

## 3. Lidar tracking fun gameplay

### 1. Program function description

When the program starts, the lidar scans for the nearest object and then locks on, the object moves and the cart follows. If the joystick node is activated, the R2 key of the joystick can pause/enable this function.

### 2. Program code reference path

After entering the docker container, the source code of this function is located at.

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser/laser_Tracker_a1_R2.py  
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_laser/yahboomcar_laser/laser_Tracker_4ROS_R2.py
```

View the source code according to the actual lidar purchased.

### 3. Program startup

#### 3.1 Startup command

After entering the docker container, according to the actual model and lidar model, terminal input, the

```
#启动小车底盘 #Start the trolley chassis  
ros2 run yahboomcar_bringup Ackman_driver_R2  
#启动A1雷达 # Activate A1 lidar  
ros2 launch sllidar_ros2 sllidar_launch.py  
#启动4ROS雷达 # Activate 4ROS lidar  
ros2 launch ydlidar_ros2_driver ydlidar_launch.py  
#启动雷达跟踪程序 A1雷达 # Activate lidar tracking program A1 lidar  
ros2 run yahboomcar_laser laser_Tracker_a1_R2  
#启动雷达跟踪程序 4ROS雷达 # Initiate lidar tracking program 4ROS lidar  
ros2 run yahboomcar_laser laser_Tracker_4ROS_R2  
#启动手柄, 如果需要的话 #Start the handle, if needed  
ros2 run yahboomcar_ctrl yahboom_joy_R2  
ros2 run joy joy_node
```

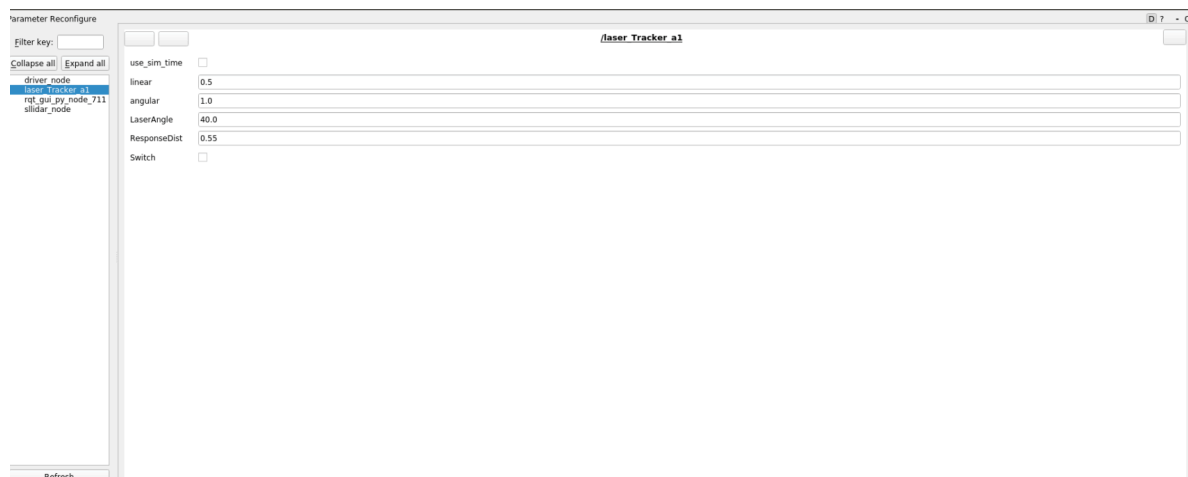
#### 3.2. View the topic communication node map

docker terminal by typing.

```
ros2 run rqt_graph rqt_graph
```



It is also possible to set the size of the parameter, the terminal input, by means of a dynamic parameter regulator, the



The meaning of each parameter is as follows.

parameter name	parameter meaning
Switch	Play switch
<u>ResponseDist</u>	Obstacle detection distance
linear	The linear velocity
angular	Angular velocity
<u>LaserAngle</u>	<u>Lidar</u> detection Angle

The above parameters are adjustable, except Switc, the other four need to be set when you need to be a decimal, modified, click on the blank before you can write.

## 4. Core Code

R2 model, A1 lidar source code as an example, mainly look at the lidar callback function, which explains how to get to each angle of the obstacle distance information, and then find out the nearest point, and then determine the distance, and then is the calculation of the speed of data, and finally released.

```

angle = (scan_data.angle_min + scan_data.angle_increment * i) * RAD2DEG
if abs(angle) > (180 - self.priorityAngle): #priorityAngle是小车优先考虑跟随范围
    #priorityAngle is the range of carts prioritized to follow
    if ranges[i] < (self.ResponseDist + offset):
        frontDistList.append(ranges[i])
        frontDistIDList.append(angle)
  
```

```

elif (180 - self.LaserAngle) < angle < (180 - self.priorityAngle):
    minDistList.append(ranges[i])
    minDistIDList.append(angle)
elif (self.priorityAngle - 180) < angle < (self.LaserAngle - 180):
    minDistList.append(ranges[i])
    minDistIDList.append(angle)
    if len(frontDistIDList) != 0:
        minDist = min(frontDistList)
        minDistID = frontDistIDList[frontDistList.index(minDist)]
    else:
        minDist = min(minDistList)
        minDistID = minDistIDList[minDistList.index(minDist)]#计算出最小距离点的
ID # Calculate the ID of the minimum distance point
    if self.Joy_active or self.Switch == True:
        if self.Moving == True:
            self.pub_vel.publish(Twist())
            self.Moving = not self.Moving
            return
        self.Moving = True
        velocity = Twist()
        if abs(minDist - self.ResponseDist) < 0.1: minDist =
self.ResponseDist#判断距离最小的点的距离
            #Determine the distance to the point with the smallest
distance
            velocity.linear.x = -self.lin_pid.pid_compute(self.ResponseDist,
minDist)#计算线速度 # Calculate the linear velocity
            ang_pid_compute = self.ang_pid.pid_compute((180 -
abs(minDistID)) / 72, 0)#计算角速度 # Calculate the angular velocity
            if minDistID > 0: velocity.angular.z = -ang_pid_compute
            else: velocity.angular.z = ang_pid_compute
            if ang_pid_compute < 0.02: velocity.angular.z = 0.0
            self.pub_vel.publish(velocity)

```