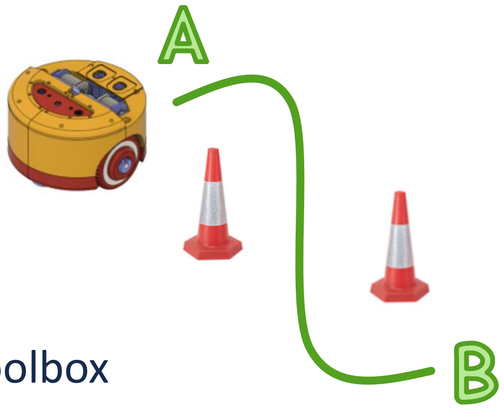
Autonomous Robot Navigation

HP Tuners 2023
Personal Project

Project Goals / Objectives



Goals

- Navigate Robot Autonomously
  - Learn ROS2 (Robot Operating System) fundamentals
  - Learn NAV2 navigation stack features
  - Learn SLAM (Simultaneous Localization and Mapping) toolbox
  - Learn how to develop / test robot in Gazebo simulation environment

Measurable Objectives / Success Criteria

- Develop robot control / being able to control and drive robot from distance
- Describe physical characteristics of the robot / Create robot description file in URDF (Unified Robot Description Format)
- Generate maps of the operational environment.
- Develop navigation system / Navigate robot autonomously in simulation world

Stretch goals:

- Navigate robot autonomously in real world
- Detect dynamic obstacles and update navigation path in real-time

System Overview - Navigation Features

- Localization

  Estimates location of the robot

- Mapping

  Global Map is used to create paths for a goal in the map or at a far-off distance

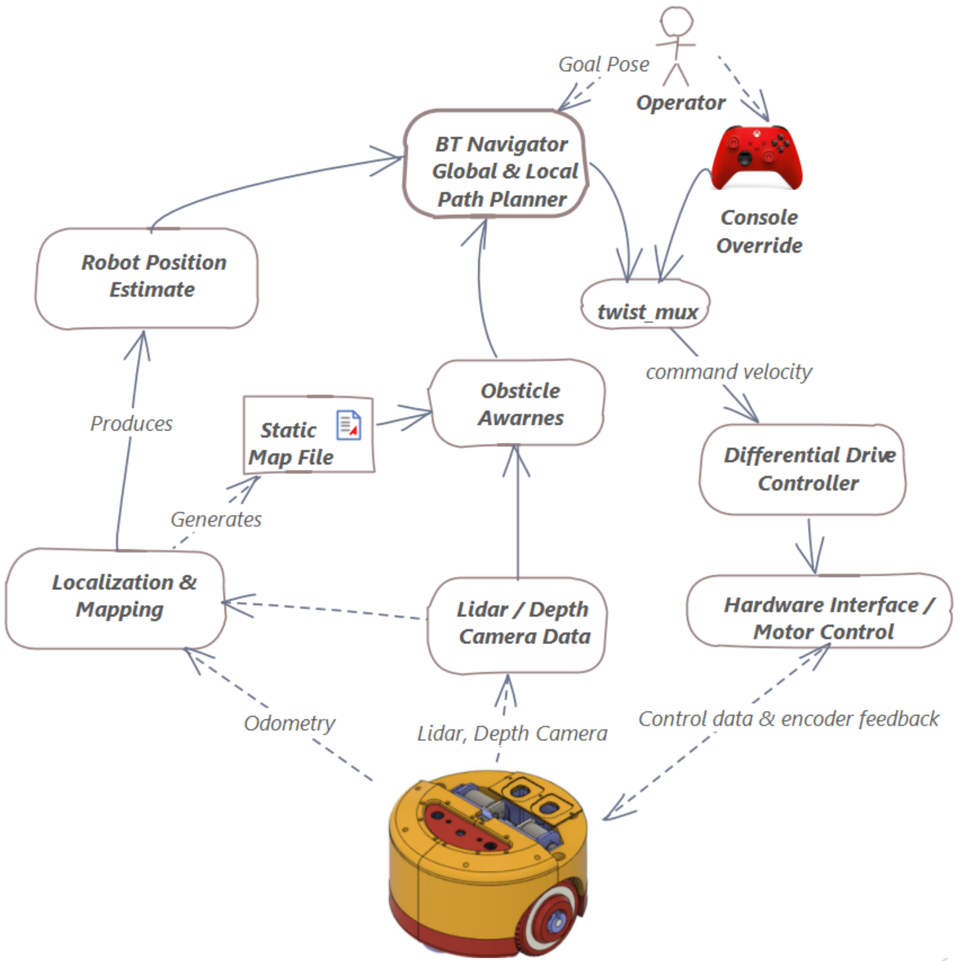
  Local Map is used to create paths in the nearby distances and avoid obstacles, for example, a square window of 2 x 2 meters around the robot

- Navigation

  Implements navigation tasks and complex robot behaviors, including recovery to get the robot out of a bad situation

- Path Planning

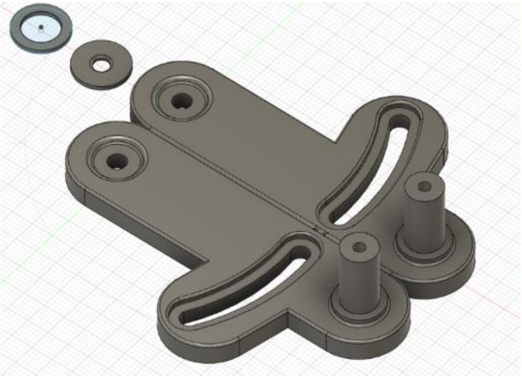
  Provides the motion plan



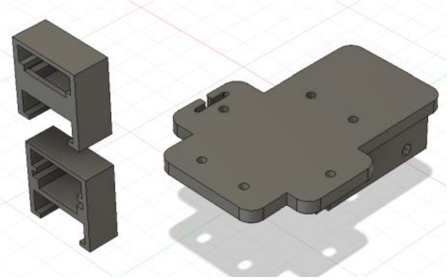
Preparation Work
Getting Ready For Development

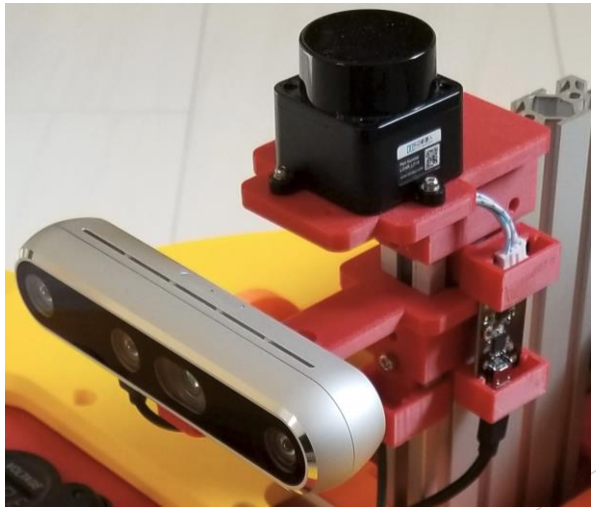
Mechanical / 3D Printing / Fusion 360 Works

- Drive Belt Tensioners to reduce backlash

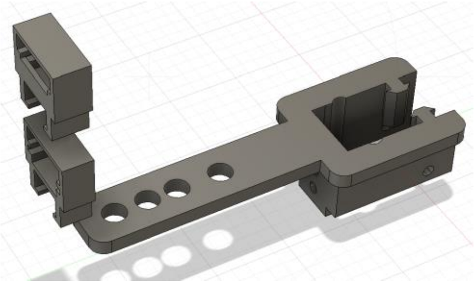


- Lidar Holder

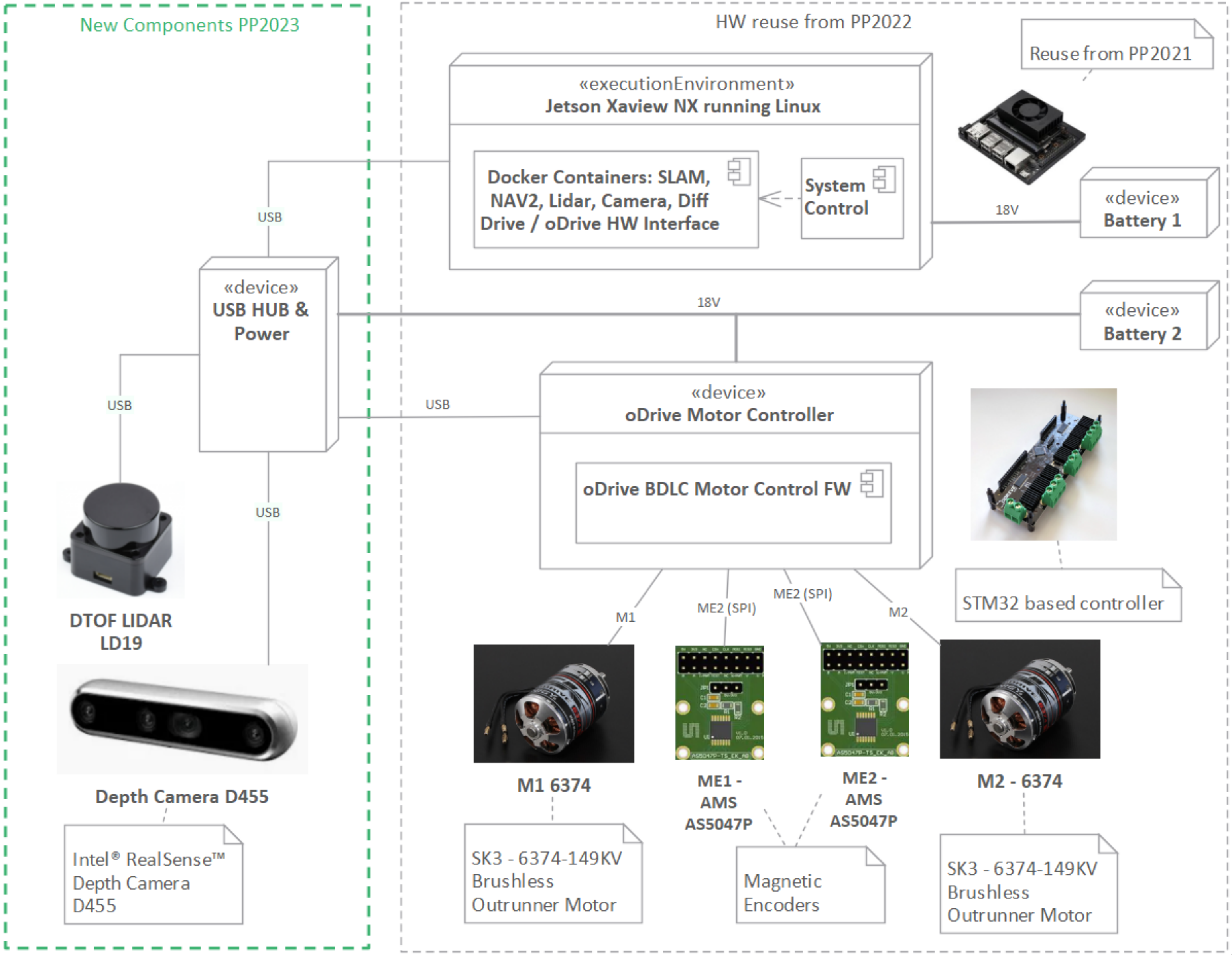




- Camera Holder

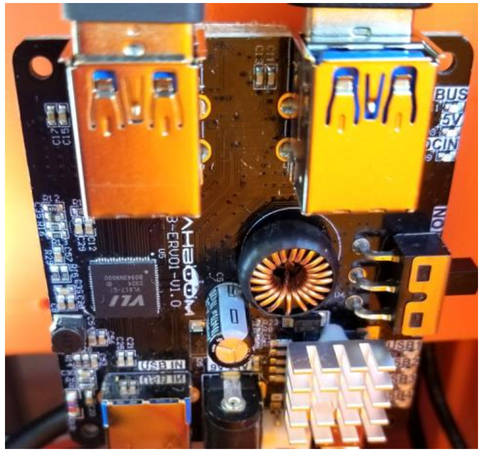


Hardware

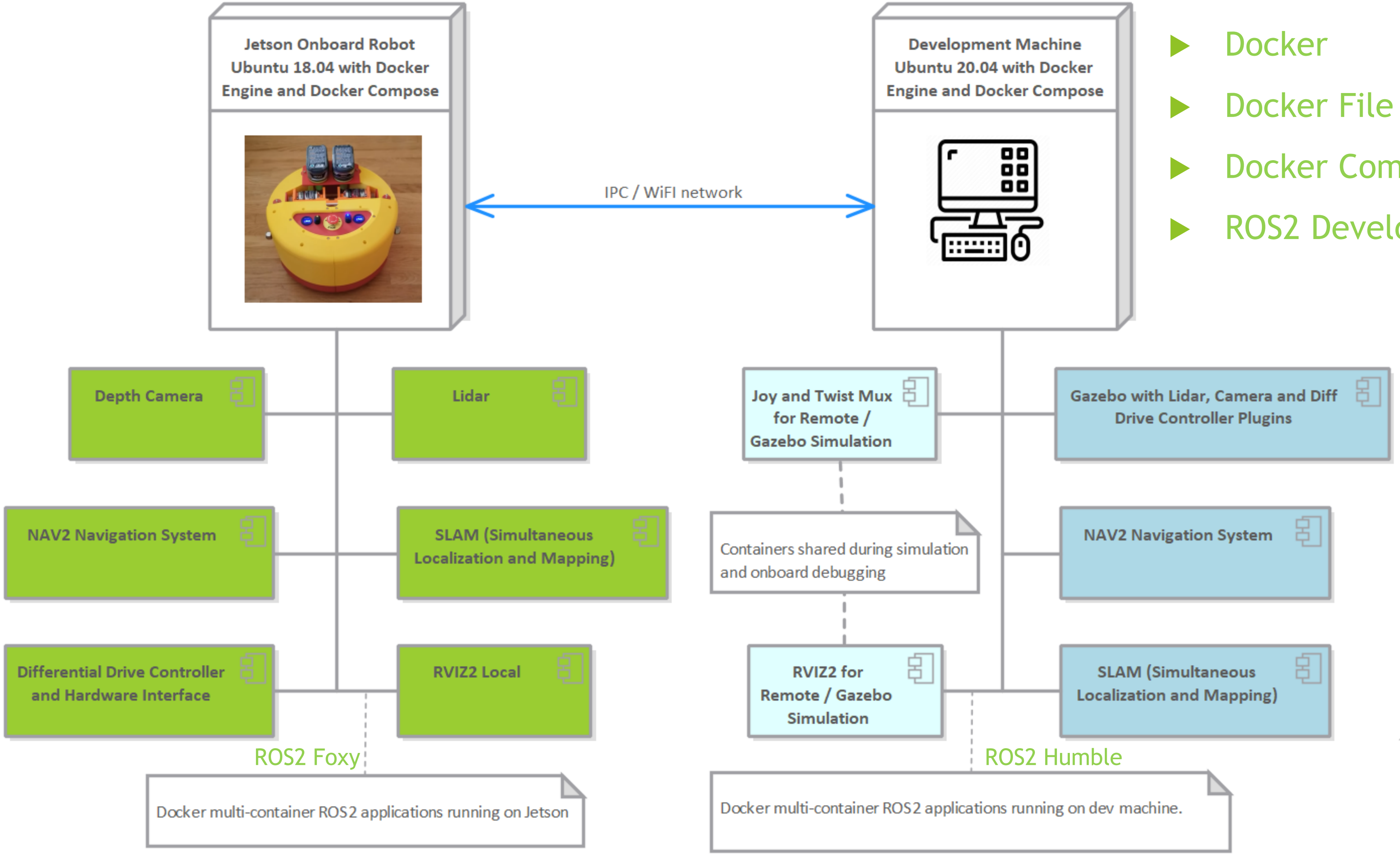

New Components PP2023
HW reuse from PP2022
Reuse from PP2021
«executionEnvironment»
Jetson Xaview NX running Linux
Docker Containers: SLAM, NAV2, Lidar, Camera, Diff Drive / oDrive HW Interface
System Control
18V
«device»
Battery 1
USB
«device»
USB HUB & Power
18V
«device»
Battery 2
USB
USB
«device»
oDrive Motor Controller
oDrive BDLC Motor Control FW
USB
DTOF LIDAR
LD19
STM32 based controller
M1
ME2 (SPI)
ME2 (SPI)
M2
Depth Camera D455
M1 6374
ME1 -
AMS
AS5047P
ME2 -
AMS
AS5047P
M2 - 6374
Intel® RealSense™
Depth Camera
D455
SK3 - 6374-149KV
Brushless
Outrunner Motor
Magnetic
Encoders
SK3 - 6374-149KV
Brushless
Outrunner Motor

Hardware / Electrical / Sensor Additions

- Lidar Sensor
- Depth Camera
- HDMI & USB Port
- USB / Power Expansion



Software



- Docker
- Docker File
- Docker Compose
- ROS2 Development

Getting Ready for Gazebo Simulation and Navigation

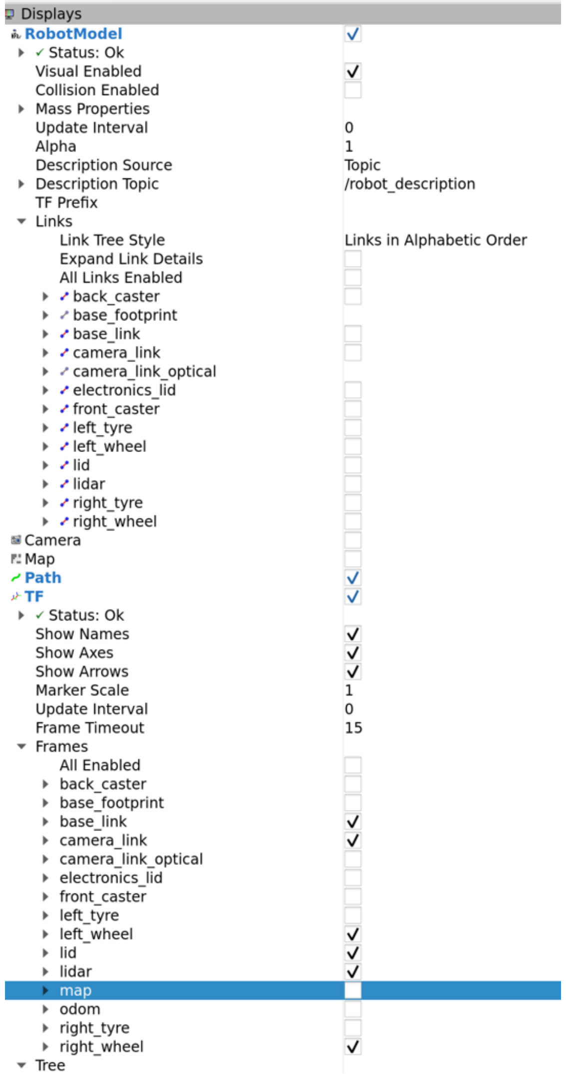
Robot Description

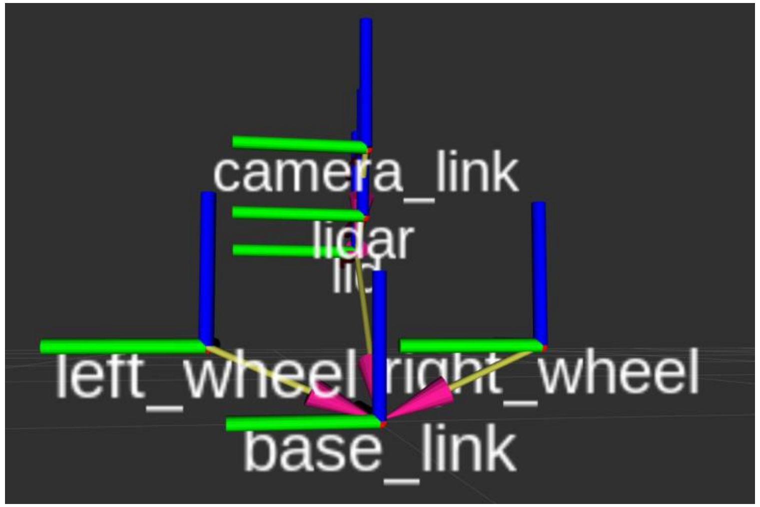
- Robot Description in URDF (Unified Robot Description Format)
  - URDF describes a robot as a tree of Links, that are connected by Joints
  - Links represent the physical components / characteristics of the robot
  - Joints represent how one link moves relative to another link
- URDF file is passed to Robot State Publisher node which makes the data available on the /robot_description topic
- Robot State Publisher also broadcasts joint transforms /tf
- ROS nodes can subscribe to /robot_description and /tf topic data to monitor robot state

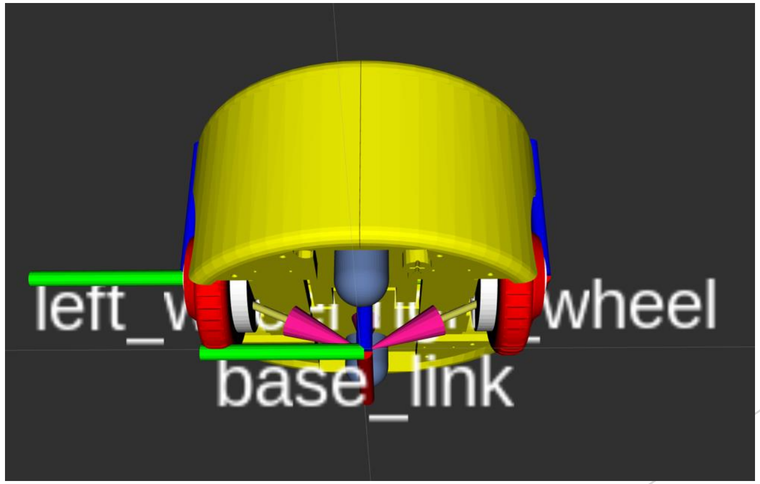
```xml
<joint name="base_footprint_joint" type="fixed">
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <parent link="base_link"/>
    <child link="base_footprint"/>
</joint>

<link name="base_footprint"></link>

<link name="base_link">
  <inertial>
    <origin/>
    <mass/>
    <inertia/>
  </inertial>
  <visual>
    <origin/>
    <geometry></geometry>
    <material/>
  </visual>
  <collision>
    <origin/>
    <geometry></geometry>
  </collision>
</link>
```

```xml
<joint name="motor_right" type="continuous">
  <origin rpy="0 0 0" xyz="0.0 -0.227 0.099839"/>
  <parent link="base_link"/>
  <child link="right_wheel"/>
  <axis xyz="0.0 1.0 0.0"/>
</joint>

<joint name="motor_left" type="continuous">
  <origin rpy="0 0 0" xyz="0.0 0.227 0.099839"/>
  <parent link="base_link"/>
  <child link="left_wheel"/>
  <axis xyz="0.0 1.0 0.0"/>
</joint>
```

Robot Description RViz

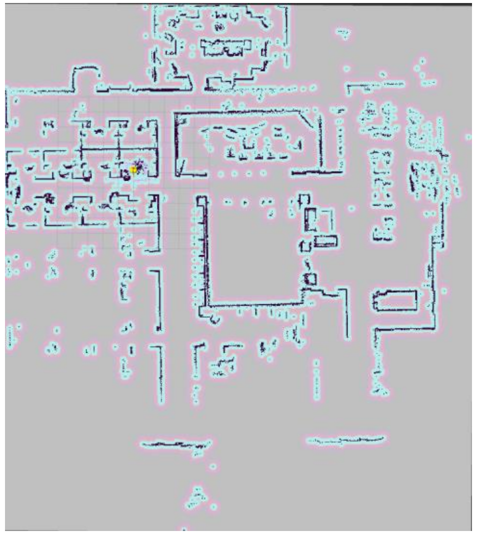
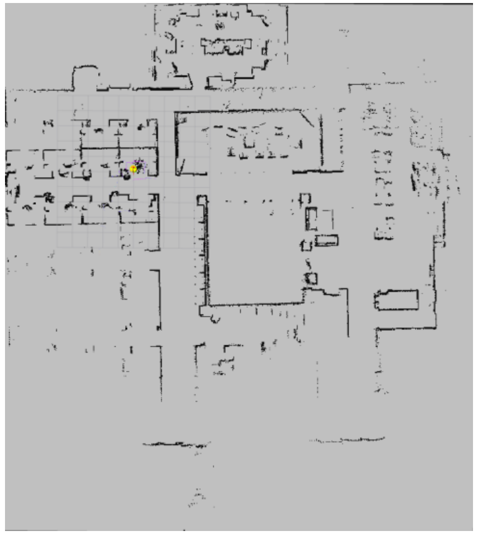
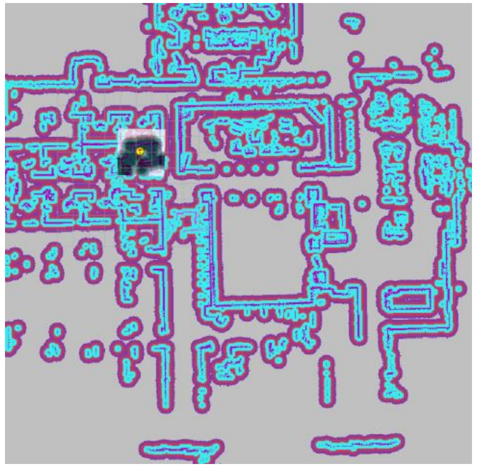
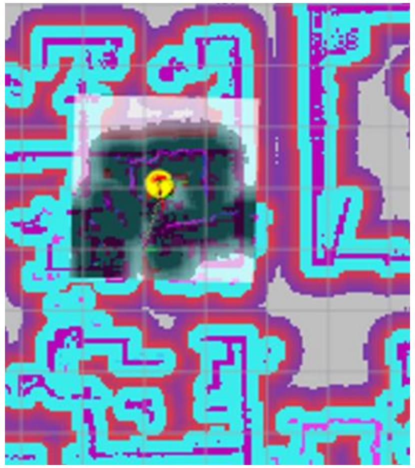




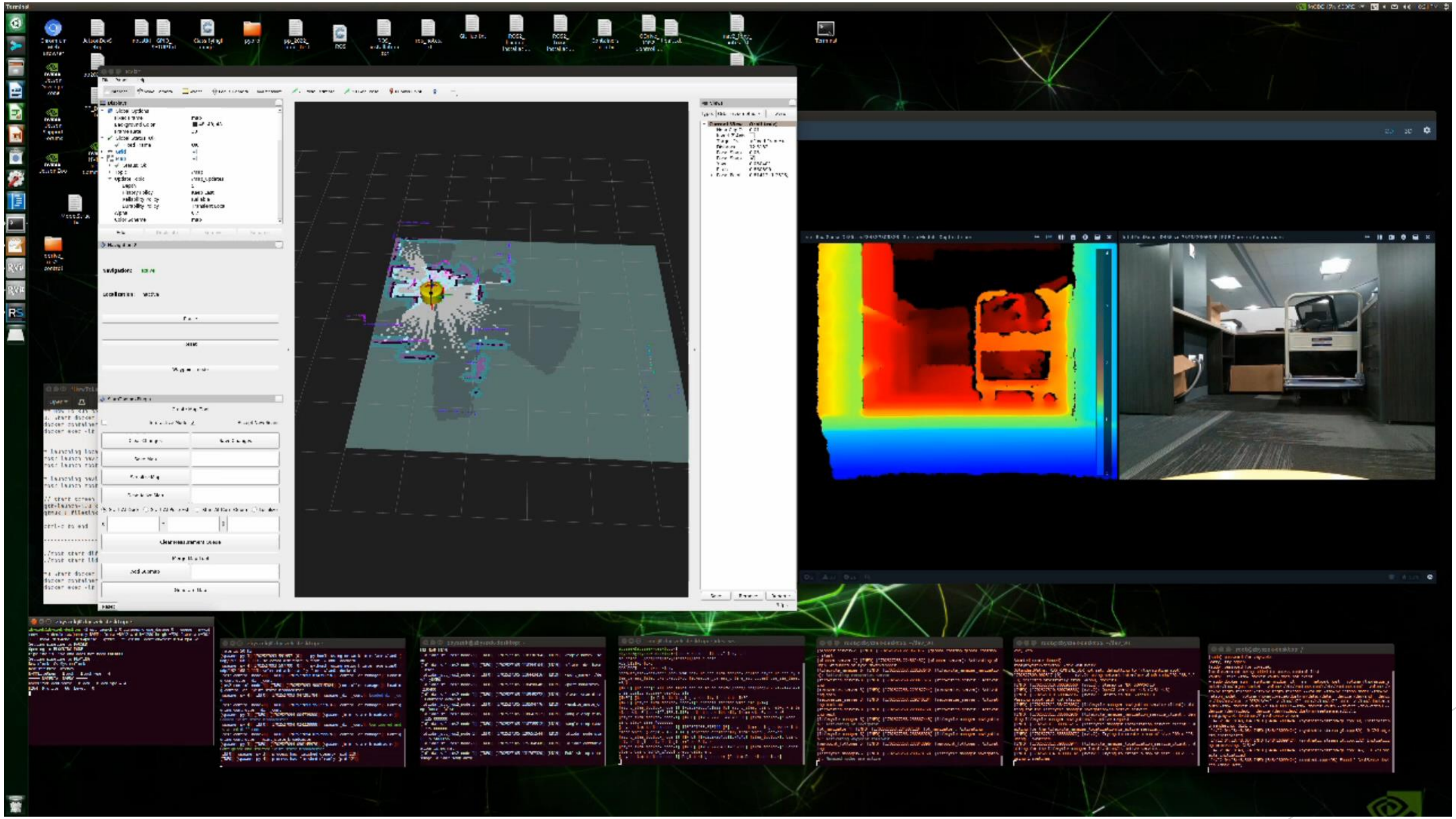


Mapping

- Global Map / Global Cost Map
  - Global map is used to create paths for a goal in the map or at a far-off distance
  - Global map is static and stored map file
- Local Cost Map
  - Local map is used to create paths in the nearby distances and avoid obstacles, for example, a square window of 3 x 3 meters around the robot
  - Dynamically updated while robot navigation is active

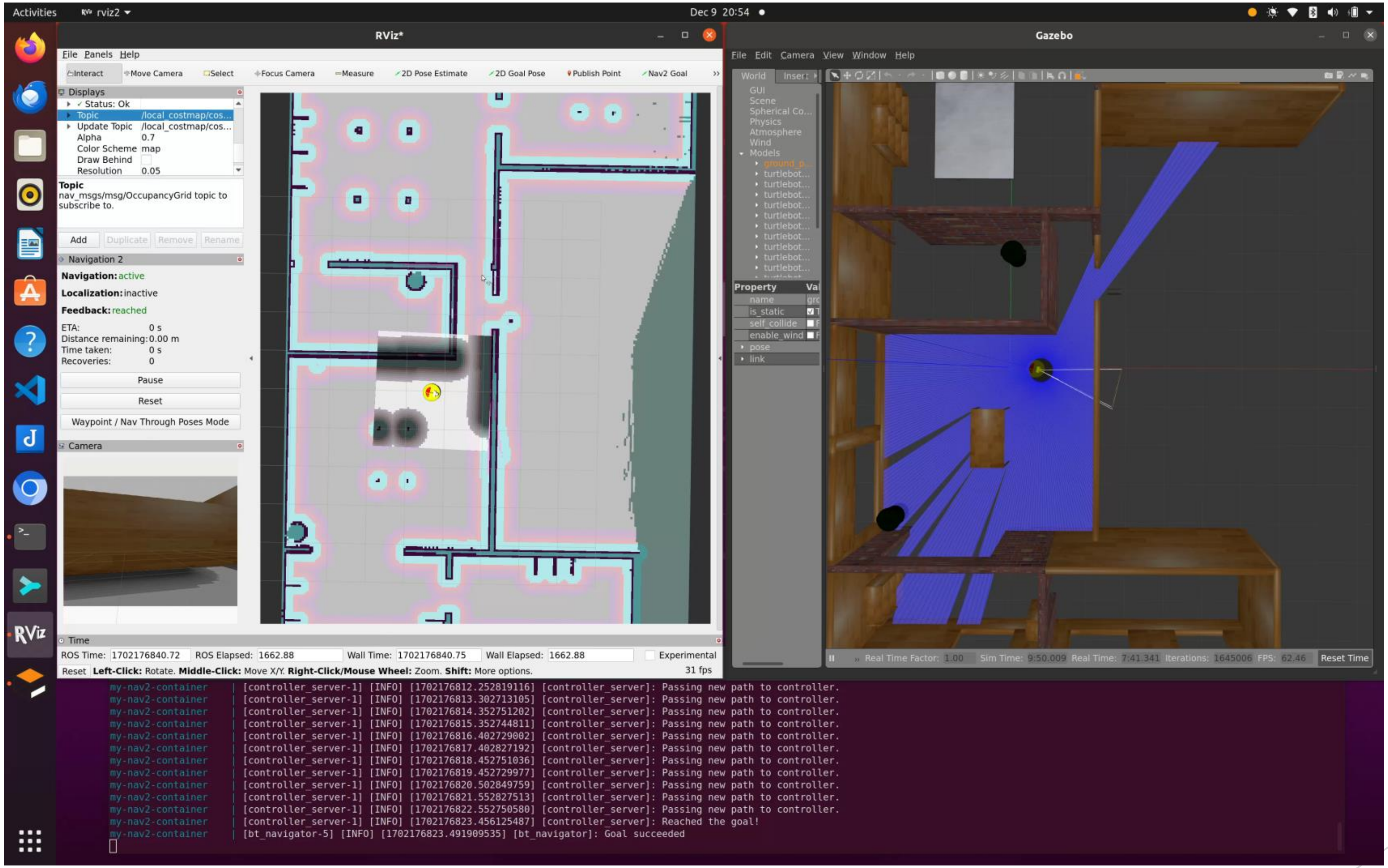





Mapping HPT Office – Creating Global Map

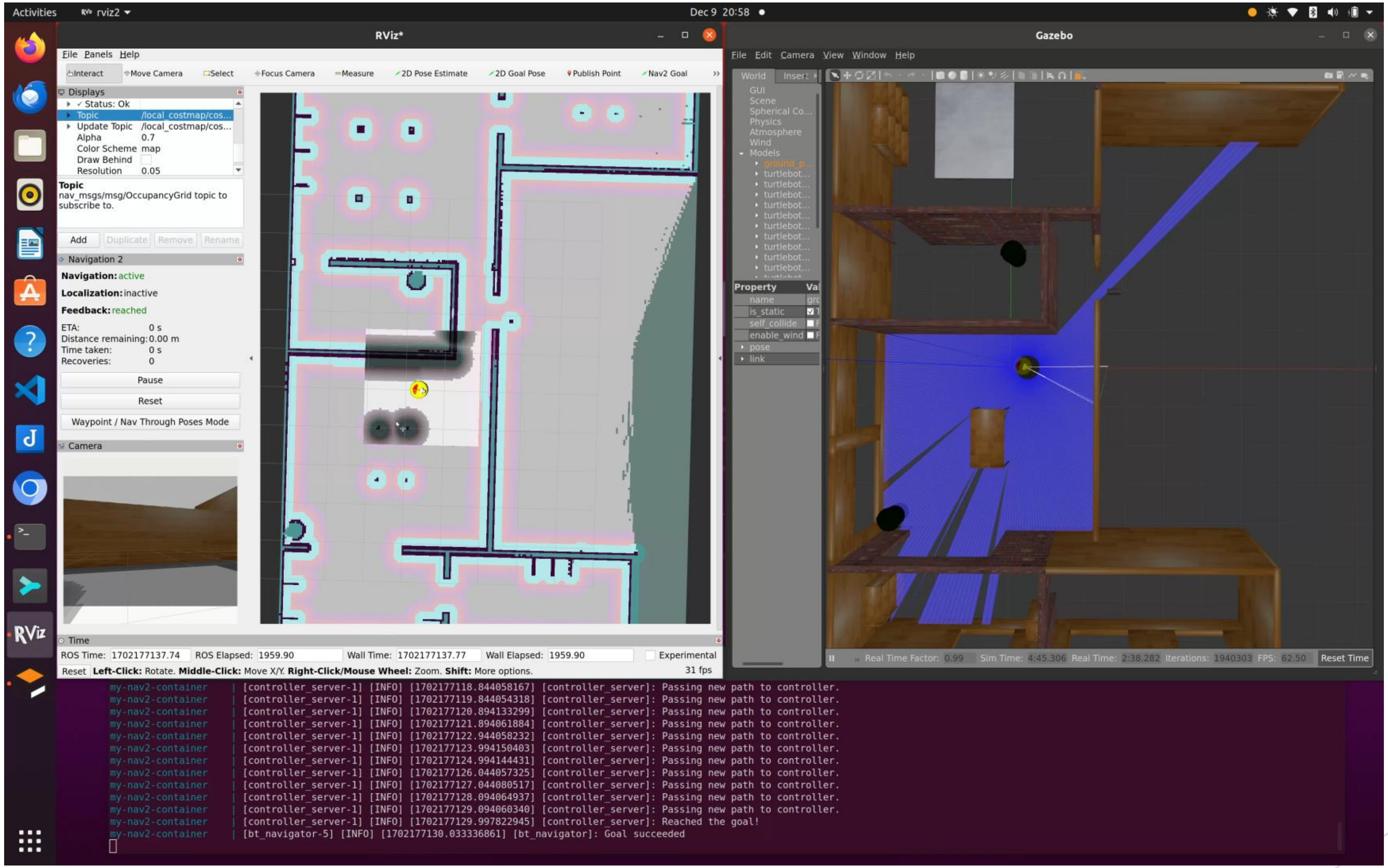


Navigation

Navigating in Gazebo - Screen Capture 1



Navigating in Gazebo Screen Capture 2



Navigating in HPT Office

Lessons Learned

- Learned how to develop robot control and navigation using ROS (Robot Operating System) libraries and tools
- Navigation concepts: mapping, localization, obstacle avoidance and path planning
- How to develop ROS nodes and packages
- How to develop multi-container Docker / ROS applications
- How to use Docker Compose to manage Docker containers

Budget Review Against Plan

Item	Plan	Actual
Lidar	100..300	70
Depth Camera	150..400	435
Display screen, Joystick	50..100	45
USB HUB, Power Expansion		**30**
Optical Encoder		**47**
Total	300..500+	500+

What would I do differently?

- Reduce scope of the project to have more time for research and development
- Select different / better lidar sensor

Q&A

Thank You

References and Links

ROSCon talks
The following ROSCon talks have been given on ROS 2 and provide information about the workings of ROS 2 and various demos. List of ROS video tutorials.

ROS Docker Tutorial

VSCode, Docker, and ROS2
by Allison Thackston
This guide covers ROS development environment setup with Docker and it is intended to be used by people that are familiar with coding and software development, but maybe not ROS.

ROS and Ubuntu on Windows (WSL2) – Video
This tutorial is based on Allison's tutorial above.
A walkthrough of using Windows, WSL2, Docker, VS Code, and an X Server to run ROS on Ubuntu

A Guide to Docker and ROS
by Sebastian Castro
Covers Docker setup for managing ROS based software projects. Shows Docker ability to create multi-stage builds, where one image stacks on top of the other. Nice step by step tutorial.

F1TENTH T01 - Tutorial on Docker and ROS2
ROS tutorial - general concepts

Making a Mobile Robot with ROS - Full list of tutorials
Excellent step by step tutorials. Cover building RPI based robot by Josh Newans.

Really Useful Robot by James Bruton

Source code and documentation - GitHub