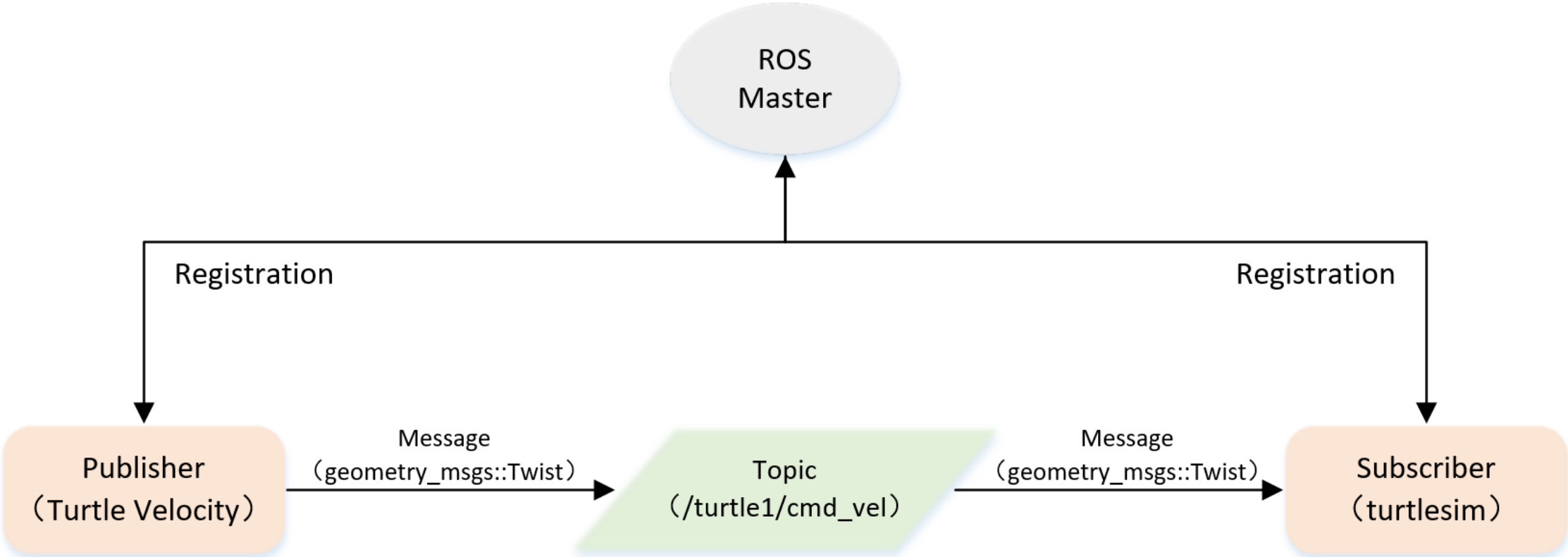
古月居 GYH.AI ROS

ROS入门
21讲

10.发布者Publisher的编程实现

主讲人：古月

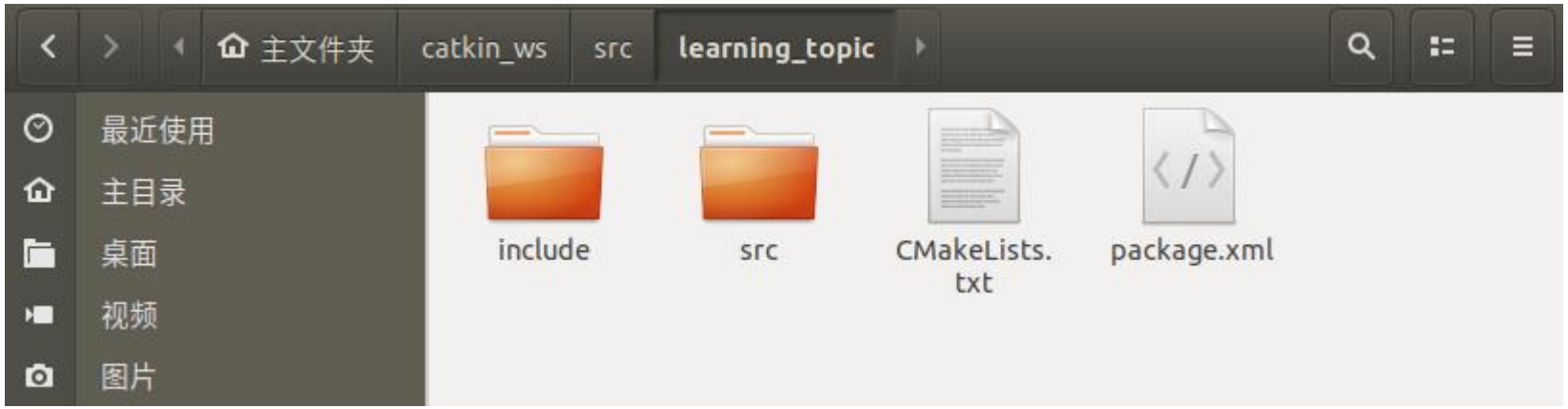
话题模型



话题模型（发布/订阅）

创建功能包

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg learning_topic roscpp rospy std_msgs geometry_msgs turtlesim
```



创建发布者代码（C++）

```cpp
/**
 * 该例程将发布turtle1/cmd_vel话题，消息类型geometry_msgs::Twist
 */

#include <ros/ros.h>
#include <geometry_msgs/Twist.h>

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

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

        // 创建一个Publisher，发布名为/turtle1/cmd_vel的topic，消息类型为geometry_msgs::Twist，队列长度10
        ros::Publisher turtle_vel_pub = n.advertise<geometry_msgs::Twist>("/turtle1/cmd_vel", 10);

        // 设置循环的频率
        ros::Rate loop_rate(10);

        int count = 0;
        while (ros::ok())
        {
            // 初始化geometry_msgs::Twist类型的消息
                geometry_msgs::Twist vel_msg;
                vel_msg.linear.x = 0.5;
                vel_msg.angular.z = 0.2;

            // 发布消息
                turtle_vel_pub.publish(vel_msg);
                ROS_INFO("Publsh turtle velocity command[%0.2f m/s, %0.2f rad/s]",
                                vel_msg.linear.x, vel_msg.angular.z);

            // 按照循环频率延时
            loop_rate.sleep();
        }

        return 0;
}
```

velocity_publisher.cpp

如何实现一个发布者

- 初始化ROS节点；
- 向ROS Master注册节点信息，包括发布的话题名和话题中的消息类型；
- 创建消息数据；
- 按照一定频率循环发布消息。

配置发布者代码编译规则

```
## 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/learning_topic_node.cpp)

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

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

如何配置CMakeLists.txt中的编译规则

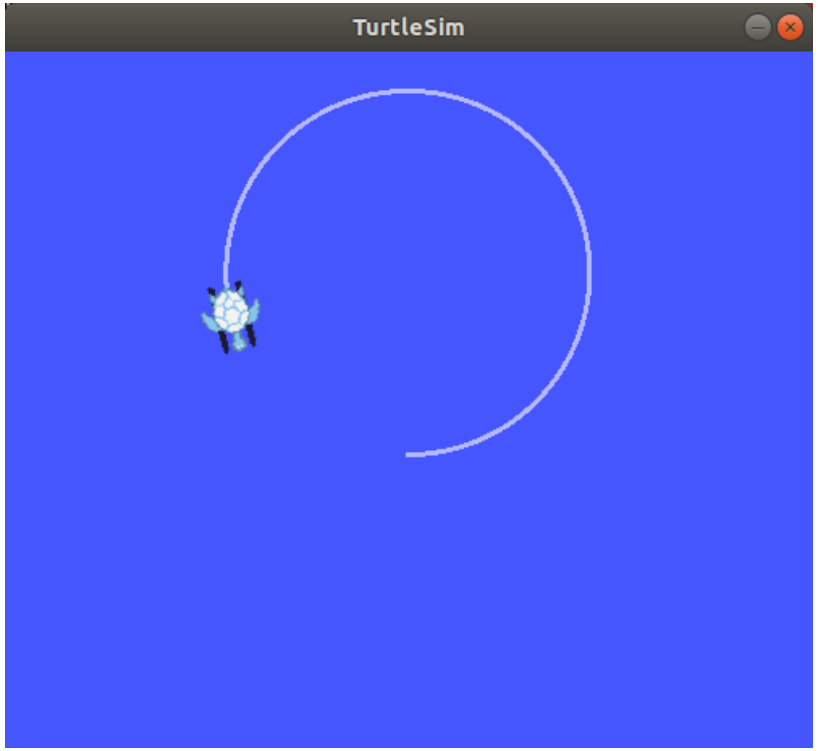
- 设置需要编译的代码和生成的可执行文件；
- 设置链接库；

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

CMakeLists.txt

编译并运行发布者

```
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
$ roscore
$ rosrun turtlesim turtlesim_node
$ rosrun learning_topic velocity_publisher
```




```
hcx@hcx-vpc:~/catkin_ws$ rosrun learning_topic velocity_publisher
[ INFO] [1562209882.139849161]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.233707855]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.333979045]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.434357523]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.533607735]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.633799235]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.733964404]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.834193121]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209882.933629197]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
[ INFO] [1562209883.034284663]: Publsh turtle velocity command[0.50 m/s, 0.20 rad/s]
```

创建发布者代码（Python）

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将发布turtle1/cmd_vel话题，消息类型geometry_msgs::Twist

import rospy
from geometry_msgs.msg import Twist

def velocity_publisher():
    # ROS节点初始化
    rospy.init_node('velocity_publisher', anonymous=True)

    # 创建一个Publisher，发布名为/turtle1/cmd_vel的topic，消息类型为geometry_msgs::Twist，队列长度10
    turtle_vel_pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)

    #设置循环的频率
    rate = rospy.Rate(10)

    while not rospy.is_shutdown():
        # 初始化geometry_msgs::Twist类型的消息
        vel_msg = Twist()
        vel_msg.linear.x = 0.5
        vel_msg.angular.z = 0.2

        # 发布消息
        turtle_vel_pub.publish(vel_msg)
        rospy.loginfo("Publsh turtle velocity command[%0.2f m/s, %0.2f rad/s]",
                vel_msg.linear.x, vel_msg.angular.z)

        # 按照循环频率延时
        rate.sleep()

if __name__ == '__main__':
    try:
        velocity_publisher()
    except rospy.ROSInterruptException:
        pass
```

velocity_publisher.py

如何实现一个发布者

- 初始化ROS节点；
- 向ROS Master注册节点信息，包括发布的话题名和话题中的消息类型；
- 创建消息数据；
- 按照一定频率循环发布消息。

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

古月春旭

bilibili 古月居GYH

ROS入门
21讲