz)

| Current View | Orbit (rviz) |
|---|---|
| Near Clip ... | 0,01 |
| Invert Z Axis | ☐ |
| Target Fra... | <Fixed Frame> |
| Distance | 10 |
| Focal Shap... | 0,05 |
| Focal Shap... | ✓ |
| Yaw | 0,785398 |
| Pitch | 0,785398 |
| Focal Point | 3; 3; 4.2426 |

ROS Time: 1509836202.22 ROS Elapsed: 150.70 Wall Time: 1509836202.25 Wall Elapsed: 150.67 ☐ Experimental

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

# ROS LIDAR NODE



```
#include <iostream>
#include <ros/ros.h>

#include <sensor_msgs/PointCloud.h>
#include <sensor_msgs/PointCloud2.h>



ros::Publisher newPointCloud;

void handlePointCloud(sensor_msgs::PointCloud2::ConstPtr scan_out)
{
    newPointCloud.publish(scan_out);
    std::cout<<"Points: "<<scan_out->height*scan_out->width<<std::endl;
}


int main(int argc, char **argv){

    ros::init(argc, argv, "strdemo");
    ros::NodeHandle nh("~");

    newPointCloud = nh.advertise<sensor_msgs::PointCloud2>("/velodyne2", 100);

    ros::Subscriber PointCloudHandlervelodyne =
nh.subscribe<sensor_msgs::PointCloud2>("/velodyne_points", 100, handlePointCloud);

    ros::Rate rate(20.0);
    while (nh.ok()){

        ros::spinOnce();
        rate.sleep();

    }

    return 1;
}
```

boost::shared_ptr

# ROS LIDAR NODE

```
#include <iostream>
#include <ros/ros.h>

#include <sensor_msgs/PointCloud.h>
#include <sensor_msgs/PointCloud2.h>
```

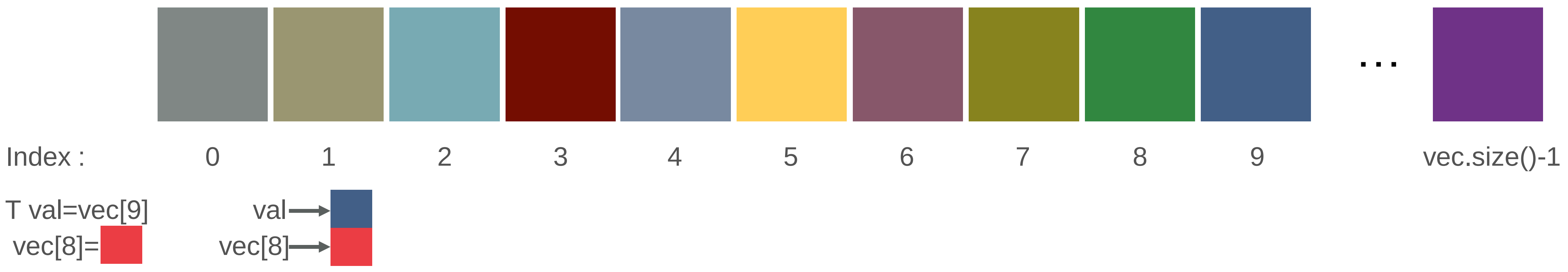
std_msgs/Header header
uint32 height
uint32 width
sensor_msgs/PointField[] fields
bool is_bigendian
uint32 point_step
uint32 row_step
uint8[] data    std::vector<unsigned char>
bool is_dense

std_msgs/Header header
geometry_msgs/Point32[] points std::vector<geometry_msgs/Point32>
sensor_msgs/ChannelFloat32[] channels

float32 x
float32 y
float32 z

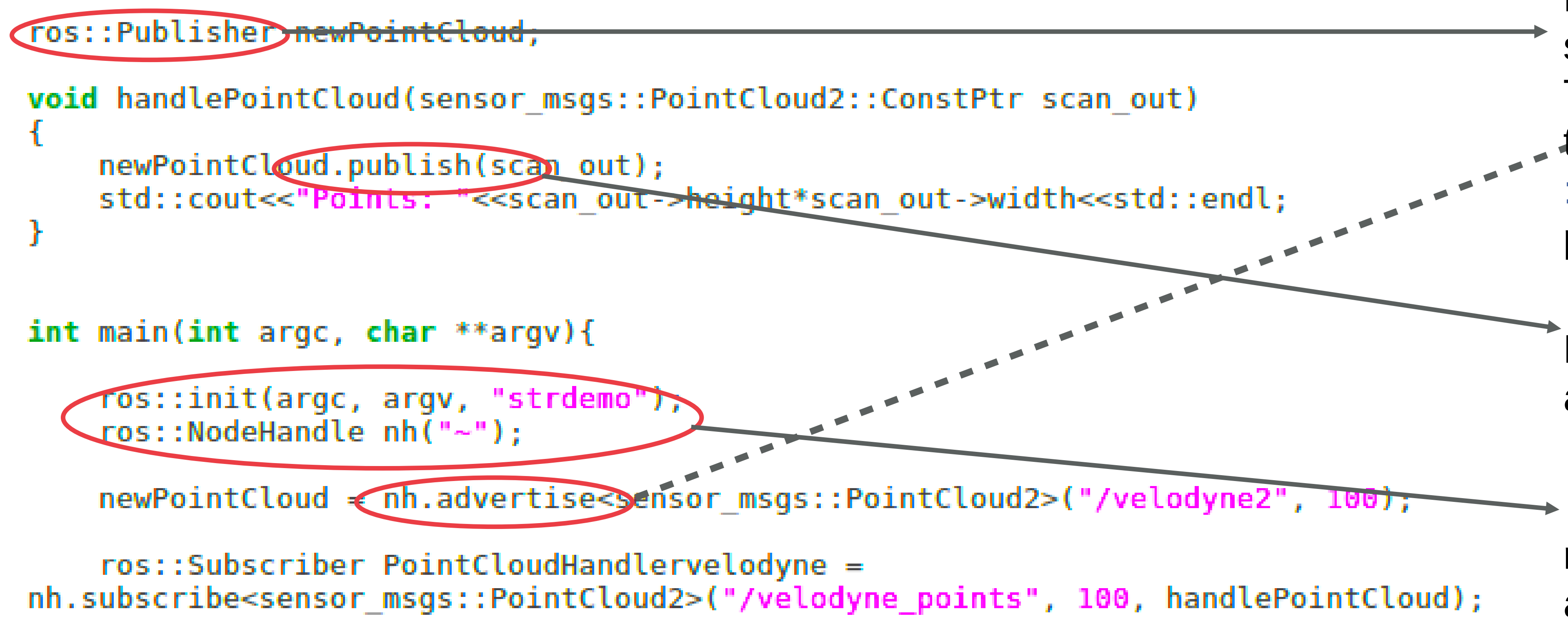
uint32 seq
time stamp
string frame_id

template <typename T> std::vector<T> vec



`push_back  pop_back  erase  reserve  resize`

# ROS LIDAR NODE



```
ros::Publisher newPointCloud;

void handlePointCloud(sensor_msgs::PointCloud2::ConstPtr scan_out)
{
    newPointCloud.publish(scan_out);
    std::cout<<"Points: "<<scan_out->height*scan_out->width<<std::endl;
}


int main(int argc, char **argv){

    ros::init(argc, argv, "strdemo");
    ros::NodeHandle nh("~");

    newPointCloud = nh.advertise<sensor_msgs::PointCloud2>("/velodyne2", 100);

    ros::Subscriber PointCloudHandlervelodyne =
nh.subscribe<sensor_msgs::PointCloud2>("/velodyne_points", 100, handlePointCloud);
```

Manages the advertisement on a specific topic.
This should always be created through a call to NodeHandle ::advertise() or copy from a previously instantiated publisher

Publish a message on the topic associated with this instance

ros::init() is called before using any other part of the ROS system. argc and argv are used by ROS tools to pass commands (remmaping).
NodeHandle is the main access point to communications with the ROS system.

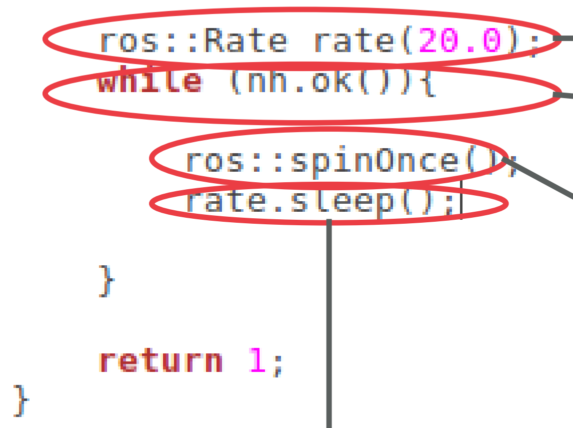
## ROS LIDAR NODE

```
ros::Publisher newPointCloud;

void handlePointCloud(sensor_msgs::PointCloud2::ConstPtr scan_out)
{
    newPointCloud.publish(scan_out);
    std::cout<<"Points: "<<scan_out->height*scan_out->width<<std::endl;
}


int main(int argc, char **argv){

    ros::init(argc, argv, "strdemo");
    ros::NodeHandle nh("~");

    newPointCloud = nh.advertise<sensor_msgs::PointCloud2>("/velodyne2", 100);

    ros::Subscriber PointCloudHandlervelodyne =
nh.subscribe<sensor_msgs::PointCloud2>("/velodyne_points", 100, handlePointCloud);
```

The subscribe() tells roscore that you want to receive messages in the specified topic. Received messages are passed to a callback function.
The second parameter to the subscribe() function is the size of the message buffer.

The advertise() tells ROS that the node will publish on a given topic name (first parameter). After the advertise() call is made, roscore will notify anyone who is trying to subscribe the topic, and in turn negotiate a peer-to-peer connection between nodes. If the Publisher object is destroyed, the topic is automatically unadvertised. The second parameter is the message buffer size.

# ROS LIDAR NODE



```
    ros::Rate rate(20.0);
    while (nh.ok()){

        ros::spinOnce();
        rate.sleep();

    }

    return 1;
}
```

Object to help run loops at a desired frequency (20 Hz).

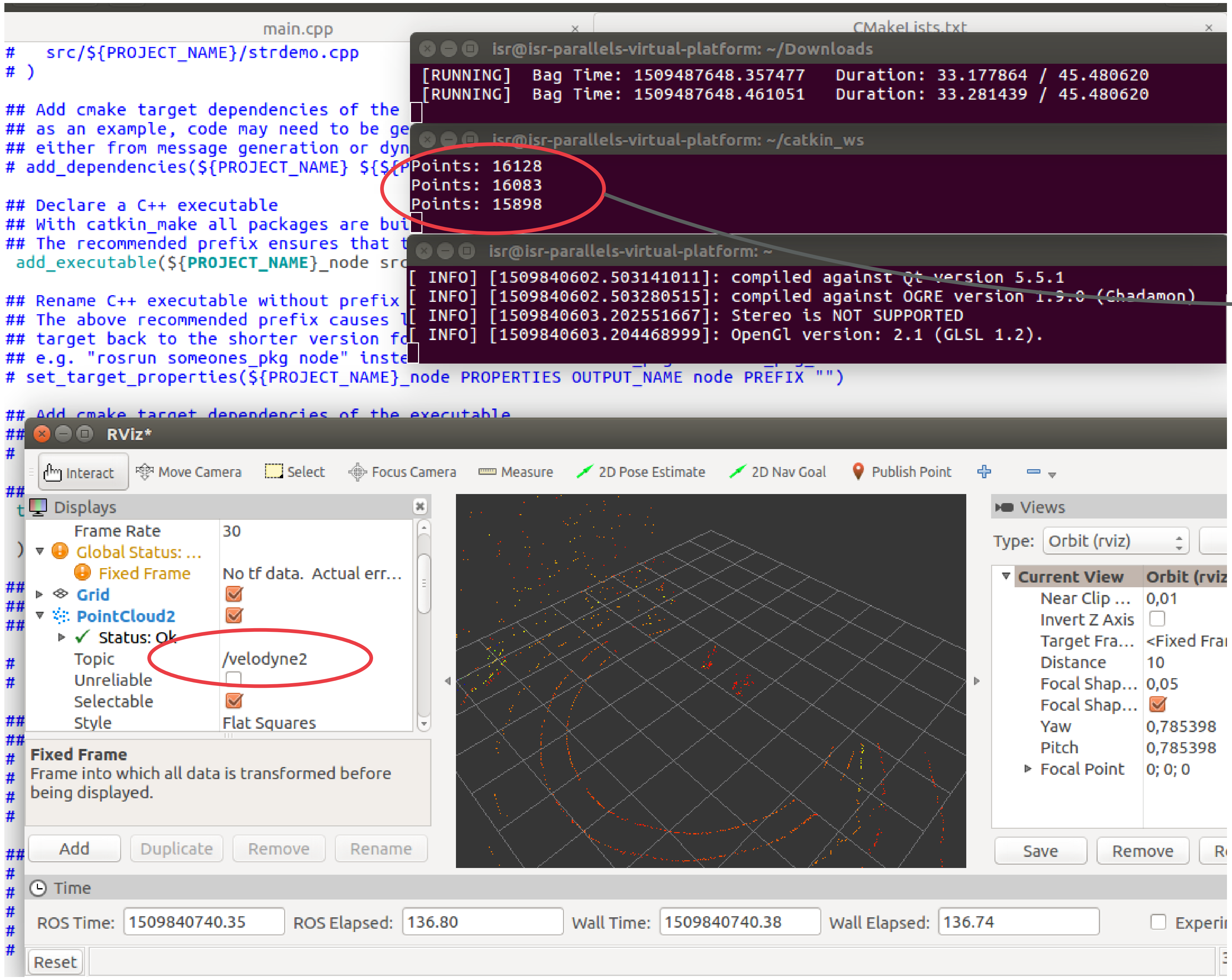
Returns false after a requested shutdown (CONTROL + C)

When a message arrives, ROS pushes your subscriber callback onto a queue. It does not call it immediately.

ROS only processes your callbacks when you tell it to with ros::spinOnce()

Sleeps for any leftover time in a cycle. Calculated from the last time sleep, reset, or the constructor was called.

# ROS LIDAR NODE



main.cpp

CMakeLists.txt

```
#   src/${PROJECT_NAME}/strdemo.cpp
# )

## Add cmake target dependencies of the
## as an example, code may need to be ge
## either from message generation or dyn
# add_dependencies(${PROJECT_NAME} ${${P

## Declare a C++ executable
## With catkin_make all packages are bui
## The recommended prefix ensures that t
 add_executable(${PROJECT_NAME}_node src

## Rename C++ executable without prefix
## The above recommended prefix causes l
## target back to the shorter version fo
## e.g. "rosrun someones_pkg node" inste
# set_target_properties(${PROJECT_NAME}_node PROPERTIES OUTPUT_NAME node PREFIX "")

## Add cmake target dependencies of the executable
```

isr@isr-parallels-virtual-platform: ~/Downloads

```
[RUNNING]  Bag Time: 1509487648.357477   Duration: 33.177864 / 45.480620
[RUNNING]  Bag Time: 1509487648.461051   Duration: 33.281439 / 45.480620
```

isr@isr-parallels-virtual-platform: ~/catkin_ws

```
Points: 16128
Points: 16083
Points: 15898
```

isr@isr-parallels-virtual-platform: ~

```
[ INFO] [1509840602.503141011]: compiled against Qt version 5.5.1
[ INFO] [1509840602.503280515]: compiled against OGRE version 1.9.0 (Ghadamon)
[ INFO] [1509840603.202551667]: Stereo is NOT SUPPORTED
[ INFO] [1509840603.204468999]: OpenGl version: 2.1 (GLSL 1.2).
```

RViz*

Displays

| | |
|---|---|
| Frame Rate | 30 |
| Global Status: ... | |
| Fixed Frame | No tf data. Actual err... |
| Grid | ✓ |
| PointCloud2 | ✓ |
| Status: Ok | |
| Topic | /velodyne2 |
| Unreliable | ☐ |
| Selectable | ✓ |
| Style | Flat Squares |

**Fixed Frame**
Frame into which all data is transformed before being displayed.

Views

Type: Orbit (rviz)

| Current View | Orbit (rviz |
|---|---|
| Near Clip ... | 0,01 |
| Invert Z Axis | ☐ |
| Target Fra... | <Fixed Fra |
| Distance | 10 |
| Focal Shap... | 0,05 |
| Focal Shap... | ✓ |
| Yaw | 0,785398 |
| Pitch | 0,785398 |
| Focal Point | 0; 0; 0 |

Time

ROS Time: 1509840740.35  ROS Elapsed: 136.80  Wall Time: 1509840740.38  Wall Elapsed: 136.74

ROS bag

STR velodyne node

The number of points per frame is not constant