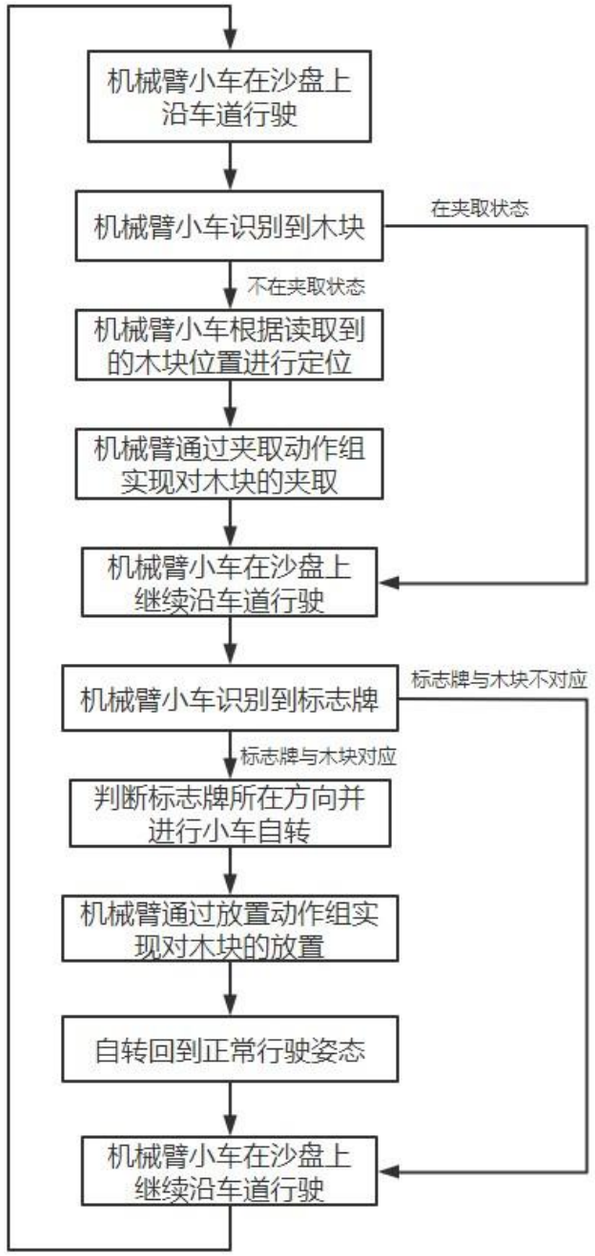
WHEELTEC
轮 趣 科 技

深度学习相关——沙盘机械臂功能使用详解



沙盘机械臂识别功能运行逻辑：
行驶——识别——夹取——行驶——识别——放置（循环）

行驶：视觉巡线
识别：深度学习
夹取放置：机械臂动作

与沙盘自动驾驶功能不同之处：加入了机械臂的协作

Step1：机械臂位置确认

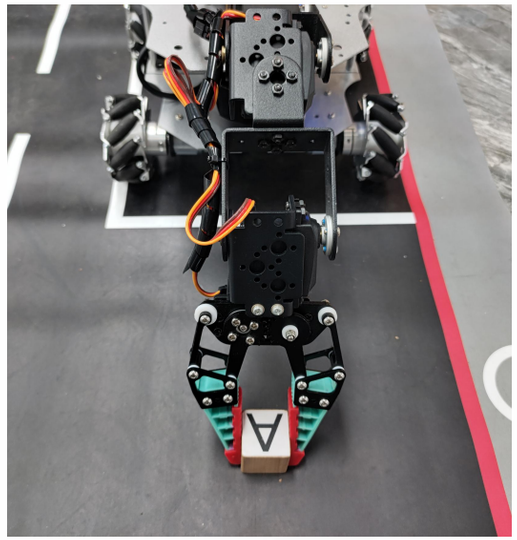
沙盘机械臂功能使用的车型——四自由度机械臂麦轮小车（相机带有加高）

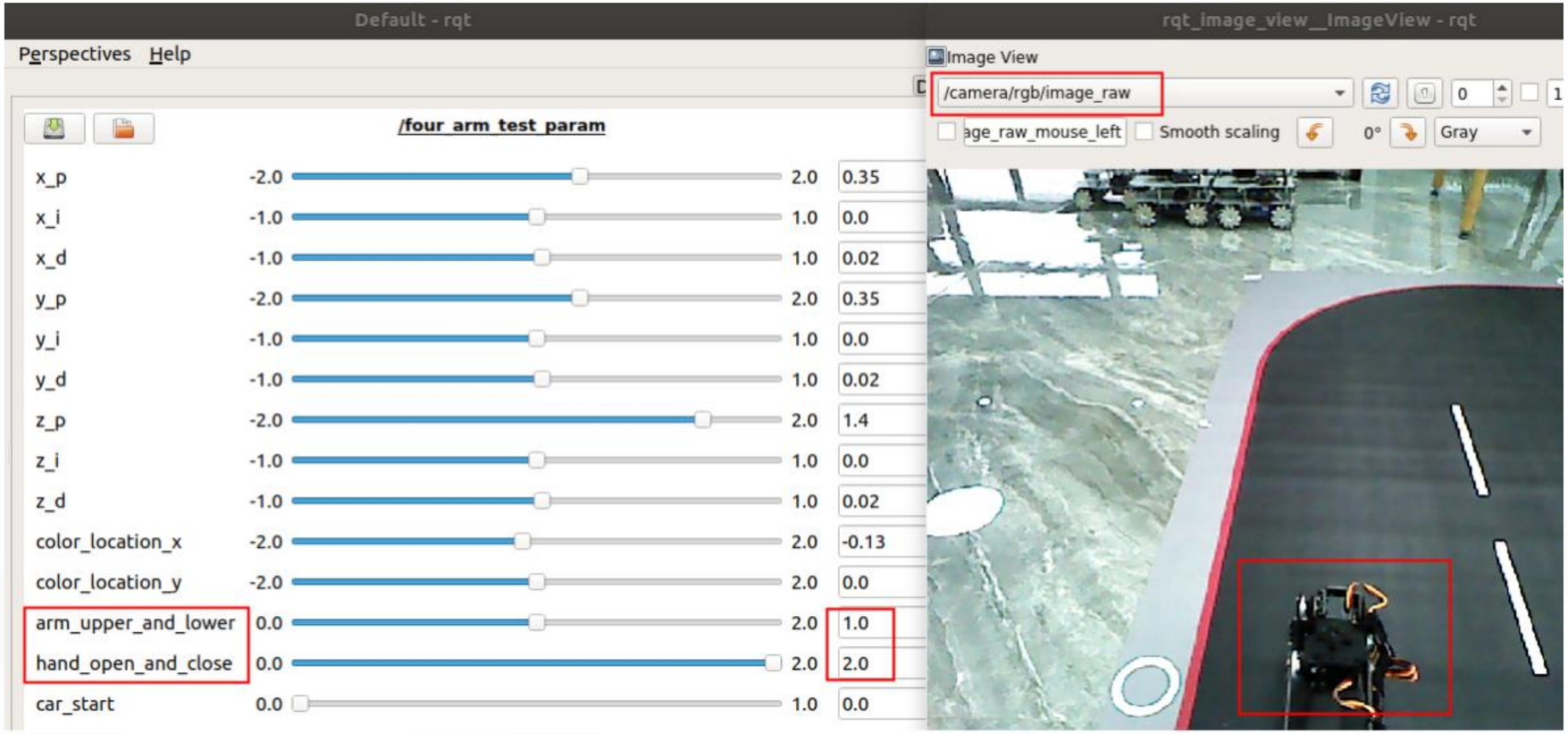
四自由度机械臂在y方向上无法平移运动 ⟶ 通过调整麦轮小车底盘的方式来进行定位（可在x、y方向上移动）
使用固定的机械臂夹取放置动作组 ⟶ 只需要控制好小车底盘与木块、标志牌的相对位置即可完成相应动作

在运行沙盘机械臂功能前需单独运行机械臂测试功能以确定最佳夹取放置位置 首先调整相机视角向下至机械限位处

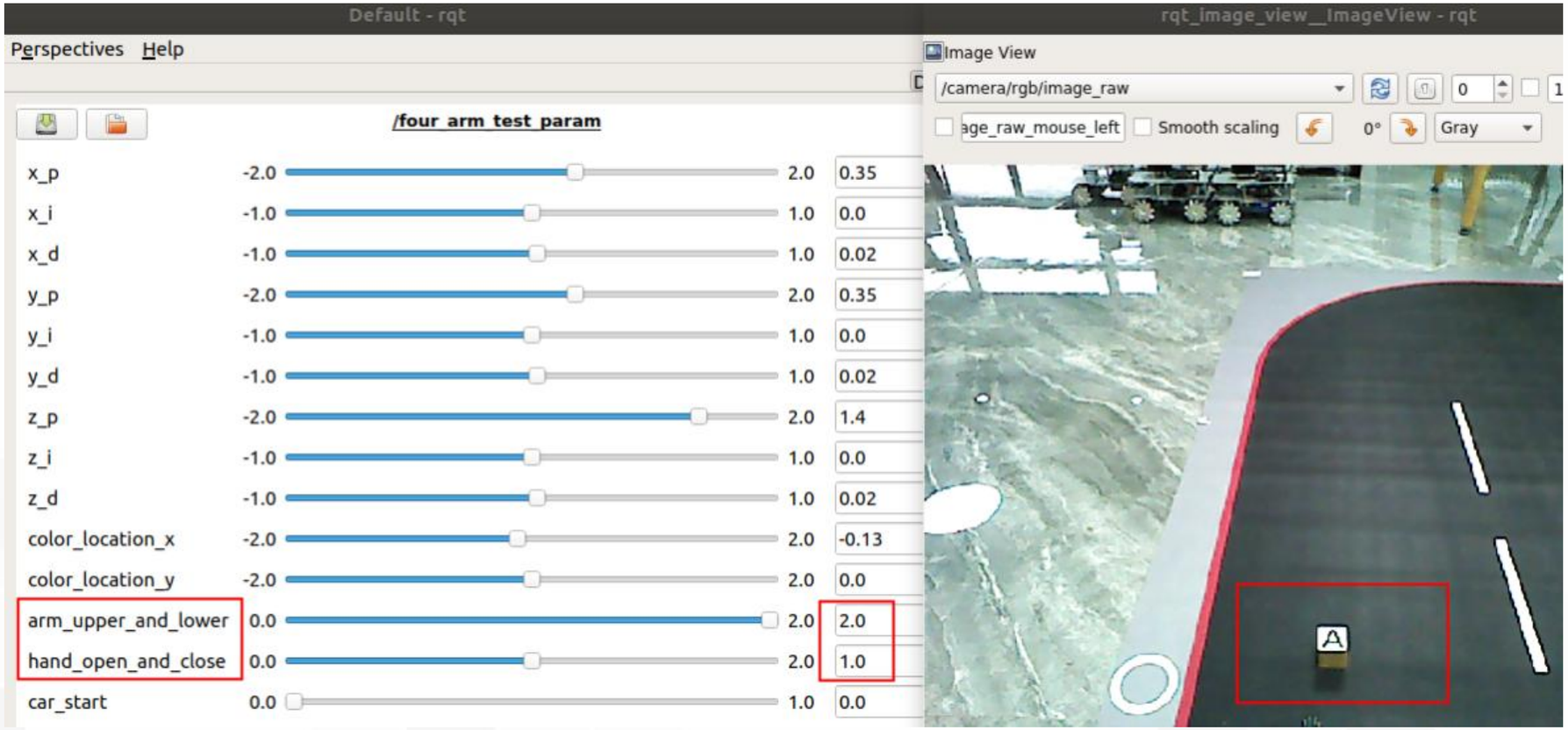
终端运行以下指令
运行机械臂测试功能：roslaunch wheeltec_arm_pick test_param.launch
打开rqt调参界面：rqt
打开rqt实时查看相机画面：rqt_image_view

在rqt调参页面将arm_upper_and_lower参数设置为1
将hand_open_and_close参数设置为2
同时将木块（任意一个可识别木块）置于夹爪刚好可以夹取到的位置 如图所示





在rqt调参页面将hand_open_and_close参数设置为1
将arm_upper_and_lower参数设置为2
确认夹爪松开机械臂收回后 刚才木块放置的最佳夹取位置是否在相机画面中 如图所示
如在画面中：结束运行机械臂的进程 进入下一步
如不在画面中：调整机械臂动作组位置——详见机械臂相关教程 此处不做赘述



Step2：根据相对位置调参

此处需保持step1中机械臂小车与木块相对位置不变，即完成step1后不移动小车或木块，进入step2

darknet_ros.launch （位于darknet_ros功能包）需修改选择沙盘机械臂识别功能

```
<!-- 沙盘机械臂识别  -->
<arg name="network_param_file"          default="$(find darknet_ros)/config/yolov3-tiny-blocks.yaml"/>
```

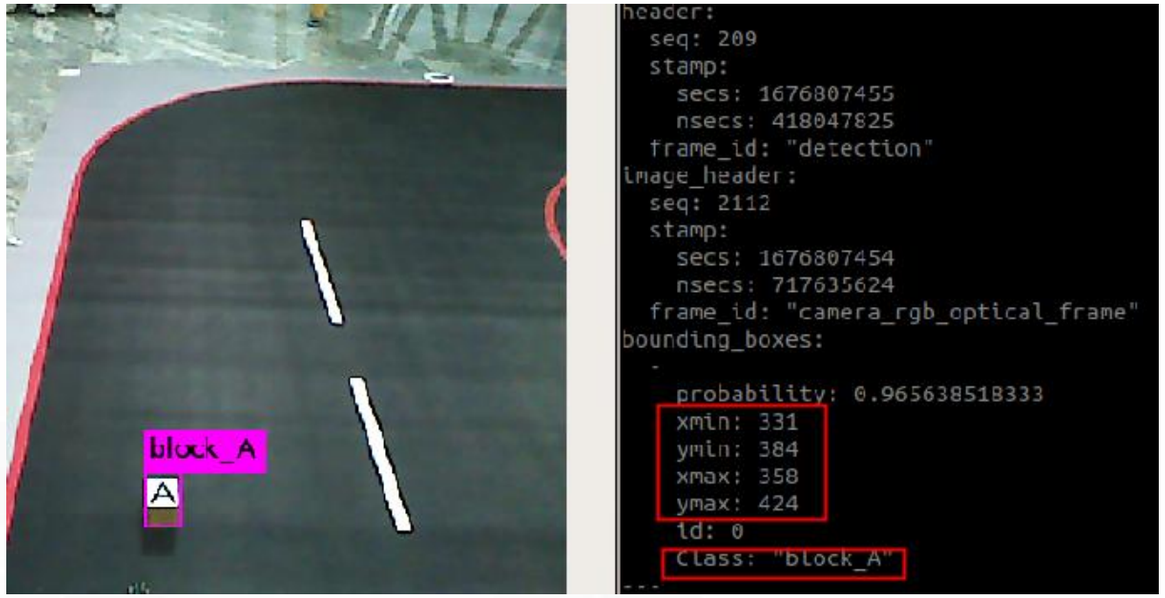
终端运行以下指令：
打开深度学习节点：roslaunch darknet_ros darknet_ros.launch
读取深度学习话题返回的参数：rostopic echo /darknet_ros/bounding_boxes

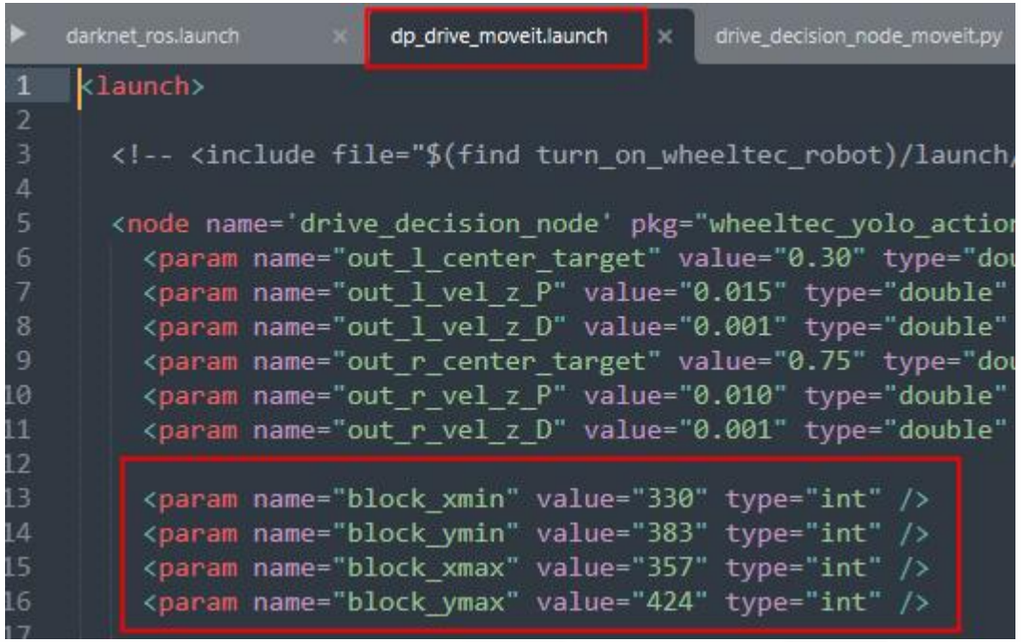

```
header:
  seq: 209
  stamp:
    secs: 1676807455
    nsecs: 418047825
  frame_id: "detection"
image_header:
  seq: 2112
  stamp:
    secs: 1676807454
    nsecs: 717635624
  frame_id: "camera_rgb_optical_frame"
bounding_boxes:
  -
    probability: 0.965638518333
    xmin: 331
    ymin: 384
    xmax: 358
    ymax: 424
    id: 0
    Class: "block_A"
---
```

根据获取的位置信息修改对应参数

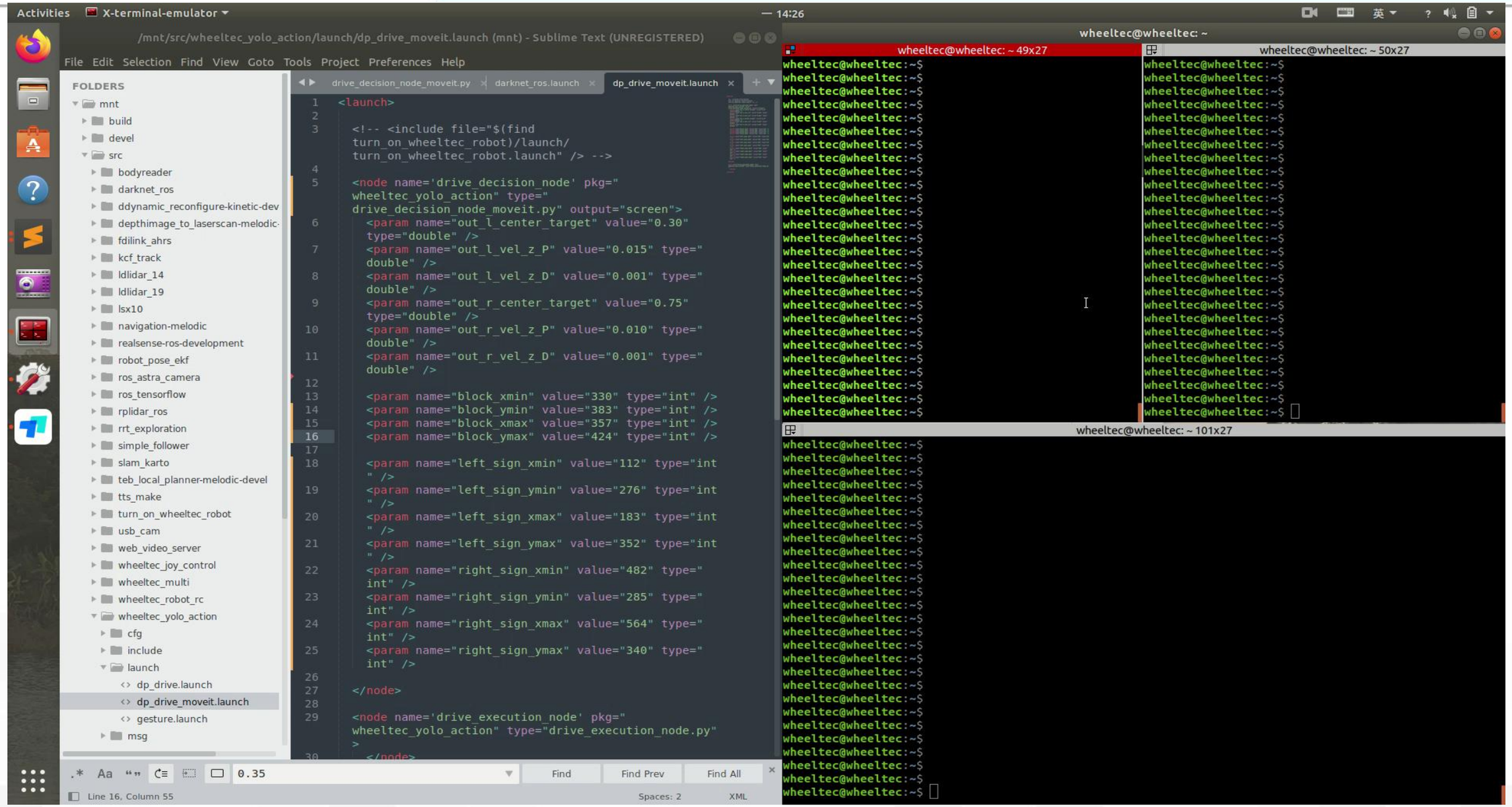
查看echo话题的终端 确认此时yolo获取到的识别框的四个端点坐标 如下左图红框
在dp_drvie_moveit.launch（位于wheeltec_yolo_action功能包）中修改对应参数 如右下图红框

四个端点坐标的作用：通过读取到画面内木块的坐标位置，确定木块最佳夹取状态下，机械臂小车与木块的相对位置





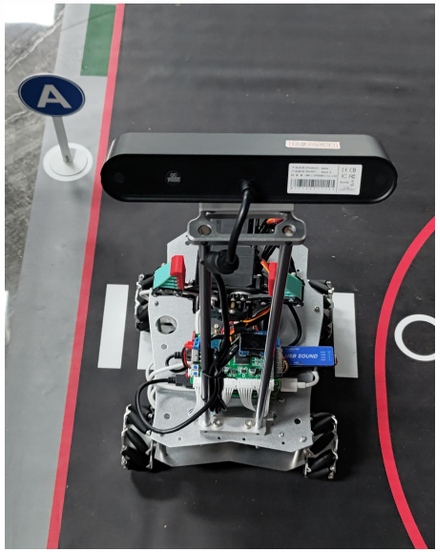
根据获取的位置信息修改对应参数



注：pdf无法插入视频 观看此演示视频请移步沙盘机械臂功能视频教程

根据获取的位置信息修改对应参数

标志牌的调整同理 将小车放置于想要进行自转的位置
查看话题读取标志牌坐标信息并进行修改
根据行驶方向选取的不同，标志牌在视野范围内的偏向也不同
如图，若行驶途中标志牌一直位于左侧，则修改文件中left_sign处对应坐标




```
wheeltec@wheeltec: ~ 50x27
    nsecs: 631923667
  frame_id: "detection"
image_header:
  seq: 6414
  stamp:
    secs: 1676804019
    nsecs:  38950674
  frame_id: "camera_rgb_optical_frame"
bounding_boxes:
  -
    probability: 0.99944883585
    xmin: 77
    ymin: 311
    xmax: 160
    ymax: 364
    id: 4
    Class: "sign_A"
  -
    probability: 0.999893784523
    xmin: 244
    ymin: 39
    xmax: 272
    ymax: 68
    id: 5
    Class: "sign_B"
```

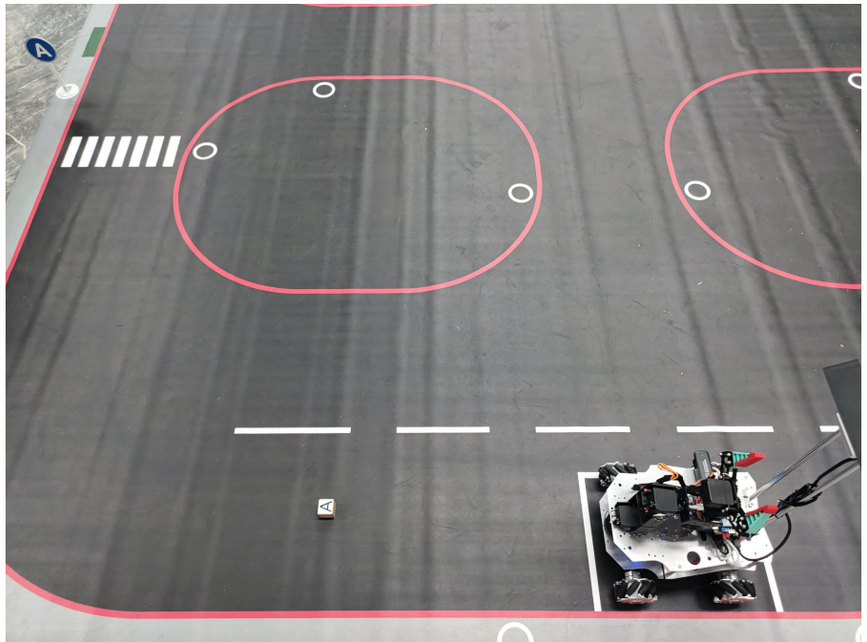
```
yolov3-tiny-blocks.yaml | drive_decision_node_moveit.py | dp_drive_moveit.launch
<launch>

  <!-- <include file="$(find turn_on_wheeltec_robot)/launch/turn_on_wheeltec_r

  <node name='drive_decision_node' pkg="wheeltec_yolo_action" type="drive_deci
    <param name="out_l_center_target" value="0.30" type="double" />
    <param name="out_l_vel_z_P" value="0.015" type="double" />
    <param name="out_l_vel_z_D" value="0.001" type="double" />
    <param name="out_r_center_target" value="0.75" type="double" />
    <param name="out_r_vel_z_P" value="0.010" type="double" />
    <param name="out_r_vel_z_D" value="0.001" type="double" />

    <param name="block_xmin" value="330" type="int" />
    <param name="block_ymin" value="383" type="int" />
    <param name="block_xmax" value="357" type="int" />
    <param name="block_ymax" value="424" type="int" />

    <param name="left_sign_xmin" value="112" type="int" />
    <param name="left_sign_ymin" value="276" type="int" />
    <param name="left_sign_xmax" value="183" type="int" />
    <param name="left_sign_ymax" value="352" type="int" />
    <param name="right_sign_xmin" value="482" type="int" />
    <param name="right_sign_ymin" value="285" type="int" />
    <param name="right_sign_xmax" value="564" type="int" />
    <param name="right_sign_ymax" value="340" type="int" />
```

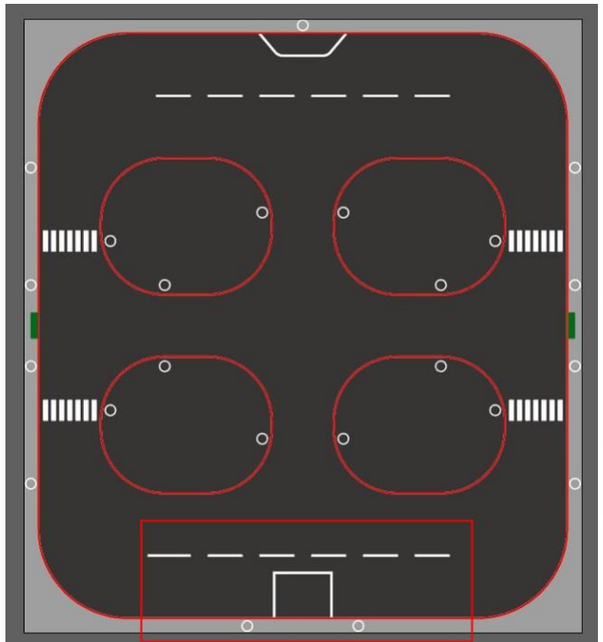
Step3：运行沙盘机械臂功能

step1与step2都是运行功能前的准备工作 通常情况下相同小车相同环境只需进行一次调整

运行沙盘机械臂功能前 需要将小车置于沙盘上 此处建议初始时将小车放置于沙盘两侧较宽的位置 如图所示 沙盘机械臂功能设定仅沿外道行走 方向可选

标志牌放置方向需注意与小车行驶方向协调 如图所示 需保证小车运行过程可以看到标志牌正面





准备工作完成后 在终端运行以下指令：
1.开启深度学习节点：roslaunch darknet_ros darknet_ros.launch
2.开启机械臂木块夹取节点：roslaunch wheeltec_arm_pick arm_pick_dp.launch
3.开启沙盘运行节点：roslaunch wheeltec_yolo_action dp_drive_moveit.launch

注意3需要在2机械臂初始化完成后再执行 机械臂节点与沙盘节点的初始化都需要一定时间

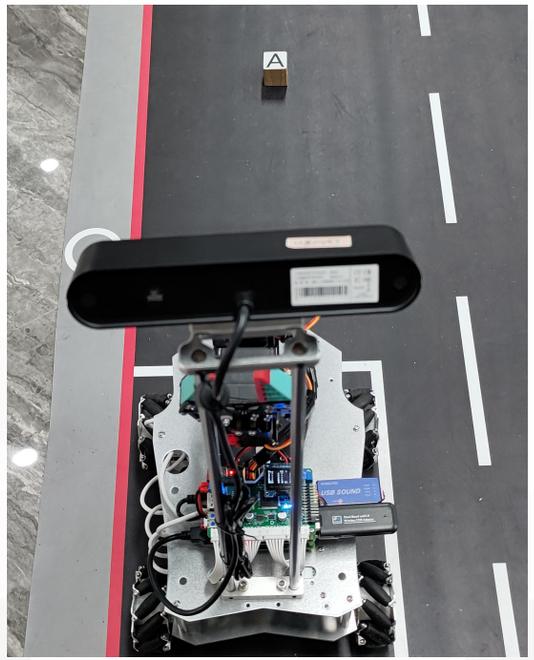
机械臂初始化完成终端提示如图：

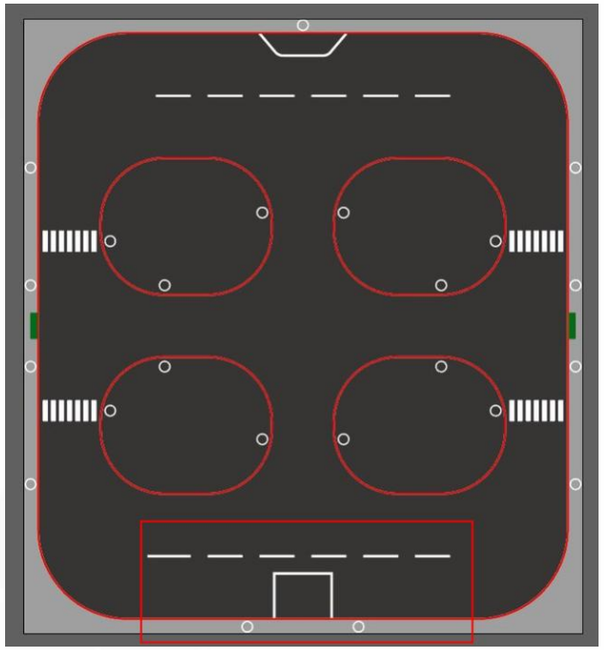
```
*      - StateValidationService
**********************************************
*******

[ INFO] [1676807674.561696075]: MoveGroup context
 using planning plugin ompl_interface/OMPLPlanner
[ INFO] [1676807674.561836752]: MoveGroup context
 initialization complete

You can start planning now!

[ INFO] [1676807675.383348835]: Ready to take com
mands for planning group arm.
[ INFO] [1676807675.741137898]: Ready to take com
mands for planning group hand.
```

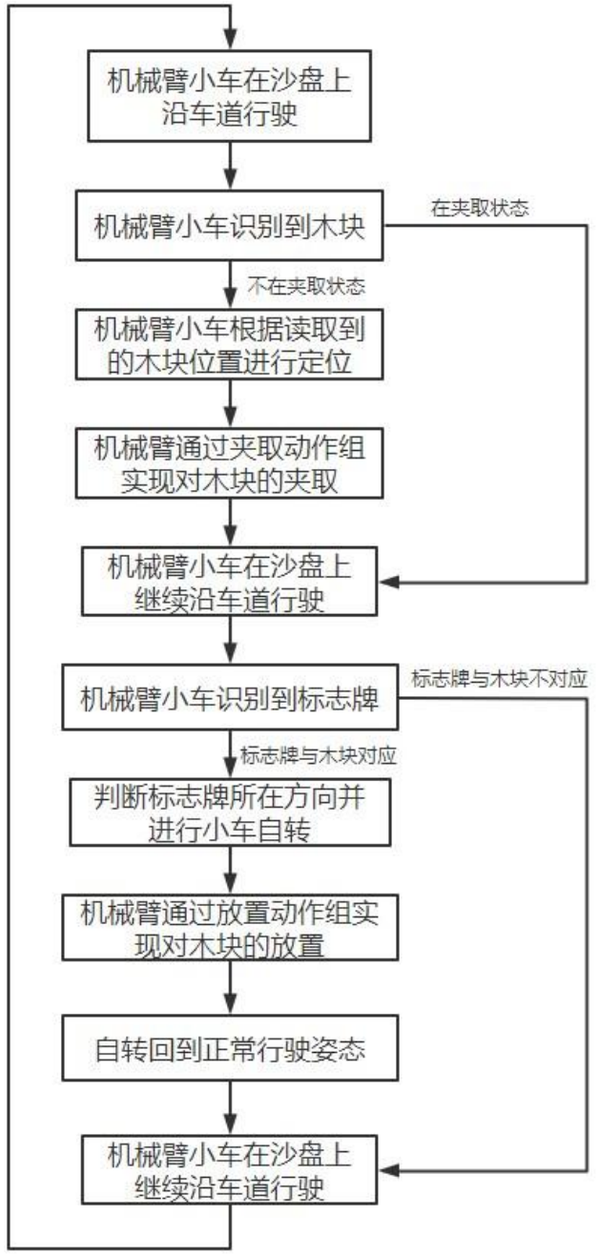
运行沙盘机械臂功能

三个节点都开启后 正常情况下小车会自动识别行驶方向 并沿外道行驶
此时可将木块置于小车前方 注意木块放置方向需与行驶方向保持一致 即保证相机视角中看到的木块中字母是可识别的 如下图所示

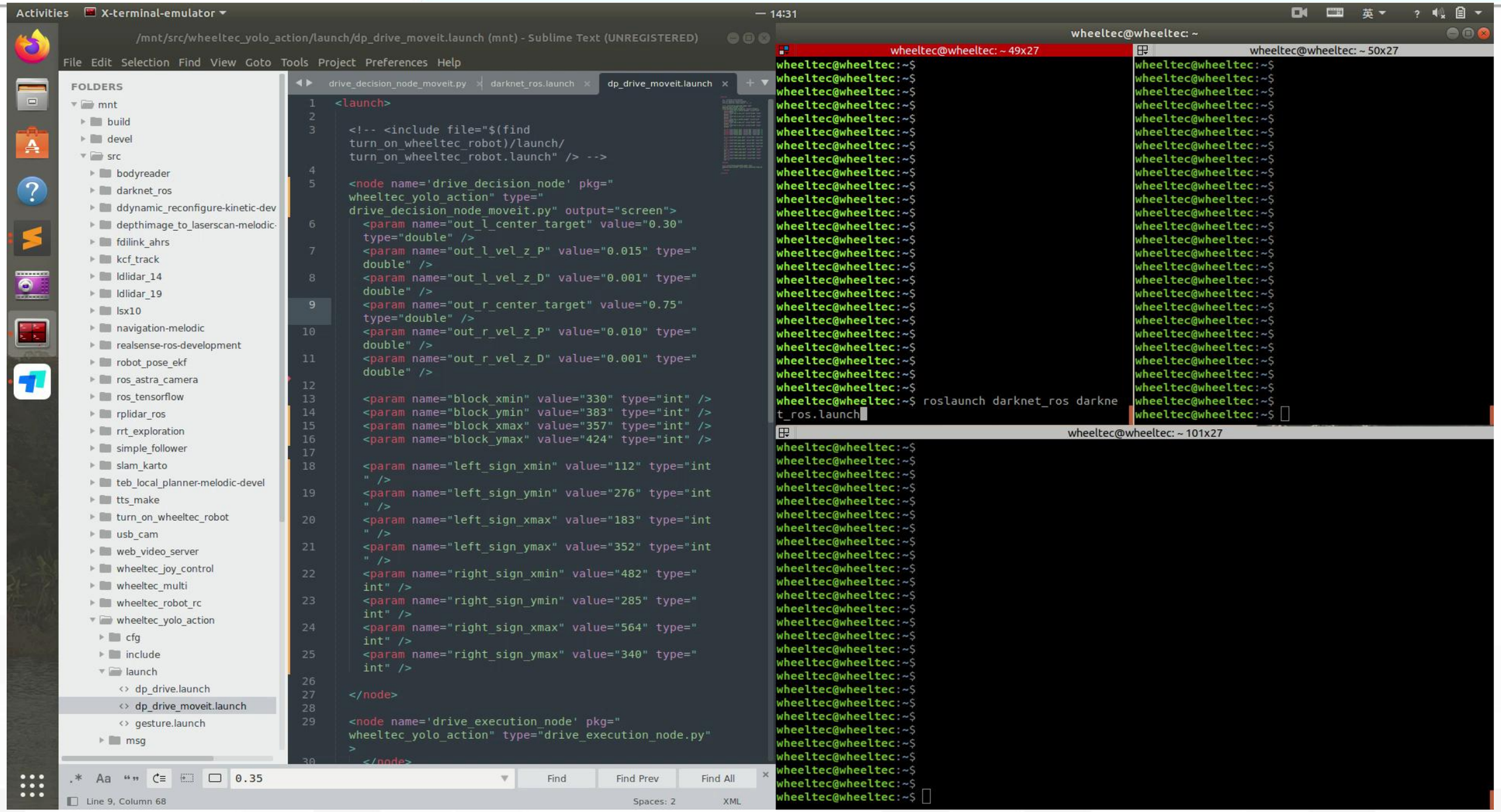
当小车识别到木块后 会自动调整底盘并进行夹取动作 完成夹取动作后继续沿沙盘行驶
当碰到与木块字母对应的标志牌时 再调整底盘并进行放置动作 完成放置动作后沿沙盘行驶 继续循环以上逻辑 如右图所示







运行沙盘机械臂功能



注：pdf无法插入视频 观看此演示视频请移步沙盘机械臂功能视频教程

Step4：代码与参数详解

沙盘机械臂功能的主要实现部分为沙盘节点（/src/wheeltec_yolo_action/drive_decision_node_moveit.py）
深度学习节点主要是调用我们所训练处理好的数据集
机械臂节点主要是执行沙盘节点发布话题中的内容

```
line_pub = rospy.Publisher("/line_judgment", Int8, queue_size=1)
cmdvel_pub = rospy.Publisher("/cmd_vel", Twist, queue_size=1)
side_flag_sub = rospy.Subscriber("/darknet_ros/bounding_boxes", BoundingBoxes, side_flag_callback)
control_flag_sub = rospy.Subscriber("/control_flag", Int8, control_flag_callback)
arm_pick_or_put_pub = rospy.Publisher("/arm_state",String ,queue_size=10 ) ##
```

drive_decision_node_moveit.py中订阅发布话题如图
订阅：图像话题——用于车道判断 深度学习识别结果话题——用于识别
发布：机械臂状态话题——控制机械臂运动 速度话题——控制底盘运动

深度学习中id对应关系可以查看
沙盘机械臂识别功能对应文件

yolov3-tiny-blocks.yaml

```yaml
yolo_model:

  config_file:
    name: yolov3-tiny-blocks.cfg
  weight_file:
    name: yolov3-tiny-blocks_last.weights
  threshold:
    value: 0.3
  detection_classes:
    names:
      - block_A
      - block_B
      - block_C
      - block_D
      - sign_A
      - sign_B
      - sign_C
      - sign_D
      - stop
```

drive_decision_node_moveit.py

```python
    for boxes in msg.bounding_boxes:
        if boxes.id == 0 and boxes.probability > 0.6:
            if boxes.ymin > 250 and boxes.xmax < 400 and boxes.xmin > 240 :
                block_a_flag = 1
                blocka_x = (boxes.xmin + boxes.xmax)/2
                blocka_y = (boxes.ymin + boxes.ymax)/2
        elif boxes.id == 1 and boxes.probability > 0.6:
            if boxes.ymin > 250 and boxes.xmax < 400 and boxes.xmin > 240 :
                block_b_flag = 1
                blockb_x = (boxes.xmin + boxes.xmax)/2
                blockb_y = (boxes.ymin + boxes.ymax)/2
        elif boxes.id == 2 and boxes.probability > 0.6:
            if boxes.ymin > 250 and boxes.xmax < 400 and boxes.xmin > 240 :
                block_c_flag = 1
                blockc_x = (boxes.xmin + boxes.xmax)/2
                blockc_y = (boxes.ymin + boxes.ymax)/2
        elif boxes.id == 3 and boxes.probability > 0.6:
            if boxes.ymin > 250 and boxes.xmax < 400 and boxes.xmin > 240 :
                block_d_flag = 1
                blockd_x = (boxes.xmin + boxes.xmax)/2
                blockd_y = (boxes.ymin + boxes.ymax)/2

        elif boxes.id == 4 and boxes.probability > 0.8:
            if boxes.ymin > 200 and boxes.ymax < 450:
                sign_a_flag = 1
                if boxes.xmax < 320:
                    side_flag = 1
                elif boxes.xmin > 320:
                    side_flag = 2
                signa_x = (boxes.xmin + boxes.xmax)/2
                signa_y = (boxes.ymin + boxes.ymax)/2
        elif boxes.id == 5 and boxes.probability > 0.8:
            if boxes.ymin > 200 and boxes.ymax < 450:
                sign_b_flag = 1
                if boxes.xmax < 320:
```

主要实现代码——识别木块部分

drive_decision_node_moveit.py | dp_drive_moveit.launch

```
elif block_a_flag == 1 and picka_flag == 0 :#recognize blockA and not picking state
    print("pick preparation")

    car_stop = CtrlData()#stop to adjust robot position
    ctrl_pub.publish(car_stop)
    erro_x = blocka_x - ((block_xmin +block_xmax)/2)
    erro_y = blocka_y - ((block_ymin +block_ymax)/2)
    #rospy.loginfo(erro_x)
    #rospy.loginfo(erro_y)
    d_erro_x = erro_x - last_erro_x
    d_erro_y = erro_y - last_erro_y
    carmove = Twist()
    carmove.linear.x = -float(erro_y)*0.0005 - float(d_erro_y)*0.0005
    carmove.linear.y = -float(erro_x)*0.0005 - float(d_erro_x)*0.0005
    last_erro_x=erro_x
    last_erro_y=erro_y
    if -0.01 < carmove.linear.x < 0.00005:
        carmove.linear.x = 0.0
    if -0.01 < carmove.linear.y < 0.00005:
        carmove.linear.y = 0.0
    if carmove.linear.x == 0 and carmove.linear.y == 0 :
        d_erro_x = 0
        d_erro_y = 0
    cmdvel_pub.publish(carmove)
    block_a_flag = 0
    count2 = i
```

涉及参数：

底盘调整速度——

```
carmove.linear.x = -float(erro_y)*0.0005 - float(d_erro_y)*0.0005
carmove.linear.y = -float(erro_x)*0.0005 - float(d_erro_x)*0.0005
```

停止速度调整判断条件

```
if -0.01 < carmove.linear.x < 0.00005:
    carmove.linear.x = 0.0
if -0.01 < carmove.linear.y < 0.00005:
    carmove.linear.y = 0.0
```

主要实现代码——夹取木块部分

```
if d_erro_x == 0 and d_erro_y == 0 and -15<erro_x<15 and -15<erro_y<15:#judge twice to improve accuracy
    time.sleep(3)
    erro_x = blocka_x - ((block_xmin +block_xmax)/2)
    erro_y = blocka_y - ((block_ymin +block_ymax)/2)
    if -15<erro_x<15 and -15<erro_y<15:#pick after adjustment
        move_flag = 1
        #print("ready to stop")
        carmove.linear.x = 0.0
        carmove.linear.y = 0.0
        cmdvel_pub.publish(carmove)

        print("stop to pick")
        arm_state = "pick"
        arm_pick_or_put_pub.publish(arm_state)
        rospy.loginfo(arm_state)
        time.sleep(30)

        block_a_flag = 0
        move_flag = 0
        print("pick over")
        picka_flag = 1#pick blockA
        #rospy.loginfo(picka_flag)
```

涉及参数：
容许误差——

```
if -15<erro_x<15 and -15<erro_y<15:
```

等待机械臂完成夹取时间——

```
print("stop to pick")
arm_state = "pick"
arm_pick_or_put_pub.publish(arm_state)
rospy.loginfo(arm_state)
time.sleep(30)
```

主要实现代码——识别标志牌部分

drive_decision_node_moveit.py | dp_drive_moveit.launch

```
elif sign_a_flag == 1 and picka_flag == 1:#recognize signA and in picking state
    print("put preparation")
    #rospy.loginfo(side_flag)
    car_stop = CtrlData()
    ctrl_pub.publish(car_stop)#stop to adjust robot position
    if signa_x > 320 :
        erro_x = signa_x - ((right_sign_xmin +right_sign_xmax)/2)
        erro_y = signa_y - ((right_sign_ymin +right_sign_ymax)/2)
    else :
        erro_x = signa_x - ((left_sign_xmin +left_sign_xmax)/2)
        erro_y = signa_y - ((left_sign_ymin +left_sign_ymax)/2)
    d_erro_x = erro_x - last_erro_x
    d_erro_y = erro_y - last_erro_y
    carmove = Twist()
    carmove.linear.x = -float(erro_y)*0.0003 - float(d_erro_y)*0.0
    carmove.linear.y = -float(erro_x)*0.0006 - float(d_erro_x)*0.0
    last_erro_x=erro_x
    last_erro_y=erro_y
    if -0.01 < carmove.linear.x < 0.0015 :
        carmove.linear.x = 0.0
    if -0.01 < carmove.linear.y < 0.0015:
        carmove.linear.y = 0.0
    if carmove.linear.x == 0 and carmove.linear.y == 0 :
        d_erro_x = 0
        d_erro_y = 0
    cmdvel_pub.publish(carmove)
    sign_a_flag = 0
    count2 = i
```

涉及参数：
底盘调整速度——

```
carmove.linear.x = -float(erro_y)*0.0003 - float(d_erro_y)*0.0
carmove.linear.y = -float(erro_x)*0.0006 - float(d_erro_x)*0.0
```

停止速度调整判断条件

```
if -0.01 < carmove.linear.x < 0.0015 :
    carmove.linear.x = 0.0
if -0.01 < carmove.linear.y < 0.0015:
    carmove.linear.y = 0.0
```

主要实现代码——放置木块部分

```
if d_erro_x == 0 and d_erro_y == 0:#put after adjustment
    move_flag = 1
    #print("ready to stop")
    carmove.linear.x = 0.0
    carmove.linear.y = 0.0

    if side_flag == 1:#if signA on left

        carmove.angular.z = -0.35
        cmdvel_pub.publish(carmove)
        time.sleep(6)
        carmove.angular.z = 0
        cmdvel_pub.publish(carmove)

        print("stop to put")
        arm_state = "put"
        arm_pick_or_put_pub.publish(arm_state)
        rospy.loginfo(arm_state)
        time.sleep(25)

        carmove.angular.z = 0.35
        cmdvel_pub.publish(carmove)
        time.sleep(6)
        carmove.angular.z = 0
        cmdvel_pub.publish(carmove)


        move_flag = 0
        print("put over")
        time.sleep(3)
        picka_flag = 0
```

涉及参数：

底盘自转速度——

```
carmove.angular.z = -0.35
cmdvel_pub.publish(carmove)
time.sleep(6)
carmove.angular.z = 0
cmdvel_pub.publish(carmove)
```

底盘自转时间——

```
carmove.angular.z = -0.35
cmdvel_pub.publish(carmove)
time.sleep(6)
carmove.angular.z = 0
cmdvel_pub.publish(carmove)
```

等待机械臂完成放置时间——

```
print("stop to put")
arm_state = "put"
arm_pick_or_put_pub.publish(arm_state)
rospy.loginfo(arm_state)
time.sleep(25)
```

WHEELTEC
轮 趣 科 技

THANK YOU

感谢聆听