古月居 GYH.AI ROS

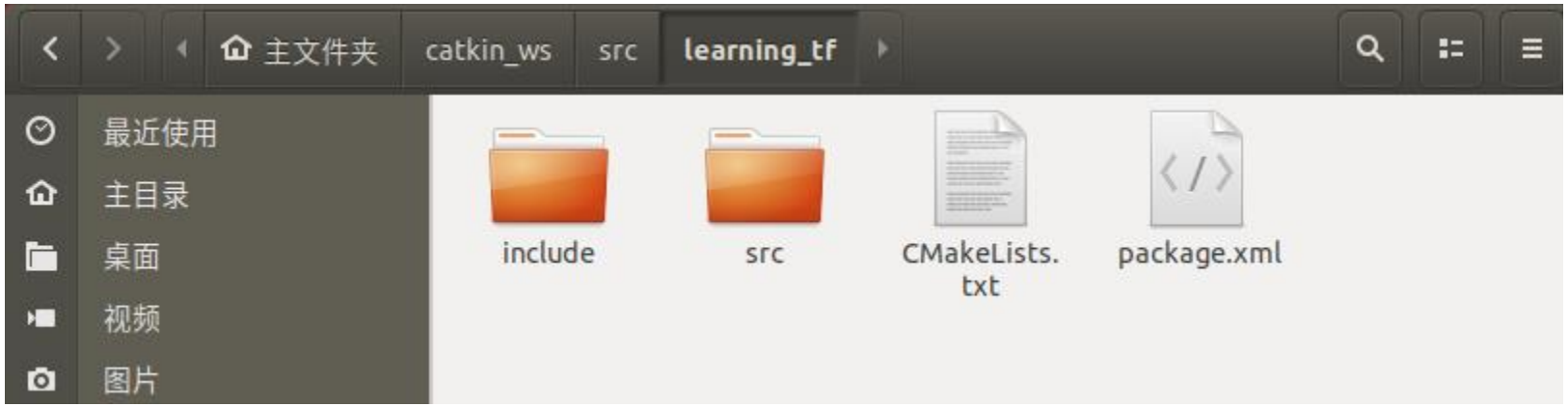
ROS入门
21讲

18.tf坐标系广播与监听的编程实现

主讲人：古月

创建功能包

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg learning_tf roscpp rospy tf turtlesim
```



创建tf广播器代码（C++）

```cpp
/**
 * 该例程产生tf数据，并计算、发布turtle2的速度指令
 */

#include <ros/ros.h>
#include <tf/transform_broadcaster.h>
#include <turtlesim/Pose.h>

std::string turtle_name;

void poseCallback(const turtlesim::PoseConstPtr& msg)
{
    // 创建tf的广播器
    static tf::TransformBroadcaster br;

    // 初始化tf数据
    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);

    // 广播world与海龟坐标系之间的tf数据
    br.sendTransform(tf::StampedTransform(transform, ros::Time::now(), "world", turtle_name));
}

int main(int argc, char** argv)
{
    // 初始化ROS节点
    ros::init(argc, argv, "my_tf_broadcaster");

    // 输入参数作为海龟的名字
    if (argc != 2)
    {
        ROS_ERROR("need turtle name as argument");
        return -1;
    }

    turtle_name = argv[1];

    // 订阅海龟的位姿话题
    ros::NodeHandle node;
    ros::Subscriber sub = node.subscribe(turtle_name+"/pose", 10, &poseCallback);

    // 循环等待回调函数
    ros::spin();

    return 0;
};
```

turtle_tf_broadcaster.cpp

如何实现一个tf广播器

- 定义TF广播器（TransformBroadcaster）
- 创建坐标变换值；
- 发布坐标变换（sendTransform）

创建tf监听器代码（C++）

```cpp
int main(int argc, char** argv)
{
    // 初始化ROS节点
    ros::init(argc, argv, "my_tf_listener");

    // 创建节点句柄
    ros::NodeHandle node;

    // 请求产生turtle2
    ros::service::waitForService("/spawn");
    ros::ServiceClient add_turtle = node.serviceClient<turtlesim::Spawn>("/spawn");
    turtlesim::Spawn srv;
    add_turtle.call(srv);

    // 创建发布turtle2速度控制指令的发布者
    ros::Publisher turtle_vel = node.advertise<geometry_msgs::Twist>("/turtle2/cmd_vel", 10);

    // 创建tf的监听器
    tf::TransformListener listener;

    ros::Rate rate(10.0);
    while (node.ok())
    {
        // 获取turtle1与turtle2坐标系之间的tf数据
        tf::StampedTransform transform;
        try
        {
            listener.waitForTransform("/turtle2", "/turtle1", ros::Time(0), ros::Duration(3.0));
            listener.lookupTransform("/turtle2", "/turtle1", ros::Time(0), transform);
        }
        catch (tf::TransformException &ex)
        {
            ROS_ERROR("%s",ex.what());
            ros::Duration(1.0).sleep();
            continue;
        }

        // 根据turtle1与turtle2坐标系之间的位置关系，发布turtle2的速度控制指令
        geometry_msgs::Twist vel_msg;
        vel_msg.angular.z = 4.0 * atan2(transform.getOrigin().y(),
                                        transform.getOrigin().x());
        vel_msg.linear.x = 0.5 * sqrt(pow(transform.getOrigin().x(), 2) +
                                      pow(transform.getOrigin().y(), 2));
        turtle_vel.publish(vel_msg);

        rate.sleep();
    }
    return 0;
};
```

turtle_tf_listener.cpp

如何实现一个TF监听器

- 定义TF监听器；

（TransformListener）

- 查找坐标变换；

（waitForTransform、 lookupTransform）

配置tf广播器与监听器代码编译规则

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

add_executable(turtle_tf_broadcaster src/turtle_tf_broadcaster.cpp)
target_link_libraries(turtle_tf_broadcaster ${catkin_LIBRARIES})

add_executable(turtle_tf_listener src/turtle_tf_listener.cpp)
target_link_libraries(turtle_tf_listener ${catkin_LIBRARIES})
```

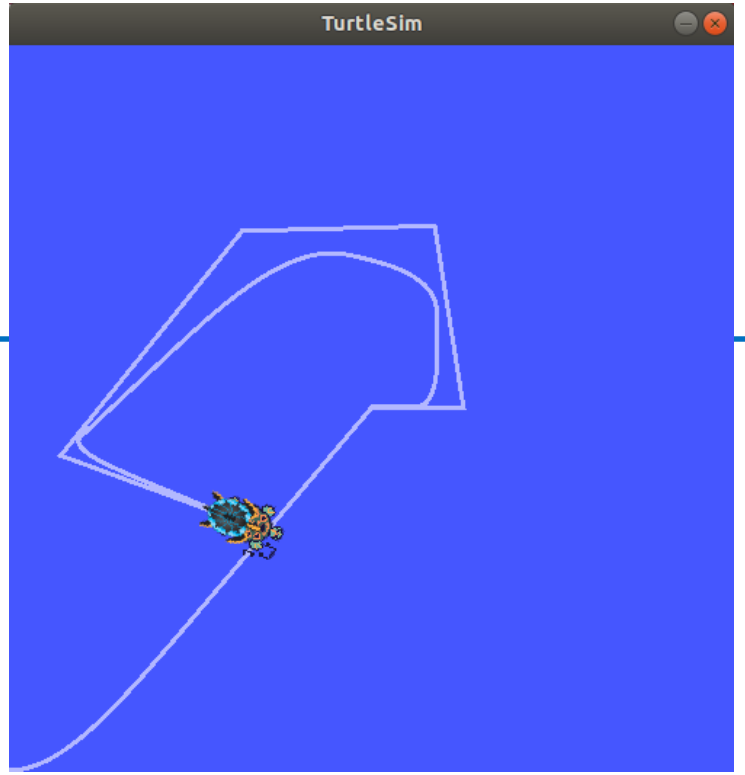
如何配置CMakeLists.txt中的编译规则

- 设置需要编译的代码和生成的可执行文件；
- 设置链接库；

```
add_executable(turtle_tf_broadcaster src/turtle_tf_broadcaster.cpp)
target_link_libraries(turtle_tf_broadcaster ${catkin_LIBRARIES})

add_executable(turtle_tf_listener src/turtle_tf_listener.cpp)
target_link_libraries(turtle_tf_listener ${catkin_LIBRARIES})
```

CMakeLists.txt

编译并运行




```
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
$ roscore
$ rosrun turtlesim turtlesim_node
$ rosrun learning_tf turtle_tf_broadcaster __name:=turtle1_tf_broadcaster /turtle1
$ rosrun learning_tf turtle_tf_broadcaster __name:=turtle2_tf_broadcaster /turtle2
$ rosrun learning_tf turtle_tf_listener
$ rosrun turtlesim turtle_teleop_key
```

创建tf广播器与监听器代码（Python）

古月居 GYH.AI

turtle_tf_broadcaster.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将请求/show_person服务，服务数据类型learning_service::Person

import roslib
roslib.load_manifest('learning_tf')
import rospy

import tf
import turtlesim.msg

def handle_turtle_pose(msg, turtlename):
    br = tf.TransformBroadcaster()
    br.sendTransform((msg.x, msg.y, 0),
                     tf.transformations.quaternion_from_euler(0, 0, msg.theta),
                     rospy.Time.now(),
                     turtlename,
                     "world")

if __name__ == '__main__':
    rospy.init_node('turtle_tf_broadcaster')
    turtlename = rospy.get_param('~turtle')
    rospy.Subscriber('/%s/pose' % turtlename,
                     turtlesim.msg.Pose,
                     handle_turtle_pose,
                     turtlename)
    rospy.spin()
```

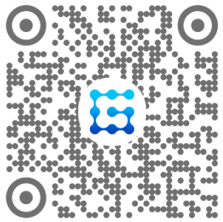
turtle_tf_listener.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将请求/show_person服务，服务数据类型learning_service::Person

import roslib
roslib.load_manifest('learning_tf')
import rospy
import math
import tf
import geometry_msgs.msg
import turtlesim.srv

if __name__ == '__main__':
    rospy.init_node('turtle_tf_listener')

    listener = tf.TransformListener()

    rospy.wait_for_service('spawn')
    spawner = rospy.ServiceProxy('spawn', turtlesim.srv.Spawn)
    spawner(4, 2, 0, 'turtle2')

    turtle_vel = rospy.Publisher('turtle2/cmd_vel', geometry_msgs.msg.Twist,queue_size=1)

    rate = rospy.Rate(10.0)
    while not rospy.is_shutdown():
        try:
            (trans,rot) = listener.lookupTransform('/turtle2', '/turtle1', rospy.Time(0))
        except (tf.LookupException, tf.ConnectivityException, tf.ExtrapolationException):
            continue

        angular = 4 * math.atan2(trans[1], trans[0])
        linear = 0.5 * math.sqrt(trans[0] ** 2 + trans[1] ** 2)
        cmd = geometry_msgs.msg.Twist()
        cmd.linear.x = linear
        cmd.angular.z = angular
        turtle_vel.publish(cmd)

        rate.sleep()
```

感谢观看

怕什么真理无穷，进一寸有一寸的欢喜

更多精彩，欢迎关注



古月居

古月学院

古月居GYH