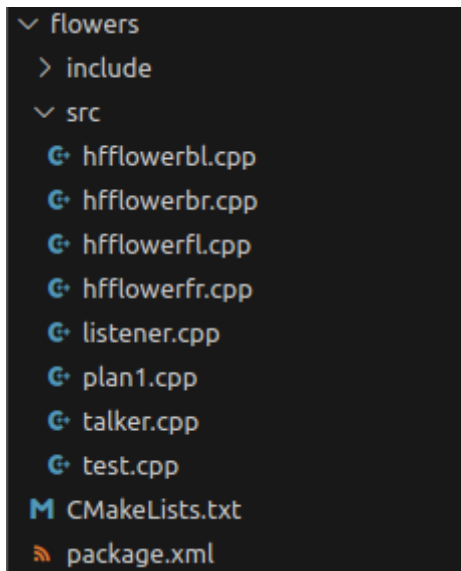
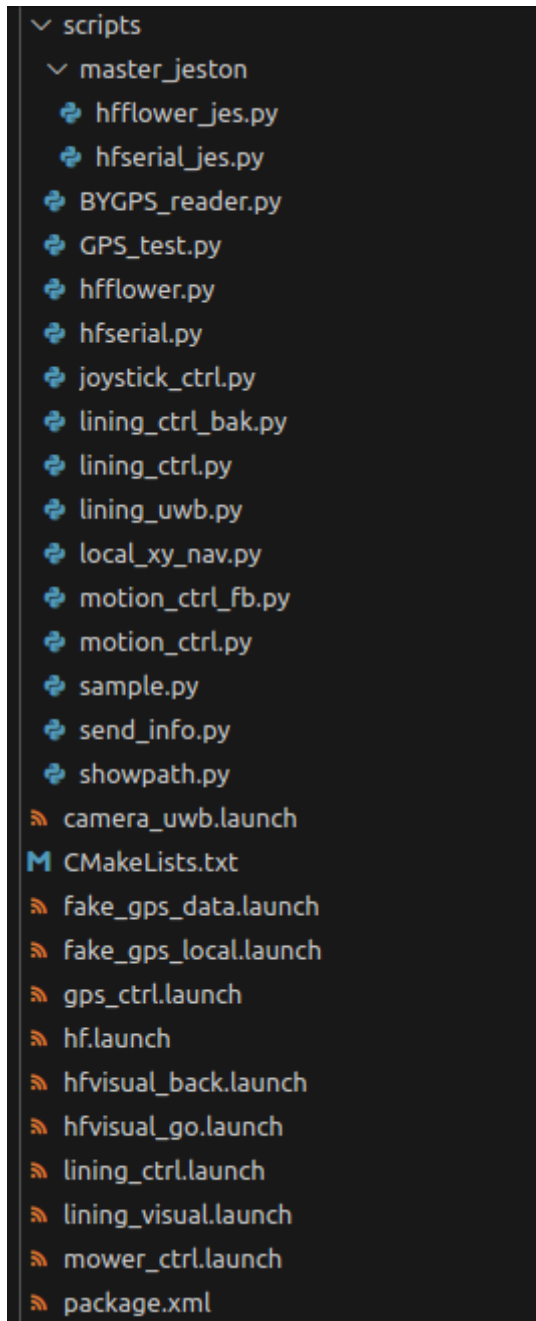


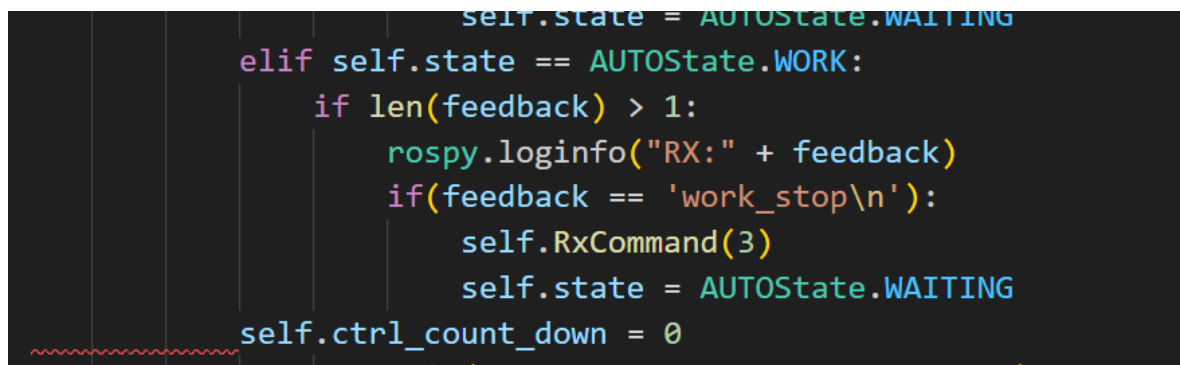
1、首先是我看到这个工作空间有个start.sh的文件，应该是一键启动程序的shell文件，但是我根据这个文件内容，发现这个shell文件只能启动一部分代码（即：hf.launch、hfflower.py、hfserial.py、hfvisual_go.launch、hfvisual_back.launch），剩下的并没有关联到，很疑惑，所以我现在的想法是发给我的可能不是最新的代码。





2、还有就是：vscode编译器报错问题：可能需要看一下工控机中的源码

vscode打开源文件报错：



进行缩进修改后：

```

158         if len(feedback) > 1:
159             rospy.loginfo("RX:" + feedback)
160             if(feedback == 'work_stop\n'):
161                 self.RxCommand(3)
162                 self.state = AUTOState.WAITING
163                 self.ctrl count down = 0
164             elif(feedback == '#set_mode,OK*\n'):
165                 self.RxCommand(4)

```

但是修改完会发现代码的逻辑就不一样了。一个在 if 语句中执行，一个在 if 语句外执行。


还有诸如这种：方法中的内容和方法缩进对齐了，这个不符合python语法。

```


231
232     def sub_mid_pos_Handler(self,Pose2D):
233         self.mid_sub_cnt = self.mid_sub_cnt + 1
234         self.beta_angle = Pose2D.theta - 36.2368
235         Pose2D.x += math.sin(self.beta_angle/180.0*math.pi)*0.9051
236         Pose2D.y += math.cos(self.beta_angle/180.0*math.pi)*0.9051
237         if(self.mid_sub_cnt == 10):
238             self.init_A_x = Pose2D.x
239             self.init_A_y = Pose2D.y
240             self.init_bis_theta = Pose2D.theta/180.0*math.pi
241             if(self.mid_sub_cnt >=20):

```


这个之前和王奇涛反映过：




王工使用什么编译器看的，我用的vscode出现了缩进问题



jetson nano里面的vim，拉到电脑的vscode是会有缩进问题



OK，王工方便再发一下那个程度吗，我把原来那个改了



电脑vscode改完缩进，nano运行就有问题，所以我一般直接再nano上写的

3、我整理的代码逻辑pdf中，之前也和老师说：有很多（基本没有说明）的变量，代码看了很多，单纯代码逻辑可以理解，但是为啥这么做很难和实物对应上，**所以想要问问老师，怎么解决这些问题？** 就比如说：

1. 自定义ros消息SensorValue.msg，可以看出这是传感器消息的传递，但是具体传递的啥，确实看不懂。

The screenshot shows the VS Code interface with the file explorer on the left displaying the project structure. The file `SensorValue.msg` is selected under the `msg` directory. The main editor shows the content of `SensorValue.msg`, which lists various ROS message fields with their data types and names.

```
src > mower_ctrl > msg > SensorValue.msg
1  uint8 col_data1
2  uint8 col_data2
3  uint8 col_data3
4  uint8 col_data4
5  uint16 ul_data1
6  uint16 ul_data2
7  uint16 ul_data3
8  uint16 ul_data4
9  uint32 distance
10 uint8 carpipe
11 uint8 rollpipe
12 float32 nowcar_speed
13
```

2. 还有代码中的变量：比如 `distance` 知道是距离，但是不知道是啥距离。

The screenshot shows the VS Code interface with the file explorer on the left displaying the project structure. The file `decision.py` is selected under the `scripts` directory. The main editor shows the content of `decision.py`, which defines a `decision` class with an `__init__` method. The class initializes various parameters and variables, including `linear_yk`, `linear_zk`, `linear_allk`, `near_ul`, `far_ul`, `rate`, `current_state`, `distance`, `last_distance`, `dis_reset_flag`, `nav_num`, `nav_num1`, `delta_d`, `delta_d1`, `speed`, `roll_speed`, `water_en`, `rx_num`, and `setmodenum`.

```
src > mower_ctrl > scripts > hfflower.py > decision > SpeedGoHandler
38 class decision:
39     def __init__(self):
40
41         self.linear_yk = rospy.get_param("~linear_yk", 0.75)
42         self.linear_zk = rospy.get_param("~linear_zk", 0.25)
43         self.linear_allk = rospy.get_param("~linear_allk", 0.5)
44         self.near_ul = rospy.get_param("~near_ul", 800)
45         self.far_ul = rospy.get_param("~far_ul", 1200)
46         self.rate = rospy.Rate(self.control_rate)
47
48
49         self.current_state = AUTOSTATE.WAITING
50
51
52         self.distance = 0
53         self.last_distance = 0
54         self.dis_reset_flag = 0
55         self.nav_num = 0
56         self.nav_num1 = 0
57         self.delta_d = 0
58         self.delta_d1 = 0
59         self.speed = 0
60         # self.roll_speed = 0
61         # self.water_en = 0
62         self.rx_num = 0
63         self.setmodenum = 0
64
```