5.Subscribers

5.1 Subscribers

The subscriber receives the data published by the publisher and then enters its callback function, where the received data is processed. The core content is the callback function, which is available for each topic subscribed to by subscribers.

5.2Create a subscriber

5.2.1 Creation steps

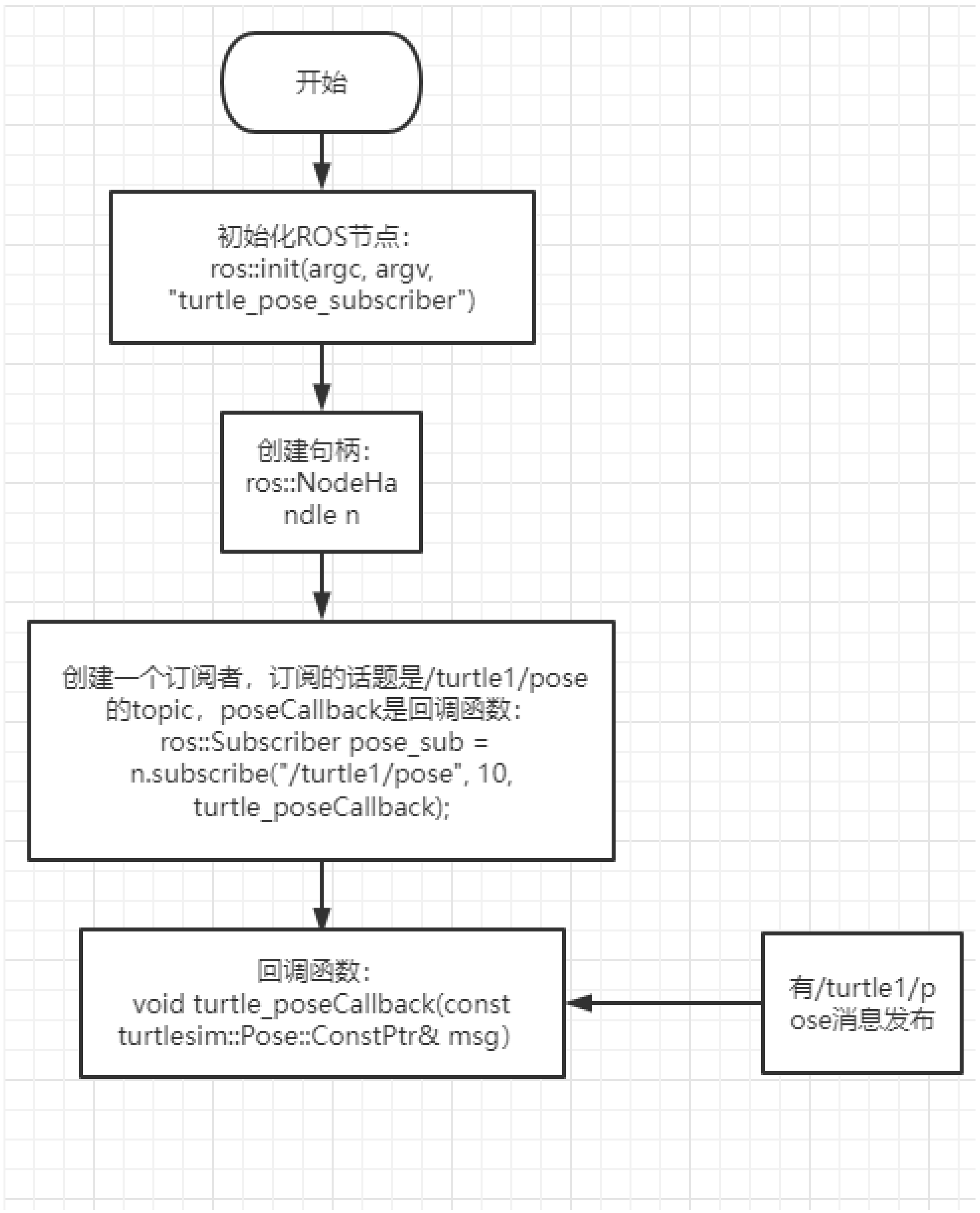
1. Initialize ROS node
2. Create handle
3. Subscribe to the required
4. topicsWait for topic messages in a loop, and upon receiving the message, enter the callback function
5. Complete message processing in the callback function.

5.2.2. C++Language Implementation

1. In the "Publisher" tutorial, create a new C++file under the src folder of the created feature pack and name it turtle_ pose_ subscriber.cpp
2. Copy and paste the program code below into the title_ pose_ In the subscriber.cpp file

```
#include <ros/ros.h>
#include "turtlesim/Pose.h"

void turtle_poseCallback(const turtlesim::Pose::ConstPtr& msg){

    ROS_INFO("Turtle pose: x:%0.3f, y:%0.3f", msg->x, msg->y);
}

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

    ros::init(argc, argv, "turtle_pose_subscriber");

    ros::NodeHandle n;


    ros::Subscriber pose_sub = n.subscribe("/turtle1/pose", 10, 
turtle_poseCallback);

    ros::spin(); // 循环等待回调函数

    return 0;
}
```

3. Program flowchart, which can be viewed according to 5.2.1 content


开始
初始化ROS节点：
ros::init(argc, argv, "turtle_pose_subscriber")
创建句柄：
ros::NodeHandle n
创建一个订阅者，订阅的话题是/turtle1/pose的topic，poseCallback是回调函数：
ros::Subscriber pose_sub = n.subscribe("/turtle1/pose", 10, turtle_poseCallback);
回调函数：
void turtle_poseCallback(const turtlesim::Pose::ConstPtr& msg)
有/turtle1/pose消息发布

4. Configure in CMakelist.txt, under the build area, add the following content

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

5. Compiling code under workspace directory

```
cd ~/catkin_ws
catkin_make
source devel/setup.bash
```

6. Run program

- Run score

```
roscore
```

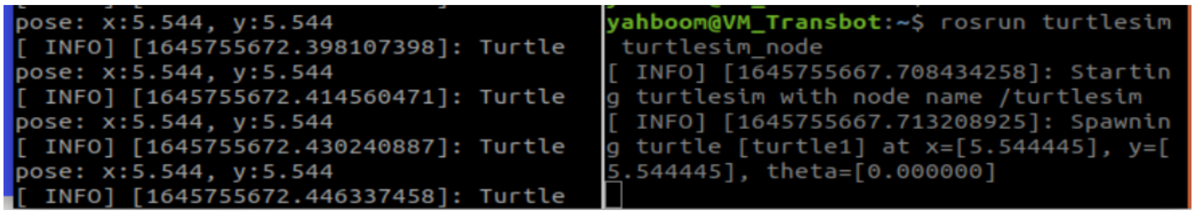
- Running the Little Turtle Node

```
rosrun turtlesim turtlesim_node
```

- Run subscription and continuously receive posture data sent by the little turtle

```
rosrun learning_topic turtle_pose_subscriber
```

7. Operation screenshot



```
pose: x:5.544, y:5.544
[ INFO] [1645755672.398107398]: Turtle
pose: x:5.544, y:5.544
[ INFO] [1645755672.414560471]: Turtle
pose: x:5.544, y:5.544
[ INFO] [1645755672.430240887]: Turtle
pose: x:5.544, y:5.544
[ INFO] [1645755672.446337458]: Turtle
```

```
yahboom@VM_Transbot:~$ rosrun turtlesim
 turtlesim_node
[ INFO] [1645755667.708434258]: Startin
g turtlesim with node name /turtlesim
[ INFO] [1645755667.713208925]: Spawnin
g turtle [turtle1] at x=[5.544445], y=[
5.544445], theta=[0.000000]
```

8. Program operation instructions

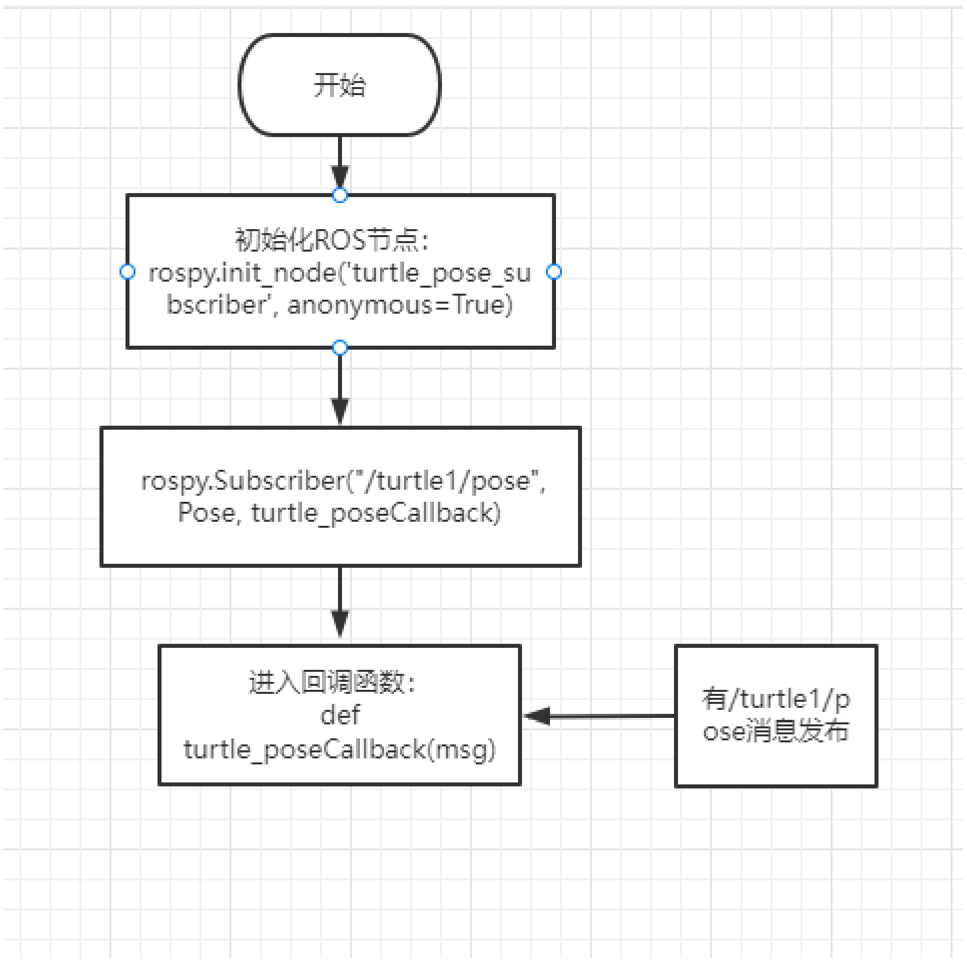
- After running the little turtle's node, the little turtle will continuously send its own pose information, and the topic name sent is/turtle1/pose
- And turnle_ pose_ After the subscriber runs, it will receive the data message sent by the turtle and print out this information in the callback function

5.2.3. Python Language Implementation

1. In the feature pack directory, create a new folder called scripts, and then create a new Python file (with the suffix. py) under the scripts folder, named 'turn'_ pose_ subscriber.py
2. Copy and paste the program code below into the title_ pose_ In the subscriber.py file

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import rospy
from turtlesim.msg import Pose

def poseCallback(msg):
    rospy.loginfo("Turtle pose: x:%0.3f, y:%0.3f", msg.x, msg.y)

def turtle_pose_subscriber():

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


    rospy.Subscriber("/turtle1/pose", Pose, poseCallback)


    rospy.spin()# 循环等待回调函数

if __name__ == '__main__':
    turtle_pose_subscriber()
```

3. Process Flow Chart


开始
初始化ROS节点：
rospy.init_node('turtle_pose_subscriber', anonymous=True)
rospy.Subscriber("/turtle1/pose", Pose, turtle_poseCallback)
进入回调函数：
def turtle_poseCallback(msg)
有/turtle1/pose消息发布

4.run a program

- Run score

```
roscore
```

- Running the Little Turtle Node

```
rosrun turtlesim turtlesim_node
```

- Run subscription and continuously receive posture data sent by the little turtle

```
rosrun learning_topic turtle_pose_subscriber
```

5.Refer to 5.2.2 for operational effects and program instructions.