

The background features a large, abstract graphic on the left side composed of dark grey, green, and yellowish-green triangles. A thin white line runs diagonally from the bottom-left corner towards the top-right.

Auto Navigating Rover

Group 8



Aline Cynthia Yiagnigni



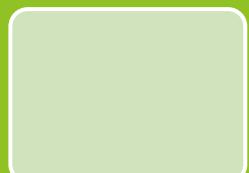
Keshav Kumar



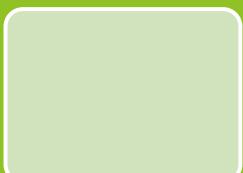
Cristobal Gallardo



Matheus Dias Cirillo



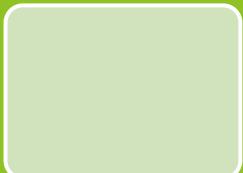
Emilio Antoine
Altamirano



Smit Nareshbhai
Kotadiya



Gustavo Moura Scarenci



Sumit Mor



Goal

- ▶ Mapping a Physical Environment and Identifying Obstacles, in order to autonomously traverse the terrain safely
- ▶ Inspired from *Perseverance Rover*

The rover



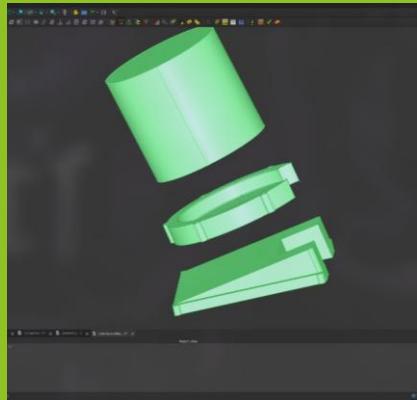
AgileX Scout Mini



Hardware

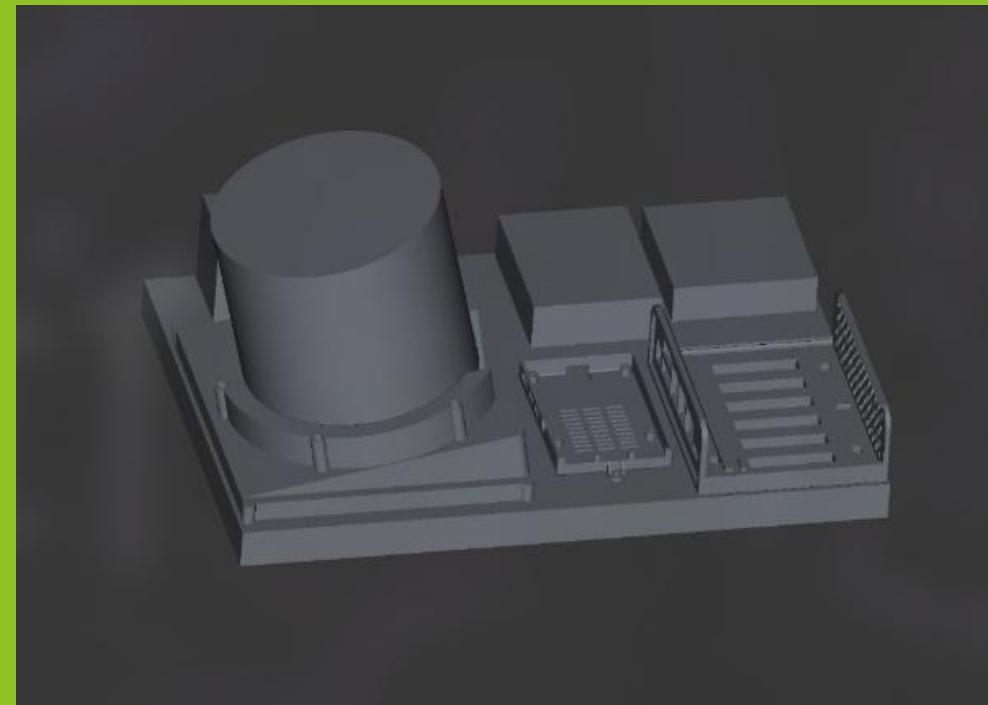
Mechanical Design & CAD Modeling

- ▶ The planning and digital creation phase.



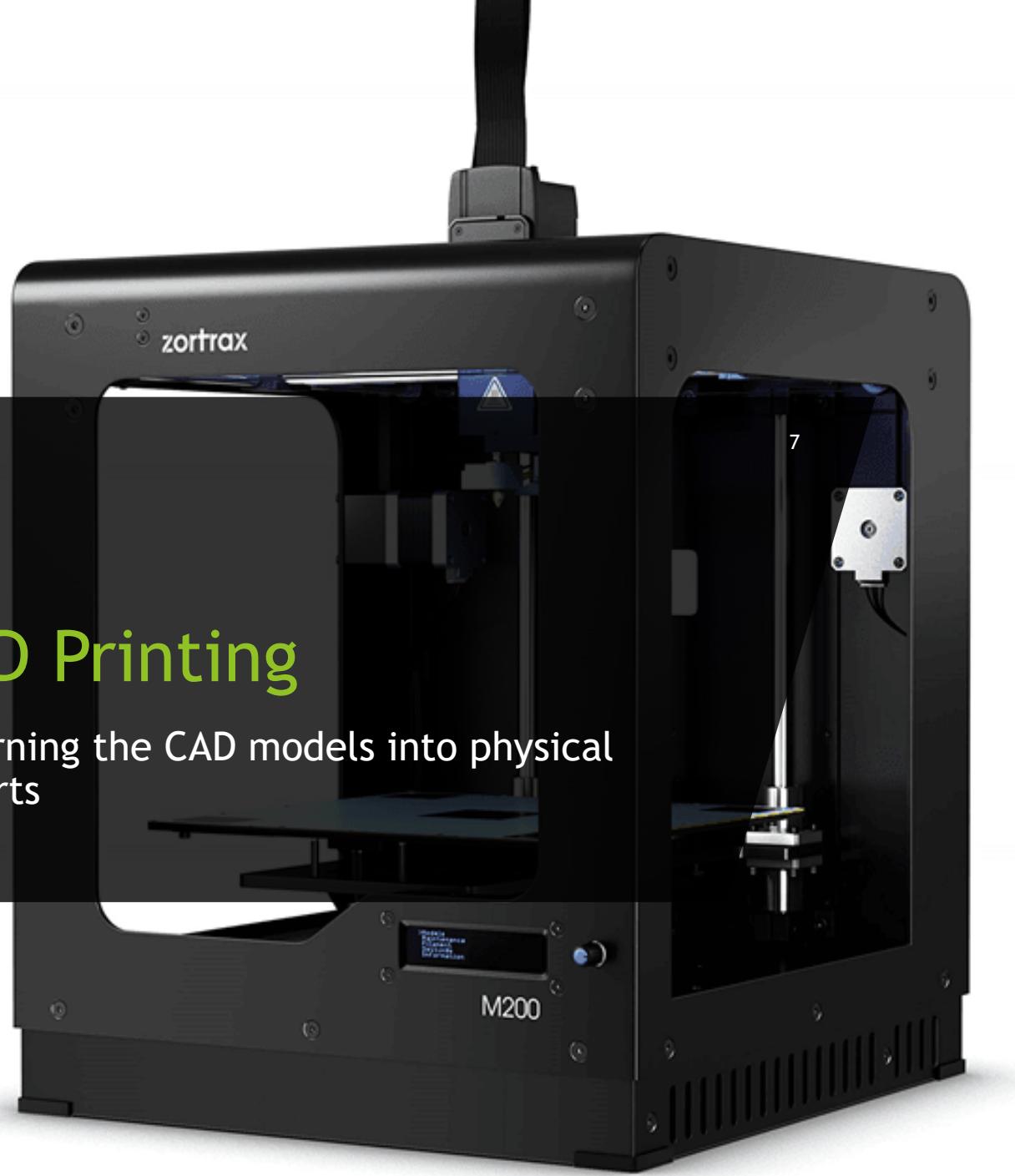
*LiDAR Mounting Assembly -
Exploded View*

Full Hardware Layout /
Digital System Assembly

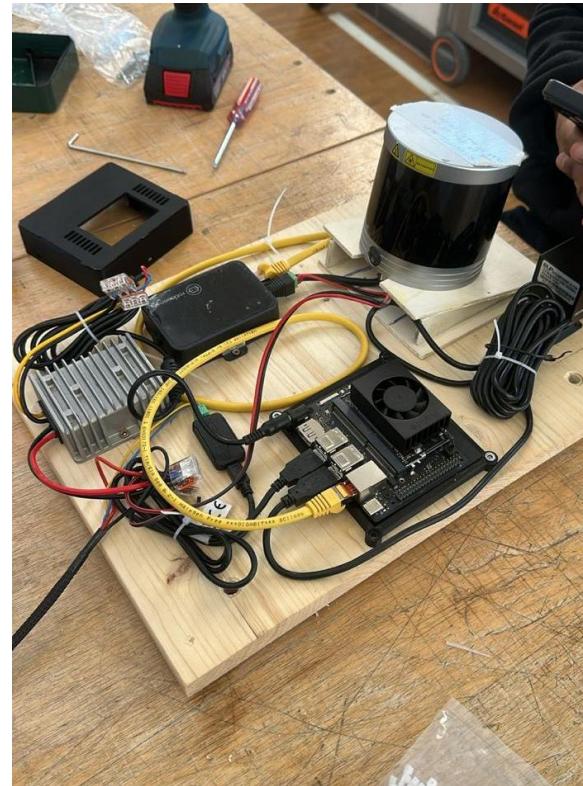


3D Printing

Turning the CAD models into physical parts

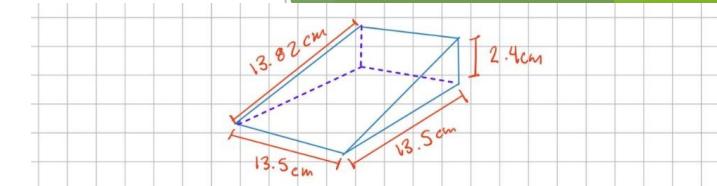


Hardware Mount Integration & Assembly



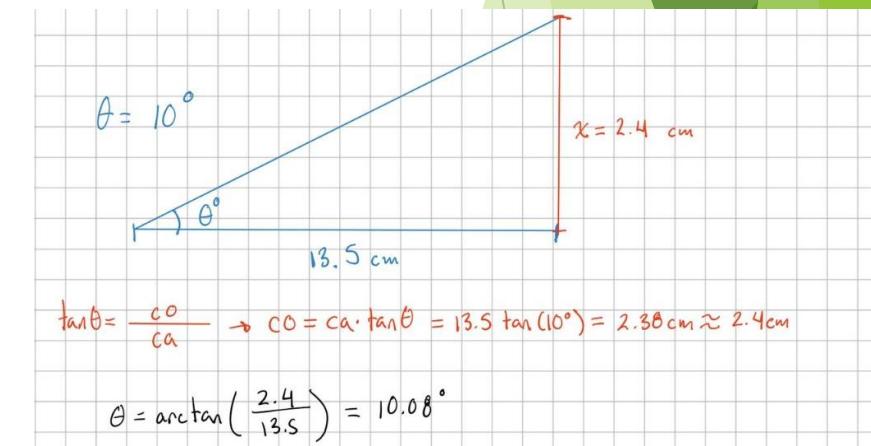
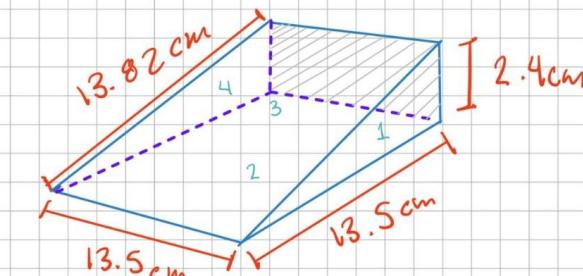
Geometric Design Parameters & Calculations

- ▶ **Sensor Angle Definition:** Defined a required inclination angle to optimize the sensor's scanning area.
- ▶ **Dimensional Analysis:** Calculated the vertical offset based on the available base length using trigonometric function
- ▶ **Slope Calculation:** Determined the hypotenuse length to ensure the 3D printed part fit the chassis perfectly.



$$\sin \theta = \frac{op}{hyp} \rightarrow h = \frac{op}{\sin \theta} = \frac{2.4 \text{ cm}}{\sin(10^\circ)} = 13.82 \text{ cm}$$

Not a wall



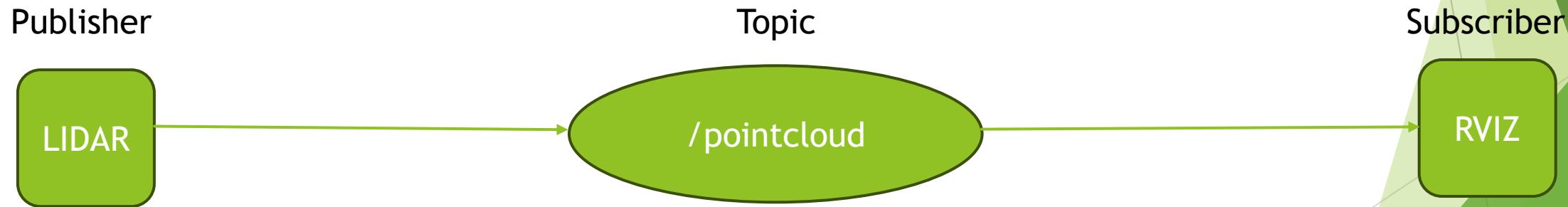
System Validation & Field Testing



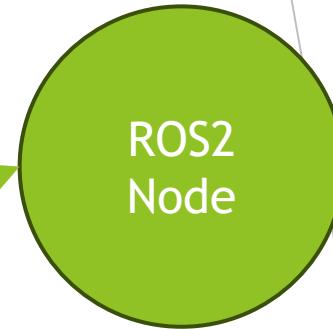
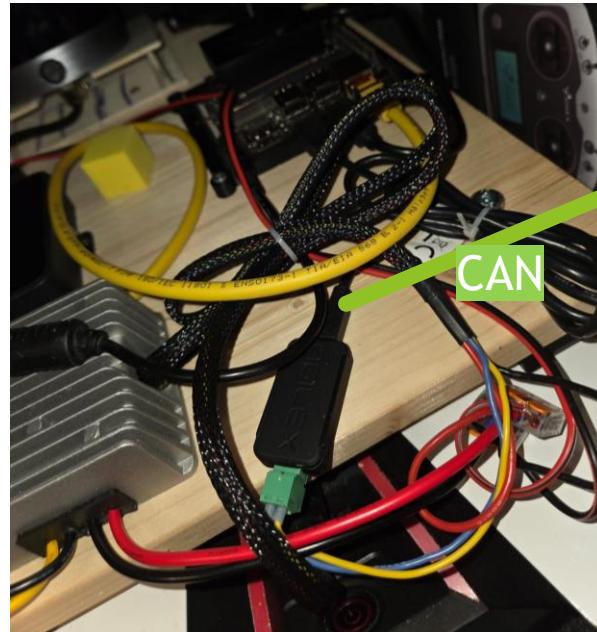
- ▶ the integrated system works under real-world conditions.
 - Structural Integrity Check
 - Vibration Assessment

ROS

- ▶ Based Pub-Sub Model
- ▶ ROS2 Humble Hawkulls
- ▶ Why?: Due to multiple Pieces of Hardware



Rover Communication



ROS2 Node

agilxrobotics / **scout_ros2** Public
forked from [westonrobot/scout_ros2](#)

Code Issues 10 Pull requests 3 Actions Projects Security Insights

scout_ros2 Public
forked from [westonrobot/scout_ros2](#)

humble 4 Branches 0 Tags Go to file Add file <> Code

This branch is 4 commits ahead of and 14 commits behind [westonrobot/scout_ros2:humble](#).

agilxrobotics add rc state bdbb904 · 10 months ago 9 Commits

scout_base add rc state 10 months ago

scout_description initial commit, code compiles but not working yet 5 years ago

scout_msgs modify msg files for scout 4 years ago

.gitignore initial commit, code compiles but not working yet 5 years ago

.gitlab-ci.yml initial commit, code compiles but not working yet 5 years ago

LICENSE Initial commit 5 years ago

README.md updated readme 5 years ago

README Apache-2.0 license

ROS2 Packages for Scout Mobile Robot

Packages

This repository contains minimal packages to control the scout robot using ROS.

- scout_base: a ROS wrapper around [ugv_sdk](#) to monitor and control the scout robot
- scout_description: URDF model for the mobile base
- scout_msgs: scout related message definitions

About

ROS2 Support Package for Scout Robot

- Readme
- Apache-2.0 license
- Activity
- 34 stars
- 1 watching
- 43 forks

Releases

No releases published

Packages

No packages published

Languages

C++ 54.5% Python 34.4%
CMake 10.9% Shell 0.2%

westonrobot / **scout_ros2** Public
Merge pull request #11 from westonrobot/fix-auto_reconnect_can_control · 34bf5df · 2 years ago 19 Commits

Code Issues 1 Pull requests 2 Actions Projects Security Insights

scout_ros2 Public

humble 2 Branches 0 Tags Go to file Add file <> Code

scout_base Launch: scout_base.launch: cleanup launch file in... 2 years ago

scout_description urdf: add scout mini meshes and urdf 2 years ago

scout_msgs initial commit, code compiles but not working yet 5 years ago

.gitignore scout_ros2: updated .gitignore 2 years ago

.gitlab-ci.yml initial commit, code compiles but not working yet 5 years ago

LICENSE Initial commit 5 years ago

README.md fixed for scoutv2, added QoS settings 4 years ago

README Apache-2.0 license

ROS2 Packages for Scout Mobile Robot

Packages

This repository contains minimal packages to control the scout robot using ROS.

- scout_base: a ROS wrapper around [ugv_sdk](#) to monitor and control the scout robot
- scout_description: URDF model for the mobile base
- scout_msgs: scout related message definitions

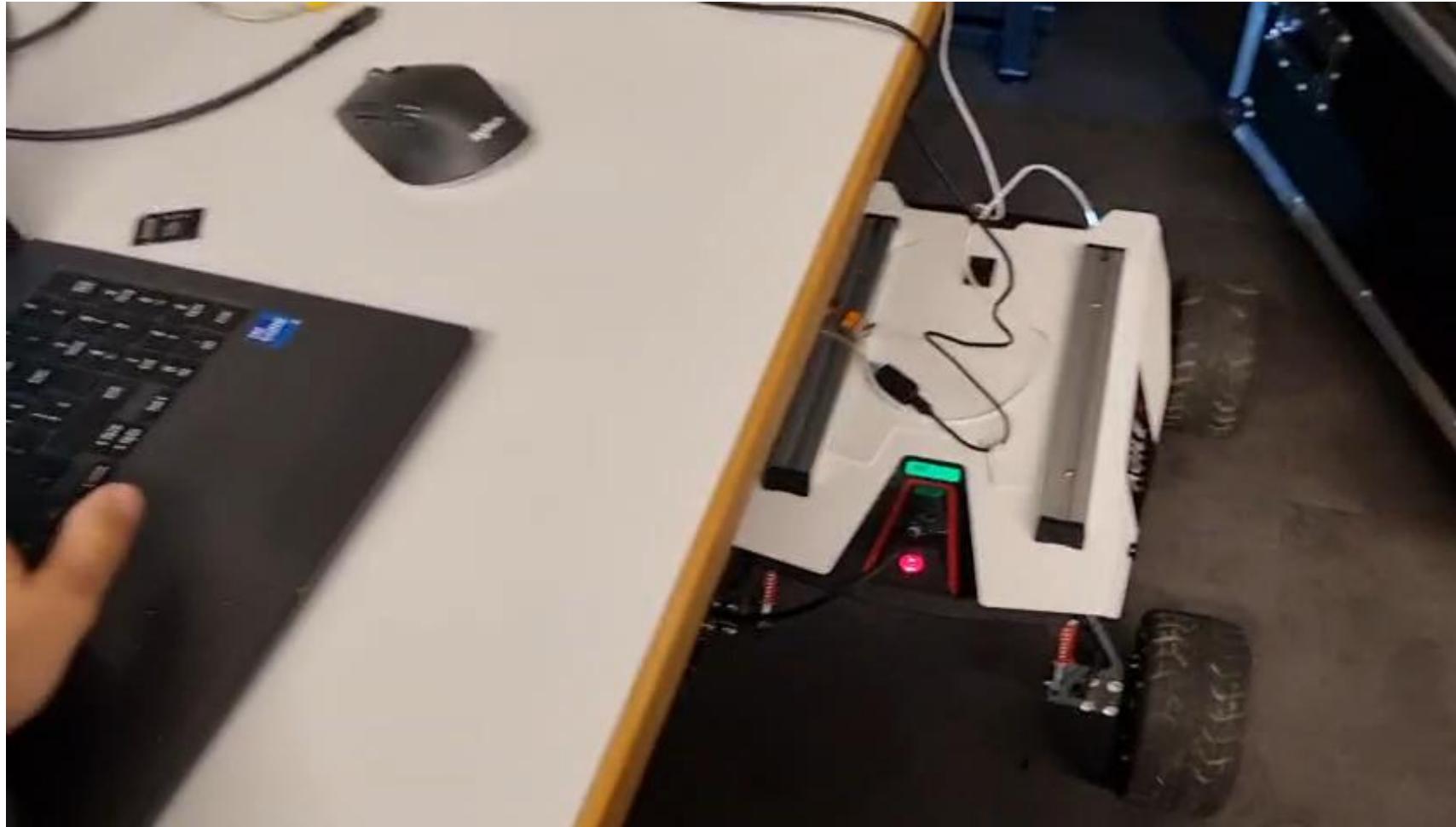
Supported Hardware

Contributors 6

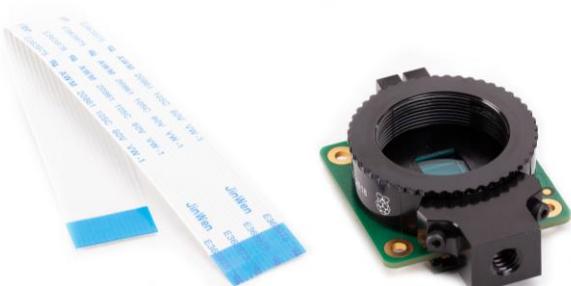
Languages

C++ 55.5% Python 34.0%
CMake 10.3% Shell 0.2%

First ROS2 operation



Camera



Camera

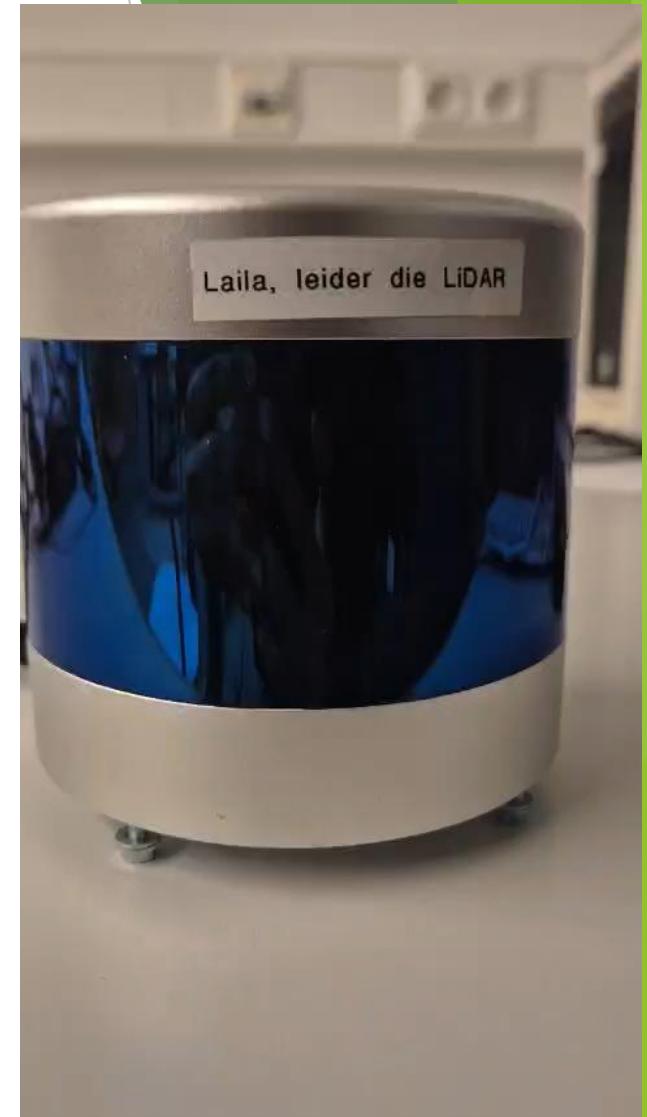


Camera



Leihsen C16 LIDAR

- ▶ Data Transmission was faulty
- ▶ Trial and Error: Check IP address, .config files, connection, Byte rate etc
- ▶ Configurations were right but problem remains unknown
- ▶ Conclusion: Change the LIDAR



RS Helios 16P

- ▶ LIDAR and Jetson are connected with Ethernet
- ▶ LIDAR send the data to a pre-configured IP address
- ▶ Sends data in UDP packets and publishes them in /rslidar_points

Source	Destination	Protocol	Length	Info
192.168.26.60	192.168.26.20	UDP	1290	6699 → 6699 Len=1248
192.168.26.60	192.168.26.20	UDP	1290	6699 → 6699 Len=1248
192.168.26.60	192.168.26.20	UDP	1290	6699 → 6699 Len=1248

In the Terminal:

1. Change the IP address of the Host Device to 192.168.26.20, LIDAR sends the data on this IP:

```
sudo ip addr add 192.168.26.20 dev eth0  
sudo ip link set up dev eth0
```

2. Check if the IP address is up:

```
ip a          # Should show IP of eth0 as set above  
ip neigh      # Should show eth0 as STALE or REACHABLE
```

3. **OPTIONAL** -> To check if the LIDAR is being detected and sending the UDP packets respectively:

```
sudo apt install wireshark    # Check if eth0 is STALE or REACHABLE  
sudo tshark -i eth0 -f "udp"
```

4. Activate ROS and Launch:

```
source /opt/ros/humble/bash  
cd rs_lidar  
source install/setup.bash  
ros2 launch rslidar_sdk start.py
```

5. Visualize the Point Cloud:

```
ros2 topic list          #You should see /rslidar_points  
rviz2
```

In RIVZ2 -> In Global Options, set Fixed Frames to **rslidar** and Under PointCloud2 set Topic to **/rslidar_points**

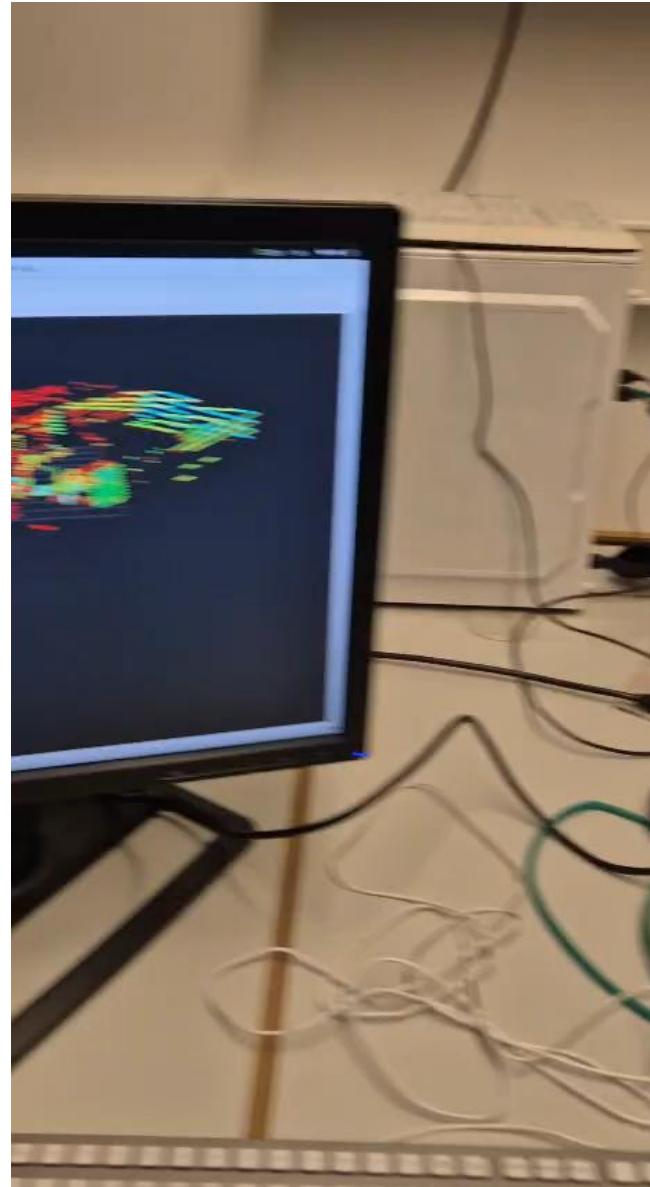
You should see the Points on the right Visualization Screen

Miscellaneous: LIDAR IP: 192.168.26.60

LIDAR Start-Up Script

RVIZ

- ▶ Visualizer
- ▶ Captures the Data from */rslidar_points*



CPUs and GPUs



Cannot handle LiDAR (low processing power)



No official support for Ubuntu 22.04



Kernel not configured with CAN by default

Startup Script & Remote Operation

```

1 #!/bin/bash
2
3 # Installation Instructions:
4 # 1. Place the script at /usr/local/bin
5 #   (e.g., sudo cp scout_control.sh /usr/local/bin/rover)
6 # 2. This script name should be rover
7 # 3. Make it executable: sudo chmod +x /usr/local/bin/rover
8 #
9 # Usage:
10 # Always launch using: rover {base|teleop|lidar|camera}
11
12 # Define the workspace path - Change this if your workspace is elsewhere
13
14 ROVER_PATH="$HOME/rosl2_ws"
15 LIDAR_WS_PATH="$HOME/rs_lidar"
16
17
18 if [ -z "$1" ]; then
19     echo "Usage: $0 {base|teleop|lidar|camera}"
20     exit 1
21 fi
22
23 v setup_can() {
24     echo "Checking CAN interface..."
25     if ip link show can0 | grep -q "UP"; then
26         echo "CAN0 is already up."
27     else
28         echo "Setting up CAN0..."
29         sudo modprobe gs_usb
30         sudo ip link set can0 up type can bitrate 500000
31
32         if [ $? -eq 0 ]; then
33             echo "CAN0 setup successful."
34         else
35             echo "Failed to setup CAN0. Check connections."
36             exit 1
37         fi
38     fi
39 }
40
41 get_wifi_ip() {
42     ip -4 addr show | grep -E 'wlan|wlp|wlx' | grep inet | awk '{print $2}' | cut -d/ -f1 | head -n 1
43 }
44
45 source /opt/ros/humble/setup.bash
46
47 case "$1" in

```

```

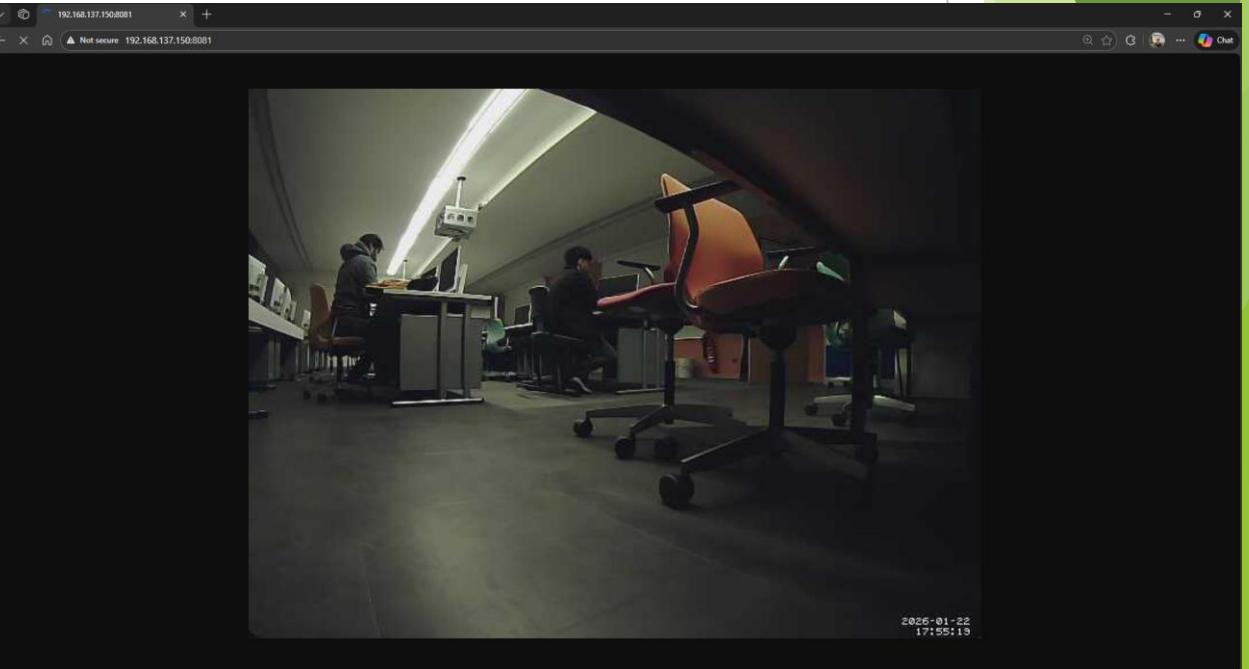
[jetson@jetson ~] [scout_base_node-1] [INFO] [1769100689.508841499] [scout]: - sim control rate: 50
[jetson@jetson ~] [INFO] [1769100689.508841499] [scout]: - Robot base: Scout
[jetson@jetson ~] [INFO] [1769100689.508841499] [scout]: Start listening to port: can0
[INFO] [robot_state_publisher-1]: process started with pid [5052]
[jetson@jetson ~] [INFO] [1769100689.571452013] [scout]: Detected protocol: AGX_V2
[jetson@jetson ~] [INFO] [1769100689.571452013] [scout]: Creating interface for Scout with AGX_V2 Protocol
[jetson@jetson ~] [INFO] [1769100689.571452013] [scout]: Robot initialised, start running...
[jetson@jetson ~] [INFO] [1769100689.571961302] [scout]: Using CAN bus to talk with the robot
[jetson@jetson ~] [INFO] [1769100689.571961302] [robot_state_publisher]: got segment base_footprint
[jetson@jetson ~] [INFO] [1769100689.641689701] [robot_state_publisher]: got segment base_link
[jetson@jetson ~] [INFO] [1769100689.641712703] [robot_state_publisher]: got segment front_left_wheel_link
[jetson@jetson ~] [INFO] [1769100689.641723231] [robot_state_publisher]: got segment front_right_wheel_link
[jetson@jetson ~] [INFO] [1769100689.641734269] [robot_state_publisher]: got segment left_wheel_link
[jetson@jetson ~] [INFO] [1769100689.641739836] [robot_state_publisher]: got segment oak_d_frame
[jetson@jetson ~] [INFO] [1769100689.641747708] [robot_state_publisher]: got segment oak_imu_frame
[jetson@jetson ~] [INFO] [1769100689.641755128] [robot_state_publisher]: got segment oak_left_camera_frame
[jetson@jetson ~] [INFO] [1769100689.641762484] [robot_state_publisher]: got segment oak_left_camera_optical_frame
[jetson@jetson ~] [INFO] [1769100689.641769807] [robot_state_publisher]: got segment oak_model_origin
[jetson@jetson ~] [INFO] [1769100689.641777195] [robot_state_publisher]: got segment oak_rgb_camera_frame
[jetson@jetson ~] [INFO] [1769100689.641784261] [robot_state_publisher]: got segment oak_rg_camera_optical_frame
[jetson@jetson ~] [INFO] [1769100689.641791423] [robot_state_publisher]: got segment oak_right_camera_frame
[jetson@jetson ~] [INFO] [1769100689.641799056] [robot_state_publisher]: got segment oak_right_camera_optical_frame
[jetson@jetson ~] [INFO] [1769100689.641806425] [robot_state_publisher]: got segment rear_left_wheel_link
[jetson@jetson ~] [INFO] [1769100689.641816169] [robot_state_publisher]: got segment rear_right_wheel_link
[jetson@jetson ~] [INFO] [1769100689.641816169] [robot_state_publisher]: Motion detected - starting event 1

[jetson@jetson ~] [motion] [N/C] [ALL] motion_startup: Motion 4.3.2 Started
[jetson@jetson ~] [motion] [N/C] [ALL] motion_startup: Using default log type (INFO)
[jetson@jetson ~] [motion] [N/C] [ALL] motion_startup: Setting log level to INFO
[jetson@jetson ~] [motion] [N/C] [STR] webu_start_ctrl: Starting all camera streams on port 8081
[jetson@jetson ~] [motion] [N/C] [STR] webu_start_ctrl: Starting webcontrol on port 8080
[jetson@jetson ~] [motion] [N/C] [ENC] ffmpeg global init: ffmpeg libavcodec version 58.134.100 libavformat version 58.76.100
[jetson@jetson ~] [motion] [N/C] [ALL] translate_init: Language: English
[jetson@jetson ~] [motion] [N/C] [ALL] motion_start_thread: Camera ID 0 is from /etc/motion/motion.conf
[jetson@jetson ~] [motion] [N/C] [ALL] motion_start_thread: Device ID 0 is from /etc/motion/motion.conf
[jetson@jetson ~] [motion] [N/C] [ALL] main: Waiting for threads to finish, pid: 5078
[jetson@jetson ~] [INFO] [N/C] [VID] vid_start: Opening VID1 device
[jetson@jetson ~] [INFO] [N/C] [VID] vid_start: Device VID1 is /dev/video0 and input -1
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_device_capability: - VIDEO CAPTURE
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_device_capability: - STREAMING
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_input_select: Name: Input - CAM0
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_input_select: Device doesn't support specifying PAL/NTSC norm
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_select: Configuration palette index 17 (YU12) for 640x480 doesn't work.
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_select: Supported palettes:
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_select: (0) YUV (Monochrome)
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_select: (1) YUV (YUV4:2:2)
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_set: Testing palette YUV (640x480)
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_pixfmt_set: Using palette YUV (640x480)
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_image_rotate: Setting rotate buffer to 1 items
[jetson@jetson ~] [INFO] [N/C] [VID] v4l2_image_rotate: Setting rotate buffer to 1 items
[jetson@jetson ~] [INFO] [N/C] [VID] event_newFile: file of type 8 saved to: ./01-20260122175235.mkv
[jetson@jetson ~] [INFO] [N/C] [ALL] motion_detected: Motion detected - starting event 1

[jetson@jetson ~] [rslidar_sdk_node-1] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: RoboSense Transform Parameters
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: dense_points: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: ts_first_point: 1
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: config from file: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: angle_path: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: start_angle: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: end_angle: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: split_Frame_mode: 1
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: split_angle: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: num_bisks_split: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: num_bisks: 0
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: For Holonomic mode (strafing), hold down the shift key:
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: U I O
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: D K L
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: M , .
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: t : up (+z)
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: b : down (-z)
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: anything else : stop
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: g/z : increase/decrease max speeds by 10%
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: w/x : increase/decrease only linear speed by 10%
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: w/c : increase/decrease only angular speed by 10%
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: CTRL-C to quit
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: currently: speed 0.30 turn 0.31
[jetson@jetson ~] [INFO] [1769100689.641762484] [rslidar_sdk_node-1]: ,j1

[jetson@jetson ~] [1754] 1754
[jetson@jetson ~] [1754] 22/01/2026

```



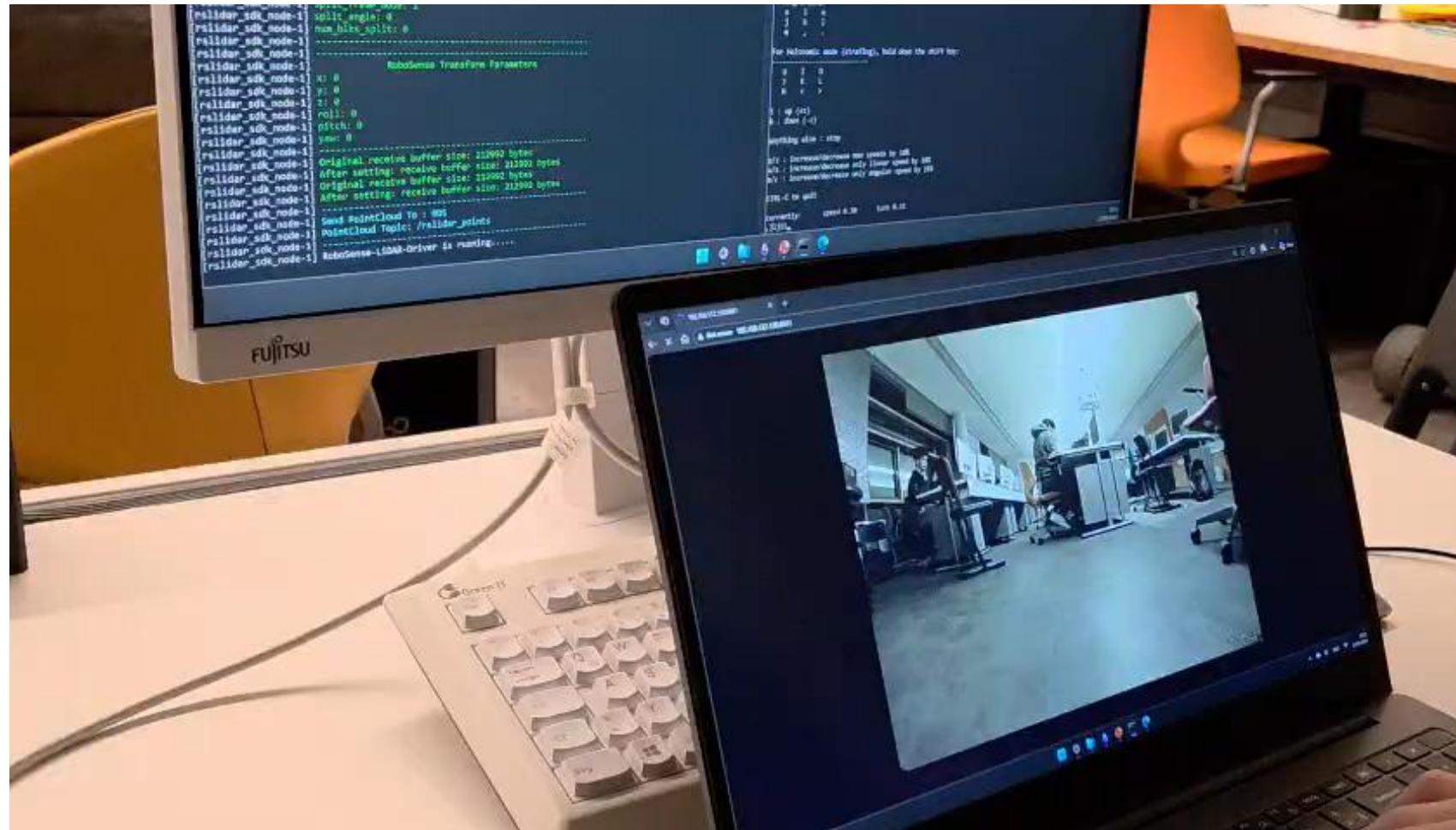
First remote operation



Final lab remote operation



Final lab remote operation



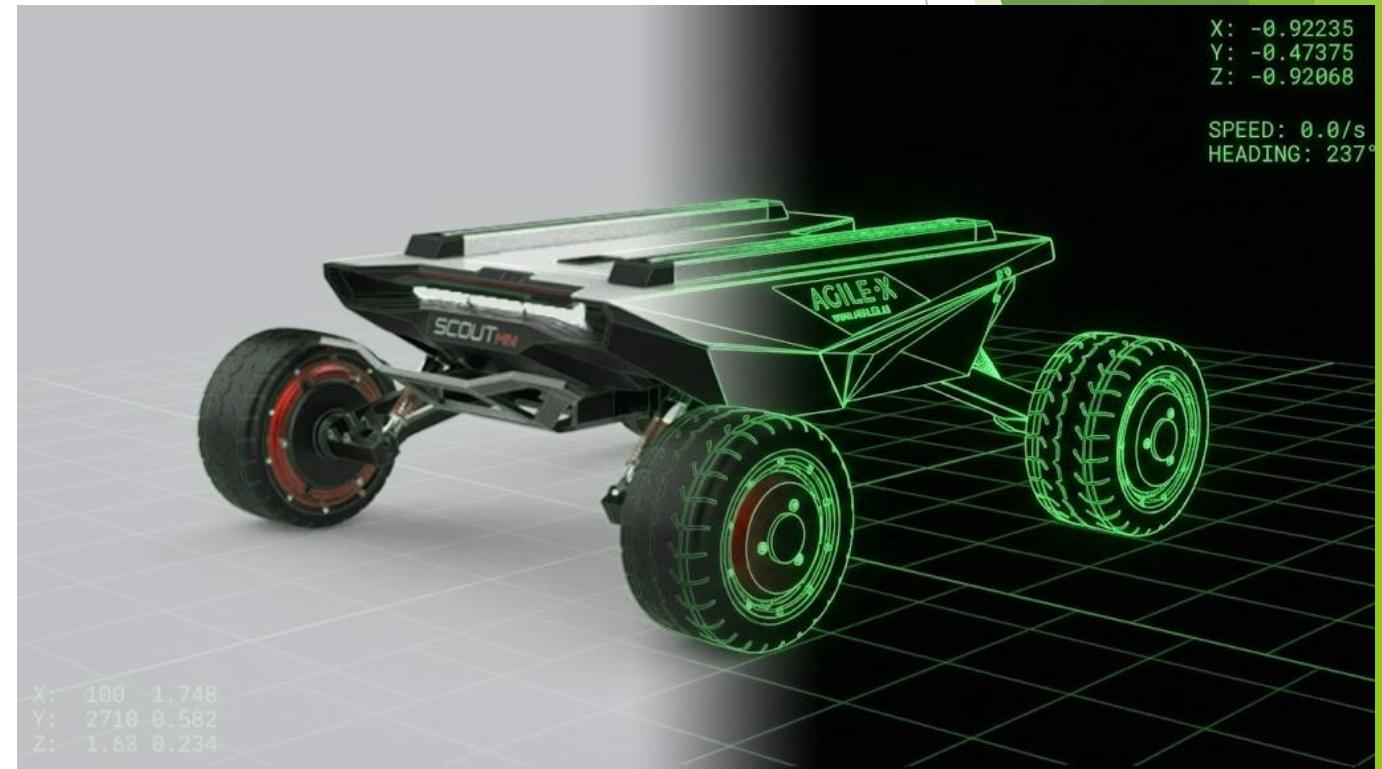
Field remote operation



Simulation

Objectives, roles and goals of the simulation in the project

- ▶ To familiarize the team with ROS 2 concepts;
- ▶ Develop a safe, reliable testing environment for the "Scout Mini" UGV;
- ▶ Isolate hardware problems from coding problems;
- ▶ Solving problems like the "Perception Disparity" (syncing movement in Gazebo vs. RViz);
- ▶ Finally achieve autonomous navigation and dynamic pathfinding.



Technologies used



Operating System: Ubuntu 22.04

- Stable Linux distribution
- Long-term support (LTS)
- Official support for ROS 2 Humble



Simulation Engine: Gazebo Fortress

- Physics-based simulation
- 3D environment rendering
- Sensor simulation



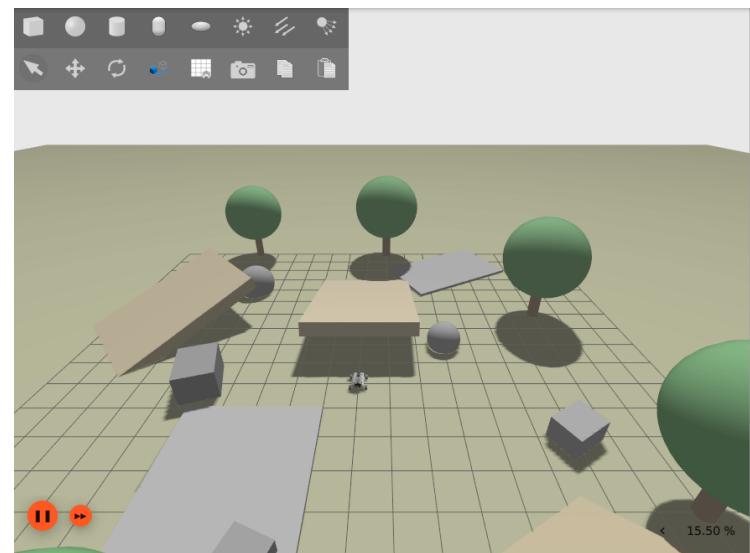
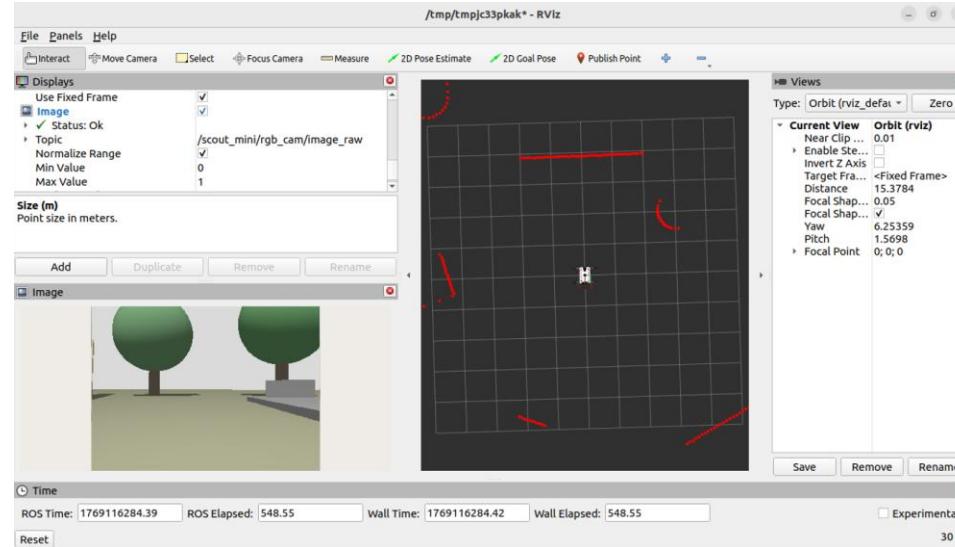
Robot Middleware : ROS 2 Humble

- Node-based architecture
- Topic-based communication
- Modular and scalable



Visualization Tools: RViz 2

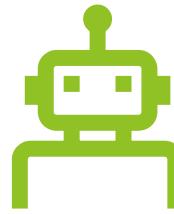
- Robot visualization
- Sensor data display
- Map visualization





Robot & Environment Description Formats

- URDF / Xacro → Robot model
- SDF → Simulation world



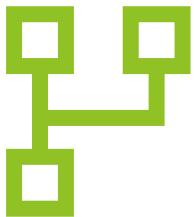
ROS–Gazebo Communication: ros_gz bridge

- .Connects ROS 2 and Gazebo
- .Sensor data → ROS
- .Control commands → Gazebo



Navigation & Mapping Tools

- .Slam toolboxes
- . Nav 2



Programming & Build Tools

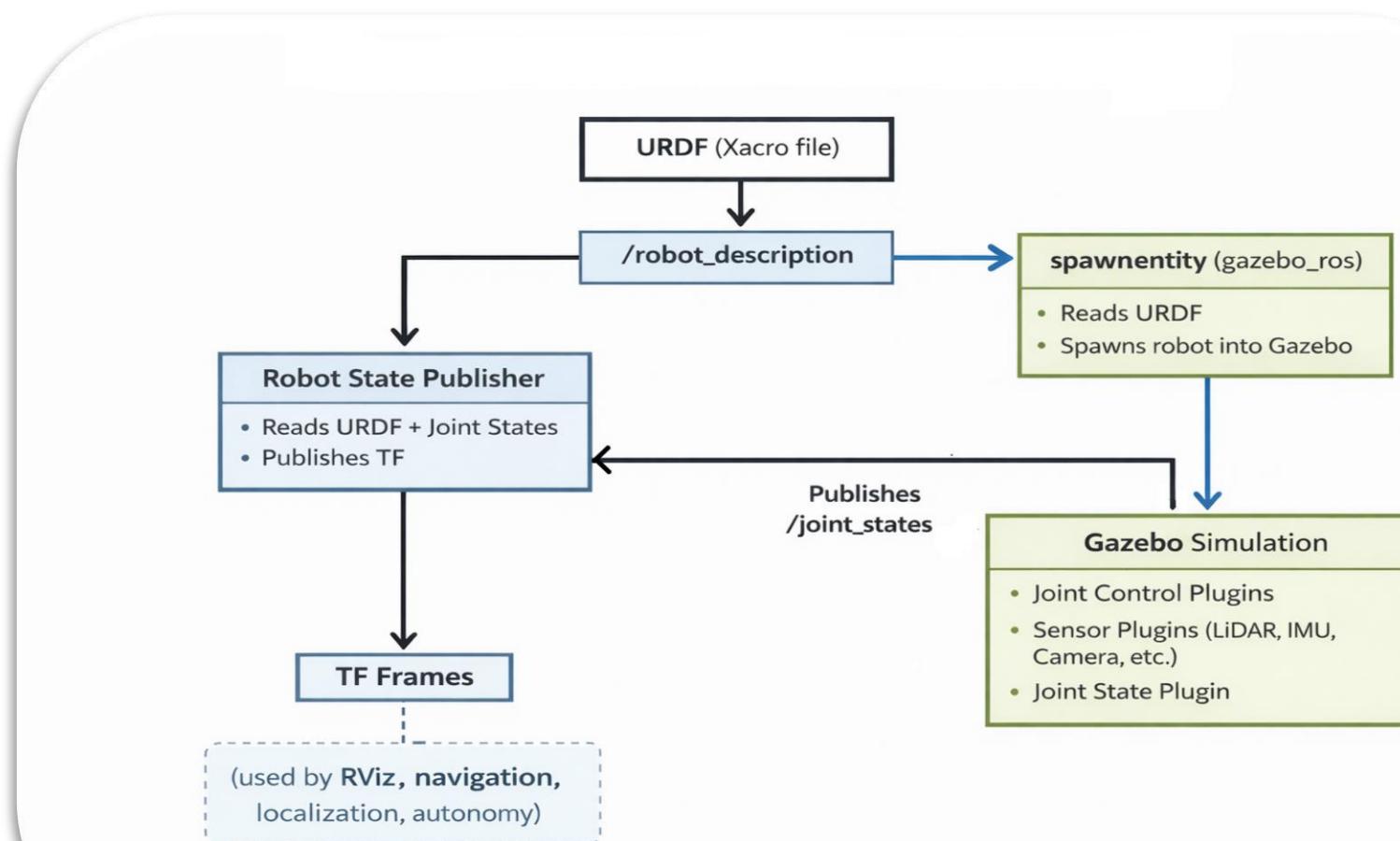
- python
- Colcon
- rosdep



languages

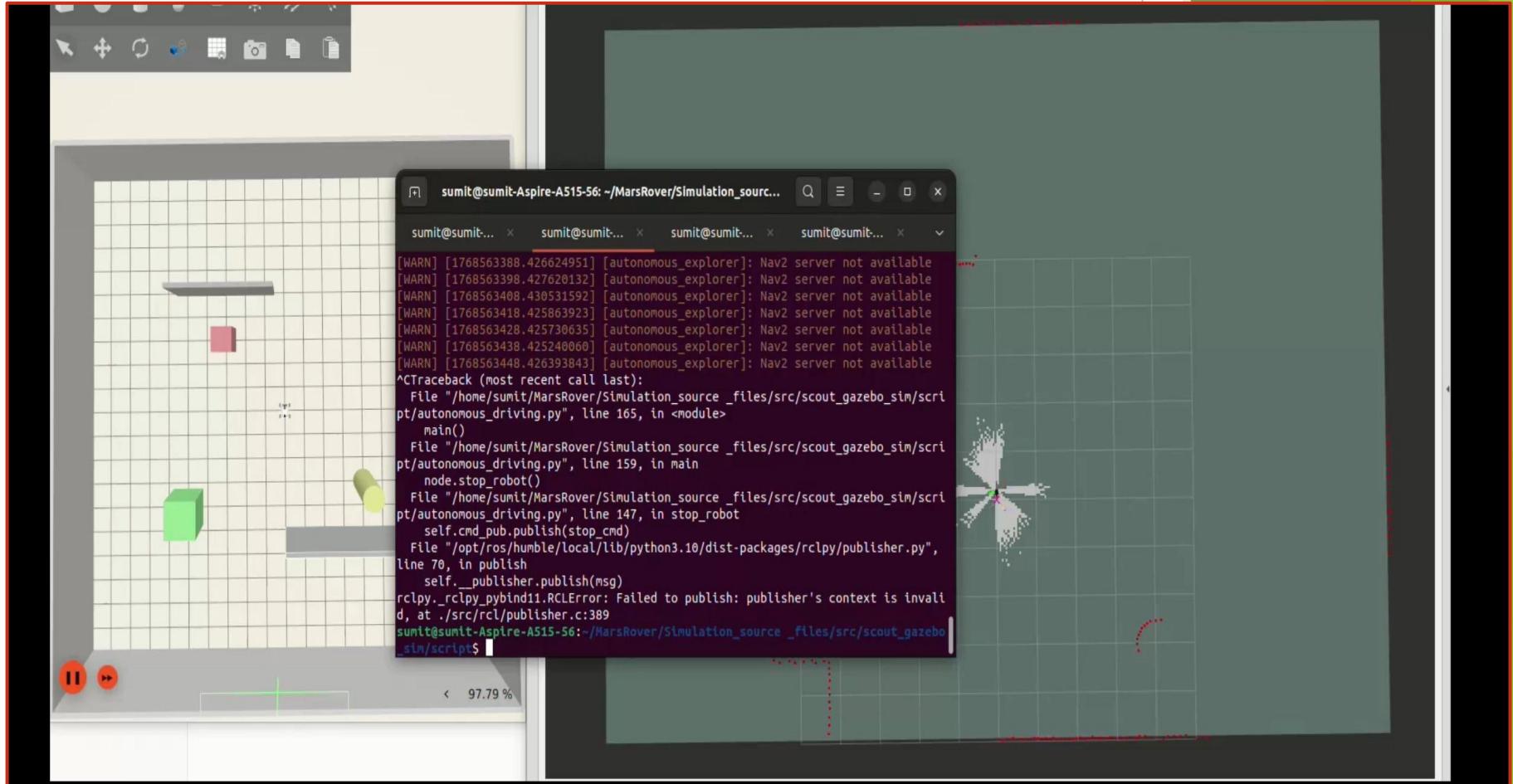
- .XML
- .python

Design of the simulation environment



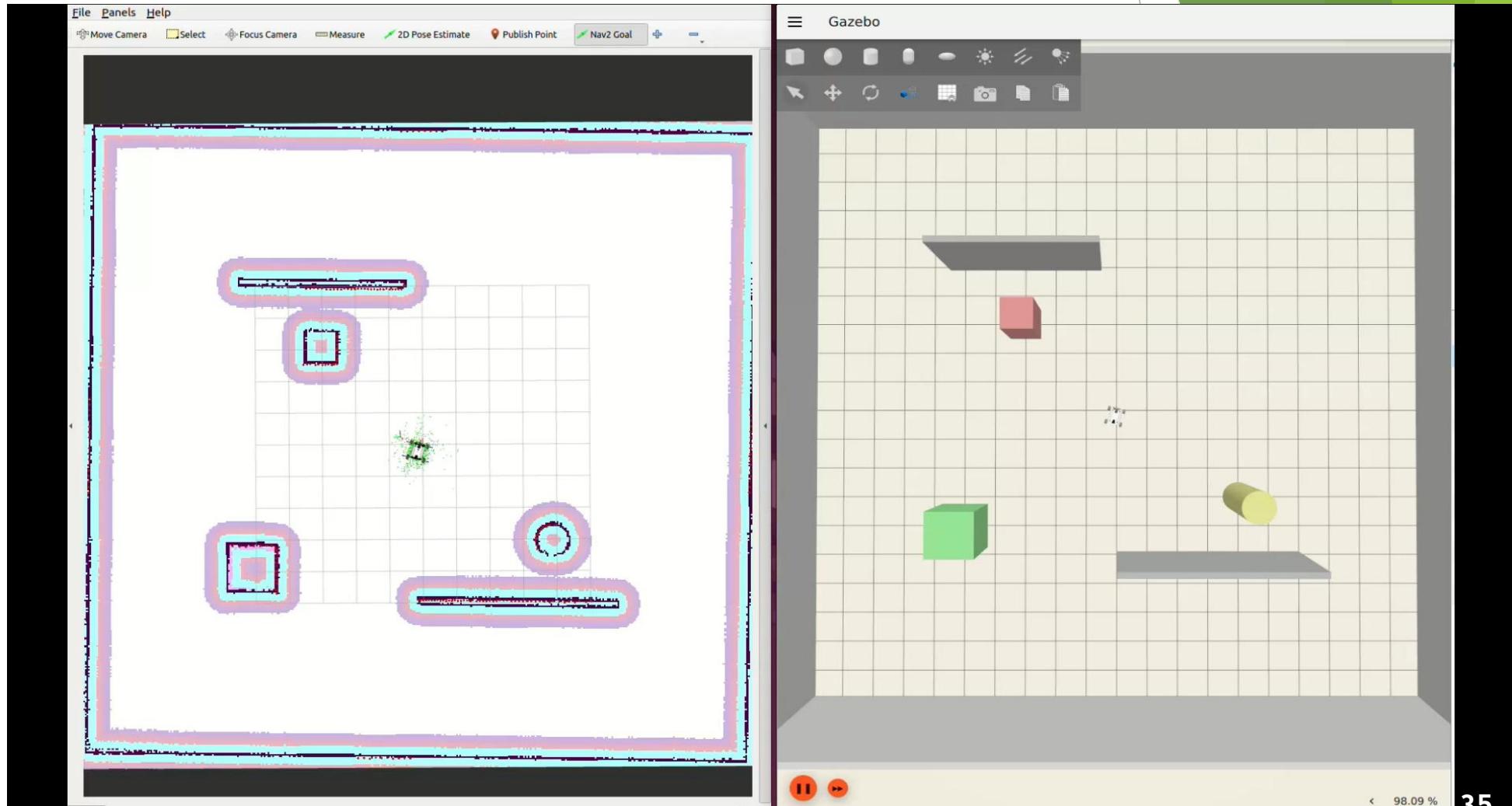
What it looks like

➤ SLAM

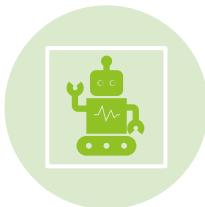


What it looks like

➤ Nav2



Challenges in the simulation



Challenge: Disparity between the Ground Truth (Gazebo) and the Robot's Perception (RViz).



Solution: Calibrated wheel radius parameter to sync physical movement with digital transforms (TF).



Challenge: Navigation in unmapped environments (no static global map).



Solution: integrated a custom Nav2 Action client with a dynamic goal generator.



Challenge: Latched Velocity Persistence (The rover won't stop).

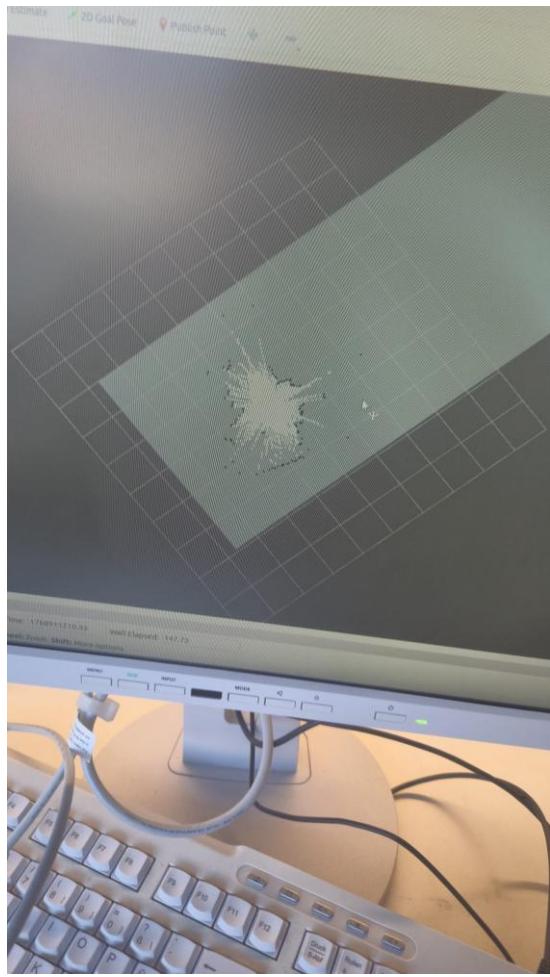


Solution: Implemented POSIX Signal Handlers and a velocity flush routine.

Integration and Final Results

Integration

- ▶ Launch all nodes together: Scout, LIDAR, SLAM
- ▶ For that there is a launch file: master_launch.py



References

- ▶ How this helicopter survived 1004 days on Mars, then disappeared..., Veritasium, available at <https://www.youtube.com/watch?v=20vUNgRdB4o>
- ▶ ROS2 Packages for Scout Mobile Robot, available at https://github.com/westonrobot/scout_ros2
- ▶ ROS2 Packages for RS Lidar, available at https://github.com/RoboSense-LiDAR/rslidar_sdk/tree/main
- ▶ Structure of the simulation environment for scout_mini available at https://github.com/agilexrobotics/ugv_gazebo_sim/tree/master/scout
- ▶ SLAM implementation in the simulation avialable at <https://youtu.be/ZaiA3hWaRzE>