Odin1 Quick Start Guide

1. Introduction

This document guides users to set up the device and run data successfully on a Linux PC.
User manual wiki: https://manifoldtechltd.github.io/wiki/odin_series/odin1/

2. Version History

No.	Version	Author (Date)	Notes
1	1.0.1	2025.9.4	Added Ubuntu 22.04 / ROS2 support
2	1.0.2	2025.10.24	Based on firmware 0.5.0 and driver 0.5.2
3	1.1.0	2025.10.30	Based on firmware 0.7.0 and driver 0.6.0
4	1.2.0	2026.1.6	Based on firmware 0.8.0 and driver 0.7.1
5	1.3.0	2026.2.25	Based on firmware 0.10.0 and driver 0.9.0
6	1.4.0	2026.4.15	Based on firmware 0.11.3 and driver 0.10.0

3. Hardware Installation

Hardware components (power adapter not included; please prepare separately. Connector: DC5521/5524, recommended input 12~24V, typical: 12V 2A):

Odin1 Main Unit

Data Cable

Power Cable

4. PC Environment Requirements

1. PC performance requirement: None
1. Recommended: discrete GPU (for rviz display), RAM >= 8 GB

5. Connecting Device to PC

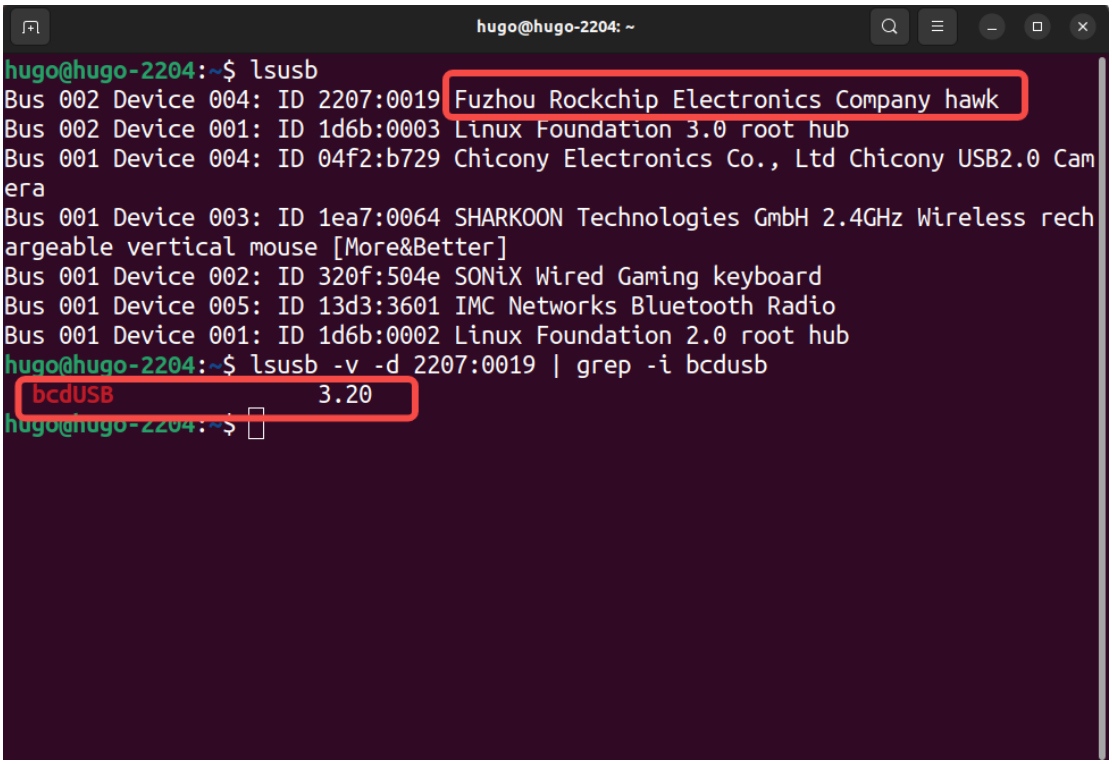
Note: USB 2.0 is supported but USB 3.0 is recommended. If using with RTK, use a USB 2.0 cable to avoid signal interference.

1. System requirement: Ubuntu 20.04 / Ubuntu 22.04

1. Environment: Ubuntu 20.04 for ROS Noetic or ROS2 Foxy / Ubuntu 22.04 for ROS2 Humble.

1. To verify if the device is recognized as USB 3.0, run the following command:

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



6. Driver Usage (Ubuntu 20.04 / ROS1)

6.1 Get the Driver

```
Shell
git clone https://github.com/manifoldsdk/odin_ros_driver.git
catkin_ws/src/odin_ros_driver
# Clone the driver from GitHub, saved to ~/catkin_ws/ by
default
```

6.2 USB Configuration

When using the device with a PC for the first time, configure the USB rules as follows:

```
Shell
sudo gedit /etc/udev/rules.d/99-odin-usb.rules    # Create
and open the rules file
SUBSYSTEM=="usb", ATTR{idVendor}=="2207",
ATTR{idProduct}=="0019", MODE="0666", GROUP="plugdev"
#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
```

```
sudo udevadm control --reload
sudo udevadm trigger      # 重新加载规则文件，并使其生效
```

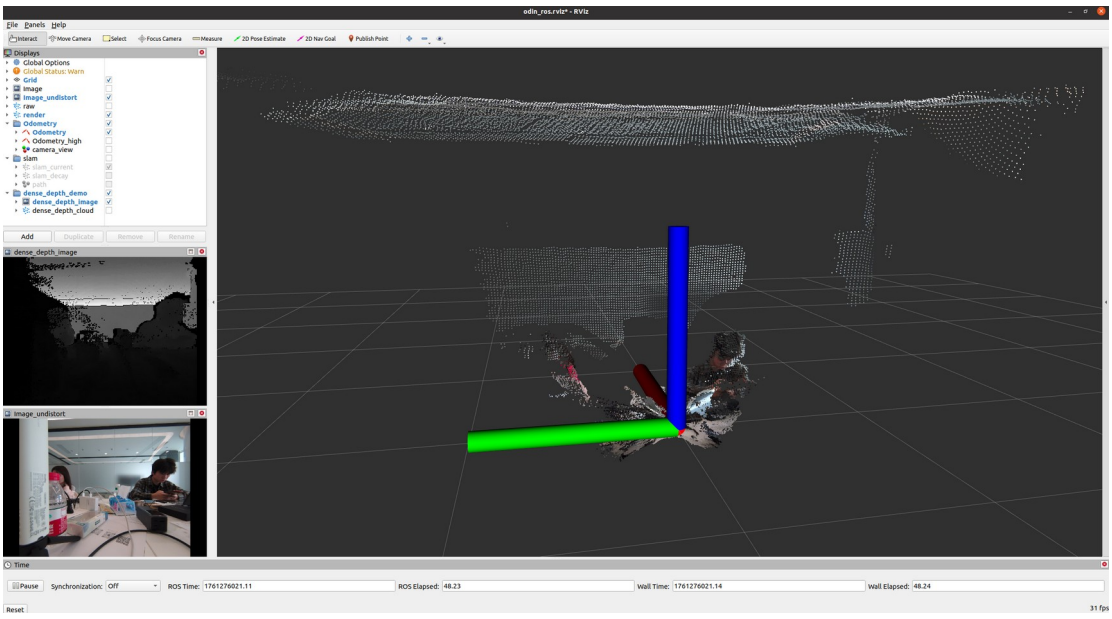
6.3 Build ROS workspace (Ubuntu 20.04 / ROS1; place driver SDK in the /src folder)

```
Shell
mkdir -p catkin_ws/src      # Skip if catkin_ws was already
created during git clone
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
cd ~/catkin_ws/src/odin_ros_driver/script/
source ~/.bashrc
./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 Run the driver (ensure Odin1 is connected to a USB 3.0 port)

```
Shell
cd ~/catkin_ws     # Navigate to your ROS workspace
source devel/setup.bash    # 声明变量
roslaunch odin_ros_driver odin1_ros1.launch       # 执行
launch 文件
```

6.5 On success, the rviz window will appear:



6.6 Driver output topics:

```
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_competetion/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. Driver Usage (Ubuntu 22.04 / ROS2)

7.1 Get the Driver (or contact MANIFOLD Tech support for the latest version)

```
Shell
git clone https://github.com/manifoldsdk/odin_ros_driver.git
catkin_ws/src/odin_ros_driver    # Clone the driver from
GitHub, saved to ~/catkin_ws/ by default
```

7.2 Build ROS2 workspace (Ubuntu 22.04 / ROS2; place driver SDK in the /src folder)

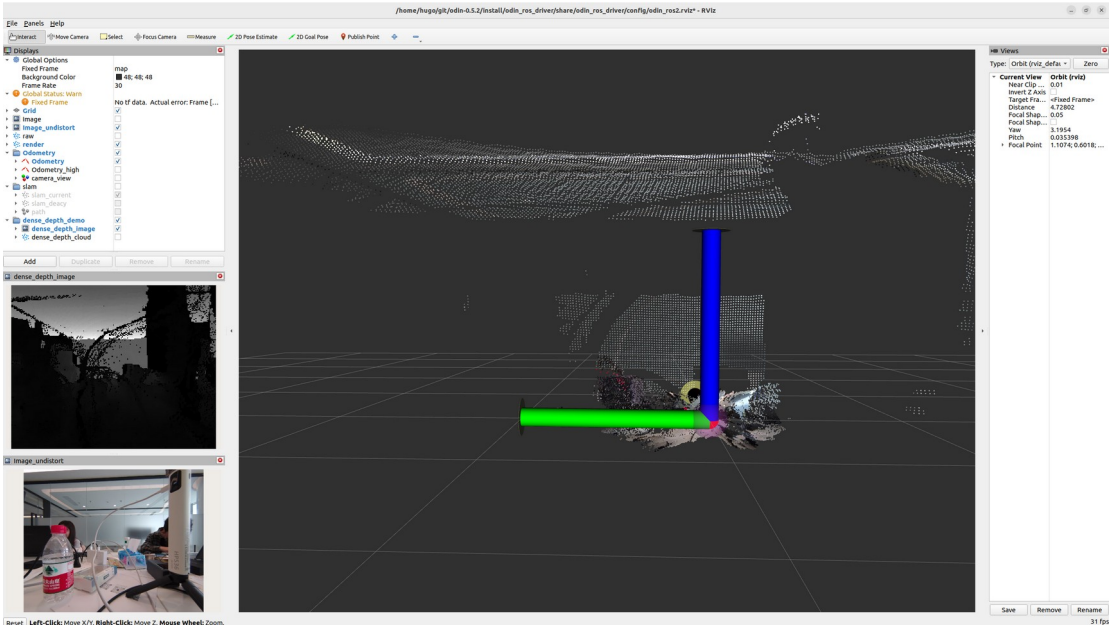
```
Shell
mkdir -p catkin_ws/src      # Skip if catkin_ws was already
created during git clone
cp -r /$PATH/odin_ros_driver/ ~/catkin_ws/src/      # 将git
下载的驱动放到 ros 工作空间中。$PATH 为下载的驱动所在位置
cd ~/catkin_ws/src/odin_ros_driver/script/
source ~/.bashrc
./build_ros2.sh                 #如果 sh 脚本没有权限可输入 sudo
chmod +x build_ros*.sh 赋予权限
```

7.3 Run the driver (ensure Odin1 is connected to a USB 3.0 port)

```
Shell
cd ~/catkin_ws     # Navigate to your ROS workspace
```

```
source install/setup.bash    # 声明变量
ros2 launch odin_ros_driver odin1_ros2.launch.py       # 执行 launch 文件
```

7.4 On success, the rviz window will appear:



7.5 Run the following command to list topics

```
Shell
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. Relocalization Mode

Odin supports three modes: Odometry, SLAM, and Relocalization. Mode switching is done via 'custom_map_mode' in /odin_ros_driver/config/control_command.yaml.

1. Odometry Mode

Set "custom_map_mode = 0" (default) to enable Odometry Mode. The map frame and odometry frame share the same coordinate system.

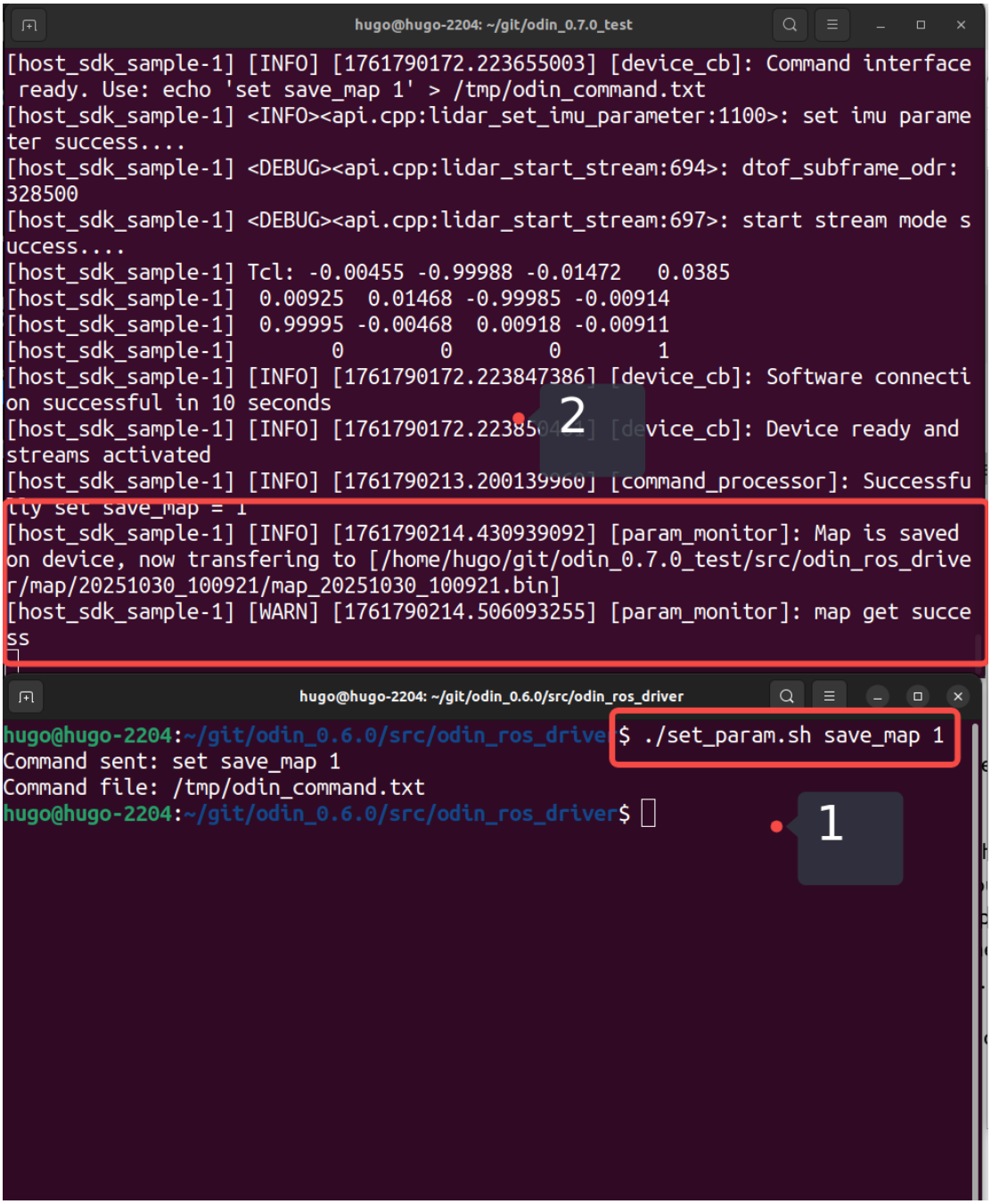
1. SLAM Mode

Set "custom_map_mode = 1" to enable SLAM Mode, which adds loop closure detection and map saving on top of Odometry Mode. Map save path:

```
Shell
/odin_ros_driver/map/
```

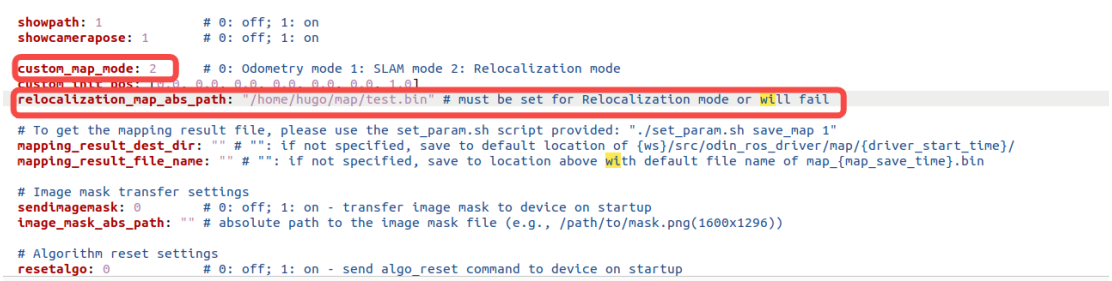
Start the driver. Odin1 will automatically build and cache the map. To stop recording:

```
Shell
cd $PATH/src/odin_ros_driver      # $PATH is your driver
source directory
./set_param.sh  save_map  1       # 执行保存脚本
```



1. Relocalization Mode

   Set "custom_map_mode = 2" to enable Relocalization Mode. Set "relocalization_map_abs_path" in /config/control_command.yaml to the absolute path of the map, e.g.:



Start the driver. On successful relocalization, rviz will show the current frame position within the map. If it fails, the system falls back to Odom mode and keeps retrying. Once successful, the TF between the map and odometry frames will be published.

*Note: The following topics are published in the odometry frame: `/odin1/cloud_slam`, `/odin1/odom`, `/odin1/highodom`, and `/odin1/path`. To use them in the map frame, apply the TF from the odometry frame to the map frame.

9. Importing Odin Recorded Data into MindCloud

9.1 Set the recorddata parameter in /src/odin_ros_driver/config/control_command.yaml:

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

9.2 Run the driver to collect map data

```
Shell
cd $PATH/catkin_ws
source install/setup.bash
ros2 launch odin_ros_driver odin1_ros2.launch.py
# After map collection, press Ctrl+C to stop the driver
```

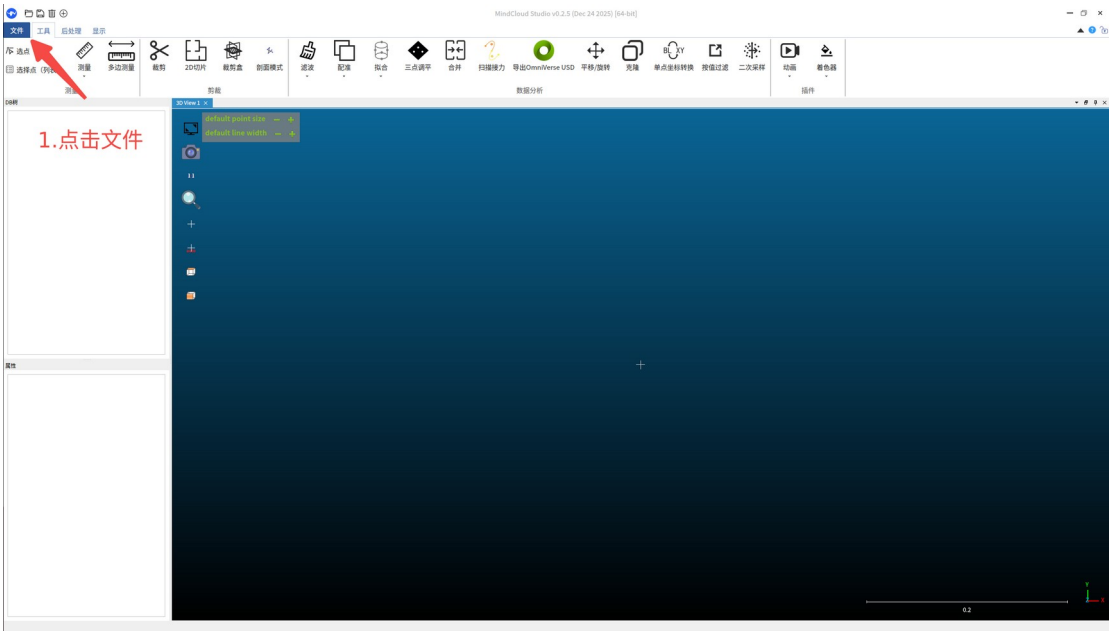
9.3 Verify /catkin_ws/src/odin_ros_driver/recorddata/***/image/cam_in_ex.txt is complete. A valid example:

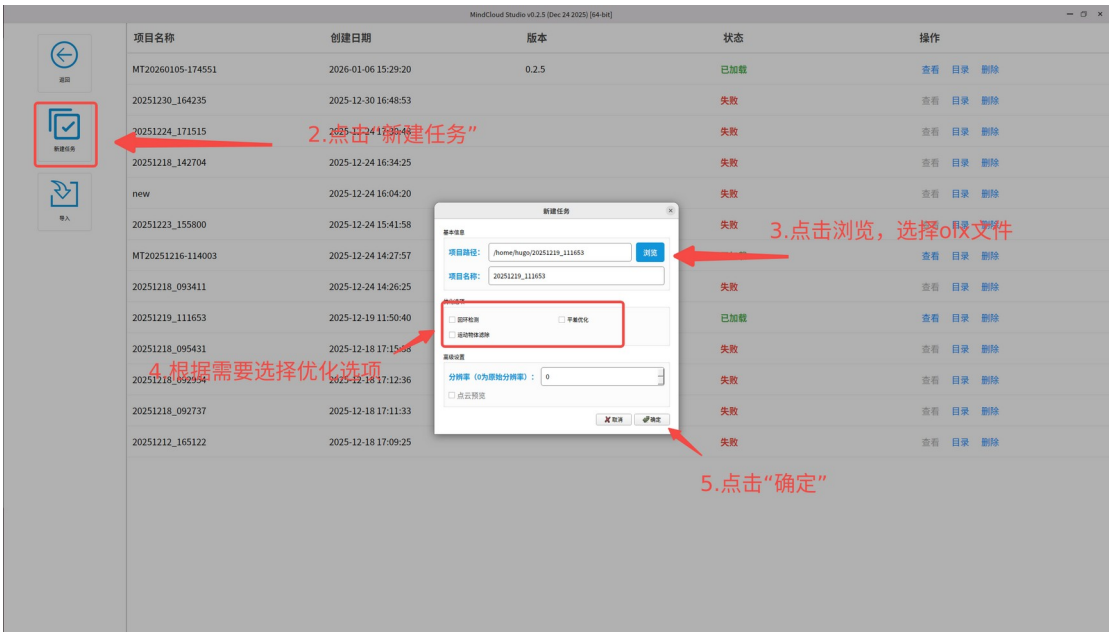
```
Shell
Tcl_0: [-0.004550, -0.999880, -0.014720, 0.038500,
        0.009250, 0.014680, -0.999850, -0.009140,
        0.999950, -0.004680, 0.009180, -0.009110,
        0.000000, 0.000000, 0.000000, 1.000000]
cam_0:
   image_width: 1600
   image_height: 1296
   k2: -0.006990
   k3: 0.003913
   k4: -0.019202
   k5: -0.001396
   k6: 0.010615
   k7: -0.005595
   p1: 0.000000
   p2: 0.000000
   A11: 731.137828
   A12: -0.682592
   A22: 730.873102
```

```
u0: 803.257996
v0: 640.726283
```

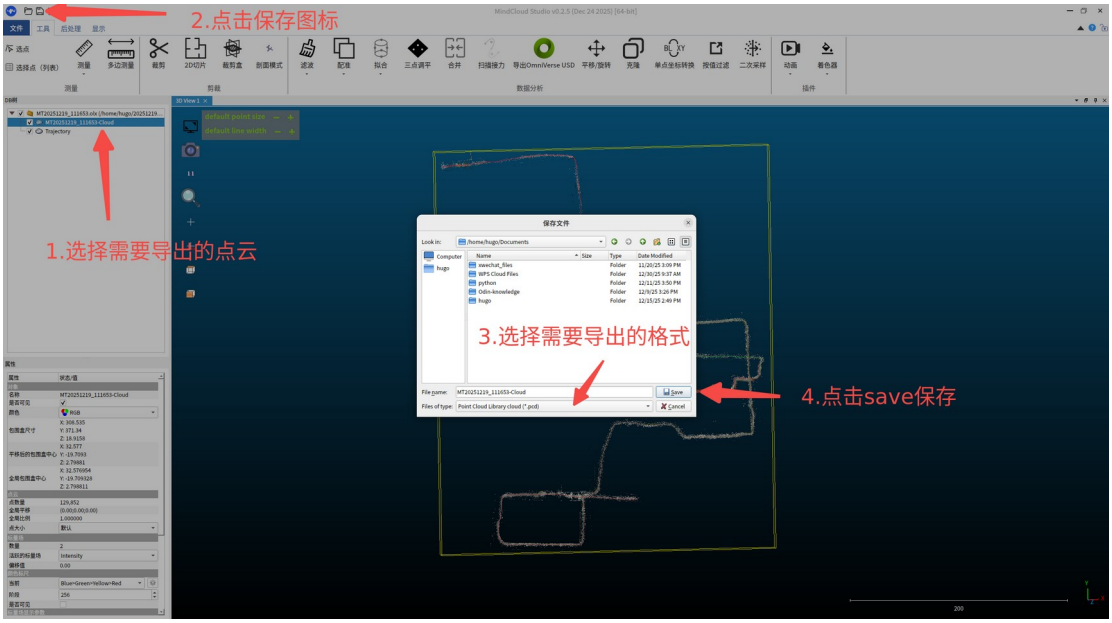
If the file is complete, the data is valid. If incomplete, the driver exited abnormally and the scan data is invalid.

9.4 Import Data into MindCloud





9.5 Export PCD File



10. Message Reference

Topic	control_command.yaml	Detailed Description
odin1/imu	sendimu	IMU Topic -- Raw IMU data at 400 Hz. Frame origin: IMU frame
odin1/image	sendrgb	RGB Camera Topic, decoded from original JPEG data, bgr8 format. Frame origin: camera frame
odin1/image_undistort	sendrgbundistort	Undistorted RGB Camera Topic, processed with calib.yaml. Frame origin: camera frame
odin1/image/compressed	sendrgbcompressed	Compressed RGB Camera Topic, original JPEG data. Frame origin: camera frame
odin1/cloud_raw	senddtof	Raw Cloud Topic -- Raw point cloud data. Frame origin: LiDAR frame
odin1/cloud_render	sendcloudrender	Render Cloud Topic -- Colored point cloud processed from raw cloud, RGB image, and calib.yaml. Frame origin: LiDAR frame
odin1/cloud_slam	sendcloudslam	SLAM PointCloud Topic -- Colored

		point cloud fusing LiDAR/camera/IMU after SLAM. Frame origin: odom frame
odin1/odometry	sendodom	Odometry Topic -- Odometry data. Frame origin: odom frame
odin1/ odometry_high	sendodom	High-frequency Odometry Topic. Frame origin: odom frame
odin1/ depth_img_competetion	senddepth	Dense Depth Image Topic. Demo, high computing power required. 1:1 with odin1/image_undistort. Subscribe directly; values are already depth data. Frame origin: camera frame
odin1/ depth_img_competetion_cloud	senddepth	Dense Depth Cloud Topic. Demo, high computing power required. Frame origin: camera frame
odin1/path	showpath	Odometry Path Topic -- Displays trajectory.
tf	sendoom	TF tree topic -- Transform between odometry and map frames
odin1/image/ intensity_gray	pubintensitygray	dToF reflectance intensity grayscale image for dToF module health evaluation
/odin1/ reprojected_image	sendreprojection	Projects 3D point cloud onto 2D image
/odin1/ overlay_image	sendoverlay	Overlays reprojected image onto undistorted image to verify projection accuracy
/odin1/wiwc	Published by default	Real-time estimated per-frame extrinsic parameters (t_cl/t_il) used by the algorithm

11. FAQ

1. How to connect the cables?

    1. Two cables are provided: a power cable with an aviation connector and a USB 3.0 data cable. The customer must provide a power adapter (DC5521/5524, output 12V 2A). The device supports 12~24V wide voltage; exceeding this range will damage internal components.

1. No display in rviz after connecting
    1. Try using a more stable power supply or slightly increase the voltage.
    1. Ensure the udev rules file from section 6.2 has been written correctly.6.2
    1. Check the terminal output for version info; ensure you are using the latest firmware and driver.
1. Can a prior map be loaded for relocalization?
    1. Yes, but the map must come from one of the following sources:
        1. Map saved using Odin's SLAM mode (map mode: 1) as a .bin file.
        1. Map collected with MANIFOLD handheld scanner Q9000, processed in MindCloud, and exported as a .bin file.
1. How to tell if relocalization succeeded?
    1. After successful relocalization, the /tf topic will publish the map frame transform, which is also visible in rviz.
    1. The terminal will print: relocalization succeeded
1. For other issues, please refer to the README.md file.