O1Lite快速启动手册-v3.3

1. 简介

本文档意在指导用户在拿到设备后在ubuntu系统中成功运行O1Lite驱动，并成功拿到数据。

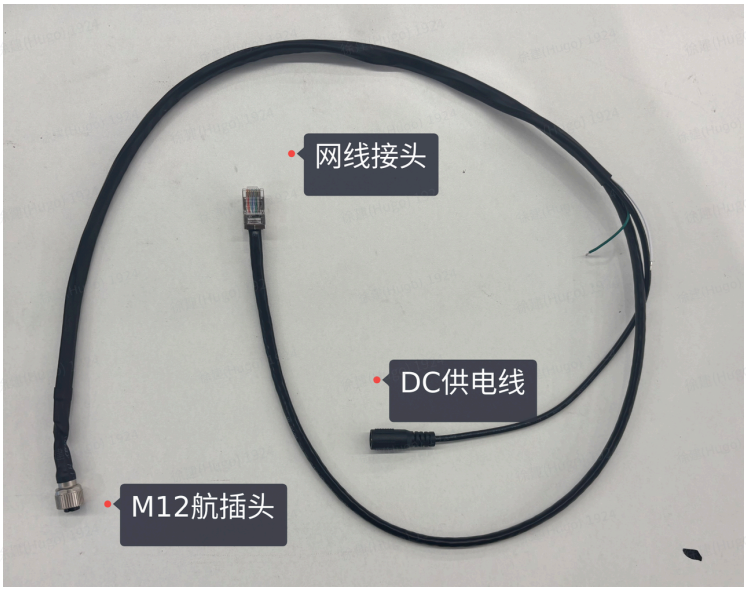
2. 固件版本/驱动版本

序号	固件版本	驱动版本	备注
1	v3.x	v3.x	增加录包说明

3. 硬件组成及安装

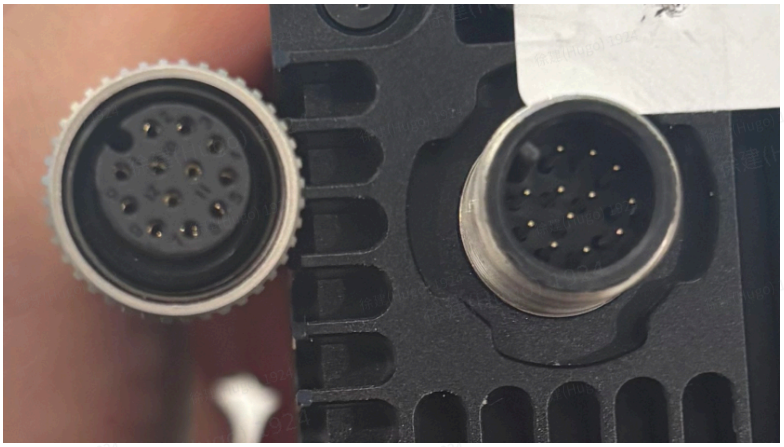


O1Lite主机



电源数据一分二航插线

- 将数据线航插头按照卡槽对应插入O1Lite，并锁紧连接



- DC头接12V2A适配器，网线水晶头插入PC

4. PC端配置

4.1 系统要求（请务必确认电脑已正确安装ros）：

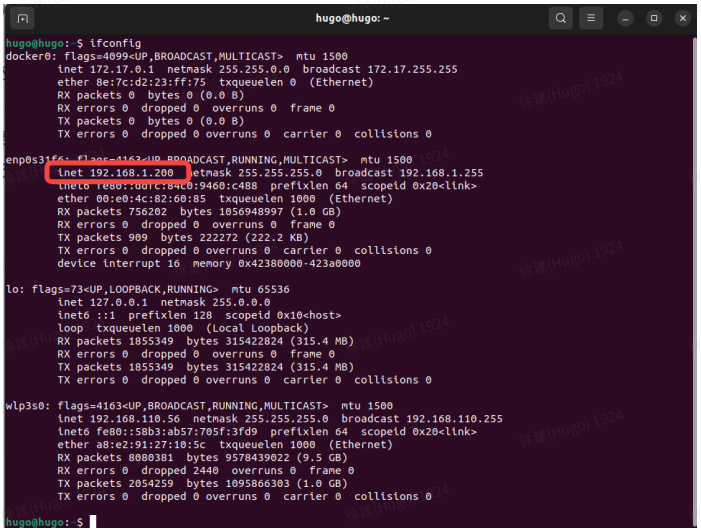
- ubuntu 20.04 / 22.04 / 24.04
- ROS1 Noetic / Iron
- ROS2 Humble / Jazzy

4.2 环境配置

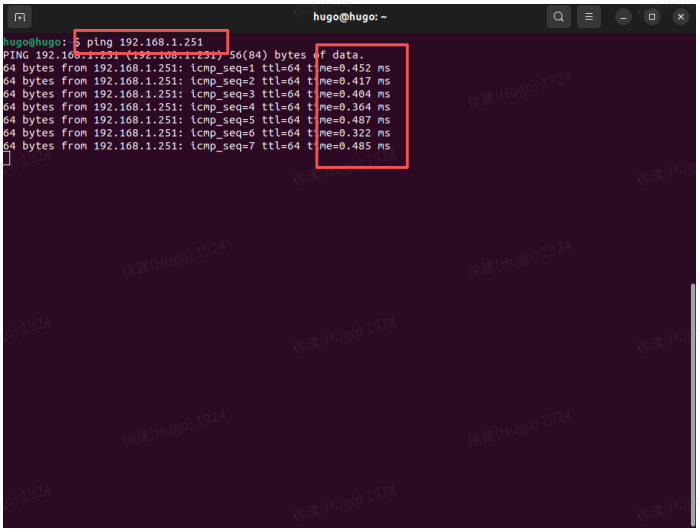
- 将网线接入PC后，配置电脑ip与O1Lite模组同一网段。O1Lite模组ip：192.168.1.251（模组ip可根据客户需求更改）



- 确定ip已生效，并检查是否与O1Lite正常通信

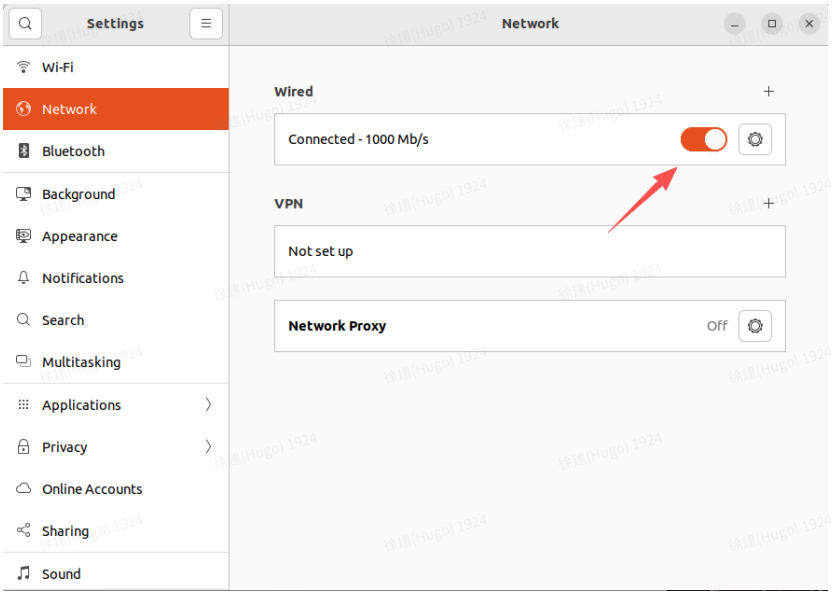


ifconfig查看ip是否生效



ping 192.168.1.251

- 若不能正常通信即ping不同，请重新开关ip设置，使其生效。



- 运行驱动之前，请设置网卡缓冲区大小，避免缓冲区过小导致与Odinlite的通信压力过大，出现连续的通信丢包影响建图/定位。同时设置DOMAIN_ID

```
代码块
1   sudo sysctl -w net.core.rmem_max=16777216
2   sudo sysctl -w net.core.rmem_default=8388608
3   sudo sysctl -w net.core.netdev_max_backlog=50000
4
5   export ROS_DOMAIN_ID=*     # *取值 1~99任一数字
```

5. 获取驱动步骤

- 联系到留形售后技术支持人员
- 提供公司信息、用户信息、邮箱
- 技术支持将提供gitlab账号
- 点击下方gitlab链接，使用提供的gitlab账号登录获取驱动（登录后请尽快修改登录密码）

驱动链接：https://git.manifoldtech.cn/manifoldtech-sdk/odin_ros_driver2

注意：上述链接中驱动版本为0.12.0，如果需要3.x的驱动（搭配3.x的固件）请联系留形技术支持

6. 驱动运行

6.1 创建ros工作空间

```
代码块
1   mkdir -p catkin_ws/src
```

6.2 将driver拷贝到~/catkin_ws/src中

- 手动拷贝：手动将ros_driver_v*_release_*.tar.xz（其中*表示版本号与对应的发布时间，请根据您手上的驱动名称拷贝）拷贝到新创建的catkin_ws/src目录下并手动解压，解压后将内部的ros_driver放到src路径下。
- 命令行拷贝（以驱动当前目录为~/Downloads为例）

```
代码块
1  cp ~/Downloads/ros_driver_v0.7.0_beta_20260415.tar.xz ~/catkin_ws/src
2  cd ~/catkin_ws/src
3  tar -xvf ros_driver_v0.7.0_beta_20260415.tar.xz
4  mv ./ros_driver_v0.7.0_beta_20260415/ros_driver ./ && rm -r
   ros_driver_v0.7.0_beta_20260415
```

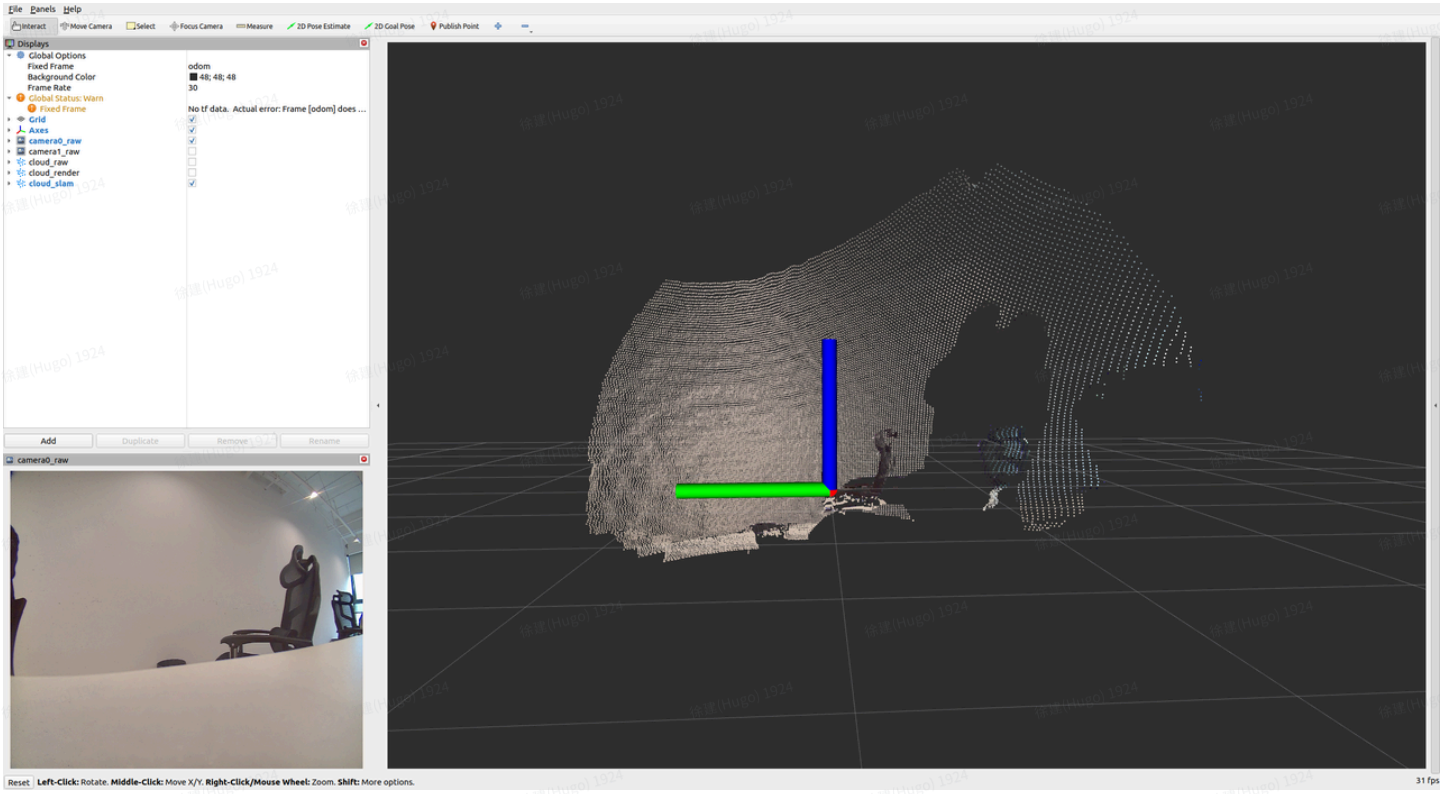
6.3 编译（以ros2为例）

```
代码块
1  cd ~/catkin_ws
2  ./src/ros_driver/script/build_ros2.sh
```

6.4 运行驱动

```
代码块
1  cd ~/catkin_ws
2  source install/setup.bash
3  ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py
```

5. 成功运行后rviz显示点云及图像



7. 参数说明

1. Confidence: 含义是 信噪比 数值越大 噪点滤除越多 但可能存在误检测 实际适用中建议在保证噪点没有的情况下 数值尽可能小，室内场景建议设置4。
2. 修改方式：

运行驱动后终端输入打开rqt，在confidence参数处进行设置。

8. 模式切换

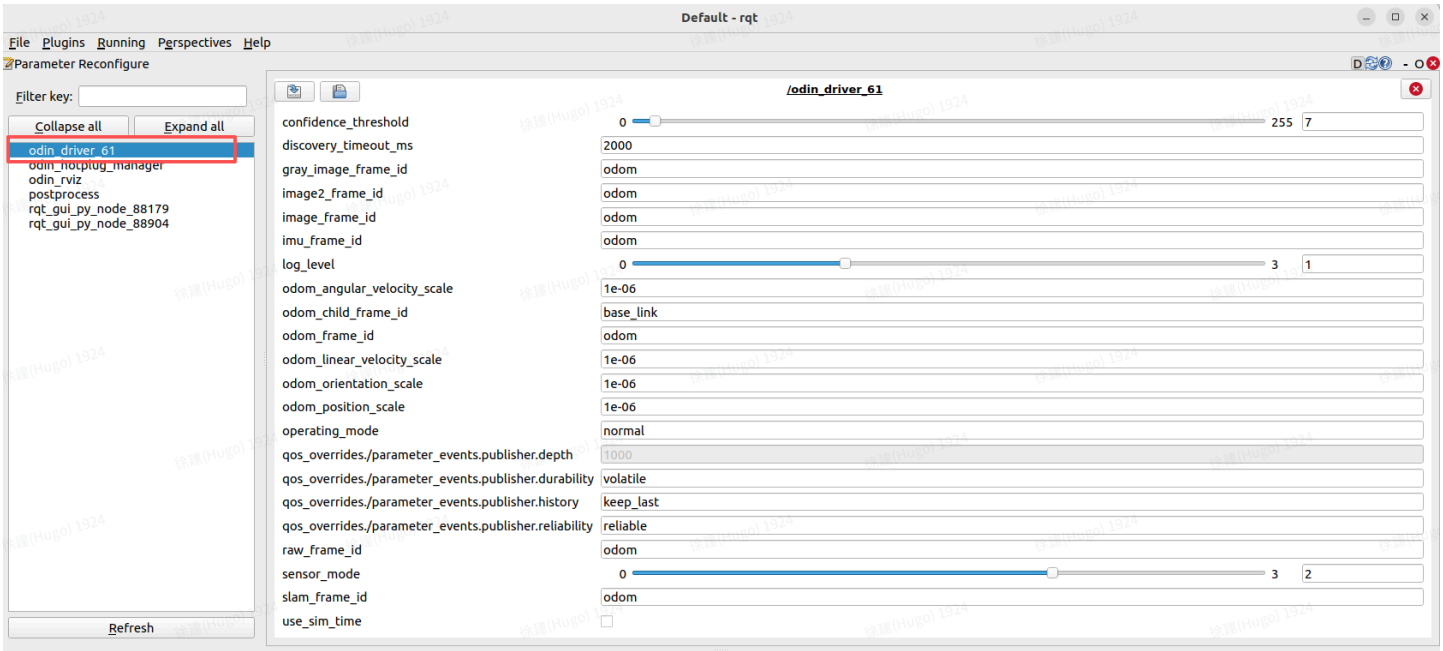
8.1 High peak/Low peak长短帧模式说明

High peak长帧模式：优化过强光照下雷达接收信号质量，近距离精度低，建议户外场景使用

Low peak短帧模式：优化近距离小空间测距不准和噪点问题，受强光影响大，建议室内场景使用

8.2 界面操作切换模式

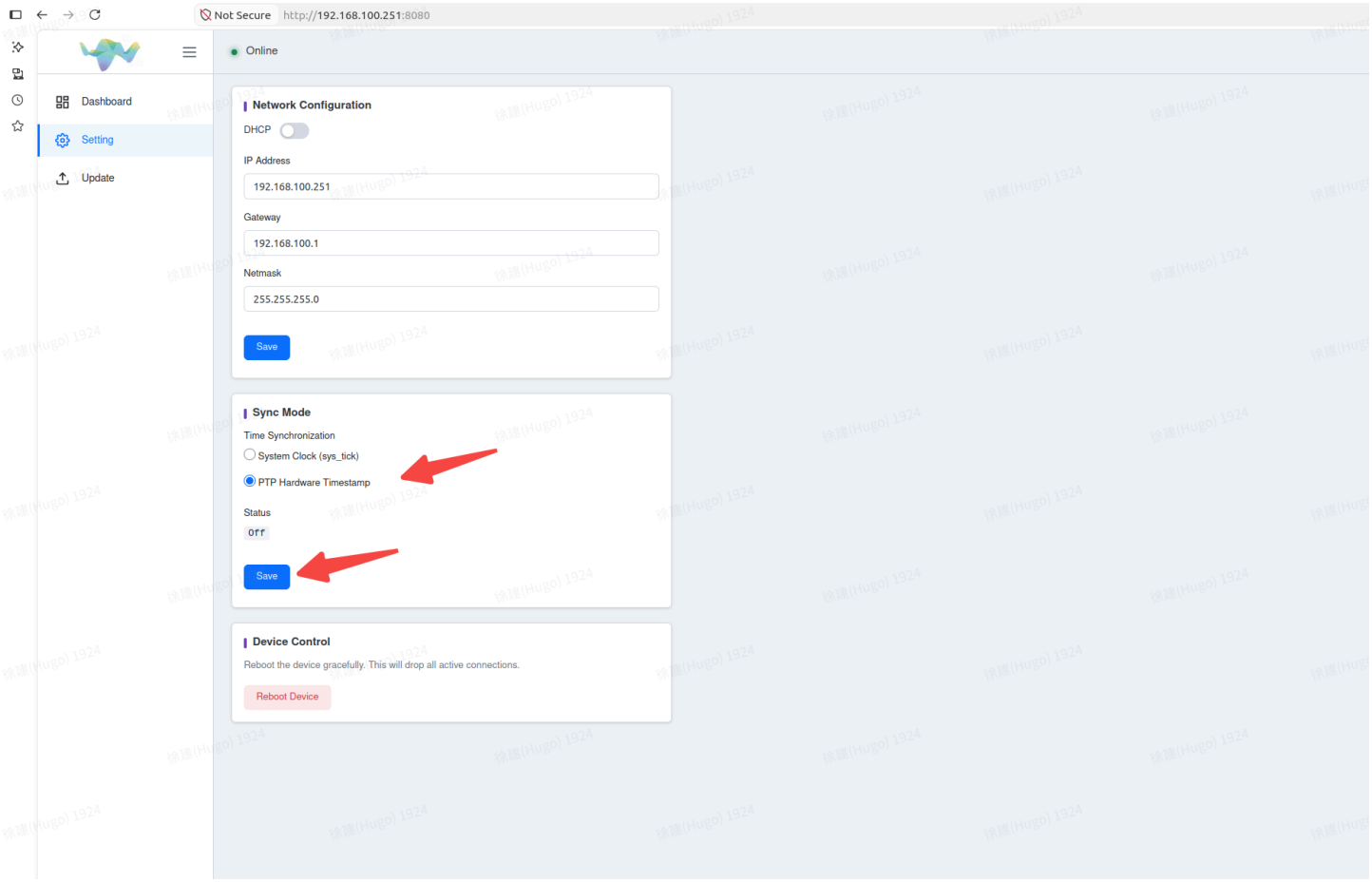
1. 命令行输入 rqt
2. 运行驱动后会自动读取odin_driver配置界面



3. 通过鼠标点击sensor_mode选择模式，1代表high peak，2代表low peak，3代表adaptive（自适应）模式。
4. 自适应模式介绍：
   a. 自适应模式为3.1.2固件新增功能，如果想使用该功能需要升级对应固件；
   b. 自适应模式会根据场景自行切换high peak和low peak模式；
   c. 自适应模式提供两个参数设置：
      i. filter_adaptive_cdf_percentile：单位 0.01%，取值范围1~10000。含义：统计整个场景中的点云数量，默认值7000，即以整个场景中所有点的70%为准，需配合outdoor_threshold使用；
      ii. filter_adaptive_outdoor_thr：单位 mm。含义：距离值，默认3000mm，即以3m为基准。搭配cdf_percentile使用。-
      iii. 上述两个参数搭配使用含义：识别整个场景中所有点云，如果70%的点的距离在3m以内，则认为是近距离/室内场景，此时会自动切换至low peak模式，如果70%的点的距离在3m以外，则认为是远距离/室外场景，此时会自动切换至high peak模式

9. PTP时间同步

1. 打开dashboard：浏览器输入：192.168.1.251:8080
2. 进入设置界面，选择PTP ，点击Save，保存完成后点击最下方的Reboot Device等待设备重启成功：



3. 请使用支持PTP时间同步的交换机。若只是测试功能，可以使用ptp4l进行软同步，操作如下：

a. PC端安装ptp4l

代码块

```
sudo apt update
sudo apt install linuxptp
```

b. 查看网卡

代码块

```
ifconfig        # 例如查询到连接O1Lite的网卡为：enp0s31f6
```

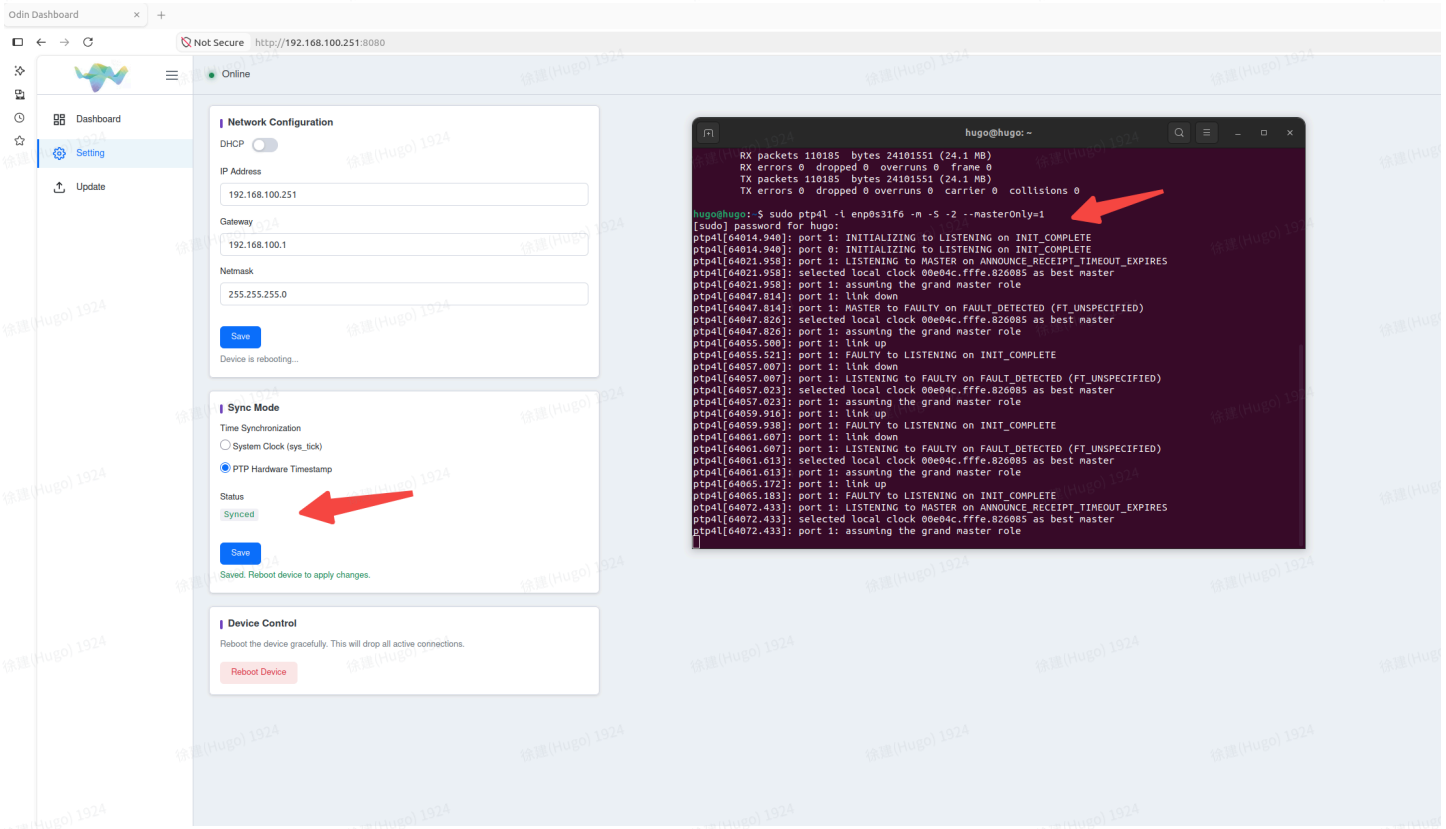
c. 运行master的ptp服务

代码块

```
sudo ptp4l -i enp0s31f6 -m -S -2 --masterOnly=1
# -i 表示指定网卡
# -m 将ptp信息直接打印到终端
# -S 软件时间戳，如果硬件支持，可以使用-H
# -2 使用 Layer 2 Ethernet 方式传输 PTP
```

```
6    # --masterOnly=1   强制本机作为 PTP Master，不参与 BMCA 选举成为 Slave
```

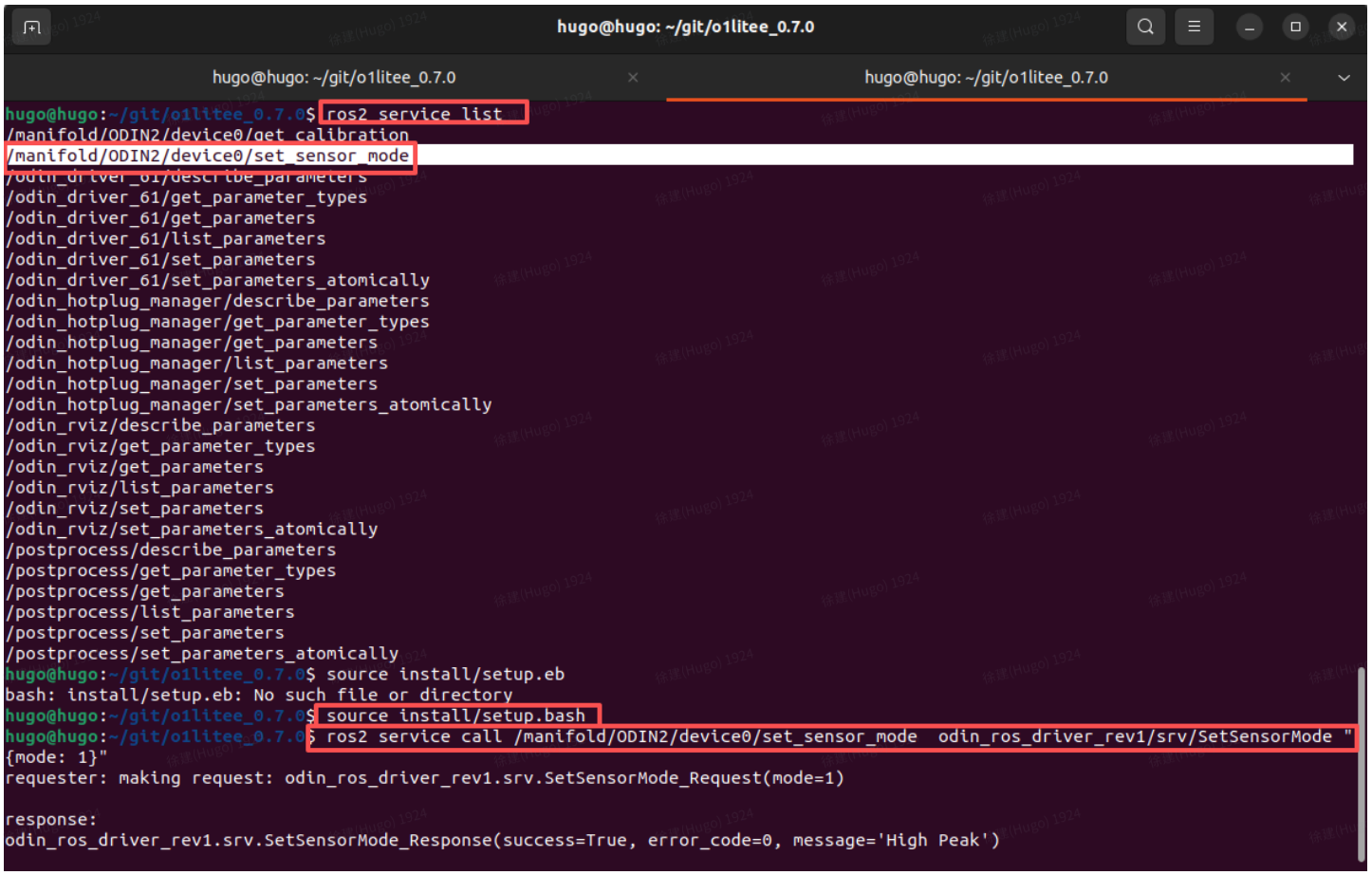
4. 同步完成后可看到dashboard显示：



5. 此时再去打印发布出来的话题时间戳已经和PC保持一致。

9.1 命令行切换模式（ros2为例）

1. 输入 ros2 service list 查看服务列表，找到/manifold/device0/set_sensor_mode是否发布，如图

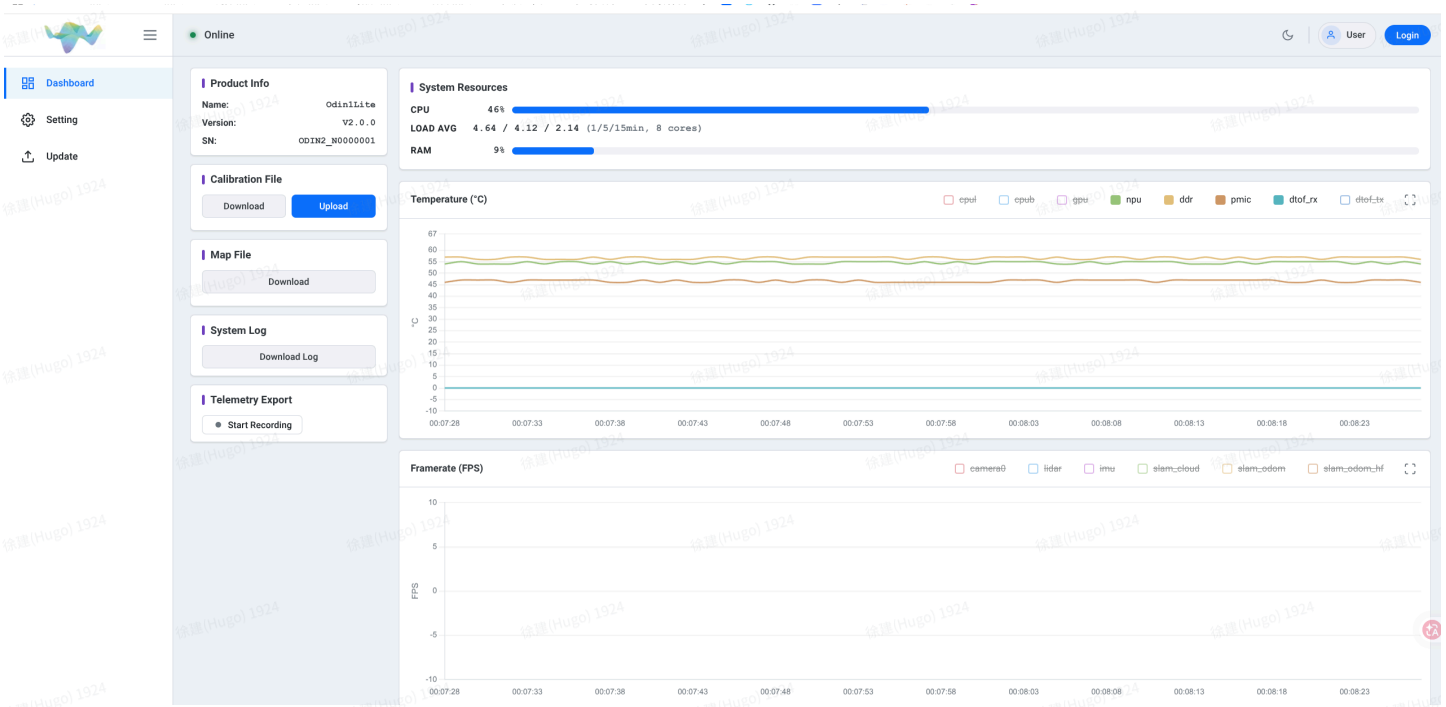


2. 在~/catkin_ws/中终端输入：source install/setup.bash
3. 终端继续输入：ros2 service call /manifold/device0/set_sensor_mode odin_ros_driver_rev1/srv/SetSensorMode "{mode: 1}" # mode: 1 high peak mode: 2 low peak mode: 3 adaptive

10. Odin HTTP Dashboard使用

10.1 访问方式

1. 请确保PC与Odin1Lite设备处于同一网段
2. 浏览器打开 `http://<设备IP>:8080` （默认 `http://192.168.1.251:8080`）
3. 推荐使用Chrome / Edge 等浏览器



10.2 功能总览

功能模块	页面	说明
设备信息	Dashboard	产品名称、固件版本、序列号
系统资源监控	Dashboard	CPU 使用率、内存使用率、系统负载（1/5/15min）
温度实时图表	Dashboard	SoC 各温区 + 传感器温度（动态折线图，1Hz 更新）
帧率实时图表	Dashboard	各传感器实时 FPS（动态折线图，1Hz 更新）
标定文件管理	Dashboard	标定文件上传 / 下载
地图文件下载	Dashboard	重定位地图文件下载
系统日志下载	Dashboard	设备日志打包下载（.tar.gz）
遥测数据录制	Dashboard	实时监控数据录制为 CSV 并下载至本地
OTA 固件升级	Update	固件上传（支持拖放）→ 自动安装 → 重启验证
网络配置	Setting	IP / 网关 / 子网掩码 / DHCP 开关
设备重启	Setting	远程重启设备

明暗主题切换	全局	亮色 / 暗色模式，自动保存偏好
管理员登录	全局	Admin 权限解锁高级操作

11. 升级固件

- 固件获取，请联系留形技术支持
- 固件获取后解压，确保可以看到bin文件解压出来
- 升级固件步骤：
  - 电脑连接odin1-lite
  - 确保PC和odin1-lite处于同一网段，并且可以Ping通
  - 打开网页端：192.168.1.251:8080
  - 左侧点击Update
  - Browse Files点击后选择解压后的bin文件
  - 等待升级结束
  - 点击左侧Dashboard，查看固件是否升级成功

12. 重定位/重定位增量建图功能使用

目前重定位/重定位增量建图功能使用需要依赖ros_driver+算法deb包的组合方式

其中重定位增量建图功能明确在ros_driver v3.7.0 +算法deb v2.1.0提供支持

- ros_driver的获取：https://git.manifoldtech.cn/manifoldtech-sdk/odin_ros_driver2
- deb获取：
  - 运行脚本：download_lc_sdk.sh （请寻求留形技术支持获取）
  - 获取后赋予执行权限并bash执行，如下：

代码块

```
chmod +x download_lc_sdk.sh          # 赋予执行权限

hugo@hugo:$ ./download_lc_sdk.sh

========================================
  Manifold LC SDK - Platform Detection
========================================

[Detection]
  Hardware  : CV15S
  CPU       : 13th Gen Intel(R) Core(TM) i7-13620H (amd64)
  GPU       : NVIDIA GeForce RTX 4060 Laptop GPU
```

```
  CUDA      : 11.5
-
[Download links]
  • manifold-lc-sdk-amd64   RECOMMENDED
      https://version.manifoldtech.cn/download/manifold-lc-sdk-amd64
========================================

```

- 脚本会自动分析硬件平台，并提供下载链接。

- 安装算法deb文件

代码块

```
#以下以amd64平台为例
# 检查当前已安装包
dpkg -l | grep manifold-lc-sdk

# 卸载旧包
sudo dpkg -P manifold-lc-sdk

# 查看deb包信息
sudo dpkg -I manifold-lc-sdk_2.1.0_amd64.deb

# 安装新包
sudo dpkg -i manifold-lc-sdk_2.1.0_amd64.deb
#注意安装新包后需要重新启动驱动才生效
```

- Deb 版本说明：

```
-rw-r--r--  1 ztyu ztyu 36246452 9月   4 09:50 manifold-lc-sdk-amd64_0.2.0.deb
-rw-r--r--  1 ztyu ztyu 84879598 9月   4 09:50 manifold-lc-sdk-amd64-openvino20250401_0.2.0.deb
-rw-r--r--  1 ztyu ztyu 35271264 9月   4 09:50 manifold-lc-sdk-arm64_0.2.0.deb
-rw-r--r--  1 ztyu ztyu 45641412 9月   4 09:50 manifold-lc-sdk-arm64-3588npu_0.2.0.deb
-rw-r--r--  1 ztyu ztyu 56016536 9月   4 09:50 manifold-lc-sdk-arm64-jetpack515_0.2.0.deb
-rw-r--r--  1 ztyu ztyu 56090428 9月   4 09:50 manifold-lc-sdk-arm64-jetpack621_0.2.0.deb
```

1. loopclosure_sdk 分为CPU版和硬件加速版本
   a. CPU版本，仅架构后缀（manifold-lc-sdk-amd64_x.x.x 和 manifold-lc-sdk-arm64_x.x.x）
   b. 硬件加速版本，除架构外还有硬件平台说明
2. 由于硬件加速和硬件、软件强绑定，因此运行环境必须和deb搭配。为此提供我们提供脚本去检测运行平台硬件和软件相关信息，其中至少会出现CPU版本下载链接，若硬件加速支持，会同时出现硬件加速版本下载链接，客户可根据需求下载使用

注意：在docker环境下安装deb进行运行会自动回退至CPU 暂时没有docker环境下的GPU版本配置参考

注意：与Q9000地图的兼容性说明：对于Q9000地图，CPU版本SDK不支持使用（manifold-lc-sdk-amd64_x.x.x 和 manifold-lc-sdk-arm64_x.x.x ）。

- 重定位功能使用
  - 打开终端：

```
代码块
1   cd $ros_workspace/
2   source install/setup.bash
```

  - 采集地图
    - 设置O1lite为SLAM建图模式：

```
代码块
1   ros2 service call /manifold/device0/device_mode
    odin_ros_driver_rev1/srv/DeviceMode "{mode : 3}"
```

    - 开始移动O1lite，开始采集地图
    - 采集地图结束后请输入如下命令保存地图：

```
代码块
1   ros2 service call /manifold/device0/save_map
    odin_ros_driver_rev1/srv/SaveMap "{path: 'odin_map.bin'}"
```

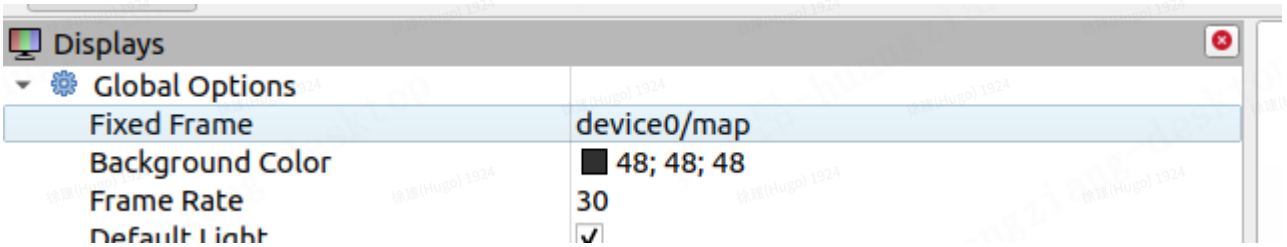
    - 注：v3.7 ros_driver驱动配合v2.1.0 算法deb包 支持在建图过程和保存过程中实时优化，优化完成后会以ROS话题形式实时发布优化结果，具体ROS话题如下

```
代码块
1   /manifold/device0/loop_closure/optimization_busy \                    优化
    状态
2   /manifold/device0/loop_closure/optimization_status_marker \           优化
    状态显示IDLE/OPTIMIZATION
3   /manifold/device0/loop_closure/optimized_path \
    local_map系下优化后累积移动路径
4   /manifold/device0/loop_closure/optimized_cloud \
    local_map系优化后全局点云
5   /manifold/device0/loop_closure/map_cloud \                            map
    系下loadmap后发布地图
```

- 运行重定位
  - 设置O1lite为relocalization模式

```
代码块
1   ros2 service call /manifold/device0/device_mode
    odin_ros_driver_rev1/srv/DeviceMode "{mode: 4}"
2   # 请确认好地图保存路径
```

  - 加载地图运行重定位
  - 先将rviz的Fixed Frame设置为device0/map

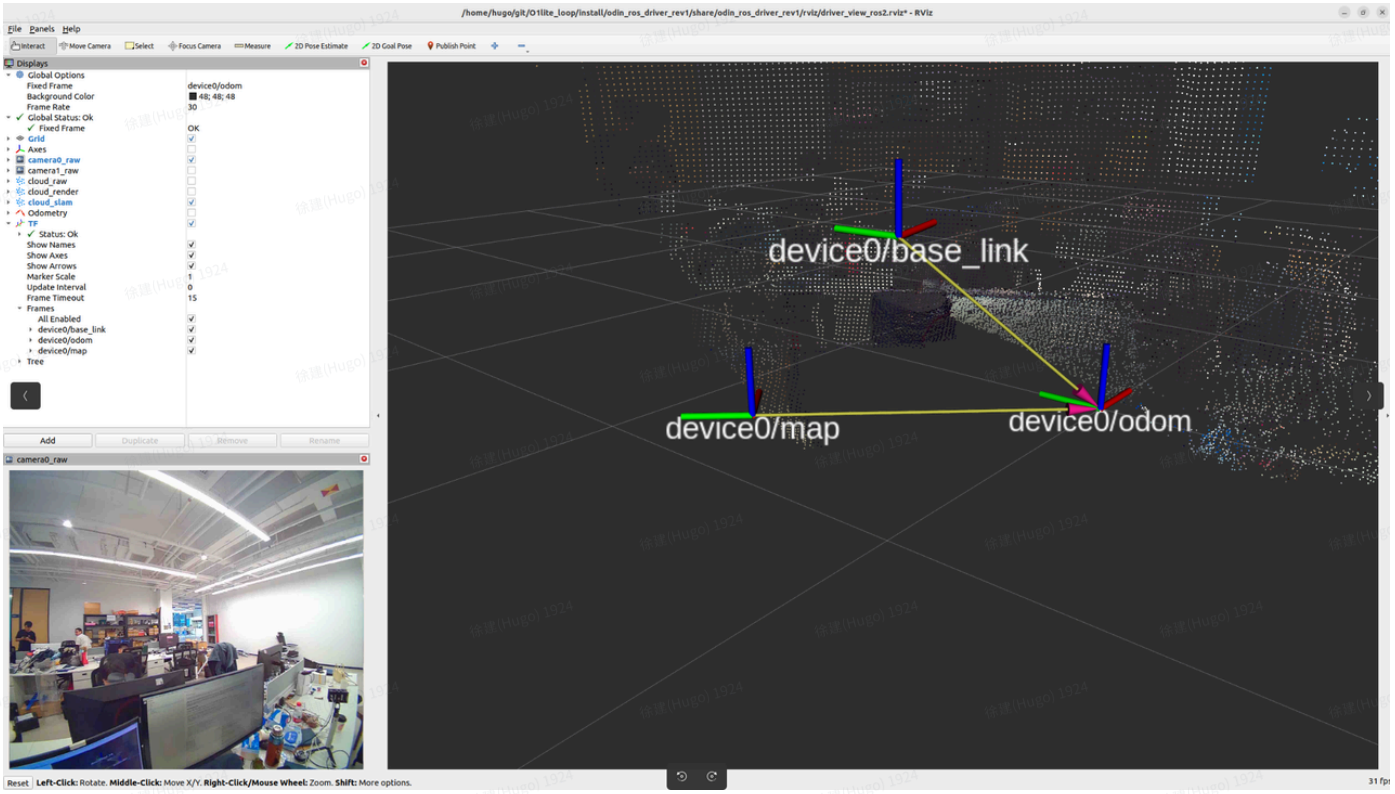


  - 然后执行以下命令可以查看载入的地图

```
代码块
1   ros2 service call /manifold/device0/load_map
    odin_ros_driver_rev1/srv/LoadMap "{path: 'odin_map.bin'}"
2   # 请确认好地图保存路径
```

- 重定位成功后：

出现device0/map坐标轴，出现导入的地图的点云



- 重定位增量建图功能使用（本功能支持在在重定位同步增量建图，支持保存增量后全局地图）
  - 打开终端：

```
代码块
1   cd $ros_workspace/
2   source install/setup.bash
```

  - 采集地图
    - 设置O1lite为SLAM建图模式：

```
代码块
1   ros2 service call /manifold/device0/device_mode
    odin_ros_driver_rev1/srv/DeviceMode "{mode : 3}"
```

    - 开始移动O1lite，开始采集地图
    - 采集地图结束后请输入如下命令保存地图：

```
代码块
1   ros2 service call /manifold/device0/save_map
    odin_ros_driver_rev1/srv/SaveMap "{path: 'odin_map.bin'}"
```

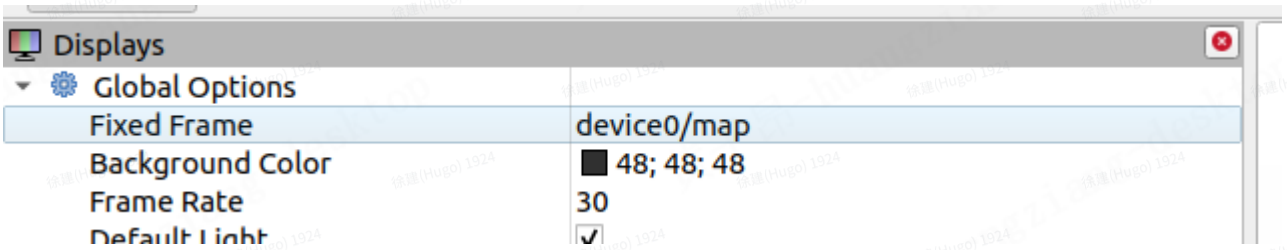
- 保存过程中会触发优化，优化完成后则会提示地图保存成功，优化结果可以在以下话题查看

```
代码块
1   /manifold/device0/loop_closure/optimization_busy \                    优化
    状态
2   /manifold/device0/loop_closure/optimization_status_marker \           优化
    状态显示IDLE/OPTIMIZATION
3   /manifold/device0/loop_closure/optimized_path \
    local_map系下优化后累积移动路径
4   /manifold/device0/loop_closure/optimized_cloud \
    local_map系优化后全局点云
5   /manifold/device0/loop_closure/map_cloud \                            map
    系下loadmap后发布地图
```

- 运行重定位增量建图
  - 设置O1lite为remap模式

```
代码块
1   ros2 service call /manifold/device0/device_mode
    odin_ros_driver_rev1/srv/DeviceMode "{mode: 5}"
2   # 请确认好地图保存路径
```

  - 加载地图运行重定位增量建图
  - 先将rviz的Fixed Frame设置为device0/map



  - 然后执行以下命令可以查看载入的地图

```
代码块
1   ros2 service call /manifold/device0/load_map
    odin_ros_driver_rev1/srv/LoadMap "{path: 'odin_map.bin'}"
2   # 请确认好地图保存路径
```

- 保存增量地图(一般情况下，增量后地图会大于原始导入地图)
  - 重定位建图结束后请输入如下命令保存地图：

```
ros2 service call /manifold/device0/save_map
odin_ros_driver_rev1/srv/SaveMap "{path: 'odin_map.bin'}"
```

- 注：v3.7 ros_driver驱动配合v2.1.0 算法deb包 支持在建图过程和保存过程中实时优化，优化完成后会以ROS话题形式实时发布优化结果，具体ROS话题如下

```
/manifold/device0/loop_closure/optimization_busy \                    优化状态
/manifold/device0/loop_closure/optimization_status_marker \           优化状态显示IDLE/OPTIMIZATION
/manifold/device0/loop_closure/optimized_path \
local_map系下优化后累积移动路径
/manifold/device0/loop_closure/optimized_cloud \
local_map系优化后全局点云
/manifold/device0/loop_closure/map_cloud \                            map系下loadmap后发布地图
```

- **重定位/增量重定位精度检测方法**
  - 操作流程(操作前保证设备正确连接，通信正常，rosdriver正常启动)
    - 运行命令

```
cd $ros_workspace/
source install/setup.bash
ros2 service call /manifold/device0/device_mode
odin_ros_driver_rev1/srv/DeviceMode "{mode: 4}"    #切换到重定位模式或切换至增量重定位模式mode:5
ros2 service call /manifold/device0/load_map
odin_ros_driver_rev1/srv/LoadMap "{path: 'odin_map.bin'}"     # 请确认好地图保存路径
```

    - 确定一个测试重定位的点位，确保设备在该点位稳定不抖动，可以运行以下命令获取位姿

```
timeout 2 ros2 run tf2_ros tf2_echo device0/map device0/imu | grep -m2 -E 'Translation|Rotation'
```

- 从该点出发，移动一段距离后回到该点，确保设备回到该点后稳定不抖动，再次运行以下命令获取位姿

```
代码块
1   timeout 2 ros2 run tf2_ros tf2_echo device0/map device0/imu | grep -m2 -
    E 'Translation|Rotation'
```

- 多次重复定位操作和记录位姿，可以在map坐标系下查看odometry_lf话题在同一点的多次重复定位位姿，比对位置的欧式距离和姿态的角度偏差可以计算重定位精度

13. 其他常见命令

13.1 先认识两类参数

驱动参数分为两类，使用时不要混淆：

参数类型	什么时候设置	怎么设置	修改后是否重启
启动前参数	启动驱动时	在 `ros2 launch` 命令末尾添加 `参数名:=值`	**必须重启驱动**
热切换参数	驱动已经运行时	使用 `ros2 param set`；批量开关可使用 ROS service	**立即生效，不需重启**

例如：`enable_pointcloud_to_depth:=true` 是启动前参数，必须在启动时写入命令；`send_undistort` 是热切换参数，可以启动时设置，也可以在运行中打开。

> “想获取什么数据”选择启动前参数；启动成功后，再按“想临时改什么”使用热切换参数。

13.2 启动前参数

目标	启动命令需要的关键参数	最终查看的话题
原始点云	无，默认就有	`{prefix}cloud/raw`
SLAM 点云	无，默认就有	`{prefix}cloud/slam`
左右相机原始图	无，默认就有	`{prefix}camera0/raw`、`{prefix}camera1/raw`
去畸变图像	`send_undistort:=true`	`{prefix}camera0/undistort`

点云重投影叠加图	`enable_reprojection:=true send_undistort:=true`	`{prefix}reprojection/overlay`
深度图	`enable_pointcloud_to_depth:=true send_undistort:=true`	`{prefix}pointcloud2depth/depth`
深度伪彩图	同上	`{prefix}pointcloud2depth/depth_color`
稠密彩色点云	深度图参数 + `enable_depth_to_pointcloud:=true`	`{prefix}depth2cloud/cloud_color`

13.3 按场景复制命令

如果不想启动时候就打开rviz，可以使用 `start_rviz:=false`；如果需要 RViz，删除该参数即可。

场景 A：只看原始点云 / 相机图

```
代码块
1  ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py start_rviz:=false
```

场景 B：看去畸变后的相机图

```
代码块
1  ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py send_undistort:=true
```

场景 C：跑原始点云重投影（标定对齐检查）

将 `cloud/raw` 投影到左相机图像上，得到叠加图。适合检查雷达点云和相机是否对齐。

```
代码块
1  ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py \
2    enable_reprojection:=true \
3    mode:=raw \
4    send_undistort:=true
```

场景 D：跑 SLAM 点云重投影

使用 `cloud/slam` 和实时里程计做重投影：

```
ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py \
  enable_reprojection:=true \
  mode:=slam \
  send_undistort:=true
```

场景 E：获取深度图

代码块

```
ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py \
  enable_pointcloud_to_depth:=true \
  send_undistort:=true
```

结果话题：

代码块

```
{prefix}pointcloud2depth/depth           # 32FC1，单位：米；0 表示无效像素
{prefix}pointcloud2depth/depth_color     # bgr8 伪彩色，便于肉眼查看
{prefix}pointcloud2depth/depth_overlay   # 深度伪彩色叠加到相机图
```

若你的有效距离只需 0.1 到 20 米，推荐同时限制距离并启用近点优先：

代码块

```
ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py \
  enable_pointcloud_to_depth:=true \
  send_undistort:=true \
  z_min:=0.1 z_max:=20.0 use_z_buffer:=true
```

场景 F：由深度图生成稠密彩色点云

此模式会同时启动深度图链路；不能只设置 `enable_depth_to_pointcloud:=true`。

代码块

```
ros2 launch odin_ros_driver_rev1 odin1Lite.launch.py \
  enable_pointcloud_to_depth:=true \
  enable_depth_to_pointcloud:=true \
  send_undistort:=true
```

13.4 热切换参数

1. 查询 `/odin_ros_driver_node` 名字

主驱动节点名也会随设备序列号变化。ROS2 运行时通常是 `/odin_driver_<SN>`，例如 `/odin_driver_P000000001`；不要把文中的旧泛称 `/odin_ros_driver_node` 当作固定节点名。用下面命令确定实际名称：

```
代码块
1  ros2 node list | grep '^/odin_driver_'
```

> 下表是热切换速查。驱动已经启动后，复制“怎么热切换”列的命令即可立即生效；`{prefix}` 请替换为你的实际设备前缀。

目标	怎么热切换	立即变化
关闭/打开原始点云	`ros2 param set /odin_ros_driver_node enable_raw_point false/true`	`{prefix}cloud/raw` 停止/恢复发布
关闭/打开 SLAM 点云	`ros2 param set /odin_ros_driver_node enable_slam_point false/true`	`{prefix}cloud/slam` 停止/恢复发布
关闭/打开左、右相机输入	`ros2 param set /odin_ros_driver_node enable_image0 false/true`；`enable_image1` 同理	相应相机数据停止/恢复
关闭/打开 IMU、里程计、TF	对应设置 `enable_imu`、`enable_odom_lf`、`enable_odom_hf`、`enable_tf` 为 `false/true`	对应数据流停止/恢复
关闭/打开左相机原始图	`ros2 param set /postprocess send_image_raw false/true`	`{prefix}camera0/raw` 停止/恢复发布
关闭/打开右相机原始图	`ros2 param set /postprocess send_image2_raw false/true`	`{prefix}camera1/raw` 停止/恢复发布
运行中打开去畸变图	`ros2 param set /postprocess send_undistort true`	发布 `{prefix}camera0/undistort`、`camera1/undistort`
运行中打开立体校正图	`ros2 param set /postprocess send_rectify true`	发布 `{prefix}camera{0,1}/rect` 与 `camera_info`

运行中打开点云渲染图	`ros2 param set /postprocess send_cloud_render true`	发布 `{prefix}cloud/render`
调整点云置信度	`ros2 param set /odin_ros_driver_node confidence_threshold 15`	低于阈值的点被过滤
切换 dToF 模式	`ros2 param set /odin_ros_driver_node sensor_mode 3`	传感器切换为 Adaptive 模式
批量开关多个数据流	调用 `set_channel_config` 或 `set_post_process_config` service	一次更新多个开关，见第 5.2 节

> 深度图、重投影的节点开关和算法参数不是热切换参数。运行中打开 `send_undistort` 只能补齐图像输入；若未在启动时设置 `enable_pointcloud_to_depth:=true` 或 `enable_reprojection:=true`，仍需重启驱动。

2. 热切换参数分为两类：

- **主驱动数据通道**在 `{driver_node}`（ROS2 中由设备序列号动态决定）
- **后处理输出**在固定节点 `/postprocess` 。

节点	可热切换参数	作用
`{driver_node}`	`confidence_threshold`、`sensor_mode`、`log_level`	点云置信度阈值、dToF 传感器模式、日志级别
`{driver_node}`	`enable_raw_point`、`enable_slam_point`、`enable_image0`、`enable_image1`、`enable_imu`、`enable_odom_lf`、`enable_odom_hf`、`enable_tf`	打开或关闭对应的设备数据通道
`/postprocess`	`send_image_raw`、`send_image2_raw`、`send_cloud_render`、`send_undistort`、`send_rectify`、`send_opencv_undistort`	打开或关闭后处理图像、着色点云、去畸变和校正输出

3. 使用方法：

代码块

```
# 查看某个节点当前参数
ros2 param list {driver_node}   # 如ros2 param list /odin_driver_P000000001
ros2 param list /postprocess

# 关闭 SLAM 点云通道；立即生效
ros2 param set {driver_node} enable_slam_point false

# 运行中打开去畸变图像发布；立即生效
ros2 param set /postprocess send_undistort true

# 将点云置信度阈值改为 15；立即生效
ros2 param set {driver_node} confidence_threshold 15

# 切换 dToF 传感器模式：0=HDR，1=High Peak，2=Low Peak，3=Adaptive
ros2 param set {driver_node} sensor_mode 3
```

4. 用 service 批量热切换（推荐一次改多个开关时使用）

除了 `ros2 param set`，也可通过 service 批量修改并直接得到全部开关状态。服务路径分两种：设备数据通道服务有设备前缀，后处理服务使用 `/{topic_prefix}/postprocess/`。

代码块

```
# 查询主驱动所有数据通道状态
ros2 service call /manifold/device0/get_channel_config \
  odin_ros_driver_rev1/srv/GetChannelConfig

# 关闭 SLAM 点云，其他数据通道保持当前状态
# 0=关闭，1=开启，-1=保持不变；未写出的字段默认是 0，必须显式填写 -1
ros2 service call /manifol/device0/set_channel_config \
  odin_ros_driver_rev1/srv/SetChannelConfig \
  "{enable_raw_point: -1, enable_slam_point: 0, enable_image0: -1,
enable_image1: -1, enable_imu: -1, enable_odom_lf: -1, enable_odom_hf: -1,
enable_tf: -1}"

# 查询所有后处理输出状态
ros2 service call /manifold/postprocess/get_post_process_config \
  odin_ros_driver_rev1/srv/GetPostProcessConfig

# 运行中打开去畸变图像，其他后处理输出保持当前状态
ros2 service call /manifold/postprocess/set_post_process_config \
  odin_ros_driver_rev1/srv/SetPostProcessConfig \
  "{send_image_raw: -1, send_image2_raw: -1, send_cloud_render: -1,
send_undistort: 1, send_rectify: -1, send_opencv_undistort: -1}"
```

> service 的 `SetChannelConfig` 和 `SetPostProcessConfig` 每个字段均为 `0=关闭`、`1=开启`、`-1=保持不变`。调用“设置”服务时请把所有不想改的字段明确写成 `-1`；否则请求的默认值为 `0`，可能意外关闭其他数据流。

> 热切换只会开关已有的输入/输出数据流。`enable_pointcloud_to_depth`、`enable_reprojection`、`z_max`、`scale`、`mode` 等处理节点和算法参数**不是**热切换参数；即使 `ros2 param set` 命令显示成功，也必须重启驱动才会进入新配置。

13.5 录制bag包命令

录包之前请注意以下几点：

- 已成功配置domain_id
- 已成功配置网卡缓冲区大小
- 使用rosbag2_qos.yaml（针对ros2）,yaml文件路径存在o1lite驱动src/config/rosbag2_qos.yaml文件

录包命令：

代码块

```
ros2 bag record -a --qos-profile-overrides-path rosbag2_qos.yaml
```