技术日志：

**网络配置：**

Roscore连接不上，改source 改ip



设置完后，车能连上vscode连不上：更换vscode版本

Rviz拉点失败：在群内发布的常见问题Q&A中，重新配置网络至小车同一子网。

**电调：**

电调以1500为油门中值，上下门限可以自己设置，包括制动、拖刹等等。在官方提供的文件中有电调调节说明。

车子在低速状态下卡顿，是电调状态较差的情况，但是能凑合

该无刷电机的三极不在乎顺序，我们按ABC安装后发现反转，所以调换了其中两根。

**编译：**

Linux编译用catkin\_make, 在本车使用catkin\_make\_isolated

C++每次修改要编译，推荐使用python

但是python创建新文件需要chmod +x 接文件名

**例程代码：**

原车只有建图和导航的代码，gmapping通过调参可以解决大部分需求，导航使用move\_base，效果奇差，故不使用。

Gmapping与move\_base都是封装好的依赖包，难以修改，可以尝试找到修改方法，或者手搓

**本次比赛代码上遇到的问题：**

**雷达避障：**需要解决找到锥桶、拟合路径、路径跟踪的问题。

找锥桶是雷达避障的关键，2D雷达数据本身包含的信息较少、且模糊，在锥桶靠近的情况数据就是一坨、难以分析，只能按照赛道摆放锥桶的规则来设计判断。

一个重要问题是雷达本身在车子的摇晃下会扫射地面，得到很多难以滤去的线，这对我们的聚类算法以及gmapping造成极大的困扰。寻锥桶算法中，以路径方向为参考，滤去了雷达扫地的干扰，但是gmapping的重定位仍然会被这个影响。

**建图：**

建图由于频率不高、雷达扫地等问题一直成功率不高，这是导致我们完赛率不高的关键。我们通过修改车子的悬挂，让车子没那么摇晃或者刻意倾斜，能解决一些问题。

部分队伍直接将悬挂胶死。

在比赛场地，我们的建图成功率较高，因为场地存在的静态参照物多，能够平衡扫地带来的重定位问题。

在国赛时，湖北汽车工业学院说使用过gmapping及cartographer，gmapping完全可以应付需求，甚至29秒两圈的建图效果都颇好，只需要调参。这是我们仍未解决的。

**各环节的衔接：**

各环节衔接是最容易出问题的地方，进赛道的姿态、速度等等都会影响下一步的运动。

事实上在国赛5分钟写出来的代码问题就在于我们修改的那部分让停车稍微靠里，从而导致记的点靠里，由此影响第二圈出直道的位置、进弯道的姿态，以及导航到大弯道时的前瞻，导致最后几次发车都擦里面锥桶没能完赛（前瞻看到之前记的点由于修改靠里）

还有记点、读点方面极容易出bug，即使没有bug也会出现小概率直接乱跳flag的问题，该问题通常在车子发车几次后出现，推测是内存产生崩溃等代码内部问题。

总之是牵一发而动全身的

**导航以及控制算法：**

经过调试没有太大问题。

导航上面可以减少搜索计算量，可以将横向误差通过更好的方式计算出来

记点可以记中点，但是需要进行坐标转换，具体不清楚好不好用，

Stanley输出的差值可以套PID

导航使用了雷达和寻线的协同策略，在解决断路区进弯以及碰锥桶上比较有效，在此基础上可以想更多的策略

模糊PID也可以使用

山大的方案有点像导航，通过几次调整，慢慢让车过第一个第二个弯道直到最后跑完，具体调试点还是其他参数并不知道，甚至有点像人工智能。他们车的姿态显然不像在寻雷达的中点，反而像知道后面几个弯怎么跑，刻意调整现在的路径，赛车车手在碰到连续弯时用斯堪的纳维亚钟摆让车子保持好的姿态以高速通过。并且电机发出的声音非常不同，看着动力澎湃，可能更换了电调或电机。他们比赛时从未降速，最慢28s完赛，也只完赛了一次，风险较大。

对于固定的赛道，是否有更好的控制方案，让小车预知应该跑的位置和应该有的姿态？

湖北汽车工业学院的雷达加导航可以到达29s，具体算法未知

**模型转换**

pt转成Onnx时可能会有问题，后面换了代码解决了。

ROS1资源下载：

在尝试安装cartographer时，使用官方给的网站，但是很多已经因为没有维护停止了。

ROS2更新：比赛方提供给我们ros2镜像，可以重新烧写在我们车上，但是很多环境配置可能出问题，所以本次比赛没使用。

**本次比赛总结与反思**

做的好的：

1、规则研究较完整

2、完赛早，但是不断修改代码，发现新问题，开发新方案，让车子鲁棒性得到很大提升，完赛率也大大增加

3、比较团结，都想做好这次比赛

4、准备时间早，准备较充足。所有方案、规则都准备好能够满足。比如提前训好了模型，在比赛前半周突然说要发车后放红绿灯，我们就改了一晚上代码解决问题，由于模型已经解决，之前做过准备，所以速度很快

不足：

1. 临场经验太少，犯了临时改代码的大忌，并且因为太想国一害怕被罚时而一直调试新代码，而且一直不完赛，大脑很懵，没有想到换回原来的
2. 没有对比赛的发车轮次进行设计，因为太自信我们代码前两发就能跑好（事实也是如此），所以只设计了如何调参跑完这套最稳的，完赛后如何修改跑更快。但是这次改代码事发突然，没有想到在三四次发车失败后就应该赶紧换回原来的代码完赛一次。
3. 比赛时我太专注于我的抓车部分，除了分工明确外，每个人都应该有全局意识，要意识到一些显而易见的错误并及时提醒队友。而且由于和之前的调试完全不同，事发突然，就很蒙圈，更想不到跳出来看问题。
4. 应该想到抓车的应急策略，之前因为碰车罚时5s，我们调试时从来没有碰过，所以在比赛时根本想不到可以手动先修正一下，让车跑完，即使罚时也能完赛。
5. 没有坚持自我，看到改规则时只想着赶紧改代码，而且对自己太过自信，想着只是小改动，没想到牵一发而动全身。
6. 应该有经验论意识，本赛场以及操场上经过多次完赛成功检验的代码不用，而是用全新的、没有检验过一次的代码。