雷达避障

注：虚拟机需要与小车处在同一个局域网下，且ROS_DOMAIN_ID，需要一致，可以查看【使用前必看】来设置板子上的IP和ROS_DOMAIN_ID。

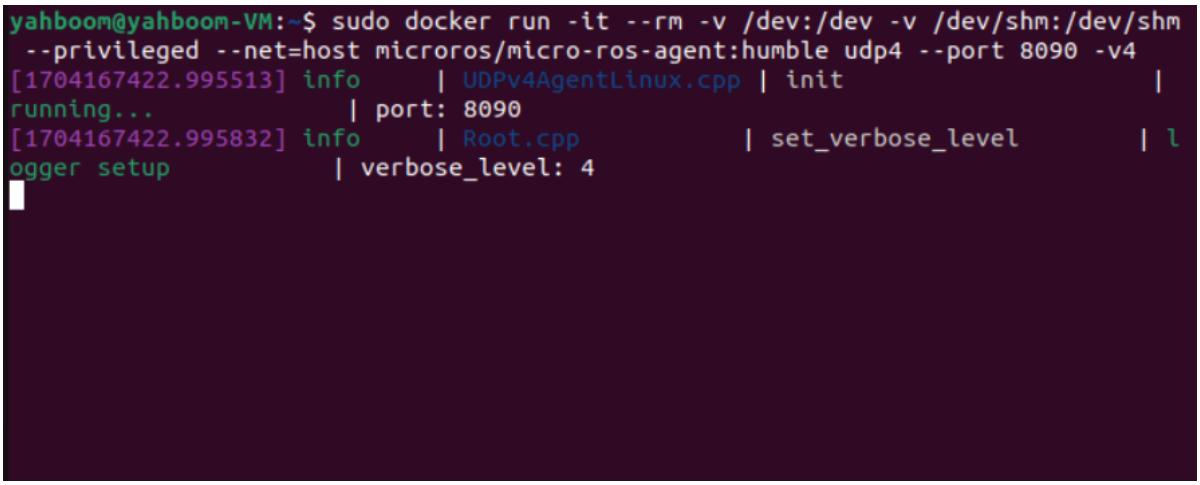
1、程序功能说明

小车连接上代理，运行程序，小车上的雷达扫描设定范围内是否有障碍物，有障碍物则会根据障碍物的位置，自动调整速度，使其自身避开障碍物。通过动态参数调节器可以调整雷达检测的范围和避障检测的距离等参数。

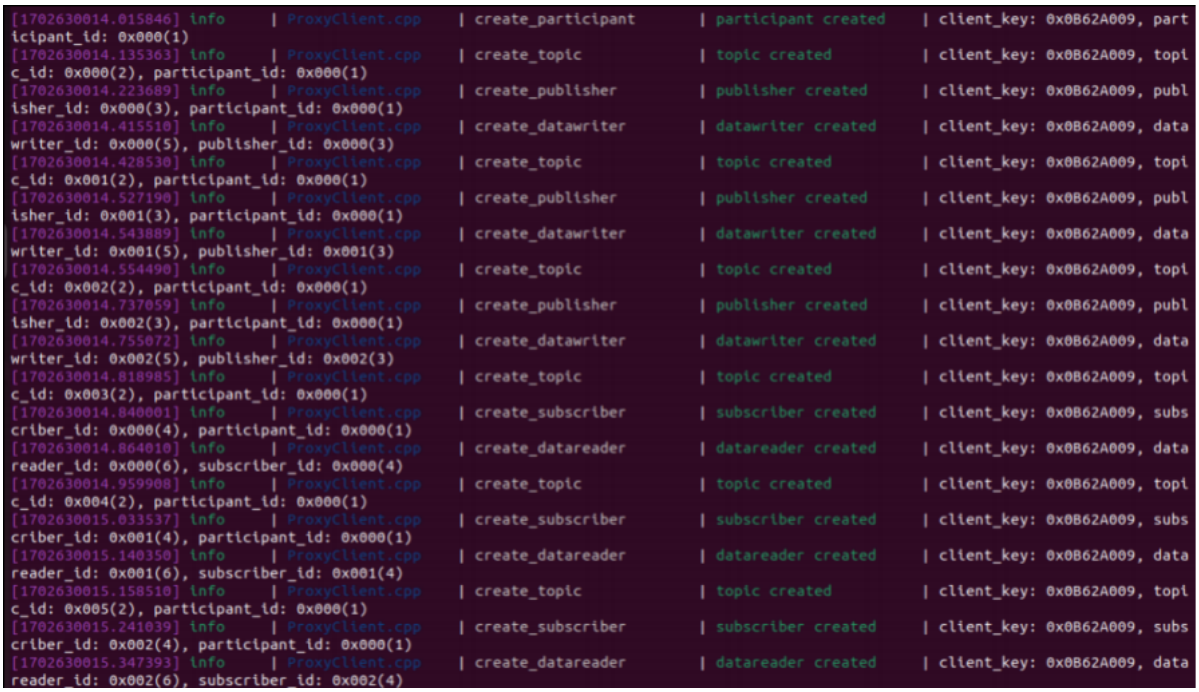
2、启动并连接代理

以配套虚拟机为例，输入以下指令启动代理，

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --
net=host microros/micro-ros-agent:humble udp4 --port 8090 -v4
```



然后，打开小车开关，等待小车连接上代理，连接成功如下图所示，



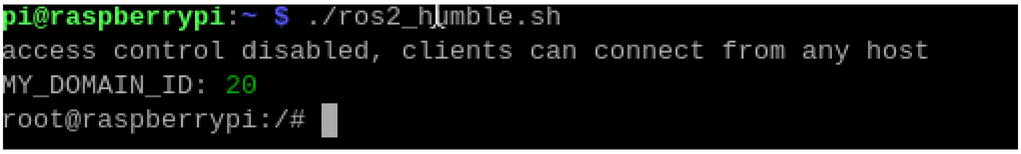
3、启动程序

3.1运行指令

如果是树莓派桌面版本和jetson nano桌面版本，需要提前进入docker，终端输入，

```
sh ros2_humble.sh
```

出现以下界面就是进入docker成功，



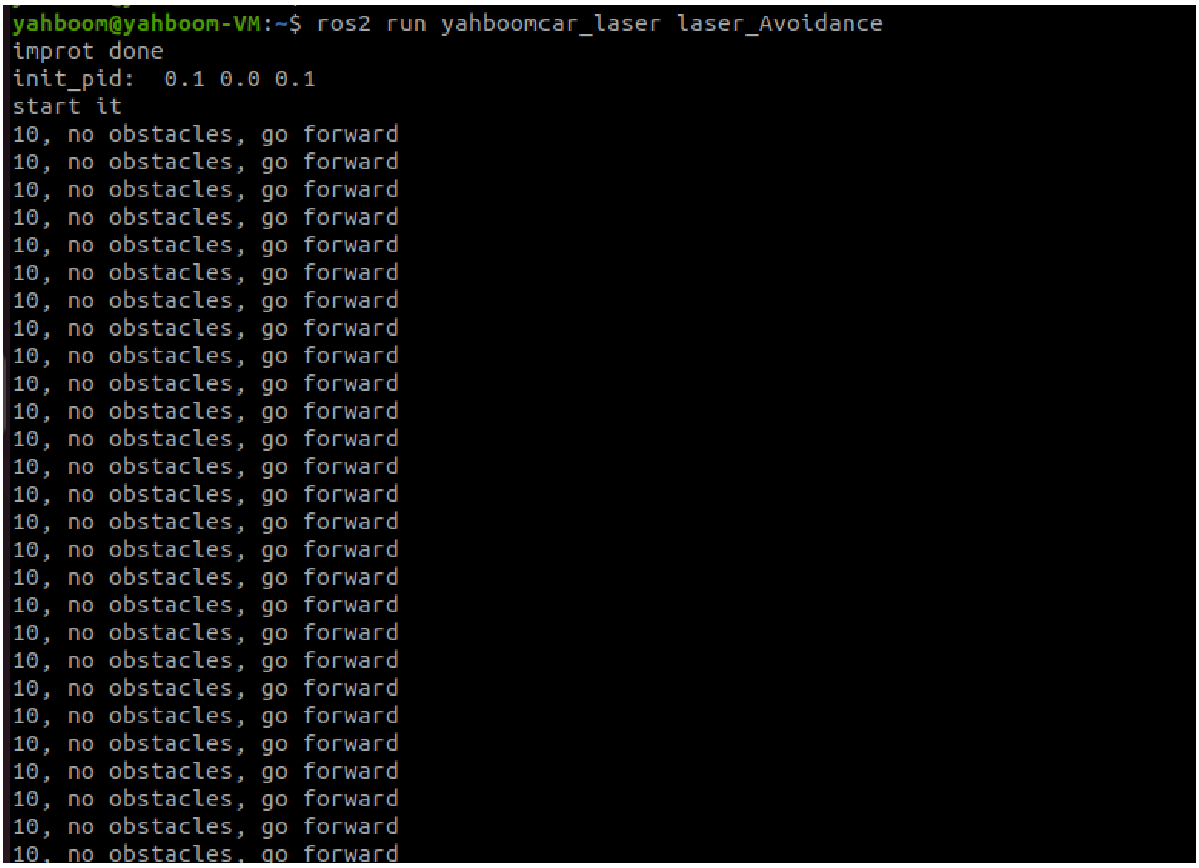
```
pi@raspberrypi:~ $ ./ros2_humble.sh
access control disabled, clients can connect from any host
MY_DOMAIN_ID: 20
root@raspberrypi:/#
```

之后在docker里输入，**（查看【docker环境】章节，如何进入同一个docker终端）**

```
ros2 run yahboomcar_laser laser_Avoidance        #雷达避障
ros2 run rqt_reconfigure rqt_reconfigure         #参数调节器
```

以配套的虚拟机为例，终端输入，

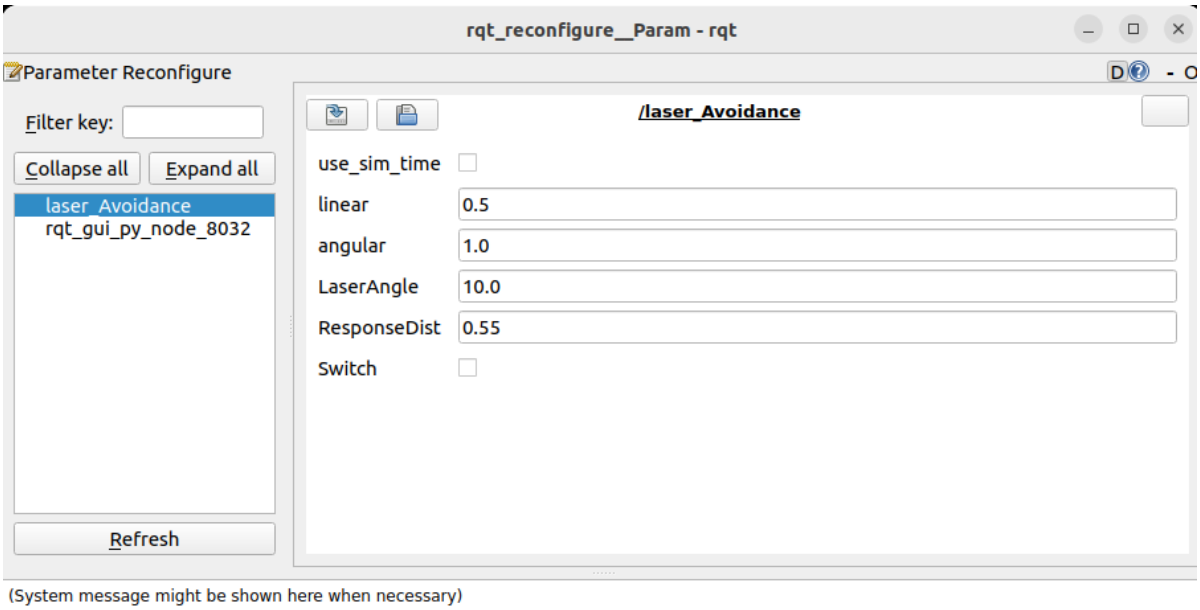
```
ros2 run yahboomcar_laser laser_Avoidance
```



```
yahboom@yahboom-VM:~$ ros2 run yahboomcar_laser laser_Avoidance
improt done
init_pid:  0.1 0.0 0.1
start it
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
10, no obstacles, go forward
```

如上图所示，如果小车上的雷达没有检测到障碍物，则会向前走。可以通过动态参数调节器去设置一些参数，终端输入，

```
ros2 run rqt_reconfigure rqt_reconfigure
```



注：刚开始打开的时候可能没有以上节点，点击Refresh刷新后可以看到全部节点。显示的laser_Avoidance就是雷达避障的节点。

以上的参数说明如下：

- linera：线速度大小
- angular：角速度大小
- LaserAngle：雷达检测角度
- ResponseDist：障碍物检测距离，当检测的物体在该范围内，则认为是障碍物
- Switch：玩法开关

修改完以上的参数，需要点击空白处，才能把参数传入程序中。

4、代码解析

源码参考路径（以配套虚拟机为例）：

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser
```

jetson nano代码路径：

```
/root/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser
```

树莓派代码路径：

```
/root/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser
```

laser_Avoidance，核心代码如下，

```
#创建雷达订阅者订阅雷达数据和遥控控制数据以及速度发布者发布速度数据
self.sub_laser = self.create_subscription(LaserScan,"/scan",self.registerScan,1)
self.sub_JoyState = self.create_subscription(Bool,'/JoyState',
self.JoyStateCallback,1)
self.pub_vel = self.create_publisher(Twist,'/cmd_vel',1)
#雷达回调函数：处理订阅到的雷达数据
```

```
ranges = np.array(scan_data.ranges)
for i in range(len(ranges)):
    angle = (scan_data.angle_min + scan_data.angle_increment * i) * RAD2DEG
#根据设定的雷达检测的角度和障碍物检测距离判断前、左、右是否有障碍物存在
if angle > 180: angle = angle - 360
if 20 < angle < self.LaserAngle:
    if ranges[i] < self.ResponseDist*1.5:
        self.Left_warning += 1
if -self.LaserAngle < angle < -20:
    if ranges[i] < self.ResponseDist*1.5:
        self.Right_warning += 1
if abs(angle) <= 20:
    if ranges[i] <= self.ResponseDist*1.5:
        self.front_warning += 1
#根据检测到障碍物，发布小车的速度让小车避开障碍物
if self.front_warning > 10 and self.Left_warning > 10 and self.Right_warning >
10:
    print ('1, there are obstacles in the left and right, turn right')
    twist.linear.x = self.linear
    twist.angular.z = -self.angular
    self.pub_vel.publish(twist)
    sleep(0.2)

elif self.front_warning > 10 and self.Left_warning <= 10 and self.Right_warning >
10:
    print ('2, there is an obstacle in the middle right, turn left')
    twist.linear.x = self.linear
    twist.angular.z = self.angular
    self.pub_vel.publish(twist)
    sleep(0.2)
.....
```