

# Odin1快速启动

## 1. 简介

本文档意在指导用户在拿到设备进行安装并在linux电脑上成功跑通数据。用户手册wiki链接如下：  
[https://manifoldtechltd.github.io/wiki/odin\\_series/odin1/](https://manifoldtechltd.github.io/wiki/odin_series/odin1/)

## 2. 版本

序号	版本	修改人（日期）	备注
1	1.0.1	2025.9.4	增加了22.04/ROS2的使用
2	1.0.2	2025.10.24	基于0.5.0固件和0.5.2驱动
3	1.1.0	2025.10.30	基于0.7.0固件和0.6.0驱动
4	1.2.0	2026.1.6	基于0.8.0固件和0.7.1驱动
5	1.3.0	2026.2.25	基于0.10.0固件和0.9.0驱动
6	1.4.0	2026.4.15	基于0.11.3固件和0.10.0驱动

## 3. 硬件设备安装

硬件组成（我们默认不提供适配器，请客户自行准备，接口为DC5521/5524，建议电源输入12~24V，典型值：12V2A）：



Odin1主机



数据线



电源线

#### 4. PC环境依赖

- PC性能需求: 无
- 建议配置: 建议配备独立显卡 (rviz显示), 内存 $\geq$ 8g

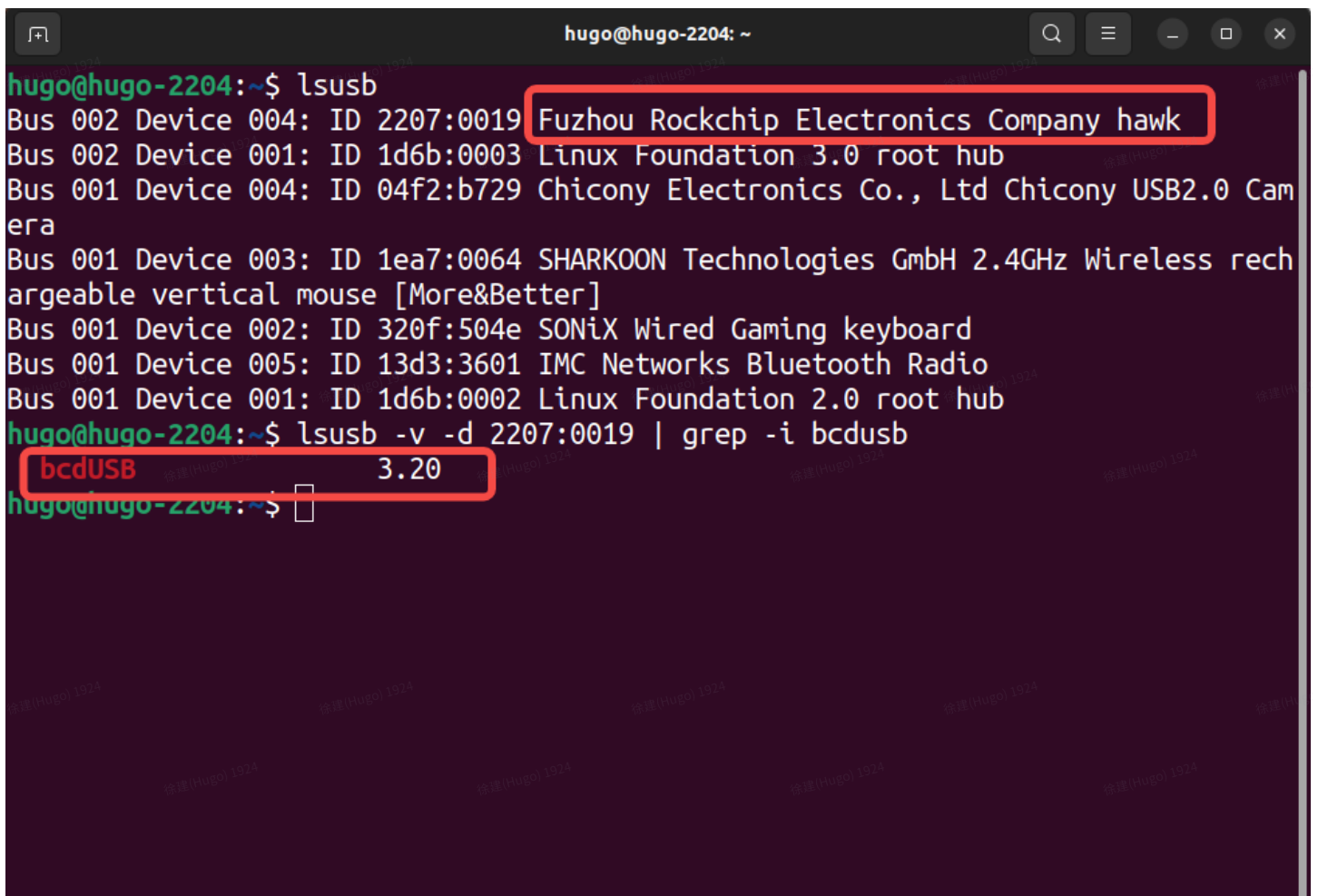
## 5. 设备连接PC

注：PC可以使用USB2.0，但更推荐插入3.0接口内，如果机器配合RTK使用，建议使用2.0的线，以防干扰信号

- 系统要求：ubuntu20.04/ubuntu22.04
- 环境要求：ubuntu 20.04 for ROS Noetic or ROS2 Foxy / ubuntu 22.04 for ROS2 Humble.
- 如果不确定是否识别成3.0接口，建议尝试如下命令查看：

代码块

```
1 lsusb -v -d 2207:0019 | grep -i bcdusb
```



```
hugo@hugo-2204: ~  
hugo@hugo-2204:~$ lsusb  
Bus 002 Device 004: ID 2207:0019 Fuzhou Rockchip Electronics Company hawk  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 004: ID 04f2:b729 Chicony Electronics Co., Ltd Chicony USB2.0 Camera  
Bus 001 Device 003: ID 1ea7:0064 SHARKOON Technologies GmbH 2.4GHz Wireless rechargeable vertical mouse [More&Better]  
Bus 001 Device 002: ID 320f:504e SONiX Wired Gaming keyboard  
Bus 001 Device 005: ID 13d3:3601 IMC Networks Bluetooth Radio  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
hugo@hugo-2204:~$ lsusb -v -d 2207:0019 | grep -i bcdusb  
bcdUSB 3.20  
hugo@hugo-2204:~$
```

## 6. 驱动使用 (ubuntu20.04/ROS1)

### 6.1 获取驱动

代码块

- 1 `git clone https://github.com/manifoldsdk/odin_ros_driver.git`  
`catkin_ws/src/odin_ros_driver`
- 2 # 使用git clone命令将github上的驱动下载到本地，并默认保存在/home下的/catkin\_ws文件夹中

## 6.2 USB配置

首次使用设备和电脑，需要对USB规则进行配置，命令如下：

代码块

```
1 sudo gedit /etc/udev/rules.d/99-odin-usb.rules # 新建.rules文件, 并使用gedit打  
   开该文件  
2 SUBSYSTEM=="usb", ATTR{idVendor}=="2207", ATTR{idProduct}=="0019",  
   MODE="0666", GROUP="plugdev"  
3 #将上述SUBSYSTEM填写至99-odin-usb.rules文件中, 建议复制粘贴, 手动输入可能导致格式错误进  
   而不能识别  
4 sudo udevadm control --reload  
5 sudo udevadm trigger # 重新加载规则文件, 并使其生效
```

## 6.3 以ubuntu20.04/ROS1为例，构建ros工程（将驱动sdk放置在/src文件夹中）

代码块

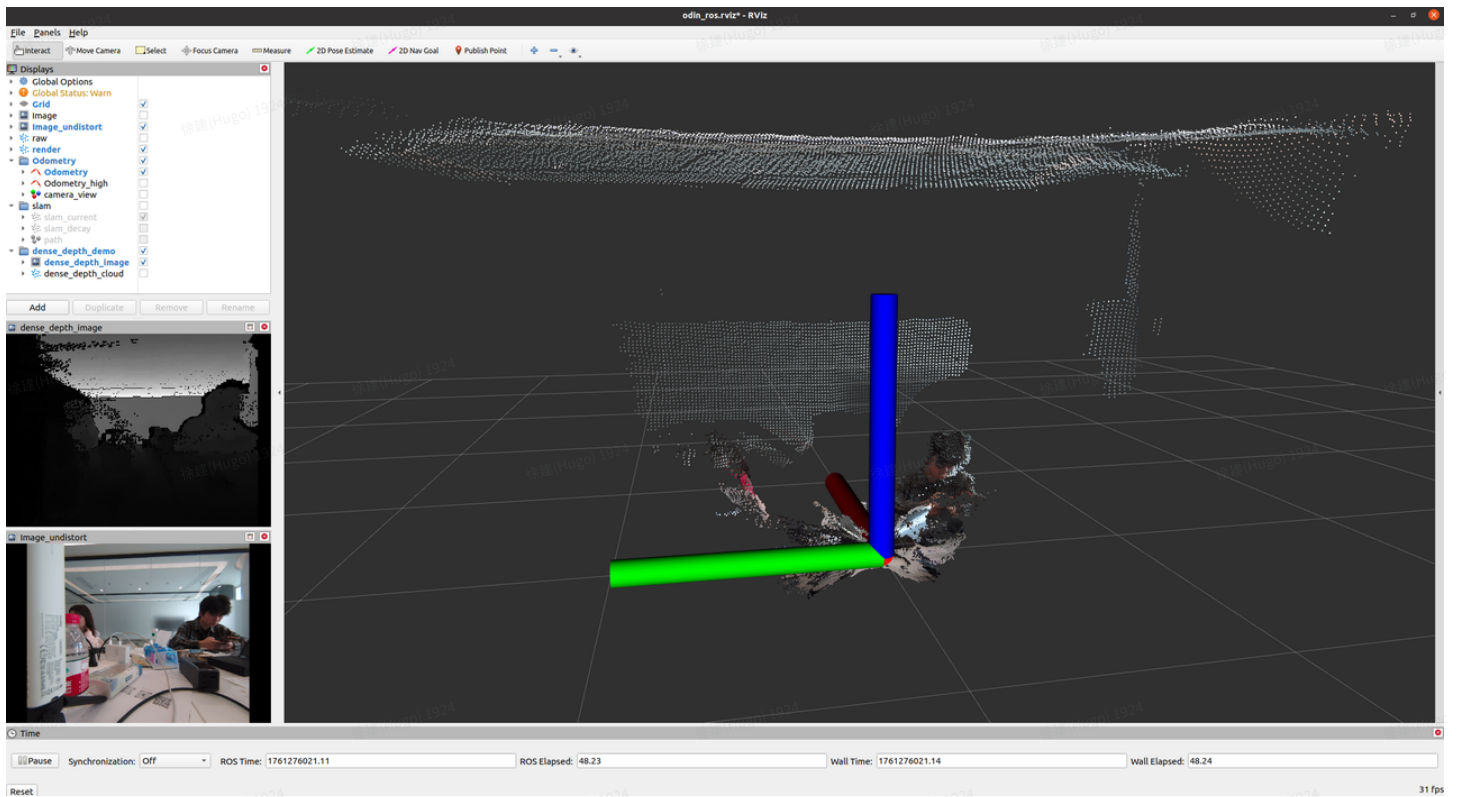
```
1 mkdir -p catkin_ws/src # 如果git clone下载驱动的时候已经新建了catkin_ws, 请省略  
   该步骤  
2 cp -r /$PATH/odin_ros_driver/ ~/catkin_ws/src/ # 将git下载的驱动放到ros工作空  
   间中。$PATH为下载的驱动所在位置  
3 cd ~/catkin_ws/src/odin_ros_driver/script/  
4 source ~/.bashrc  
5 ./build_ros.sh #如果sh脚本没有权限可输入sudo chmod +x build_ros*.sh  
   赋予权限
```

## 6.4 运行驱动（保证Odin1连接到电脑USB3.0接口上）

代码块

```
1 cd ~/catkin_ws # 进入ros工作空间, 具体路径根据存放位置确定  
2 source devel/setup.bash # 声明变量  
3 roslaunch odin_ros_driver odin1_ros1.launch # 执行launch文件
```

## 6.5 运行成功可看到rviz界面：



## 6.6 驱动输出topic包含如下内容:

```

root@hugo:/# rostopic list
/clicked_point
/initialpose
/move_base_simple/goal
/odin1/camera_pose_visual
/odin1/cloud_raw
/odin1/cloud_render
/odin1/cloud_slam
/odin1/depth_img_competition/mouse_click
/odin1/image
/odin1/image/compressed
/odin1/image/intensity_gray
/odin1/image/mouse_click
/odin1/image/undistorted
/odin1/image/undistorted/mouse_click
/odin1/imu
/odin1/odometry
/odin1/odometry_highfreq
/odin1/overlay_image
/odin1/path
/odin1/reprojected_image
/odin1/reprojected_image/mouse_click
/odin1/wiwc
/rosout
/rosout_agg
/tf
/tf_static
root@hugo:/#

```

## 7. 驱动使用 (ubuntu22.04/ROS2)

## 7.1 获取驱动（亦可联系留形科技技术支持获取最新版本）

代码块

```
1 git clone https://github.com/manifoldsdk/odin_ros_driver.git
   catkin_ws/src/odin_ros_driver # 使用git clone命令将github上的驱动下载到本地，并
   默认保存在/home下的/catkin_ws文件夹中
```

## 7.2 以ubuntu22.04/ROS2为例，构建ros2工程（将驱动sdk放置在/src文件夹中）

代码块

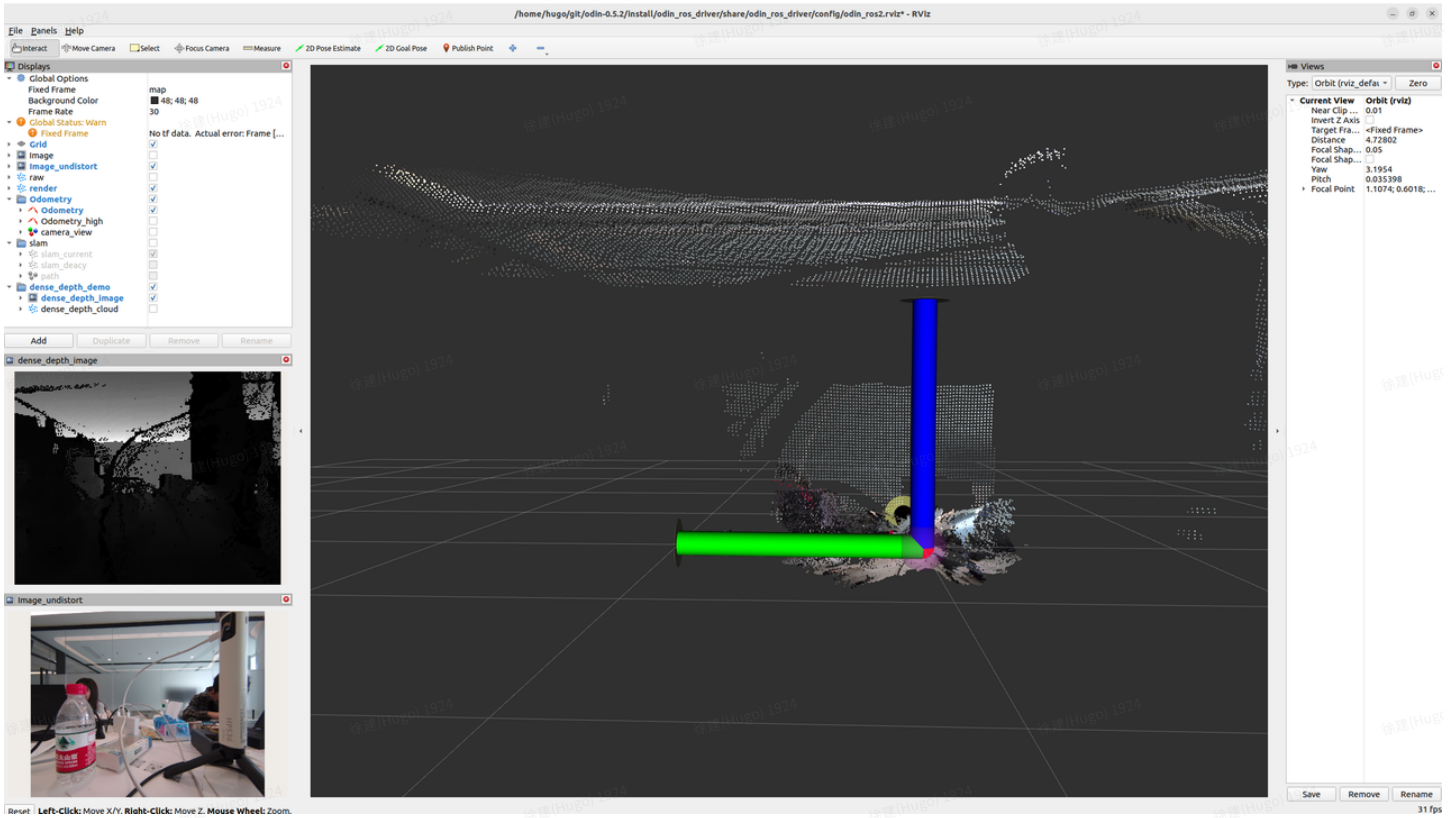
```
1 mkdir -p catkin_ws/src # 如果git clone下载驱动的时候已经新建了catkin_ws，请省略
   该步骤
2 cp -r /$PATH/odin_ros_driver/ ~/catkin_ws/src/ # 将git下载的驱动放到ros工作空
   间中。$PATH为下载的驱动所在位置
3 cd ~/catkin_ws/src/odin_ros_driver/script/
4 source ~/.bashrc
5 ./build_ros2.sh #如果sh脚本没有权限可输入sudo chmod +x
   build_ros*.sh赋予权限
```

## 7.3 运行驱动（保证Odin1连接到电脑USB3.0接口上）

代码块

```
1 cd ~/catkin_ws # 进入ros工作空间，具体路径根据存放位置确定
2 source install/setup.bash # 声明变量
3 ros2 launch odin_ros_driver odin1_ros2.launch.py # 执行launch文件
```

## 7.4 运行成功可看到rviz界面：



## 7.5 键入如下命令查看topic

代码块

```
1 ros2 topic list
```

```
hugo@hugo:~/git/odin_0.10.0$ ros2 topic list
/clicked_point
/goal_pose
/initialpose
/odin1/cloud_raw
/odin1/depth_img_competetion
/odin1/depth_img_competetion_cloud
/odin1/image
/odin1/image/compressed
/odin1/overlay_image
/parameter_events
/rosout
hugo@hugo:~/git/odin_0.10.0$
```

## 8. 重定位模式使用说明

目前Odin使用分为三种模式：里程计模式、SLAM模式、重定位模式。以下会就三种模式操作进行简单介绍。模式切换均通过修改/odin\_ros\_driver/config/control\_command.yaml中的‘custom\_map\_mode’参数进行配置

- 里程计模式

设置“custom\_map\_mode = 0”（默认）启动里程计模式，该模式下，地图坐标系和里程计坐标系享用相同坐标系，与之前版本效果一致。

- SLAM模式

设置“custom\_map\_mode = 1”启动SLAM模式，此模式提供一个完整的SLAM系统，在里程计模式的基础上增加了SLAM系统，即增加“回环检测”和“地图保存”功能。map保存路径：

代码块

```
1 /odin_ros_driver/map/
```

设置好参数启动驱动程序，Odin1将自动执行建图并缓存地图数据。结束录制操作：

代码块

```
1 cd $PATH/src/odin_ros_driver # $PATH为用户保存驱动的源目录
2 ./set_param.sh save_map 1 # 执行保存脚本
```

```
hugo@hugo-2204: ~/git/odin_0.7.0_test
[host_sdk_sample-1] [INFO] [1761790172.223655003] [device_cb]: Command interface
ready. Use: echo 'set save_map 1' > /tmp/odin_command.txt
[host_sdk_sample-1] <INFO><api.cpp:lidar_set_imu_parameter:1100>: set imu parame
ter success....
[host_sdk_sample-1] <DEBUG><api.cpp:lidar_start_stream:694>: dtof_subframe_odr:
328500
[host_sdk_sample-1] <DEBUG><api.cpp:lidar_start_stream:697>: start stream mode s
uccess....
[host_sdk_sample-1] Tc1: -0.00455 -0.99988 -0.01472 0.0385
[host_sdk_sample-1] 0.00925 0.01468 -0.99985 -0.00914
[host_sdk_sample-1] 0.99995 -0.00468 0.00918 -0.00911
[host_sdk_sample-1] 0 0 0 1
[host_sdk_sample-1] [INFO] [1761790172.223847386] [device_cb]: Software connecti
on successful in 10 seconds
[host_sdk_sample-1] [INFO] [1761790172.223850425] [device_cb]: Device ready and
streams activated
[host_sdk_sample-1] [INFO] [1761790213.200139960] [command_processor]: Successfu
lly set save_map = 1
[host_sdk_sample-1] [INFO] [1761790214.430939092] [param_monitor]: Map is saved
on device, now transferring to [/home/hugo/git/odin_0.7.0_test/src/odin_ros_drive
r/map/20251030_100921/map_20251030_100921.bin]
[host_sdk_sample-1] [WARN] [1761790214.506093255] [param_monitor]: map get succe
ss
]

hugo@hugo-2204: ~/git/odin_0.6.0/src/odin_ros_driver
hugo@hugo-2204:~/git/odin_0.6.0/src/odin_ros_driver$ ./set_param.sh save_map 1
Command sent: set save_map 1
Command file: /tmp/odin_command.txt
hugo@hugo-2204:~/git/odin_0.6.0/src/odin_ros_driver$
```

- 重定位模式

设置“custom\_map\_mode = 2”启动重定位模式，设置/config/control\_command.yaml中的“relocalization\_map\_abs\_path”为指定的绝对路径，如：

```

showpath: 1          # 0: off; 1: on
showcamerapose: 1   # 0: off; 1: on
custom_map_mode: 2   # 0: Odometry mode 1: SLAM mode 2: Relocalization mode
custom_map_mode_pos: 0.0 0.0 0.0 0.0 0.0 1.0
relocalization_map_abs_path: "/home/hugo/map/test.bin" # must be set for Relocalization mode or will fail

# To get the mapping result file, please use the set_param.sh script provided: "./set_param.sh save_map 1"
mapping_result_dest_dir: "" # "": if not specified, save to default location of {ws}/src/odin_ros_driver/map/{driver_start_time}/
mapping_result_file_name: "" # "": if not specified, save to location above with default file name of map_{map_save_time}.bin

# Image mask transfer settings
sendimagemask: 0     # 0: off; 1: on - transfer image mask to device on startup
image_mask_abs_path: "" # absolute path to the image mask file (e.g., /path/to/mask.png(1600x1296))

# Algorithm reset settings
resetalgo: 0         # 0: off; 1: on - send algo_reset command to device on startup

```

启动驱动，重定位成功后rviz将显示当前帧在map中的位置关系。如果重定位失败，将会退回Odom模式，后台将持续进行尝试重定位，一旦成功，地图帧和里程计帧之间的TF将会发布。

**\*注：**以下主题在里程计坐标系中发布：`/odin1/cloud\_slam`、`/odin1/odom`、`/odin1/highodom`和`/odin1/path`。若要在地图坐标系中获取这些主题，请应用从里程计坐标系到地图坐标系的TF。

## 9. Odin录制数据导入MindCloud使用

### 9.1 设置/src/odin\_ros\_driver/config/control\_command.yaml文件中recorddata参数：

代码块

```

1 # /src/odin_ros_driver/config/control_command.yaml
2 recorddata: 1          # 0: off; 1: on

```

### 9.2 设置好参数后运行驱动进行地图采集

代码块

```

1 cd $PATH/catkin_ws
2 source install/setup.bash
3 ros2 launch odin_ros_driver odin1_ros2.launch.py
4 # 地图采集结束后Ctrl + c 结束驱动

```

### 9.3 查

看/catkin\_ws/src/odin\_ros\_driver/recorddata/\*\*\*/image/cam\_in\_ex.txt是否完整，完整信息示例如下：

代码块

```

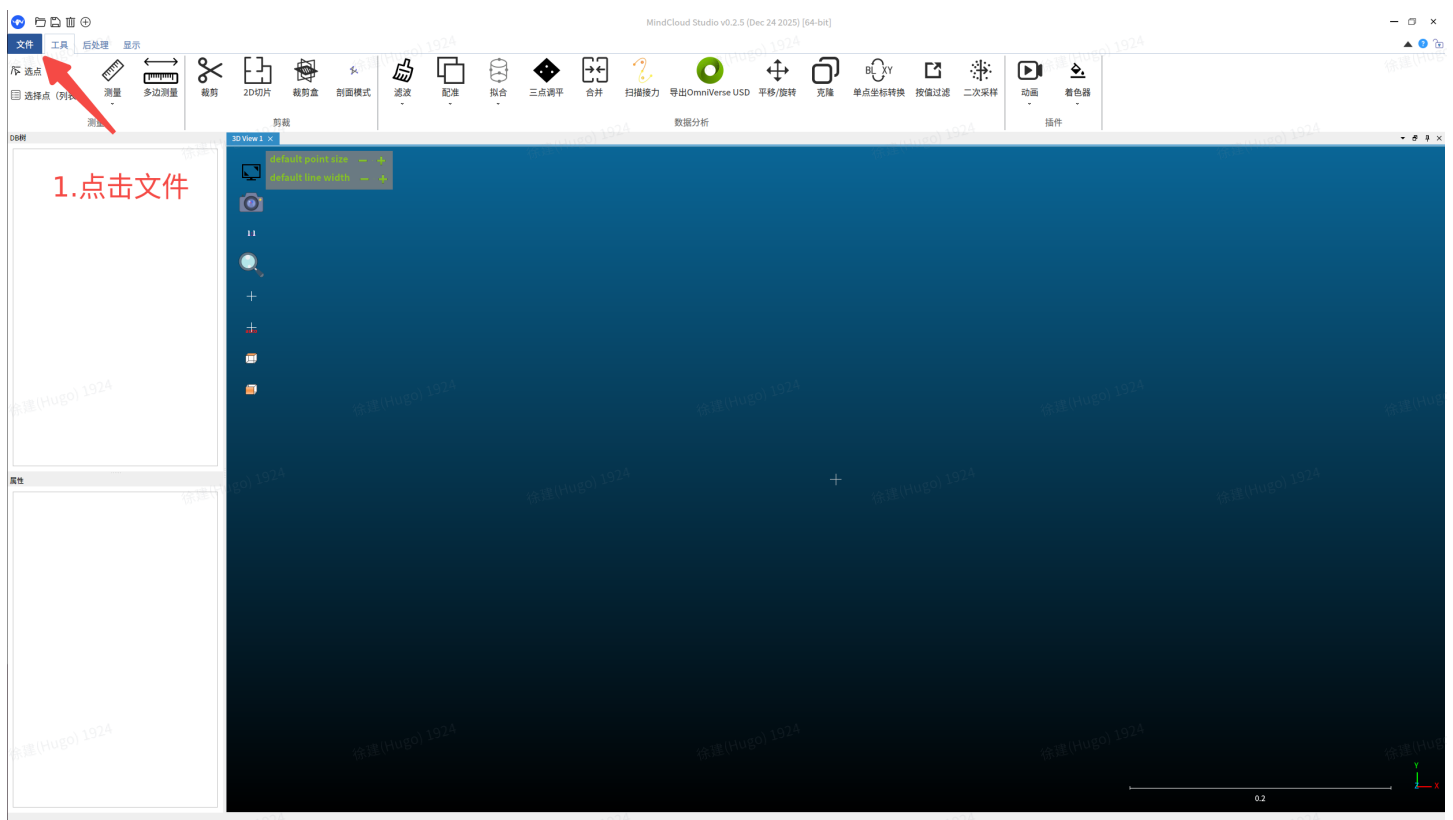
1 Tc1_0: [-0.004550, -0.999880, -0.014720, 0.038500,
2         0.009250, 0.014680, -0.999850, -0.009140,
3         0.999950, -0.004680, 0.009180, -0.009110,

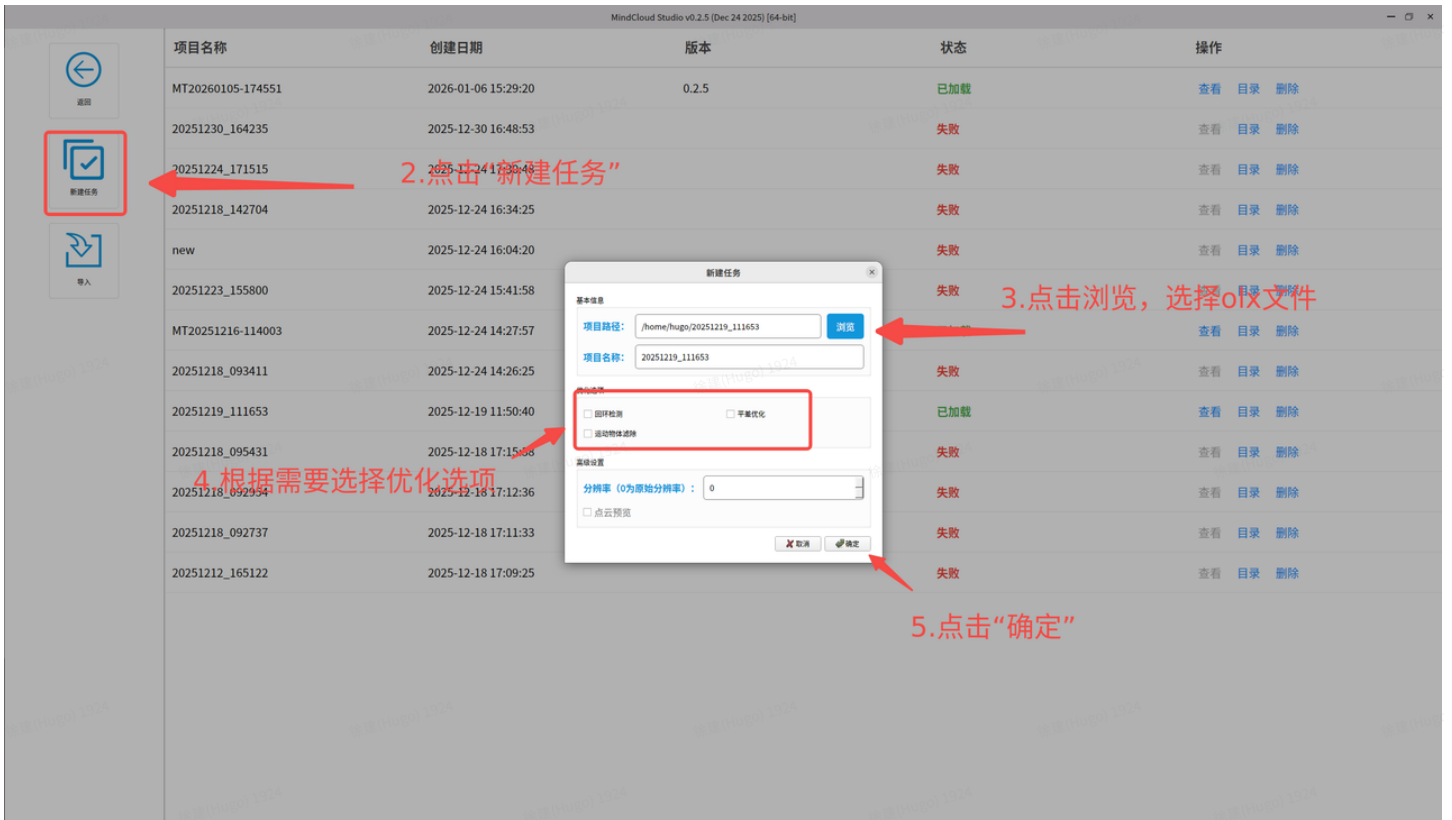
```

```
4          0.000000, 0.000000, 0.000000, 1.000000]
5  cam_0:
6    image_width: 1600
7    image_height: 1296
8    k2: -0.006990
9    k3: 0.003913
10   k4: -0.019202
11   k5: -0.001396
12   k6: 0.010615
13   k7: -0.005595
14   p1: 0.000000
15   p2: 0.000000
16   A11: 731.137828
17   A12: -0.682592
18   A22: 730.873102
19   u0: 803.257996
20   v0: 640.726283
```

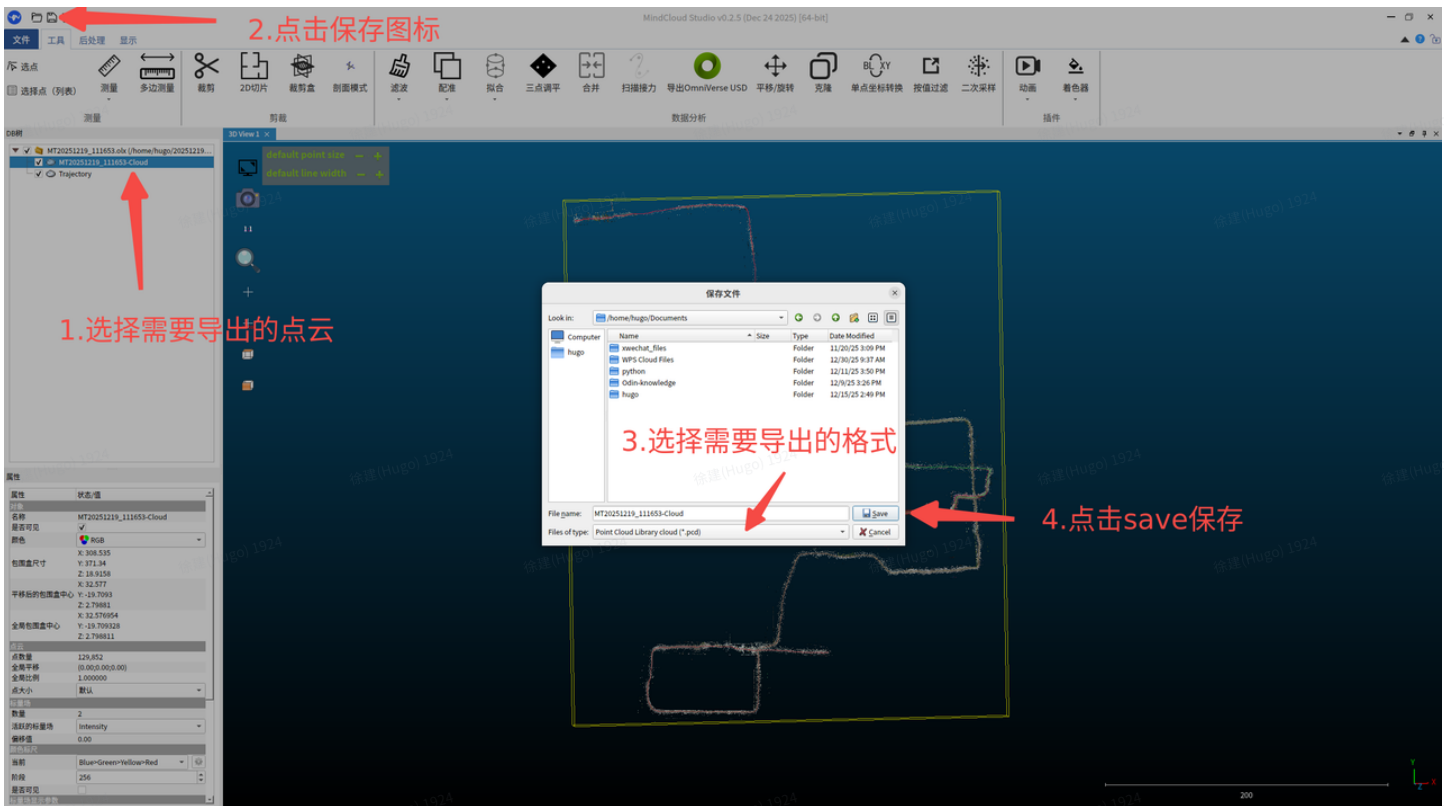
如果显示完整代表该数据可用，如果显示不完整表示驱动异常结束，此次数据扫描无效。

## 9.4 数据导入Mindcloud





## 9.5 导出pcd文件



## 10. 消息说明

Topic	control_command.yaml	Detailed Description
odin1/imu	sendimu	Imu Topic--原始imu数据400hz。坐标系原点: imu系

odin1/image	sendrgb	RGB Camera Topic, decoded from original jpeg data from device, bgr8 format--原始图像数据。坐标系原点：camera系
odin1/image_undistort	sendrgbundistort	undistorted RGB Camera Topic, processed with calib.yaml from device--去畸变后的图像数据。坐标系原点：camera系
odin1/image/compressed	sendrgbcompressed	RGB Camera compressed Topic, original jpeg data from device--压缩后的图片数据。坐标系原点：camera系
odin1/cloud_raw	senddtof	Raw_Cloud Topic--原始点云数据。坐标系原点：lidar系
odin1/cloud_render	sendcloudrender	Render_Cloud Topic, processed with raw point cloud, rgb image, and calib.yaml from device--着色后的点云数据。坐标系原点：lidar系
odin1/cloud_slam	sendcloudslam	Slam_PointCloud Topic--经过slam算法后的彩色点云数据，融合了LiDAR/camera/imu。坐标系原点：odom系
odin1/odometry	sendodom	Odom Topic--里程计数据。坐标系原点：odom系
odin1/odometry_high	sendodom	high frequency Odom Topic--高频里程计数据。坐标系原点：odom系
odin1/depth_img_competition	senddepth	Dense depth image Topic. Demo, high computing power required. One-to-one with odin1/image_undistort. To utilize the data please directly subscribe to this topic instead of echoing it. Original value is already depth data, no need for further convert.--与去畸变后分辨率一致的深度图数据。坐标系原点：camera系
odin1/depth_img_competition_cloud	senddepth	Dense Depth_Cloud Topic. Demo, high computing power required--深度点云。坐标系原点：camera系
odin1/path	showpath	Odom Path Topic--显示轨迹点。
tf	sendodom	Tf tree topic--用于里程计坐标系与map坐标系的转换
odin1/image/intensity_gray	pubintensitygray	dtof反射强度灰度图像，用于评判dtof模组是否异常

/odin1/reprojected_image	sendreprojection	将3D点云投影到2D图片上
/odin1/overlay_image	sendoverlay	将重投影图片投影到去畸变图片上，用于查看点云投影是否准确
/odin1/wiwc	默认发出	算法使用的实时估算的每一帧的设备外参信息 t_cl/t_il

## 11. FAQ

- 设备线束如何连接？
  - 出厂提供两根线，航插头的供电线和usb3.0的数据线，客户需要自己准备适配器，输出12V2A，DC接头5521/5524。设备支持宽压12~24V，超过量程会烧毁Odin内部芯片。
- 线束连接后rviz无显示
  - 建议更换稳定电源，或者适当增加电源电压
  - 确保已经将 6.2 rules文件准确写入系统文件
  - 查看终端打印版本信息，确保是最新版的固件和最新版的驱动
- 重定位可以导入先验地图么？
  - 支持，但是地图来源只能是如下来源：
    - 使用Odin提供的建图功能（map mode : 1）保存bin文件
    - 使用留形手持扫描仪Q9000采集的地图，通过mindcloud处理，在后处理功能中导出地图功能获取的bin文件
- 如何判断重定位成功？
  - 重定位成功后，/tf topic会输出map系关系。rviz中也可以看到map系的显示。
  - 终端会打印提示relocalizaion sucesseed
- 其他相关问题亦可参阅README.md文件。