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 通过话题实现两个节点间的通信 实验项目编号 01

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 10 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 学会使用话题进行节点间的通信，明白话题通信的特点与用途

二、实验环境

2.1 Linux 虚拟机，版本为 Ubuntu22.0.4

三、实验内容

3.1 通过话题通信实现节点 A 打开摄像头并发布图像数据，节点 B 订阅节点 A 发布的数据并进行图像处理，
实现目标识别、图像灰度化等等，并把处理的结果显示

四、实验及分析

4.1 创建工作空间

```
mkdir -p topic_ws/src
```

4.2 进入工作空间

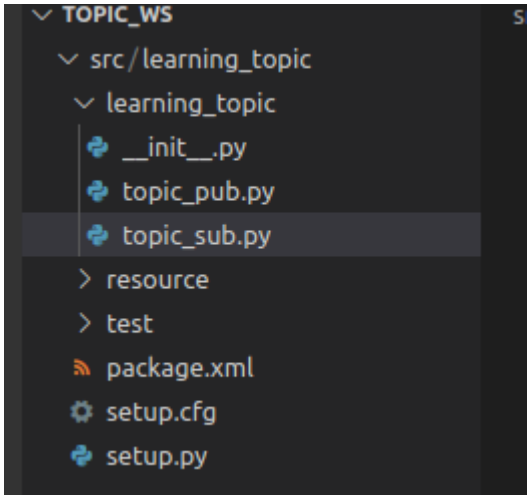
```
cd topic_ws/src
```

4.3 创建 learning_topic 功能包

```
ros2 pkg create learning_topic --build-type ament_python --dependencies rclpy
```

4.4 创建订阅者和发布者

在目录下创建节点文件，如下图所示：



其中发布者代码如下：

```python
import rclpy                        # ROS2 Python接口库
from rclpy.node import Node         # ROS2 节点类
from sensor_msgs.msg import Image   # 图像消息类型
from cv_bridge import CvBridge      # ROS与OpenCV图像转换类
import cv2                          # Opencv图像处理库

"""
创建一个发布者节点
"""
class ImagePublisher(Node):

    def __init__(self, name):
        super().__init__(name)                                          # ROS2节点父类初始化
        self.publisher_ = self.create_publisher(Image, 'image_raw', 10)  # 创建发布者对象（消息类型、话题名、队列长度）
        self.timer = self.create_timer(0.1, self.timer_callback)        # 创建一个定时器（单位为秒的周期，定时执行的回调函数）
        self.cap = cv2.VideoCapture(0)                                  # 创建一个视频采集对象，驱动相机采集图像（相机设备号）
        self.cv_bridge = CvBridge()                                     # 创建一个图像转换对象，用于稍后将OpenCV的图像转换成ROS

    def timer_callback(self):
        ret, frame = self.cap.read()                        # 一帧一帧读取图像

        if ret == True:                                     # 如果图像读取成功
            self.publisher_.publish(
                self.cv_bridge.cv2_to_imgmsg(frame, 'bgr8')) # 发布图像消息

        self.get_logger().info('Publishing video frame')     # 输出日志信息，提示已经完成图像话题发布

def main(args=None):                                # ROS2节点主入口main函数
    rclpy.init(args=args)                           # ROS2 Python接口初始化
    node = ImagePublisher("open_camera_pub")        # 创建ROS2节点对象并进行初始化
    rclpy.spin(node)                                # 循环等待ROS2退出
    node.destroy_node()                             # 销毁节点对象
    rclpy.shutdown()                                # 关闭ROS2 Python接口
```

其中订阅者代码如下：

```python
创建一个订阅者节点
"""
class ImageSubscriber(Node):
    def __init__(self, name):
        super().__init__(name)                                  # ROS2节点父类初始化
        self.sub = self.create_subscription(
            Image, 'image_raw', self.listener_callback, 10)     # 创建订阅者对象（消息类型、话题名、订阅者回调函数、队列长度）
        self.cv_bridge = CvBridge()                             # 创建一个图像转换对象，用于OpenCV图像与ROS的图像消息的互相转换

    def object_detect(self, image):
        hsv_img = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)        # 图像从BGR颜色模型转换为HSV模型
        mask_red = cv2.inRange(hsv_img, lower_red, upper_red)   # 图像二值化
        contours, hierarchy = cv2.findContours(
            mask_red, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)     # 图像中轮廓检测

        for cnt in contours:                                    # 去除一些轮廓面积太小的噪声
            if cnt.shape[0] < 150:
                continue

            (x, y, w, h) = cv2.boundingRect(cnt)                # 得到苹果所在轮廓的左上角xy像素坐标及轮廓范围的宽和高
            cv2.drawContours(image, [cnt], -1, (0, 255, 0), 2)  # 将苹果的轮廓勾勒出来
            cv2.circle(image, (int(x+w/2), int(y+h/2)), 5,
                       (0, 255, 0), -1)                         # 将苹果的图像中心点画出来
        cv2.imshow("object", image)                             # 使用OpenCV显示处理后的图像效果
        cv2.waitKey(10)


    def listener_callback(self, data):
        self.get_logger().info('Receiving video frame')         # 输出日志信息，提示已进入回调函数
        image = self.cv_bridge.imgmsg_to_cv2(data, 'bgr8')      # 将ROS的图像消息转化成OpenCV图像
        self.object_detect(image)                               # 苹果检测
```

在 python 中打入代码后，需要在 setup 上添加下面内容：

```
'console_scripts': [
    'topic_pub = learning_topic.topic_pub:main',
    'topic_sub = learning_topic.topic_sub:main',
]
```

然后下面编译，过程是：

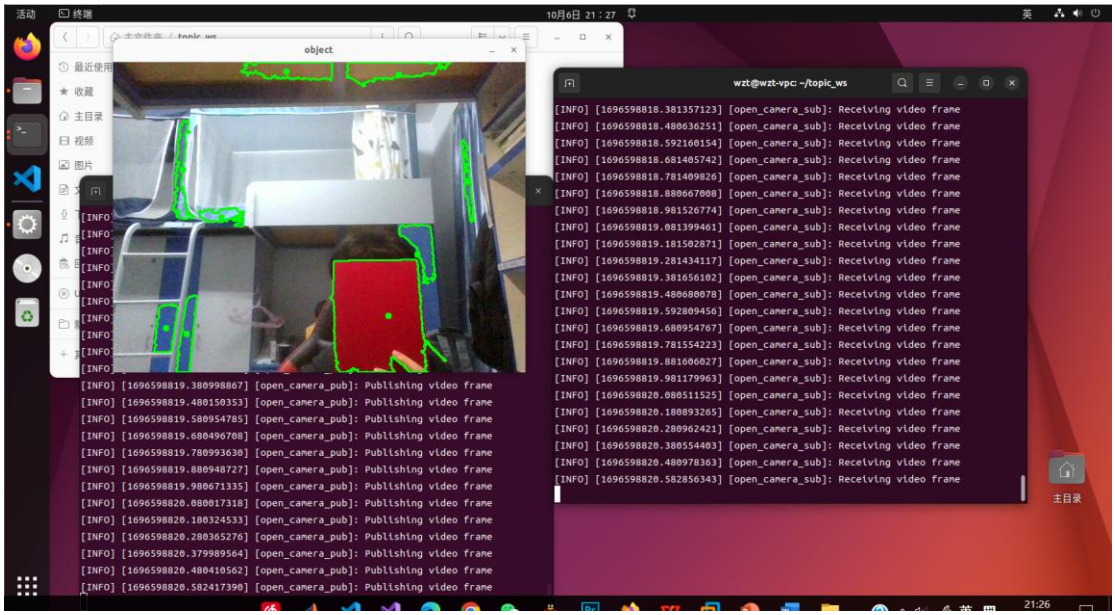
```
(base) wzt@wzt-vpc:~/topic_ws$ colcon build
```

```
(base) wzt@wzt-vpc:~/topic_ws$ source install/setup.bash
```

然后根据代码运行即可：

话题通信	ros2 run learning_topic topic_helloworld_pub ros2 run learning_topic topic_helloworld_sub

4.5 结果展示：



五、实验总结

5.1 如果想要实现一个发布者，那么流程如下：

- 编程接口初始化
- 创建节点对象并初始化
- 创建发布者对象
- 创建并填充话题消息
- 发布话题消息
- 销毁节点并关闭接口

5.2 如果想要实现一个订阅者，那么流程如下：

- 编程接口初始化
- 创建节点并初始化
- 创建订阅者对象
- 回调函数处理话题数据
- 销毁节点并关闭接口