## 物体追踪

注：虚拟机、ROS-wifi图传模块需要与microROS控制板ROS*DOMAIN*ID需要一致，都要设置成20，可以查看【MicroROS控制板参数配置】来设置microROS控制板ROS*DOMAIN*ID，查看教程【连接MicroROS代理】判断ID是否一致。

### 1、程序功能说明

程序启动后，通过鼠标选中需要跟踪的物体，按下空格键，小车的云台舵机进入跟踪模式。小车云台会跟随 被跟踪的物体移动，并且时刻保证被追踪的物体保持在画面中心。

### 2、操作步骤

#### 2.1、启动

进入docker，终端输入，

ros2 run yahboom\_esp32ai\_car mono\_Tracker

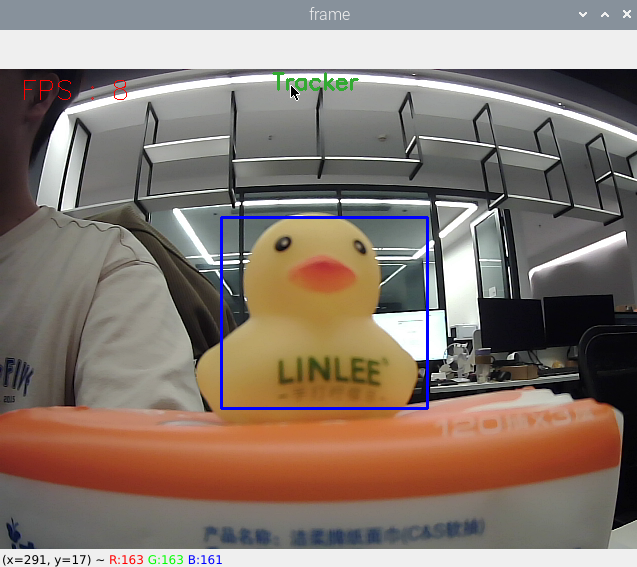
**如果摄像头的角度不是处于该角度，请按CTRL+C结束程序，重新运行一下，这是因为网络延迟导致发送舵机的角度丢包导致**



**如果摄像头的画面图像出现倒置**，需要看**3.摄像头画面纠正(必看)**文档自己纠正，该实验不再阐述。

#### 2.2、识别

启动之后，进入选择模式，用鼠标选中目标所在位置，如下图所示，松开即开始识别。



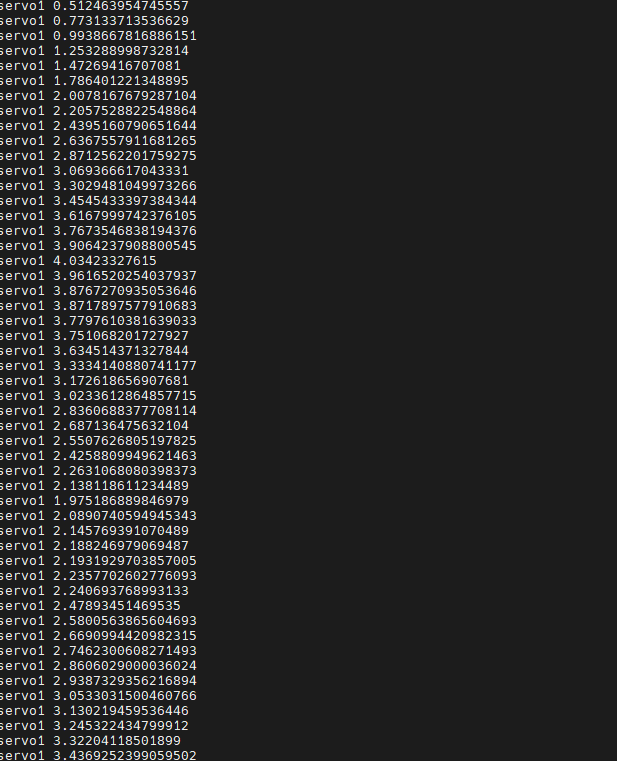
键盘按键控制：

【r】：选择模式，可用鼠标选择要识别目标的区域，如上图。

【q】：退出程序。

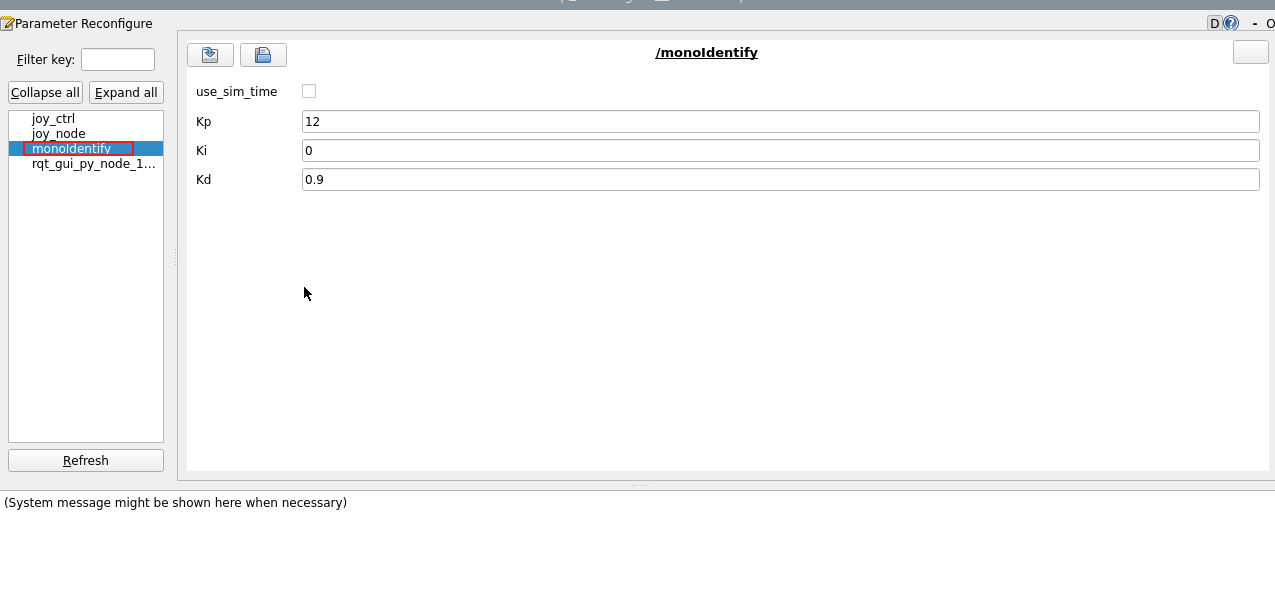
【空格键】：目标追踪；在跟随的时候缓慢移动目标即可，移动太快将会丢失目标。

按下空格能看到通过pid算出来的运动角度。



还可以使用动态参数调节器对PID参数进行调试，Docker终端输入，

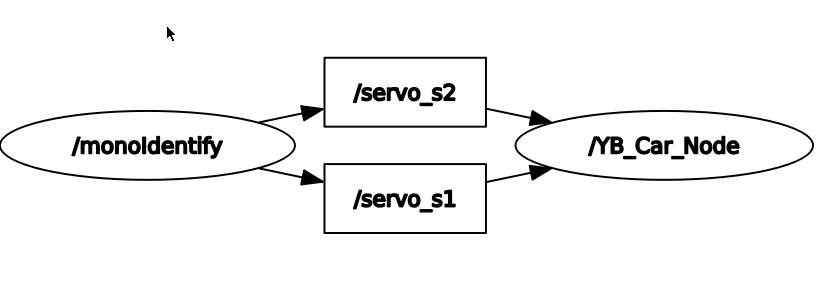
ros2 run rqt\_reconfigure rqt\_reconfigure



可调节的参数有小车云台的PID参数 ，修改参数后，点击“refresh”刷新数据。

#### 2.3、查看节点话题通讯图

可以通过以下命令查看节点之间的话题通讯，



### 3、核心代码

功能实现的原理与颜色追踪差不多，都是根据目标的中心坐标和来计算s1、s2舵机转动角度，然后发布给底盘，部分代码入下，

#这部分是选择物体后，得出中心坐标，用于计算舵机角度  
if len(contours) != 0:  
 areas = []  
 for c in range(len(contours)): areas.append(cv.contourArea(contours[c]))  
 max\_id = areas.index(max(areas))  
 max\_rect = cv.minAreaRect(contours[max\_id])  
 max\_box = cv.boxPoints(max\_rect)  
 max\_box = np.int0(max\_box)  
 (color\_x, color\_y), color\_radius = cv.minEnclosingCircle(max\_box)  
#这部分是计算出center\_x，distance的值，  
center\_x = targEnd\_x / 2 + targBegin\_x / 2  
center\_y = targEnd\_y / 2 + targBegin\_y / 2  
width = targEnd\_x - targBegin\_x  
high = targEnd\_y - targBegin\_y  
self.point\_pose = (center\_x, center\_y, min(width, high))  
if self.Track\_state == 'tracking':  
 if self.circle[2] != 0: threading.Thread(target=self.execute, args=(self.circle[0], self.circle[1])).start()  
#通过pid计算舵机运动角度  
[x\_Pid, y\_Pid] = self.PID\_controller.update([point\_x - 320, point\_y - 240])  
 if self.img\_flip == True:  
 self.target\_servox -= x\_Pid  
 self.target\_servoy += y\_Pid  
 else:  
 self.target\_servox -= x\_Pid  
 self.target\_servoy += y\_Pid  
 if self.target\_servox >= 45:  
 self.target\_servox = 45  
 elif self.target\_servox <= -45:  
 self.target\_servox = -45  
 if self.target\_servoy >= 40:  
 self.target\_servoy = 40  
 elif self.target\_servoy <= -90:  
 self.target\_servoy = -90