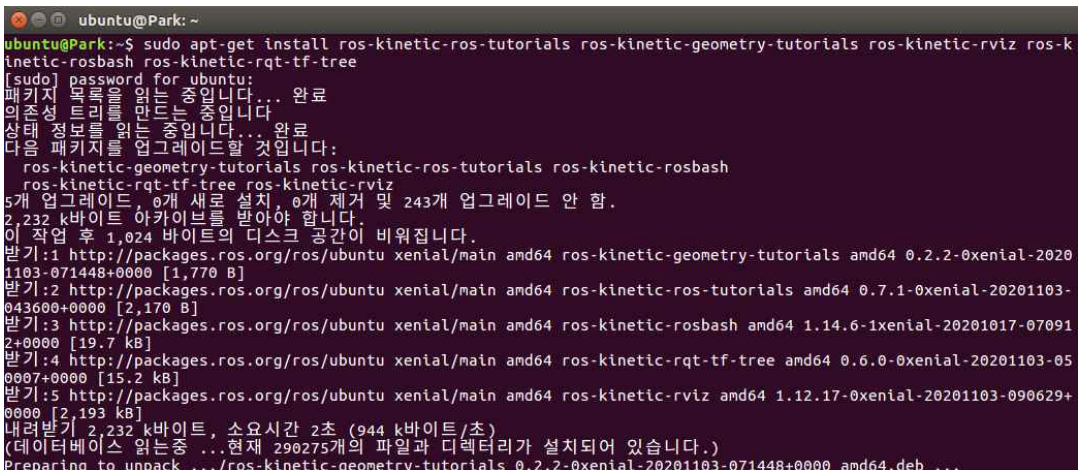
ROS Rviz,bag,sensor_msg,tf(2)

학과 : 산업인공지능학과
학번 : 2020254016
이름 : 박민우

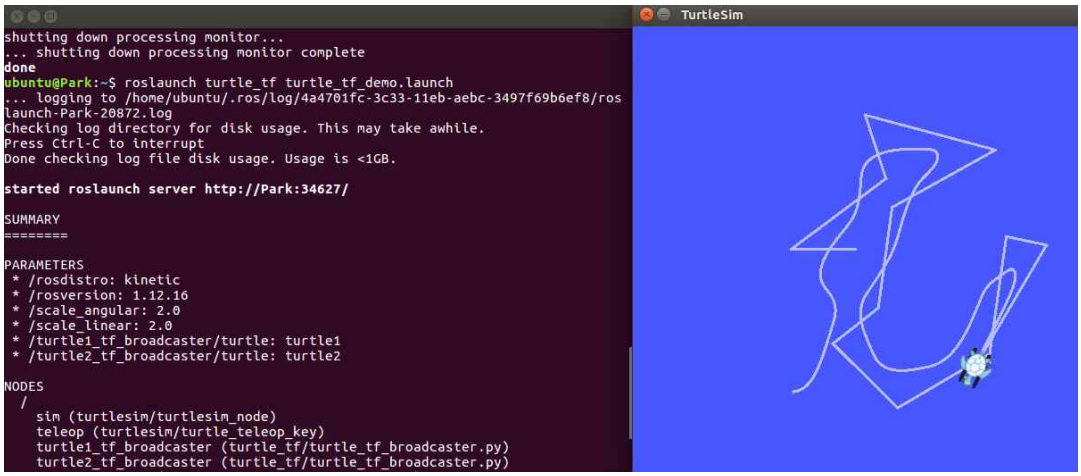
1. tf

1) tf - turtle demo

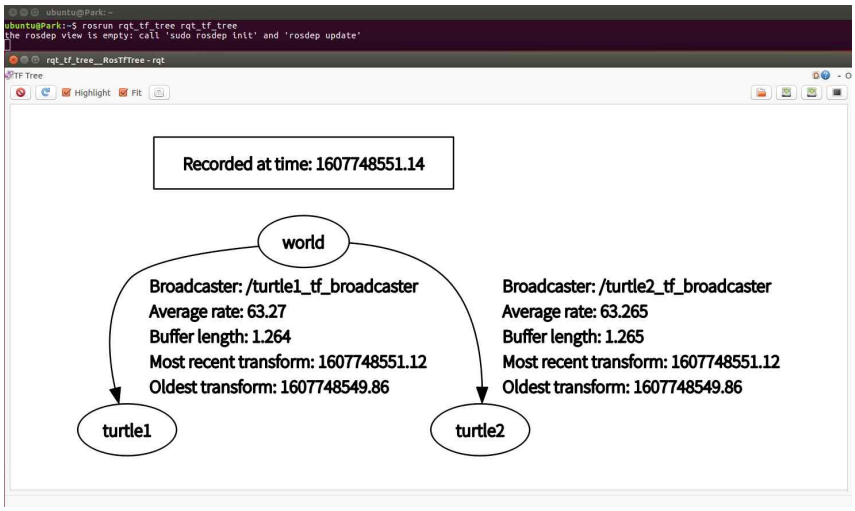
- Dependency 설치



- Turtle_tf demo 실행



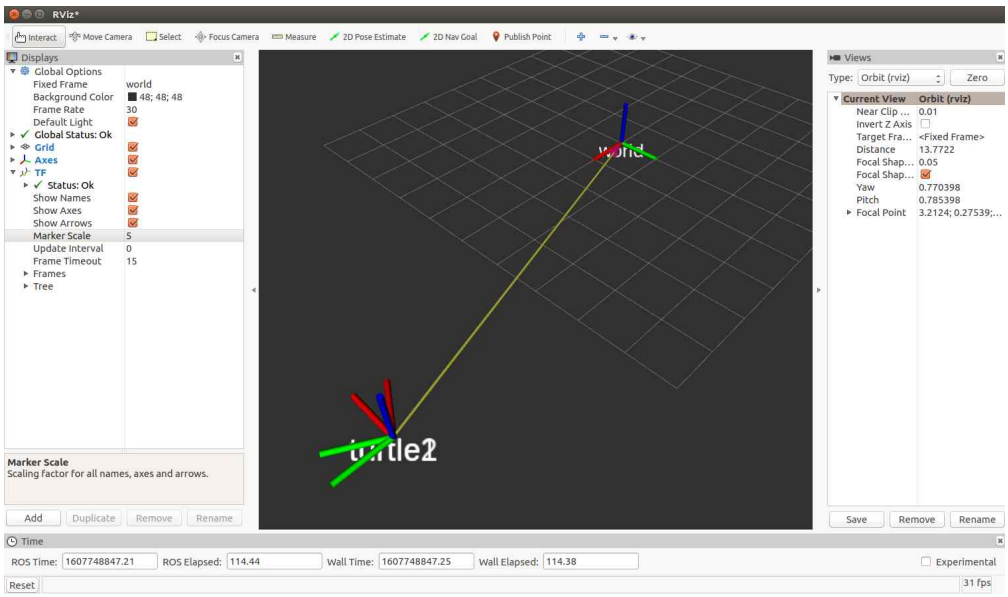
2) tf - rqt_tf_tree



3) tf - tf echo

```
ubuntu@Park: ~
ubuntu@Park:~$ rosrun tf tf_echo turtle1 turtle2
At time 1607748629.154
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.263, 0.965]
            in RPY (radian) [0.000, 0.000, 0.532]
            in RPY (degree) [0.000, 0.000, 30.457]
At time 1607748629.890
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.263, 0.965]
            in RPY (radian) [0.000, 0.000, 0.532]
            in RPY (degree) [0.000, 0.000, 30.457]
At time 1607748630.881
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.263, 0.965]
            in RPY (radian) [0.000, 0.000, 0.532]
            in RPY (degree) [0.000, 0.000, 30.457]
At time 1607748631.890
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.263, 0.965]
            in RPY (radian) [0.000, 0.000, 0.532]
            in RPY (degree) [0.000, 0.000, 30.457]
At time 1607748632.882
- Translation: [0.000, 0.000, 0.000]
- Rotation: in Quaternion [0.000, 0.000, 0.263, 0.965]
```

4) tf - rviz



5) tf - broadcaster 예제

- 새로운 패키지를 생성

```
ubuntu@Park: ~/test_ws/src
ubuntu@Park:~$ cd test_ws/src
ubuntu@Park:~/test_ws/src$ catkin_create_pkg learning_tf tf roscpp rospy turtlesim
Created file learning_tf/package.xml
Created file learning_tf/CMakeLists.txt
Created folder learning_tf/include/learning_tf
Created folder learning_tf/src
Successfully created files in /home/ubuntu/test_ws/src/learning_tf. Please adjust the values in package.xml.
ubuntu@Park:~/test_ws/src$
```

- 새로운 패키지를 빌드

```
ubuntu@Park: ~/test_ws
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/ubuntu/test_ws/build"
####
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
Scanning dependencies of target custommsg_subnode
[  0%] Built target std_msgs_generate_messages_py
Scanning dependencies of target testsrv_client
Scanning dependencies of target testsrv_server
Scanning dependencies of target paramtestnode
[  0%] Built target std_msgs_generate_messages_lisp
[  0%] Built target std_msgs_generate_messages_cpp
[  0%] Built target std_msgs_generate_messages_eus
[  0%] Built target std_msgs_generate_messages_nodejs
[  0%] Built target _testpkg_generate_messages_check_deps_testsrv
[  0%] Built target _testpkg_generate_messages_check_deps_testmsg
Scanning dependencies of target testnode
```

```
ubuntu@Park: ~/test_ws
[ 31%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/testnode
[ 34%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/paramtestnode
[ 37%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/testsrv_client
[ 41%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/talkernode
[ 44%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/testsrv_server
[ 48%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/custommsgnode
[ 51%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/listenernode
[ 55%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/custommsg_subnode
[ 55%] Built target paramtestnode
[ 55%] Built target talkernode
[ 55%] Built target custommsgnode
[ 55%] Built target testnode
[ 55%] Built target testsrv_client
[ 55%] Built target testsrv_server
[ 55%] Built target listenernode
[ 55%] Built target custommsg_subnode
[ 75%] Built target testpkg_generate_messages_nodejs
[ 75%] Built target testpkg_generate_messages_lisp
[ 82%] Built target testpkg_generate_messages_py
[ 93%] Built target testpkg_generate_messages_eus
[100%] Built target testpkg_generate_messages_cpp
[100%] Built target testpkg_generate_messages
ubuntu@Park:~/test_ws$ source ./devel/setup.bash
ubuntu@Park:~/test_ws$
```

- tf_broadcaster.cpp 생성

```
tf_broadcaster.cpp (~/test_ws/src/learning_tf/src) - gedit
열기(O)  저장(S)
#include <ros/ros.h>
#include <tf/transform_broadcaster.h>
#include <turtlesim/Pose.h>
std::string turtle_name;
void poseCallback(const turtlesim::PoseConstPtr& msg){
  static tf::TransformBroadcaster br;
  tf::Transform transform;
  transform.setOrigin(tf::Vector3(msg->x, msg->y, 0.0));
  tf::Quaternion q;
  q.setRPY(0, 0, msg->theta);
  transform.setRotation(q);
  br.sendTransform(tf::StampedTransform(transform, ros::Time::now(), "world", turtle_name));
}
int main(int argc, char** argv){
  ros::init(argc, argv, "my_tf_broadcaster");
  if (argc != 2){ROS_ERROR("need turtle name as argument");
return -1;};
  turtle_name = argv[1];
  ros::NodeHandle node;
  ros::Subscriber sub = node.subscribe(turtle_name+"/pose", 10, &poseCallback);
  ros::spin();
  return 0;
};
'/home/ubuntu/test_ws/src/learning_tf/src/tf_broadcaster.cpp' 파일을 읽어들이는 중입니다...  C++  탭 너비: 8  8행, 57열  삽입
```

- CMakeList에 add_executable, target_link_library를 추가 후 빌드

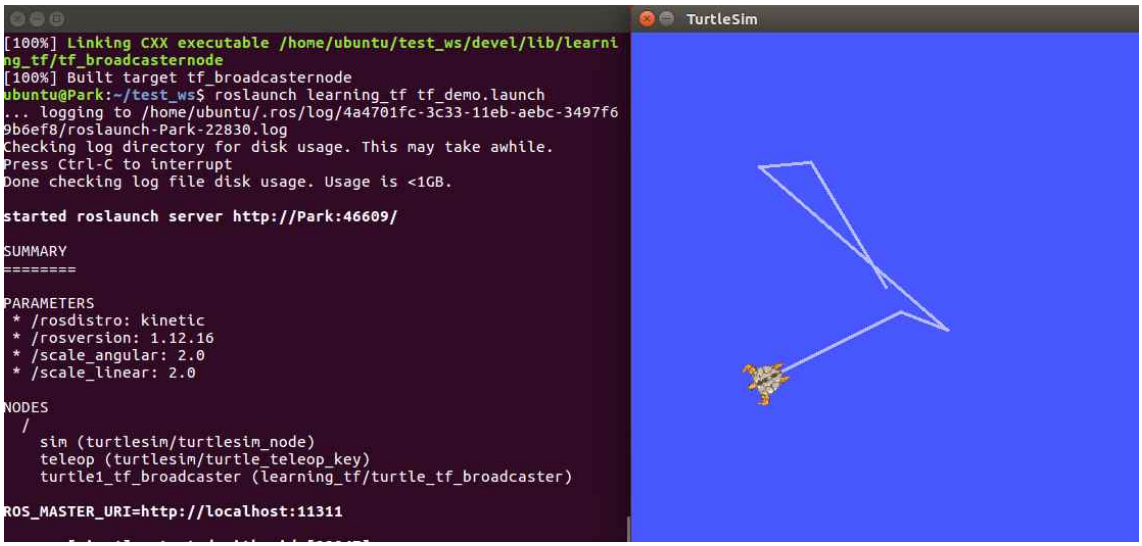
```
CMakeLists.txt (~/test_ws/src/learning_tf) - gedit
열기(O)  저장(S)
# catkin_add_nosetests(test)

add_executable(tf_broadcasternode src/tf_broadcaster.cpp)
target_link_libraries(tf_broadcasternode ${catkin_LIBRARIES})
CMake  탭 너비: 8  210행, 62열  삽입
```

```
ubuntu@Park: ~/test_ws
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/ubuntu/test_ws/build"
####
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
[  0%] Built target std_msgs_generate_messages_lisp
[  0%] Built target std_msgs_generate_messages_py
[  6%] Built target testsrv_client
[ 12%] Built target paramtestnode
[ 19%] Built target custommsg_subnode
[ 25%] Built target testsrv_server
[ 25%] Built target std_msgs_generate_messages_cpp
[ 25%] Built target _testpkg_generate_messages_check_deps_testsrv
[ 25%] Built target _testpkg_generate_messages_check_deps_testmsg
[ 25%] Built target std_msgs_generate_messages_nodejs
```

- learning_tf/launch에 tf_demo.launch를 생성 후 실행

```
*tf_demo.launch (~/test_ws/src/learning_tf) - gedit
열기(O)  저장(S)
<launch>
  <!-- Turtlesim Node-->
  <node pkg="turtlesim" type="turtlesim_node" name="sim"/>
  <node pkg="turtlesim" type="turtle_teleop_key" name="teleop" output="screen"/>
  <!-- Axes-->
  <param name="scale_linear" value="2" type="double"/>
  <param name="scale_angular" value="2" type="double"/>
  <node pkg="learning_tf" type="turtle_tf_broadcaster" args="/turtle1" name="turtle1_tf_broadcaster"/>
</launch>
일반 텍스트  탭 너비: 8  9행, 10열  삽입
```


[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/learning_tf/tf_broadcasternode
[100%] Built target tf_broadcasternode
ubuntu@Park:~/test_ws$ roslaunch learning_tf tf_demo.launch
... logging to /home/ubuntu/.ros/log/4a4701fc-3c33-11eb-aebc-3497f69b6ef8/roslaunch-Park-22830.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://Park:46609/
SUMMARY
PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.16
* /scale_angular: 2.0
* /scale_linear: 2.0
NODES
/
sim (turtlesim/turtlesim_node)
teleop (turtlesim/turtle_teleop_key)
turtle1_tf_broadcaster (learning_tf/turtle_tf_broadcaster)
ROS_MASTER_URI=http://localhost:11311
TurtleSim