

# 暨南大学本科实验报告专用纸

课程名称 机器人操作系统理论与实践 指导教师 李德平 成绩     

实验项目名称 通过动作实现机器人画圆 实验项目编号 03

实验项目实验地点 验证 学院 智能科学与工程学院 专业 人工智能

学生姓名 王志涛 学号 2021102259 实验时间 2023 年 10 月 31 日

## 一、实验目的

- 1.1 熟练掌握建立工作空间、建立功能包的命令
- 1.2 熟练掌握编译功能包、建立节点的方式
- 1.3 实现一个机器人画圆的动作

## 二、实验环境

2.1 Linux 虚拟机，版本为 Ubuntu22.0.4

## 三、实验内容

3.1 假设我们有一个机器人，我们希望通过动作的通信方法，让机器人转个圈，请编程实现动作通信中，客户端和服务端的实现过程

## 四、实验及分析

### 4.1 建立工作空间

```
wzt@wzt-vpc:~/ros_experience$ mkdir -p ~/my_ws_action/src
wzt@wzt-vpc:~/ros_experience$
```

### 4.2 建立功能包

#### ● 建立源代码功能包

```
wzt@wzt-vpc:~/ros_experience/my_ws_action/src$ ros2 pkg create --build-type ament_python learning_action
going to create a new package
package name: learning_action
destination directory: /home/wzt/ros_experience/my_ws_action/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['wzt <wzt@todo.todo>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: []
creating folder ./learning_action
creating ./learning_action/package.xml
creating source folder
creating folder ./learning_action/learning_action
creating ./learning_action/setup.py
creating ./learning_action/setup.cfg
creating folder ./learning_action/resource
```

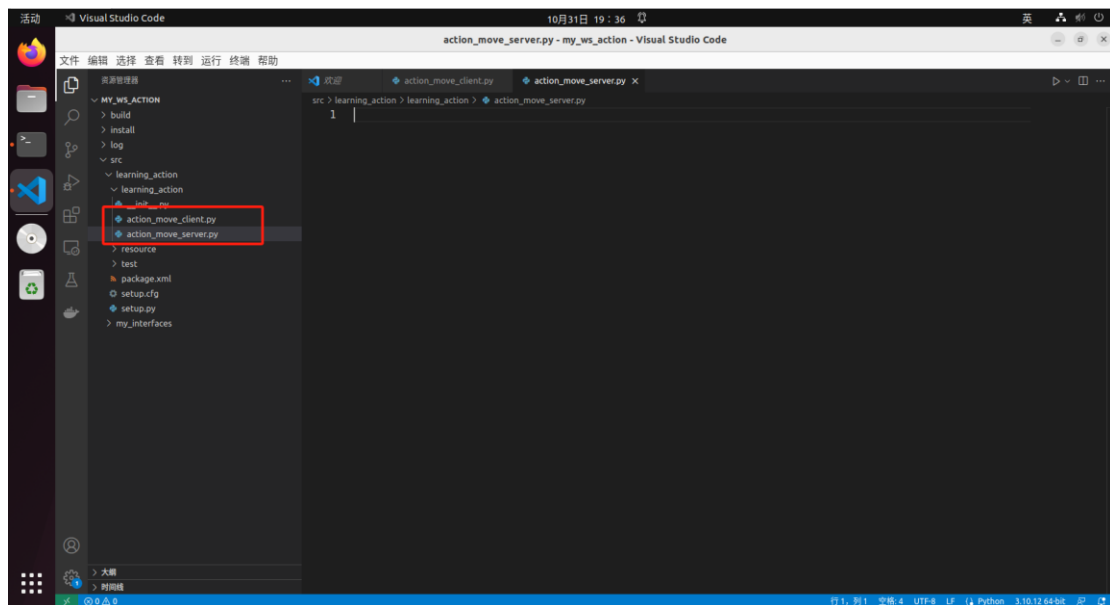
#### ● 建立接口功能包

```
wzt@wzt-vpc:~/ros_experience/my_ws_action/src$ ros2 pkg create --build-type ament_cmake my_interfaces
going to create a new package
package name: my_interfaces
destination directory: /home/wzt/ros_experience/my_ws_action/src
package format: 3
version: 0.0.0
description: TODO: Package description
```

#### 4.3 添加节点和接口文件

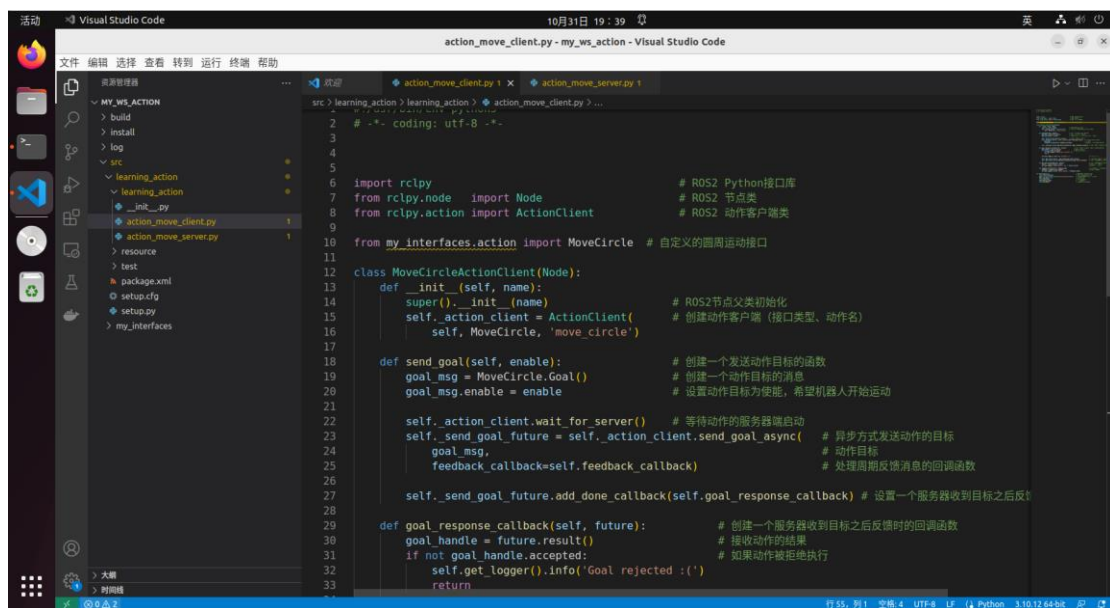
- 用 vscode 打开 my\_ws\_action 文件夹:

在 learning\_action 文件夹下添加两个文件, action\_move\_client(客户端), action\_move\_server(服务端)

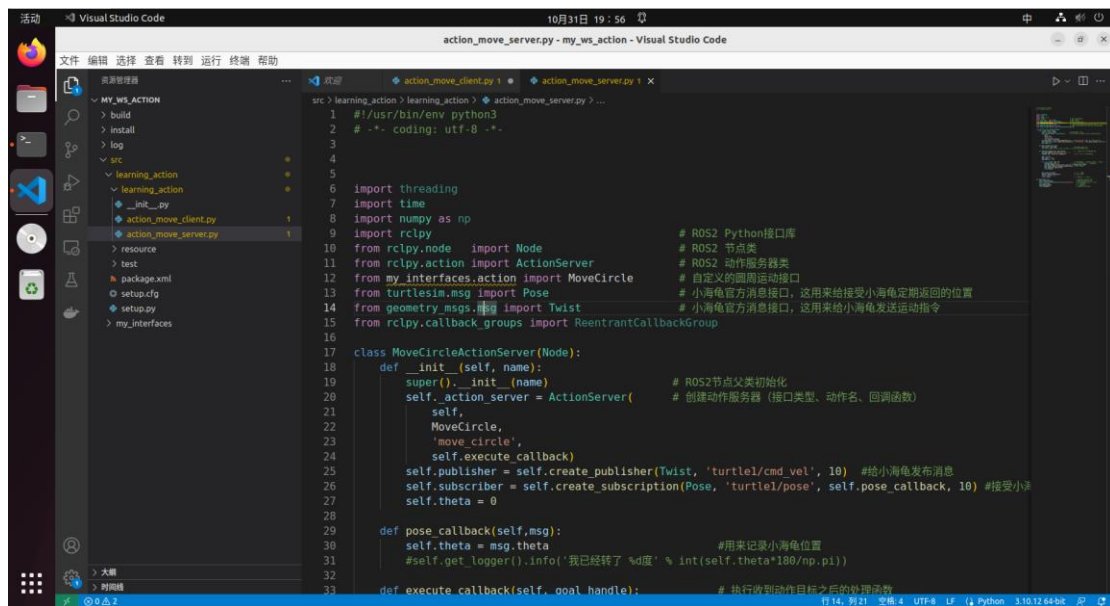


添加代码:

action\_move\_client(客户端):

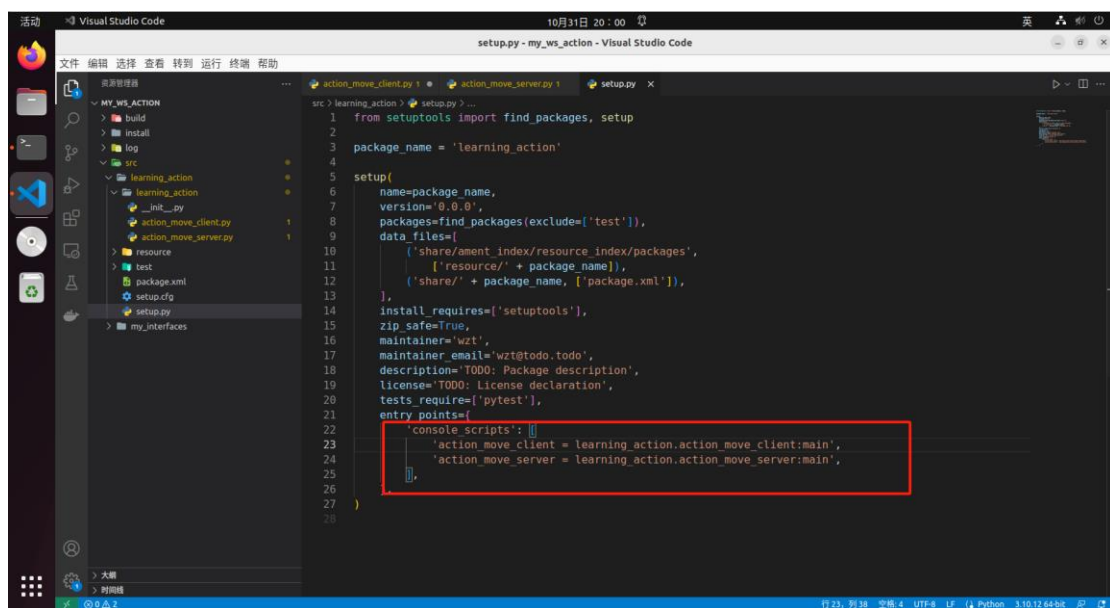


## action\_move\_server(服务端):



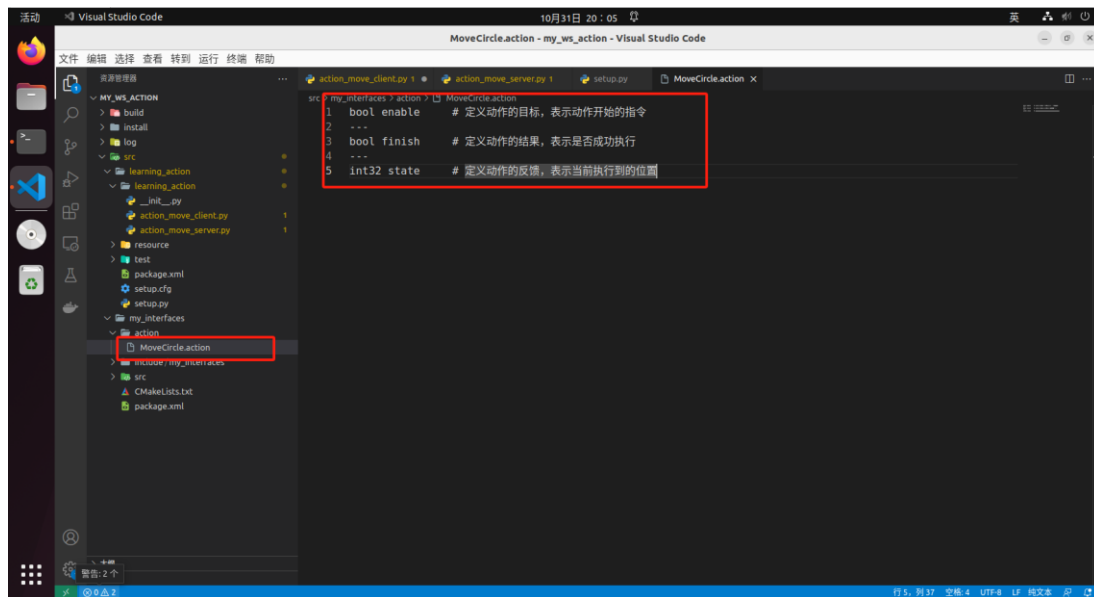
```
src > learning_action > learning_action > action_move_server.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5
6  import threading
7  import time
8  import numpy as np
9  import rcipy
10 from rcipy.node import Node
11 from rcipy.action import ActionServer
12 from my_interfaces.action import MoveCircle
13 from turtlesim.msg import Pose
14 from geometry_msgs.msg import Twist
15 from rcipy.callback_groups import ReentrantCallbackGroup
16
17 class MoveCircleActionServer(Node):
18     def __init__(self, name):
19         super().__init__(name)
20         self.action_server = ActionServer(
21             self,
22             MoveCircle,
23             'move_circle',
24             self.execute_callback)
25         self.publisher = self.create_publisher(Twist, 'turtle/cmd_vel', 10)
26         self.subscriber = self.create_subscription(Pose, 'turtle/pose', self.pose_callback, 10)
27         self.theta = 0
28
29     def pose_callback(self, msg):
30         self.theta = msg.theta
31         self.get_logger().info('我已经转了 %d度' % int(self.theta*180/np.pi))
32
33     def execute_callback(self, goal_handle):
```

- 在 setup 文件中设立节点的名称与节点的入口函数

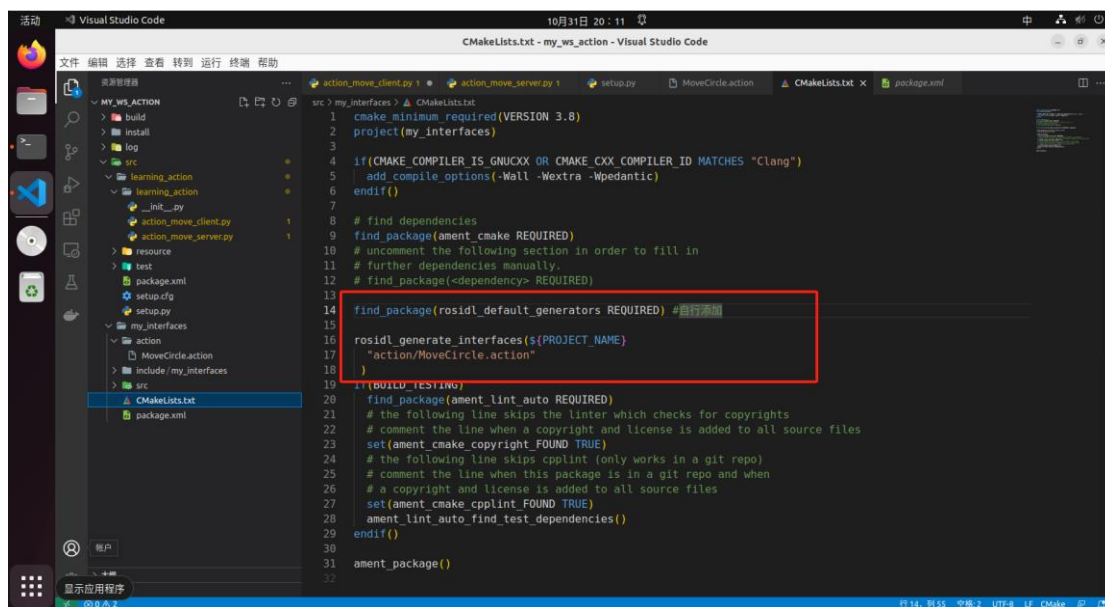


```
src > learning_action > setup.py > ...
1  from setuptools import find_packages, setup
2
3  package_name = 'learning_action'
4
5  setup(
6      name=package_name,
7      version='0.0.0',
8      packages=find_packages(exclude=['test']),
9      data_files=[
10         ('share/ament_index/resource_index/packages',
11          ['resource/' + package_name]),
12         ('share/' + package_name, ['package.xml']),
13     ],
14     install_requires=['setuptools'],
15     zip_safe=True,
16     maintainer='wzt',
17     maintainer_email='wzt@todo.todo',
18     description='TODO: Package description',
19     license='TODO: License declaration',
20     tests_require=['pytest'],
21     entry_points={
22         'console_scripts': [
23             'action_move_client = learning_action.action_move_client:main',
24             'action_move_server = learning_action.action_move_server:main',
25         ],
26     },
27 )
```

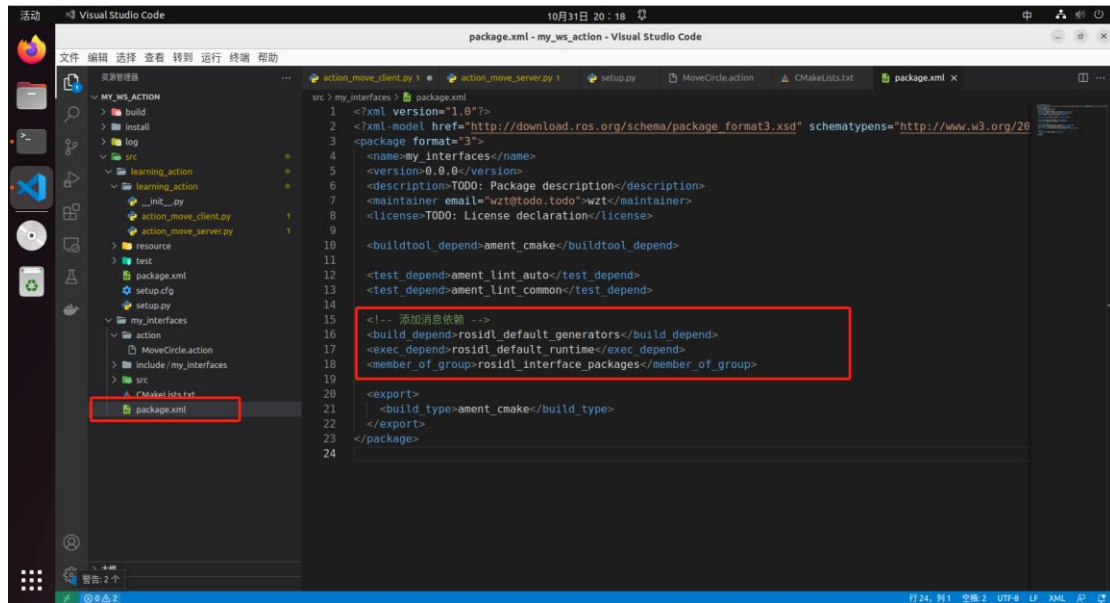
- 打开 my\_interfaces 功能包，在 my\_interfaces 中添加 action 文件夹



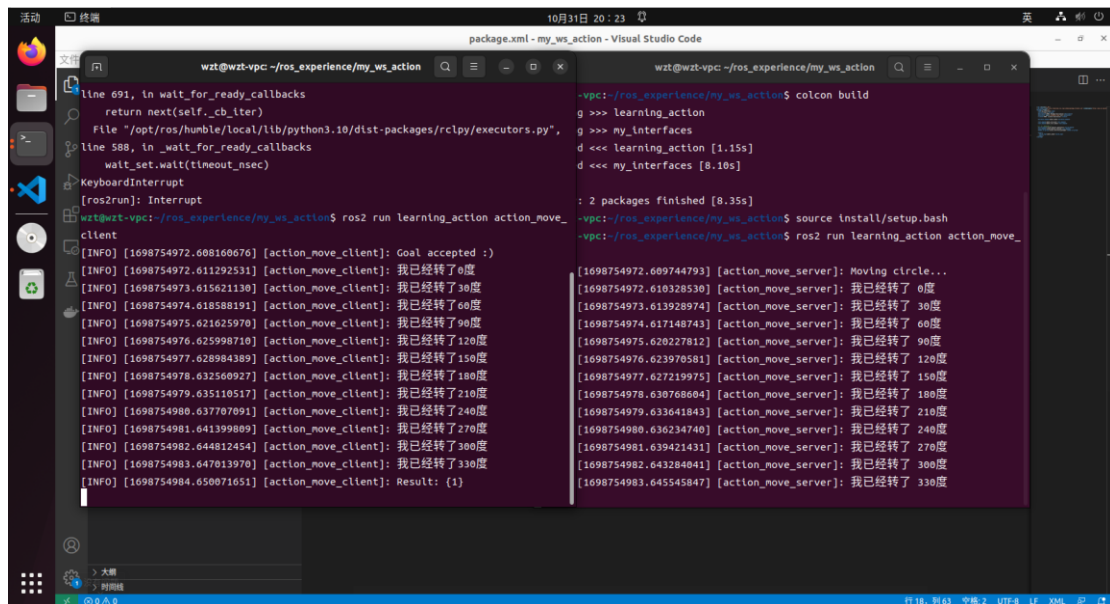
- 在 CMakeLists.txt 文件中添加代码, 让文件夹能获取到动作信息:



- 在 package.xml 文件中添加依赖



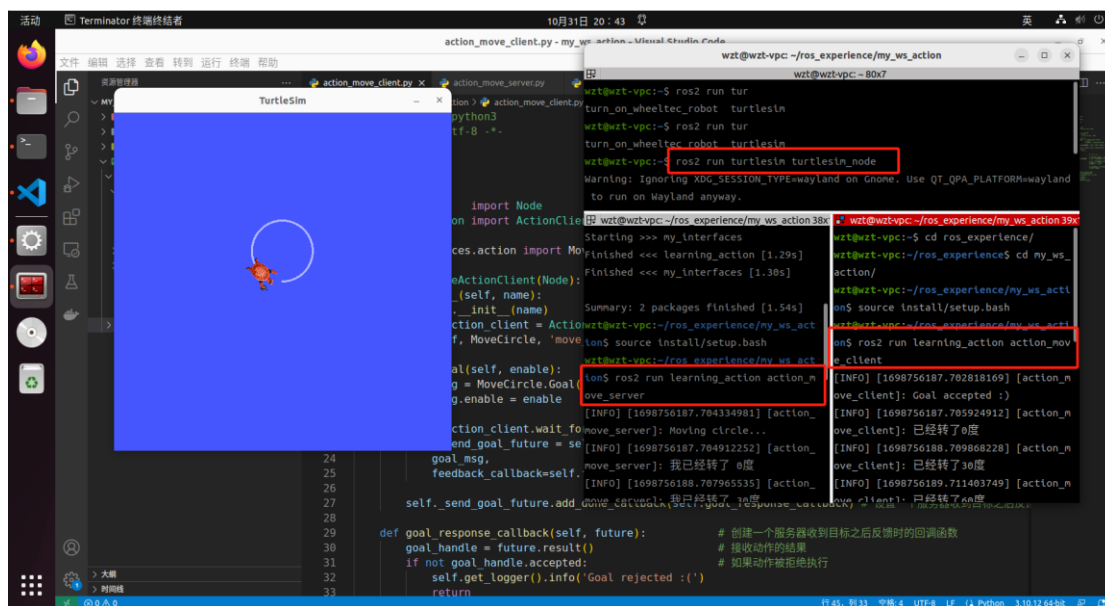
#### 4.4 编译测试:



运行成功。

#### 4.5 运行小海龟:

如图运行三个命令行:



实现了使用机器人画圆，并且周期返回位置信息

## 五、实验总结

动作和服务类似，使用的也是客户端和服务端模型，客户端发送动作的目标，想让机器人干什么，服务端执行动作过程，控制机器人达到运动的目标，同时周期反馈动作执行过程中的状态