

2022年11月19日 星期六 16:50

创建订阅者代码 (C++)

```

1 // 使用std::string的typedef类型, 消息类型为Pose
2 #include <string>
3 #include <iostream>
4 #include "std_msgs/Pose.h"
5
6 // 回调函数: 接收消息, 打印位置
7 void callback(const std_msgs::PoseConstPtr& msg)
8 {
9     // 输出收到的消息内容
10     std::cout << "收到消息: (" << msg->x << ", " << msg->y << ", " << msg->z << ")" << std::endl;
11 }
12
13 int main(int argc, char** argv)
14 {
15     // 初始化ROS
16     ros::init(argc, argv, "pose_subscriber");
17
18     // 创建订阅者
19     ros::Subscriber pose_subscriber = ros::Subscriber::withCallback(callback);
20
21     // 阻塞主线程
22     ros::spin();
23 }

```

如何实现一个订阅者

- 初始化ROS节点;
- 订阅需要的话题;
- 循环等待话题消息, 接收回调消息后进入回调函数;

在回调函数中完成消息处理。

pose\_subscriber.cpp

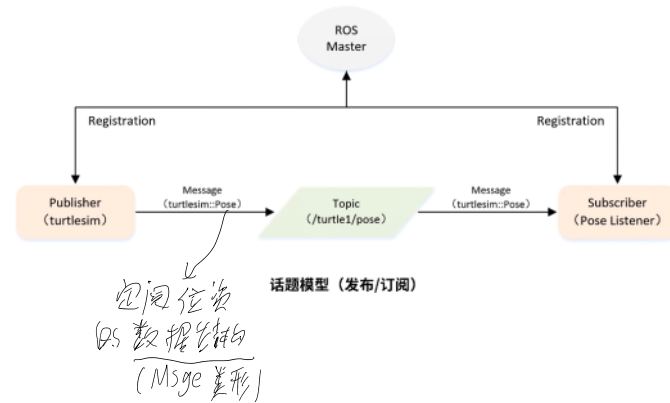
```
work space/src/ package
```

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

```

c:\xchex-ipc>cat nss_wss_rsnr
learning topic pos subscriber
INFO [1562215557.322225871]: Turtle pose: x16.3890005, y16.3960278
INFO [1562215557.339997278]: Turtle pose: x16.381475, y16.398730
INFO [1562215557.345421218]: Turtle pose: x16.379338, y16.401410
INFO [1562215557.351045158]: Turtle pose: x16.377201, y16.404090
INFO [1562215557.387085434]: Turtle pose: x16.388384, y16.406965
INFO [1562215557.482710847]: Turtle pose: x16.351273, y16.409363
INFO [1562215557.48886939]: Turtle pose: x16.343701, y16.411885
INFO [1562215557.434496988]: Turtle pose: x16.336121, y16.414443
INFO [1562215557.440120938]: Turtle pose: x16.328541, y16.416967
INFO [1562215557.465994943]: Turtle pose: x16.326931, y16.419487
INFO [1562215557.46213454]: Turtle pose: x16.313333, y16.421971

```



二. 仿 Sim 包  
 NT  
 功能包      显示 RT 位置  
 posrun      learning\_topic      pose-subscriber  
 ↓  
 NT  
 posrun turtleSim turtle\_teleop-keg