

7.ROS topic publisher

As mentioned in the previous section, a node program has both publishing and subscribing messages. In this section, we will explain how to declare a publisher in the node program and publish topic messages. We proceed based on the previously established workspace `ros_ws` and `learn_topic` function packages.

7.1 Create a publisher

General creation steps are as follows:

- Initialize ROS nodes
- Create handle
- Register node information with ROS Master, including the published topic name, message type in the topic, and queue length
- Create and initialize message data
- Send messages cyclically with a certain frequency

7.2 C++ version

7.2.1 Writing source code

In the `src` folder of the function package `learn_topic`, create a C++ file (the file suffix is `.cpp`), name it `turtle_velocity_publisher.cpp`, and paste the following content into `turtle_velocity_publisher.cpp`.

```
/*Create a small turtle speed publisher*/
#include <ros/ros.h>
#include <geometry_msgs/Twist.h>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "turtle_velocity_publisher");//ROS node initialization
    ros::NodeHandle n;//Here is create handle
    //Create a Publisher, publish a topic named /turtle1/cmd_vel, the message
    type is geometry_msgs::Twist, and the queue length is 10
    ros::Publisher turtle_vel_pub = n.advertise<geometry_msgs::Twist>
    ("/turtle1/cmd_vel", 10);
    ros::Rate loop_rate(10);//Set the frequency of the loop
    while (ros::ok())
    {
        //Initialize the message to be published, the type must be consistent
        with Publisher
        geometry_msgs::Twist turtle_vel_msg;
        turtle_vel_msg.linear.x = 0.8;
        turtle_vel_msg.angular.z = 0.6;
        turtle_vel_pub.publish(turtle_vel_msg);// Publish speed news
        //Print published speed content
        ROS_INFO("Publsh turtle velocity command[%0.2f m/s, %0.2f rad/s]",
        turtle_vel_msg.linear.x, turtle_vel_msg.angular.z);
        loop_rate.sleep();//Delay according to cycle frequency
    }
    return 0;
}
```

7.2.2 Modify CMakeList.txt file

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

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

add_executable shows that the generated executable program file is turtle_velocity_publisher, and the compiled source code is turtle_velocity_publisher.cpp in the src directory.

target_link_libraries specifies the libraries that need to be linked when compiling and generating an executable file.

7.2.3 Compile

Terminal input,

```
cd ~/ros_ws
catkin_make
```



```
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Found gmock sources under '/usr/src/gmock': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.8.10")
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~ traversing 1 packages in topological order:
-- ~~~ - learn_topic
-- ~~~ processing catkin package: 'learn_topic'
-- ==> add_subdirectory(learn_topic)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/yahboom/ros_ws/build
####
#### Running command: "make -j4 -l4" in "/home/yahboom/ros_ws/build"
####
Scanning dependencies of target turtle_velocity_publisher
[ 50%] Building CXX object learn_topic/CMakeFiles/turtle_velocity_publisher.dir/src/turtle_velocity_publisher.cpp.o
[100%] Linking CXX executable /home/yahboom/ros_ws/devel/lib/learn_topic/turtle_velocity_publisher
[100%] Built target turtle_velocity_publisher
yahboom@yahboom-virtual-machine:~/ros_ws$
```

After the compilation is passed, you need to re-source the current environment variables to find or update the program. Enter in the terminal.

```
cd ~/ros_ws
source devel/setup.bash
```

7.2.4 Running the program

Open roscore,

```
roscore
```

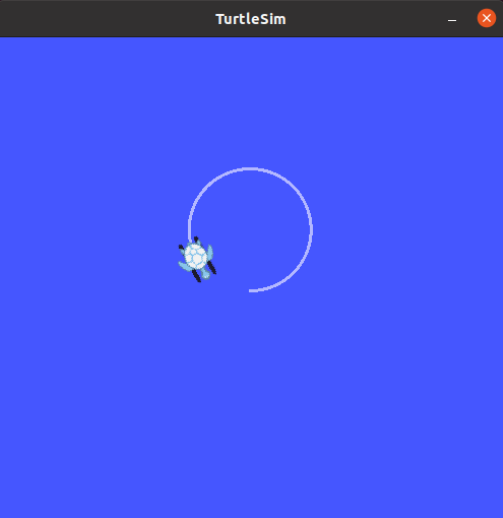
Run the little turtle node program,

```
roslaunch turtlesim turtlesim_node
```

Run the publisher node program and continue to send speed to the little turtle.

```
roslaunch learn_topic turtle_velocity_publisher
```

```
yahboom@yahboom-virtual-machine:~/ros_ws$ rosrn learn_topic turtle_velocity_publisher
[INFO] [1698046957.816703307]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046957.917566391]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.016812029]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.117630093]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.217381996]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.317062602]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.417188114]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.516812278]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.617437320]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.717833977]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.817813754]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046958.917787904]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.017210273]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.117143148]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.216844756]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.317006740]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.417129303]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.517108764]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.617350924]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.717637459]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.817160281]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046959.917528129]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.016916007]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.116963436]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.217457588]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.316811390]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.417118362]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.516943855]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.617111732]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.716866366]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.816945668]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046960.917062329]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
[INFO] [1698046961.016816934]: Publish turtle velocity command[0.80 m/s, 0.60 rad/s]
```



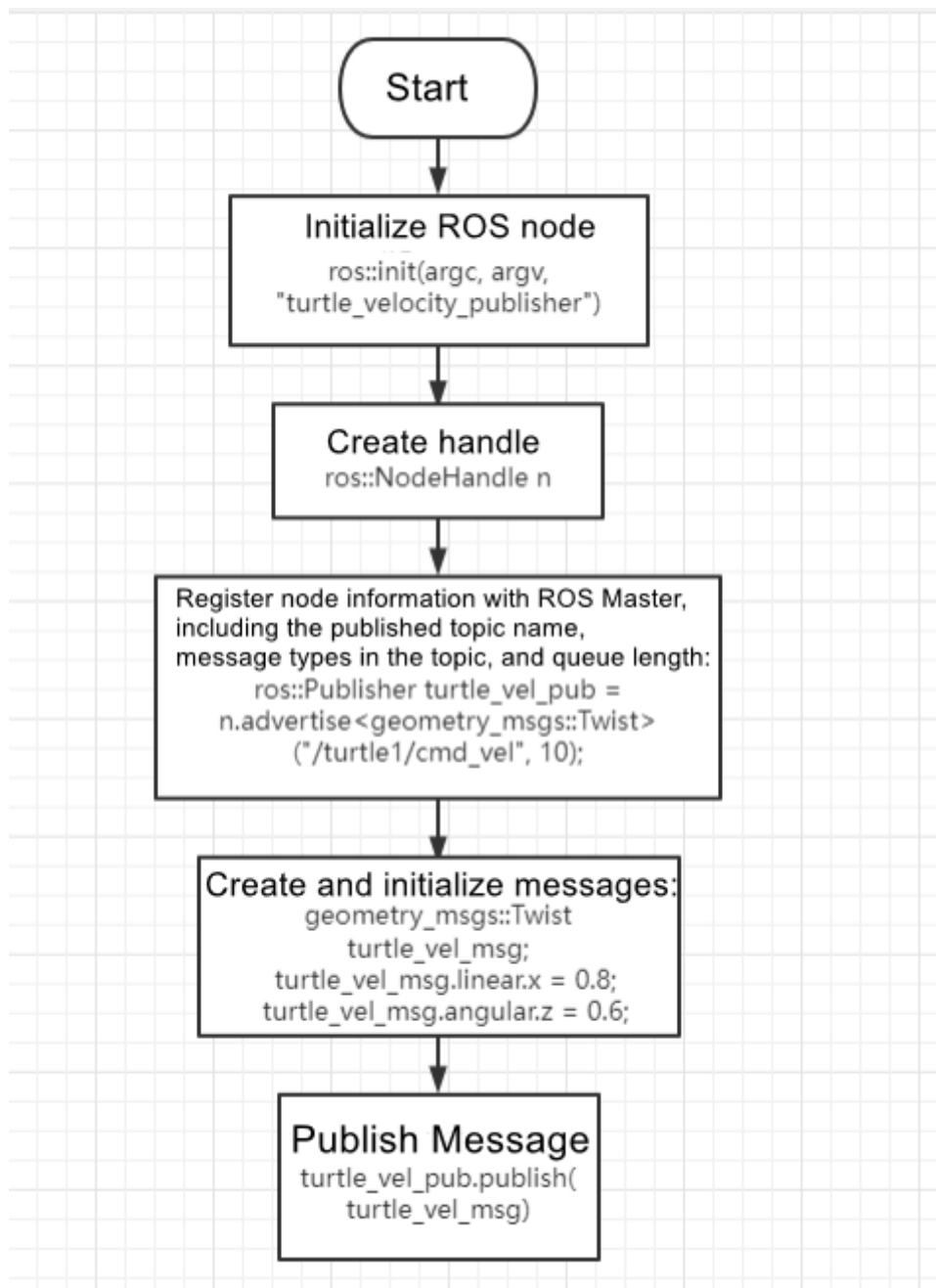
As shown in the figure above, after receiving the published message, the little turtle will move at the specified speed. We can check which nodes are running through `rostopic list` and enter in the terminal,

```
rostopic list

yahboom@yahboom-virtual-machine:~/ros_ws$ rostopic list
/rosout
/turtle_velocity_publisher
/turtlesim
```

`/turtle_velocity_publisher` is the program we write, compile, and run. The node name here is consistent with `ros::init(argc, argv, "turtle_velocity_publisher")` in the code.

7.2.5 Program flow chart



7.3 Python version

7.3.1 Writing source code

Create a new python file (file suffix .py) in the scripts folder under the function package learn_topic, name it `turtle_velocity_publisher.py`, copy and paste the following program code into the `turtle_velocity_publisher.py` file,

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# This routine will publish to the turtle1/cmd_vel topic, message type
# geometry_msgs::Twist
import rospy
from geometry_msgs.msg import Twist
def turtle_velocity_publisher():
    rospy.init_node('turtle_velocity_publisher', anonymous=True) # ROS node
    # initialization
```

```

# Create a small turtle speed publisher and publish a topic named
/turtle1/cmd_vel. The message type is geometry_msgs::Twist, and 8 represents the
message queue length.
turtle_vel_pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=8)
rate = rospy.Rate(10) #Set the frequency of the loop
while not rospy.is_shutdown():
    # Initialize geometry_msgs::Twist type message
    turtle_vel_msg = Twist()
    turtle_vel_msg.linear.x = 0.8
    turtle_vel_msg.angular.z = 0.6
    # release the news
    turtle_vel_pub.publish(turtle_vel_msg)
    rospy.loginfo("linear is :%0.2f m/s, angular is :%0.2f rad/s",
    turtle_vel_msg.linear.x, turtle_vel_msg.angular.z)
    rate.sleep()# Delay according to cycle frequency
if __name__ == '__main__':
    try:
        turtle_velocity_publisher()
    except rospy.ROSInterruptException:
        pass

```

The python program does not need to be compiled, but it needs to add executable permissions and enter it in the terminal.

```

cd ~/ros_ws/src/learn_topic/scripts
sudo chmod a+x turtle_velocity_publisher.py

```

7.3.2 Run

Open roscore,

```
roscore
```

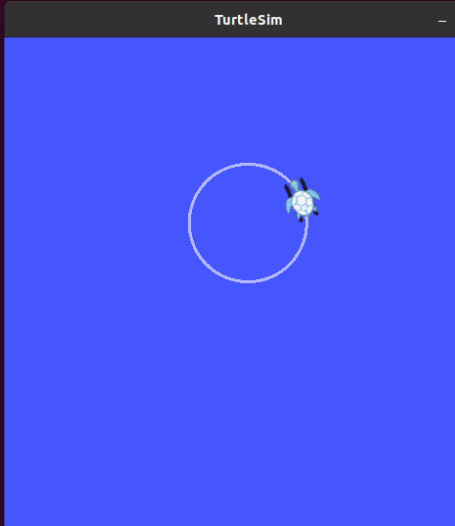
Run the little turtle node,

```
roslaunch turtlesim turtlesim_node
```

Run the publisher node program and continue to send speed to the little turtle.

```
roslaunch learn_topic turtle_velocity_publisher.py
```

```
yahboom@yahboom-virtual-machine:~/ros_ws$ roslaunch learn_topic turtle_velocity_publisher.py
[INFO] [1698048074.521173]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048074.622458]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048074.722430]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048074.823043]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048074.922365]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.022650]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.122369]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.222820]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.322470]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.422261]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.521658]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.622267]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.721954]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.821788]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048075.922228]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.021847]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.122250]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.221647]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.321924]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.422476]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.522333]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.622248]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.721722]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.821695]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048076.922414]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.022253]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.122383]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.221824]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.322050]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.422328]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.521906]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.621662]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.722392]: linear is :0.80 m/s, angular is :0.60 rad/s
[INFO] [1698048077.821953]: linear is :0.80 m/s, angular is :0.60 rad/s
```



7.3.3 Program flow chart

