

🔗

开源！手把手教你搭建Arduino+树莓派的ROS小车（下）

原创

Lee

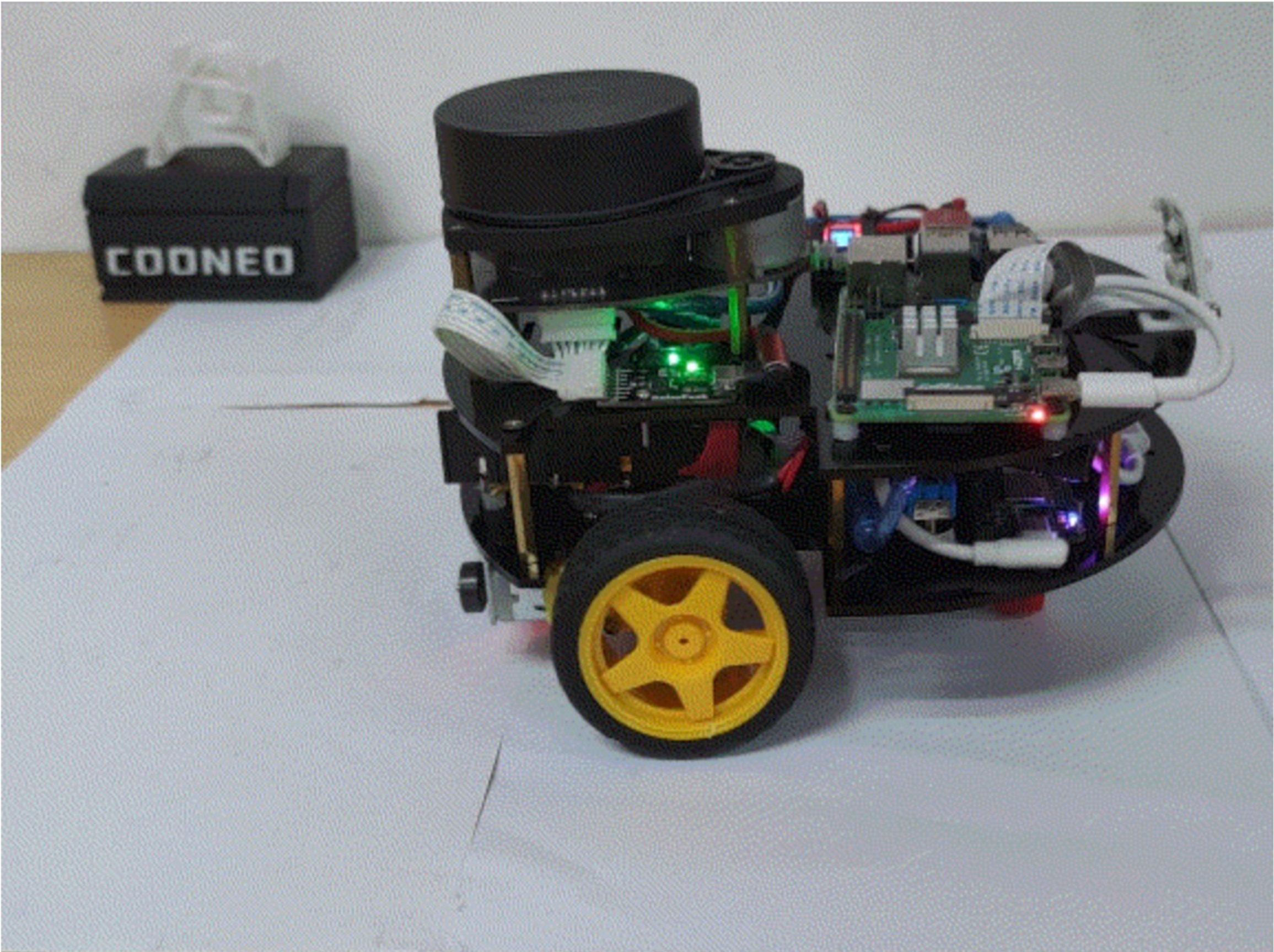
COONEO

2021年12月25日 21:20



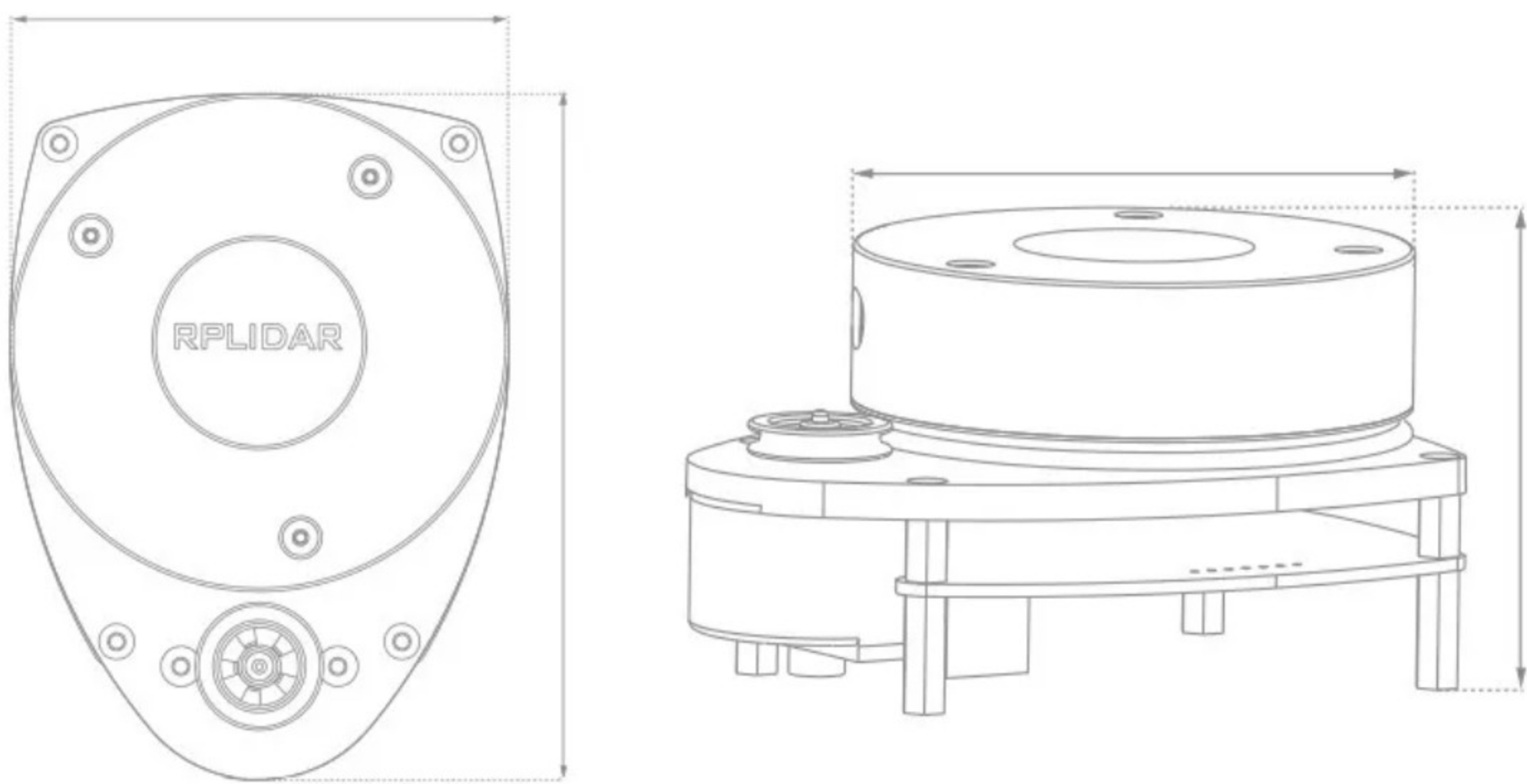
一、引言

在上一篇推文中，我们讲解了如何使用Arduino和树莓派等开源硬件搭建一台搭载ROS（开源机器人操作系统）的实车《开源！手把手教你搭建Arduino+树莓派的ROS小车（上）》。那么这一篇将继续手把手教你如何将ROS那些特别厉害的ROS功能包复现在咱的这一台实体小车上。



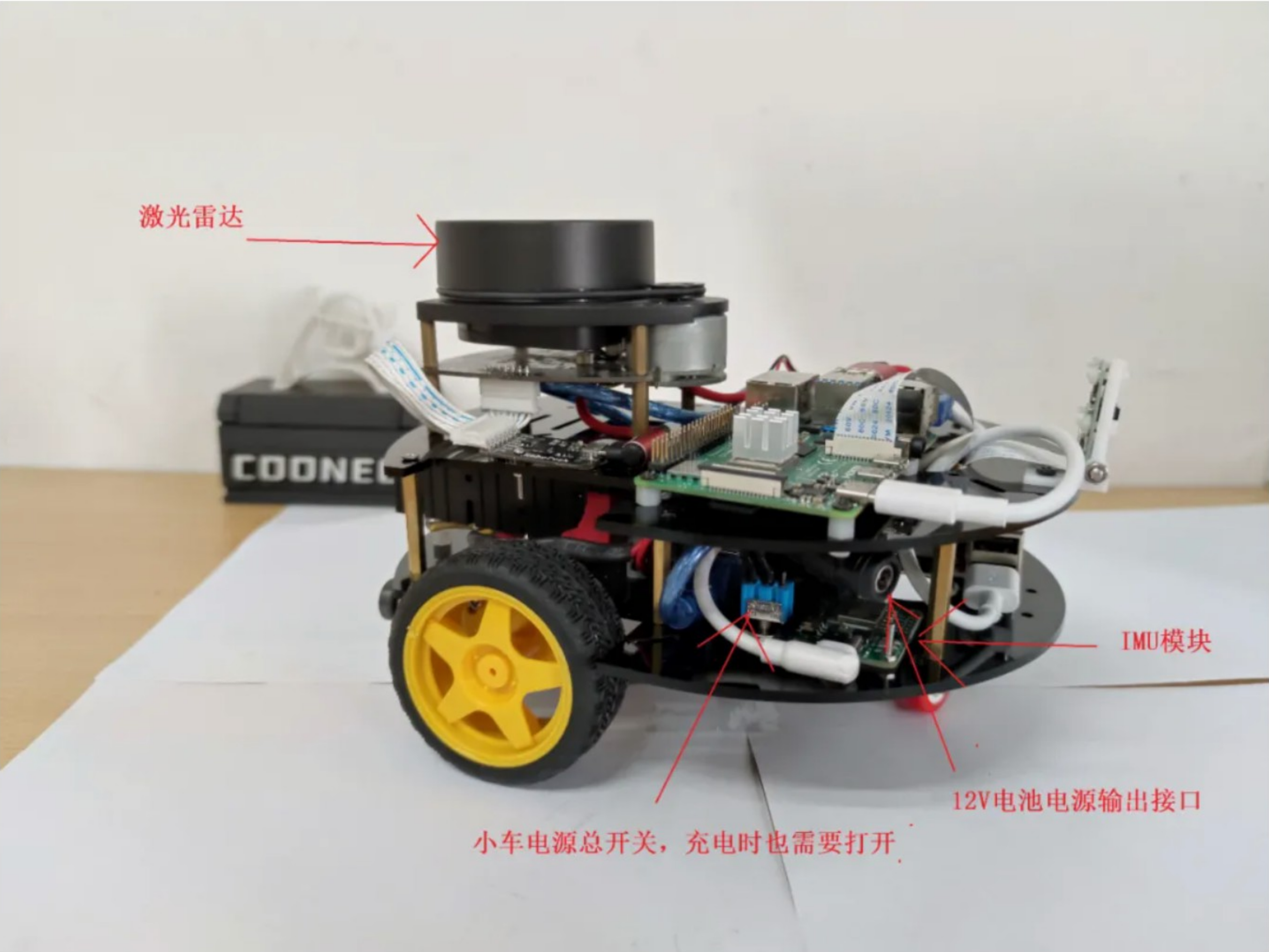
二、增加的传感器

1、激光雷达传感器



(From: SLAMTEC)

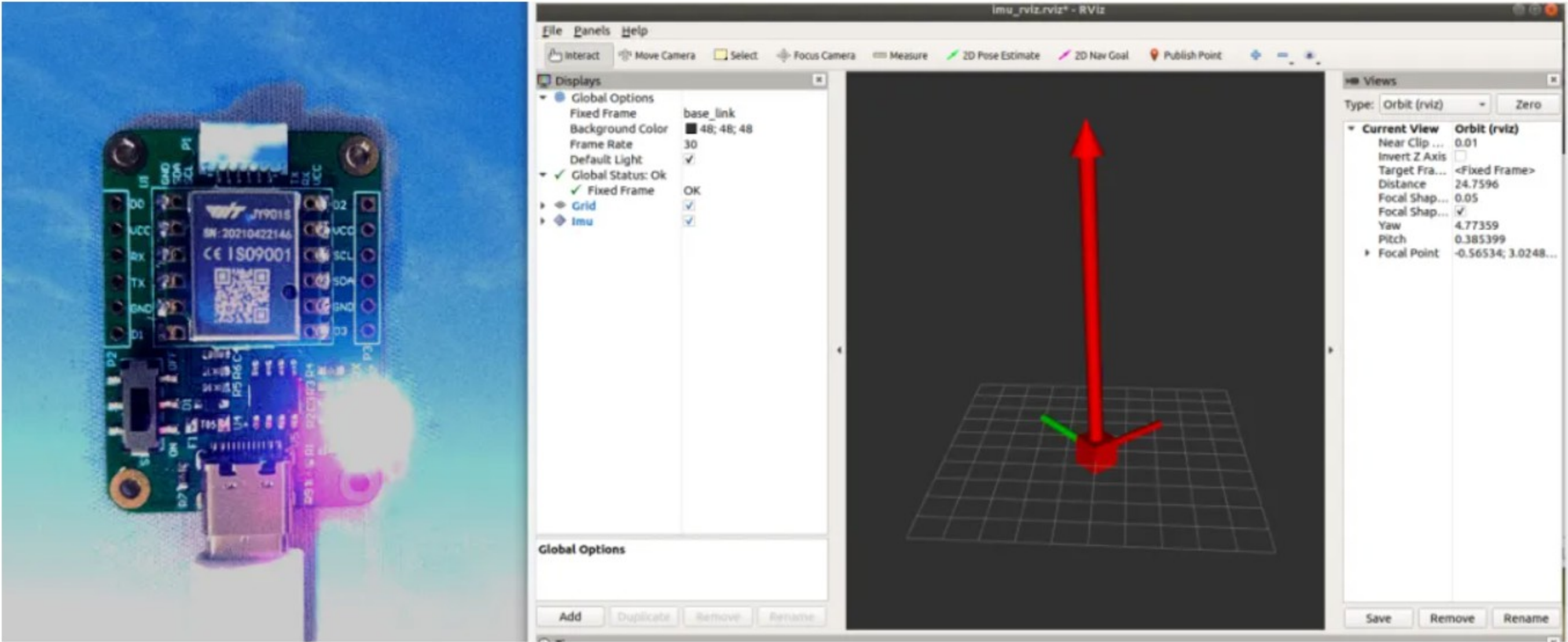
这里使用的是思岚科技的单线激光雷达A1M8，测距长度是8m，可测角度是360度，采用的是串口的通讯方式。安装位置如图所示：



2、IMU传感器

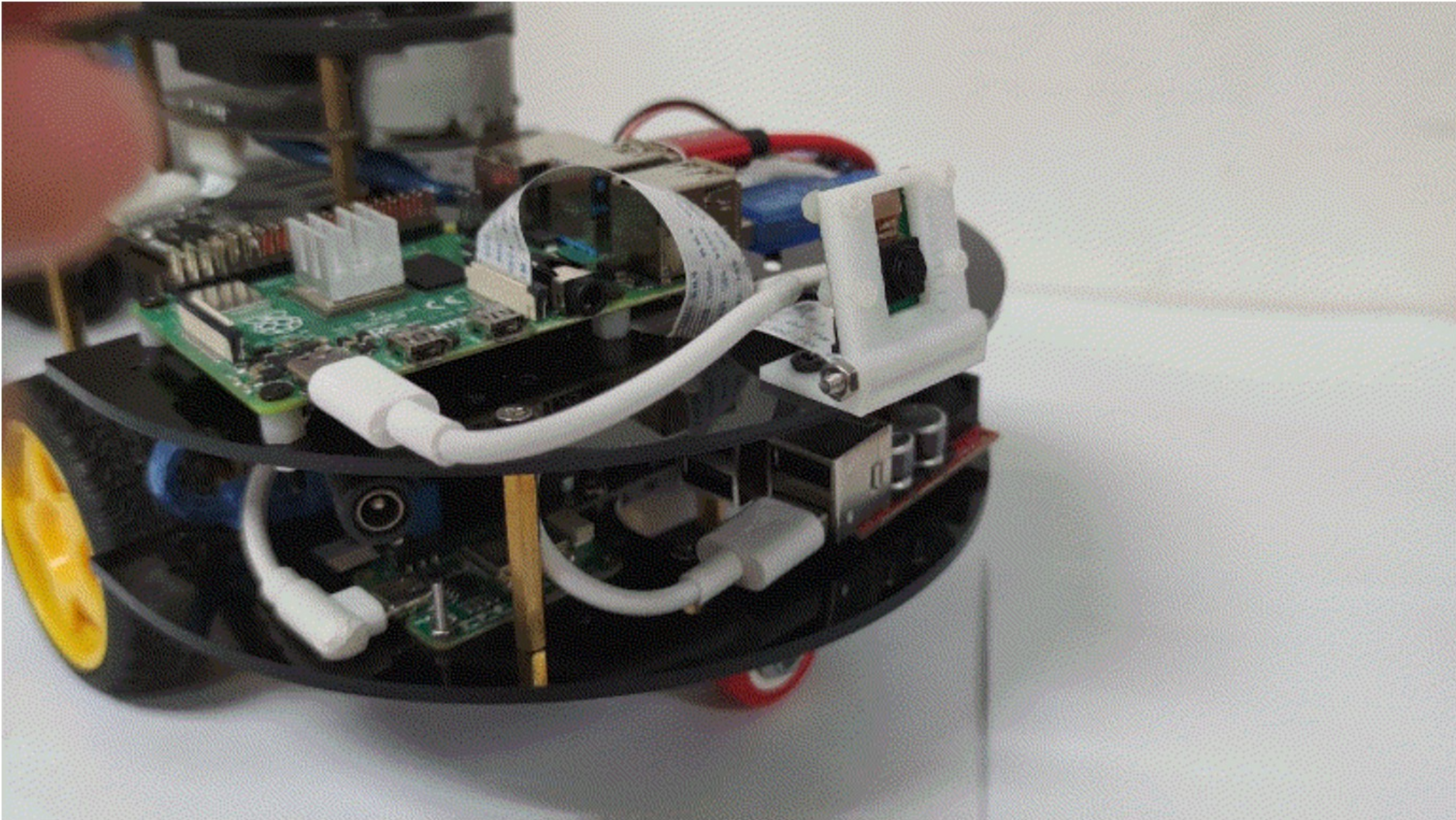
为了测量机器人小车的姿态和融合编码器里程计数据，这里使用了惯性测量单元IMU，其内部集成了三轴加速度计、三轴磁力计、陀螺仪以及温度补偿单元，输出的数据已经经过了一次数据滤波，能直接拿来供机器人使用。这里使用的是维特的9轴IMU模块，并根据其数据输出

通讯协议编写了对应的ROS解算程序，可直接发布Imu和mag话题数据。并将其安装在了小车前方靠右的位置（如上图箭头所示）。



3、摄像头传感器

在智能小车中，摄像头扮演的作用有：目标识别与检测、图像传输、视觉循迹。针对不同的应用环境，摄像头需要以不同的视角对准物体。为了达到这个目的，这里3D打印了一个简易的可调节角度的摄像头支架；使用的树莓派中常用的CSI接口摄像头。并将两者安装在车的最前端。





可旋转树莓派摄像头支架

COONEO旗舰店

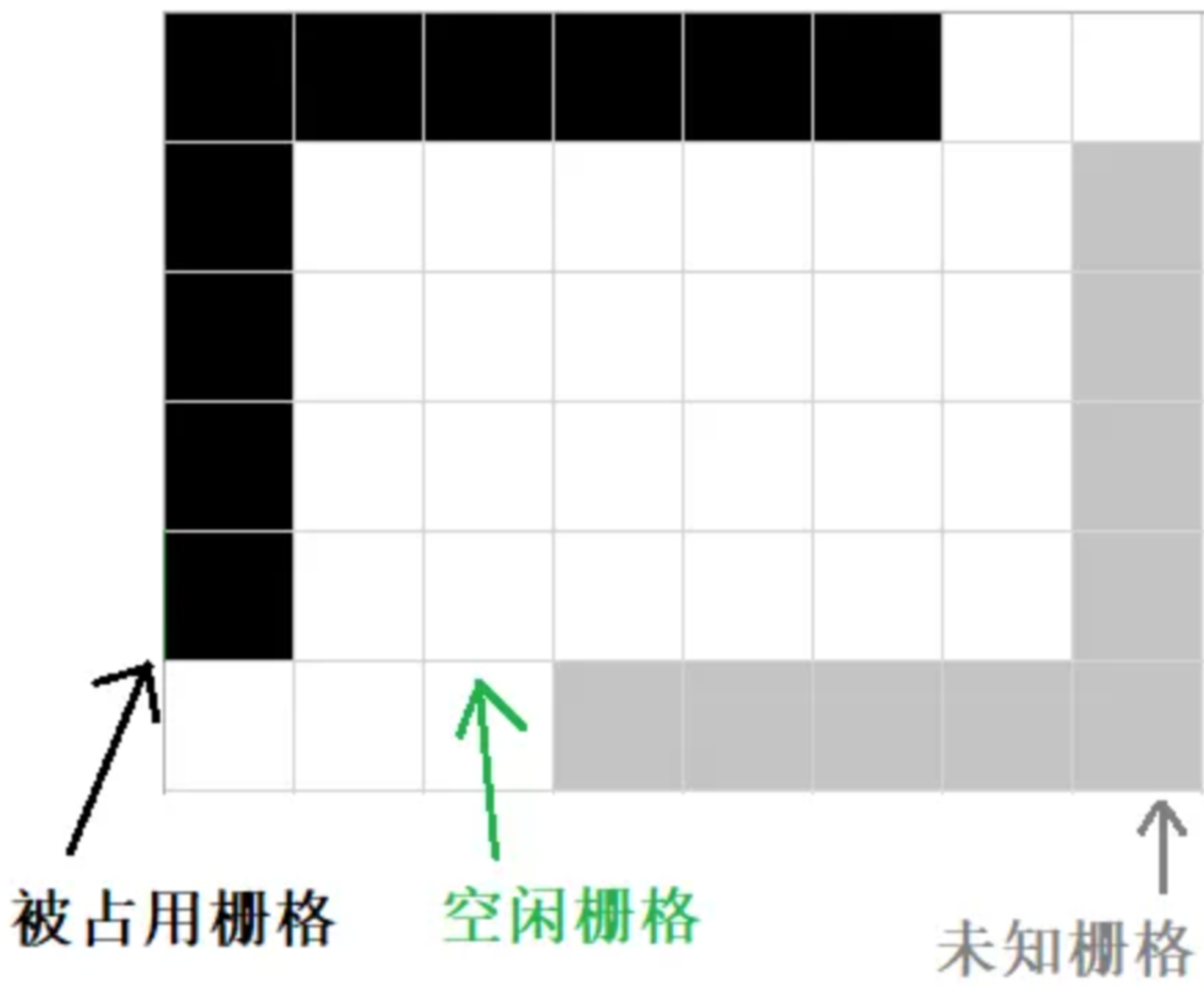
¥15

购买

三、Gmapping 建图功能复现

Gmapping是一种称为：“Rao-Blackwellized”的比较高的粒子滤波器，它可以通过激光雷达反馈的距离信息构建“栅格地图”。栅格地图是ROS机器人中的一种常用地图表示方式，其每个栅格具有三种状态：

- 1. 被占用栅格：表示该位置有障碍物
- 2. 空闲栅格：表示该位置没有障碍物
- 3. 未知栅格：地图还未更新的区域



根据Gmapping 在Wiki中的说明，该ROS功能包节点会订阅如下类型数据：
订阅：

- tf(tf/tfMessage)：静态的坐标变换，即需要订阅得到：激光雷达相当于与小车基坐标系的相对空间位置关系，里程计和小车基坐标之间的变换关系
- scan(sensor_msgs/LaserScan)：激光雷达的测距信息。

发布：

- map_metdata(nav_msgs/MapMetaData)：一种类型的地图数据
- map(nva_msgs/OccupancyGrid)：构建生成的栅格地图数据

实际操作如下：

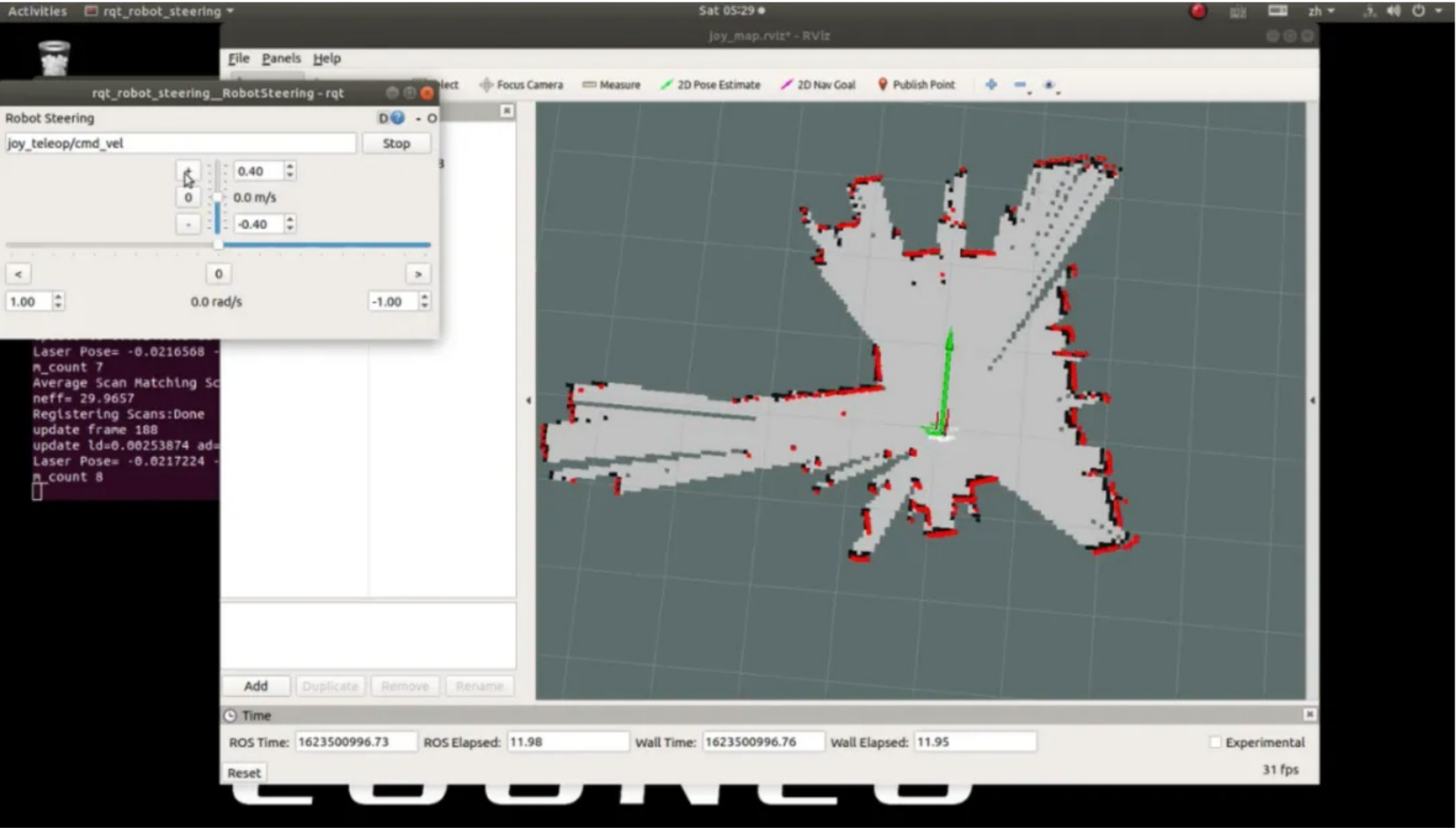
- 1. ssh进树莓派，然后运行gmapping建图程序，

开源！手把手教你搭建Arduino+树莓派的ROS小车（下）
https://mp.weixin.qq.com/s?__biz=MzIxNTAxMTg0Ng==&mid=2247486165&idx=3&sn=cf3136c3b...

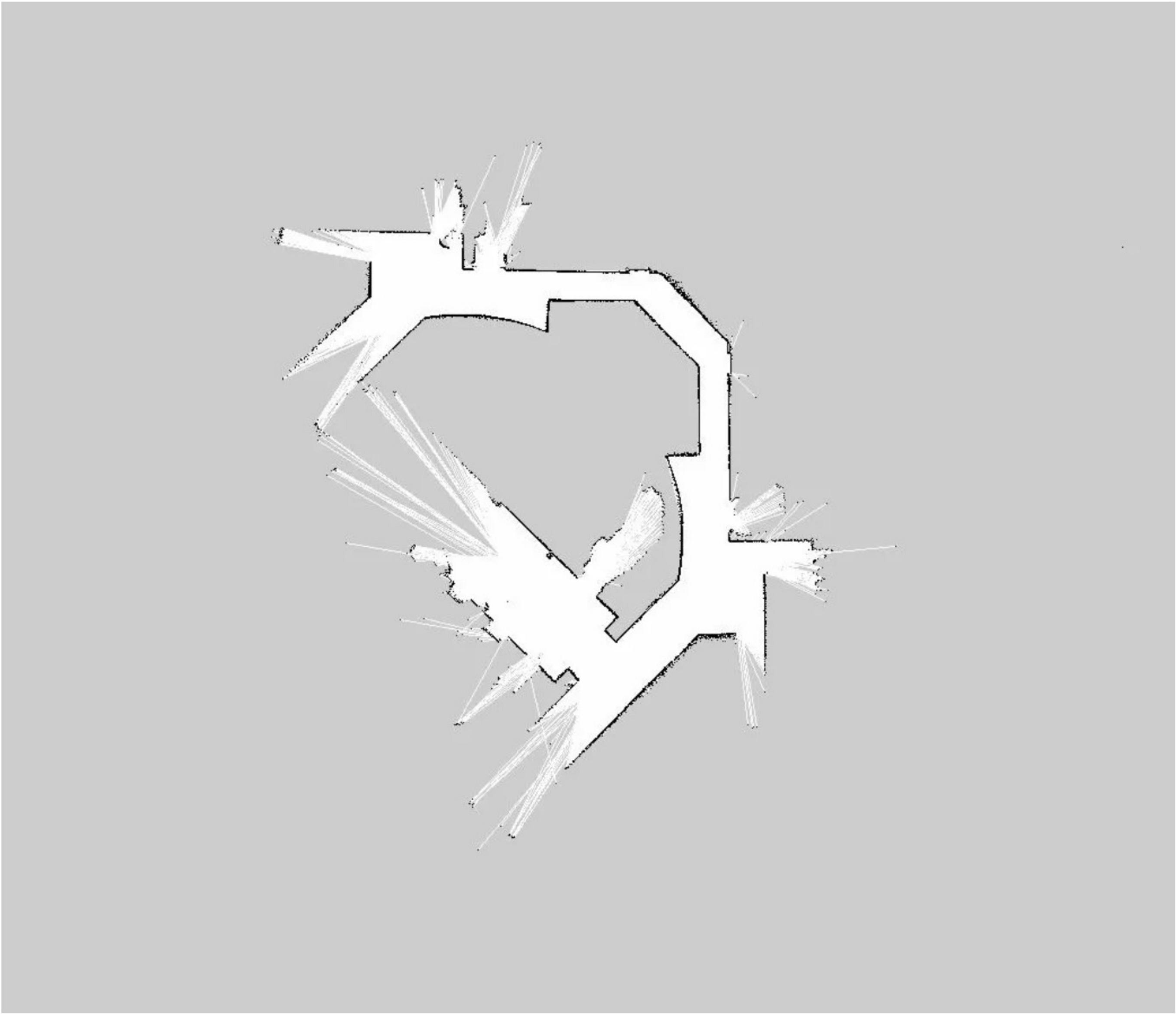
```
1 ssh cooneo@10.42.0.1      # 密码: cooneo
2 cd catkin_ws
3 roslaunch launch_file gmapping_ekf.launch
```

2. 在虚拟机终端中运行建图的观看程序

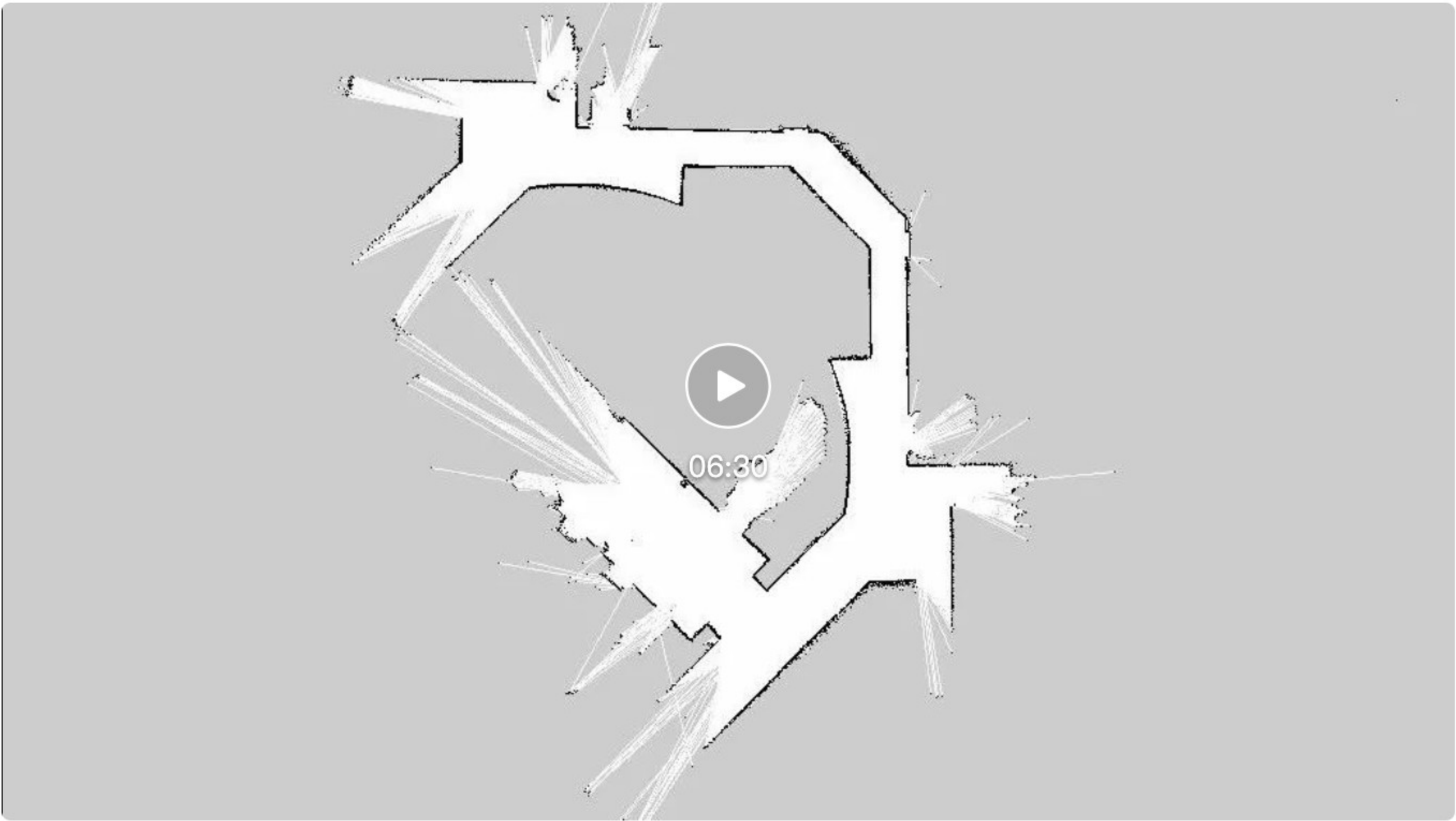
```
1 cd catkin_ws
2 source devel/setup.bash
3 roslaunch remote_gmapping joy_gmapping.launch
```



我们测试过一个比较大的场地，使用Gmapping建图效果如下图：

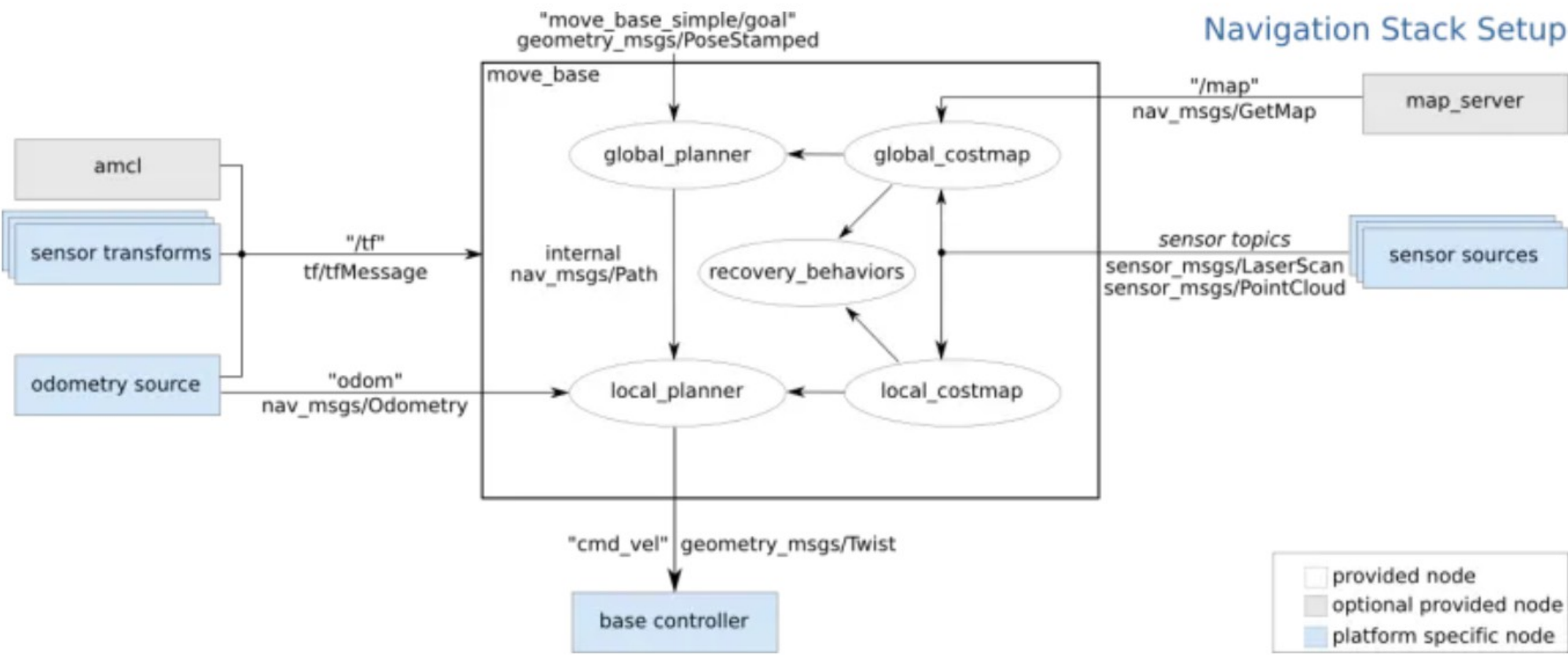


过程视频如下：



四、Navigation 导航功能复现

先看看 navigation/Tutorials/RobotSetup 中关于ROS导航栈的数据输入输出介绍：



(From navigation wiki)

我们从左往右开始看，amcl在这里主要是用于定位的ROS节点；下面是传感器之间的坐标变换、里程计数据输入；右侧上方的map_server节点主要是用来加载已将构建好的栅格地图、下方是激光传感器或者是其他点云数据输入；这些数据输入到navigation导航堆栈中，最后得出的输出是面向底层车控制器的节点的，输出的数据是包含线速度和角速度的速度话题数据，它直接控制小车行走。navigation内部集成了全局、局部规划器、地图膨胀、全局、局部代价地图，涉及的内容比较多，本次推文不深入分享，留到后面特别写一篇推文进行详解。

实际操作如下：

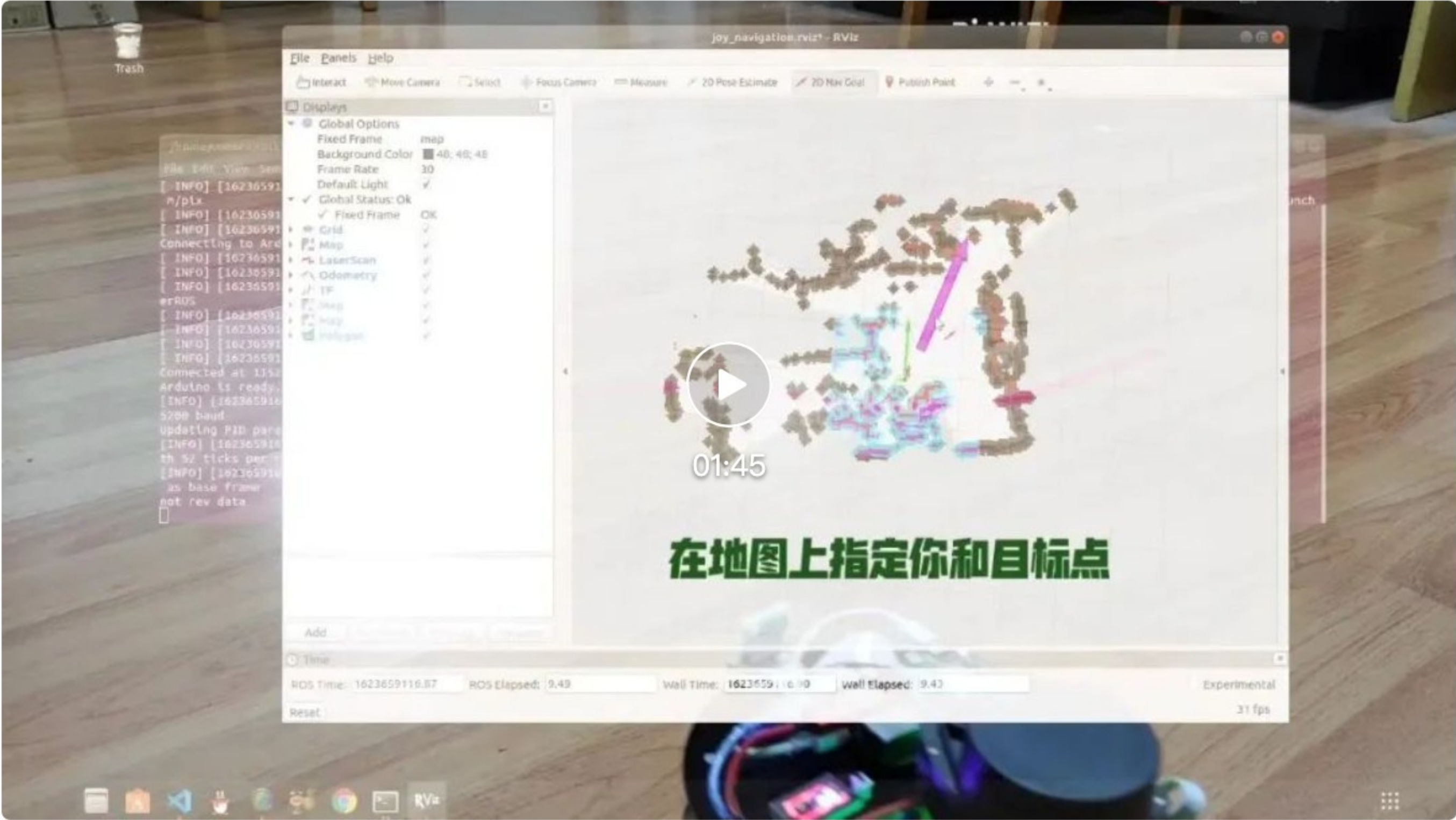
1. ssh进树莓派，启动导航程序

```
1 ssh cooneo@10.42.0.1      # 密码: cooneo
2 cd catkin_ws
3 roslaunch launch_file navigation_ekf.launch
```

2. 打开虚拟机终端中的导航观看程序

```
1 cd catkin_ws
2 source devel/setup.bash
3 roslaunch remote_gmapping joy_navigation.launch
```

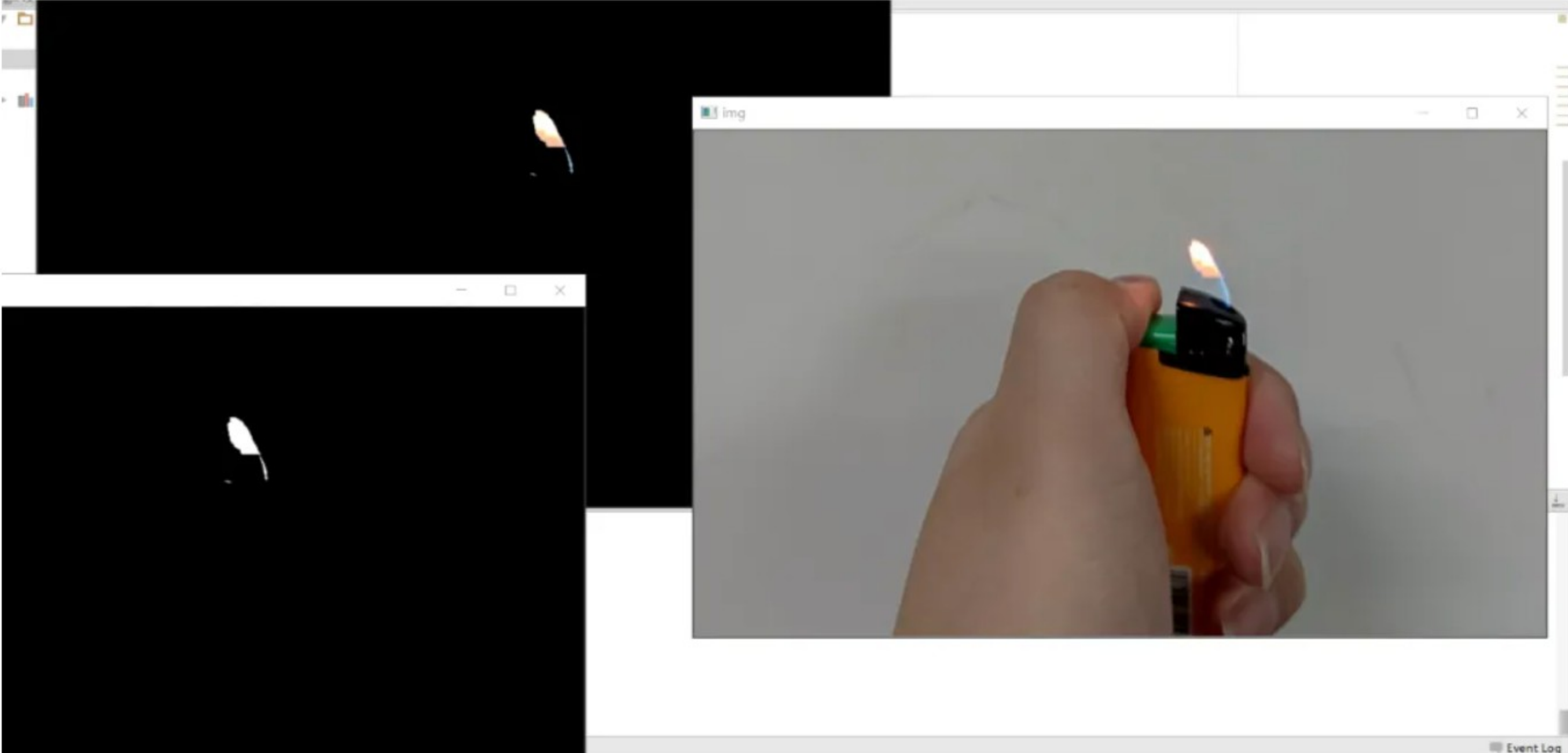

其实际运行效果如下视频所示：



五、火焰识别与局域网图像传输的实现

opencv是一个很好的关于图像处理的库，多年来，许多无私奉献的大神们已经编写好了许多底层的库函数，让初学者可以直接调用，快速实现复杂的功能。我们这次就是利用opencv中的色彩空间转换、图像掩膜等API，使用python ROS语法规则，编写了一个基于颜色的火焰识别ROS节点程序。

该程序通过订阅树莓派摄像头话题数据，然后对其进行数据处理，提取出火焰部分的颜色，通过求解该部分颜色的有无和所占像素点的大小作为判断是否有火产生，过程如下：



实际操作如下：

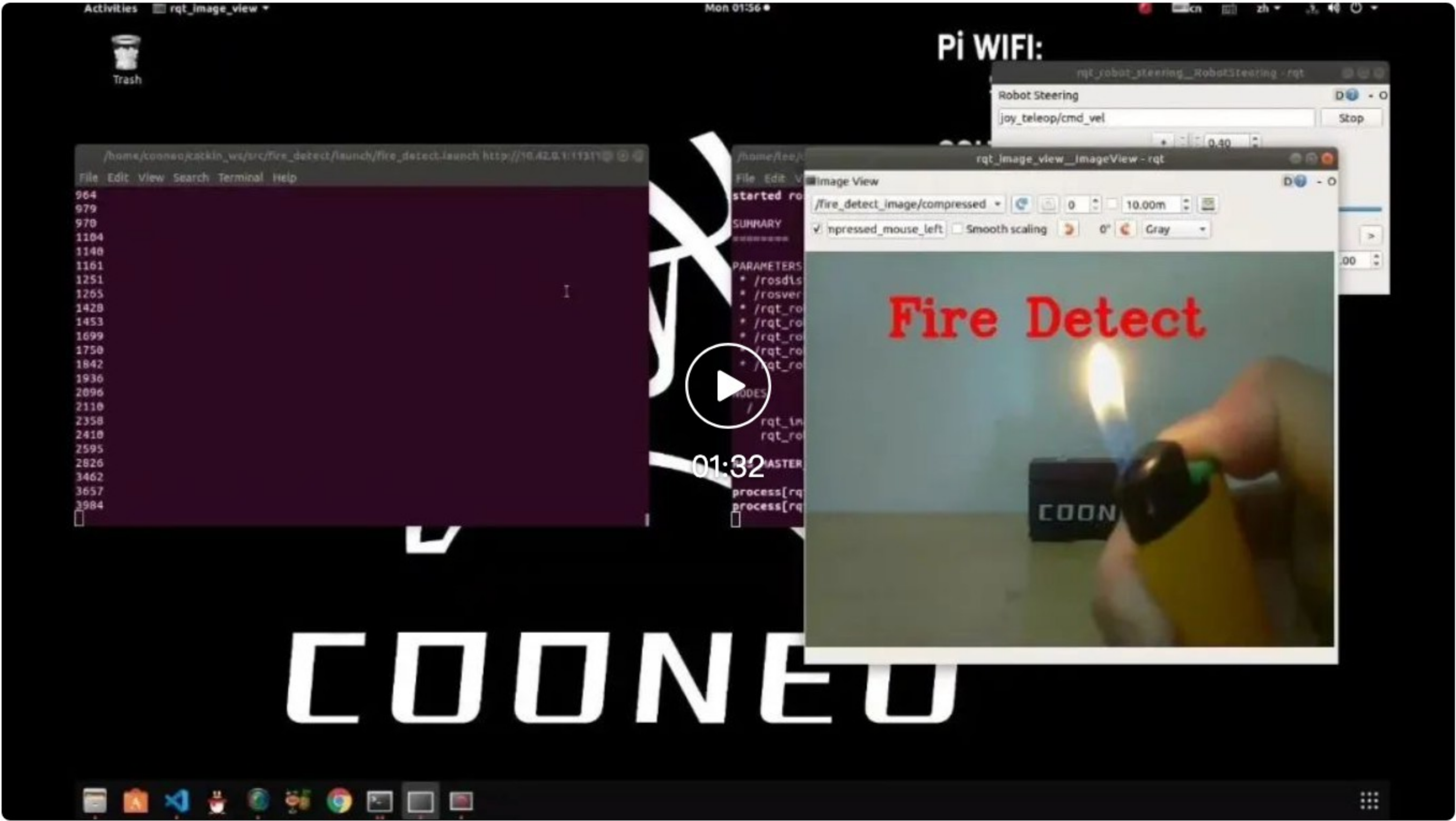
1. ssh进树莓派打开火焰检测识别程序

```
1 ssh cooneo@10.42.0.1      # 密码：cooneo
2 cd catkin_ws
3 roslaunch fire_detect fire_detect.launch
```

2. 打开虚拟机终端，启动火焰识别效果观看程序


```
1 cd catkin_ws
2 source devel/setup.bash
3 roslaunch remote_gmapping joy_fire_detect.launch
```

整个过程的演示视频如下：



想快人一步玩起来？我们提供少量文中提到的套件、转接板和可旋转树莓派支架，可进入小店购买：





ROS小车套件

COONEO旗舰店

¥549

购买

六、展望与彩蛋

展望：

- 在后面的推文中，我们将分享：
1. 在ROS下，如何使用树莓派摄像头进行地面线条循迹；
 2. 通过树莓派实现ROS应用的过程中常见的配置方法总结；
 3. navigation 导航栈中的众多配置文件参数修改方法与建议；