MYBOTSHOP

MOBILE ROBOTICS

Ridgeback

Mobile Manipulator — User Manual

CONTACT

info@mybotshop.de

support@mybotshop.de

mybotshop.de

Contents

1 Attention

Important: Carefully review this document. If there are uncertainties, it is crucial to seek guidance from specialists, experts, or the manufacturers of the assemblies employed. The robot should not be activated until clarification is obtained. If any doubts persist, refer to the provided guides or get in touch with a specialist, expert, or the manufacturer of the assemblies used. Exercise caution and do not permit individuals unfamiliar with robotics or these instructions to operate the robot — untrained users pose a risk when handling robots.

Warning: The robot supplied by MYBOTSHOP GmbH is intended for research and development purposes and does not bear the CE Marking and/or Certificate of Incorporation. A fundamental understanding of ROS / ROS 2 (Robot Operating System) is necessary — if you lack familiarity, we recommend consulting the ROS documentation as a first step. The robot is categorized as a partially completed machine according to the Machinery Directive 2006/42/EC and does not carry a CE marking.

2 Disclaimer

The Ridgeback Mobile Manipulator (MMP) documented here is designed, integrated and supported by *MYBOTSHOP GmbH* around a Clearpath Robotics Ridgeback (R100) omnidirectional base. This manual covers all five customer configurations built on that shared base — MMP DAR, NÜ, DRE, MPI and BI — each with a different robotic arm and sensor loadout, detailed in their own chapters.

The client acknowledges and agrees that any information or materials provided by MYBOTSHOP GmbH are exclusively for research and development purposes. All services are provided on an “AS IS” basis without any representation or warranty, whether express or implied, including but not limited to merchantability, fitness for a particular purpose, non-infringement, or any other warranty. MYBOTSHOP GmbH shall not be held liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising from or in connection with the use or inability to use the provided information or materials. This limitation on liability applies irrespective of the form of action, whether in contract, tort, or otherwise.

MMP DAR is the current, most actively developed configuration (ROS 2 Jazzy) and receives the deepest treatment in this manual. MMP NÜ, DRE, MPI and BI remain in service; some of their documentation predates ROS 2 migration and is preserved here as originally written, including ROS 1-style commands (`roslaunch`, `rosrun`, `rostopic`) where that is what the deployed unit actually runs.

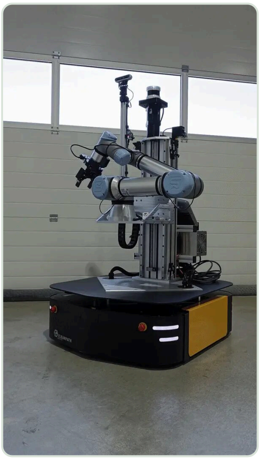
3 About the Ridgeback Mobile Manipulator

Mobile Manipulation Platform Ridgeback

Note: This package supplies Sphinx-based tutorial content to assist you with setting up and using MBS mobile manipulation platform. The tutorials topics are listed in the left column, and presented in the suggested reading order.

MMP DAR	MMP NÜ	MMP DRE
MMP MPI	MMP BI	

Attention: These tutorials assume that you are comfortable working with ROS. We recommend starting with ROS tutorial if you are not familiar with ROS already.



_images/mmp_ru_front.webp

The Mobile Manipulation Platform (MMP) Ridgeback in action, demonstrating mobility and dual-arm coordination.

Clearpath Ridgeback

The Clearpath Ridgeback is a highly adaptable, mid-sized robotic platform, engineered to meet the demanding requirements of both research and development as well as industrial environments. Its low-profile design makes it ideal for navigating tight spaces, while its robust build ensures stability and durability. The Ridgeback's versatility stems from its advanced features and extensive customization options, making it a go-to solution for a wide array of applications.

Key Features:

- Omni-Directional Movement: The Ridgeback is equipped with advanced omni-directional wheels, allowing it to move seamlessly in any direction. This capability provides unparalleled maneuverability, making it perfect for dynamic and cluttered environments.

- Precise Control: With its high-precision control systems, the Ridgeback can execute complex movements and tasks with exceptional accuracy. This precision is crucial for applications requiring delicate handling and exact positioning.
- 360-Degree Field of View: The platform features a comprehensive 360-degree field of view, ensuring that it can perceive and navigate its surroundings effectively. This all-encompassing vision is essential for obstacle avoidance and environment mapping.

Customization and Components:

The Ridgeback's current configuration is tailored to enhance its functionality and adaptability, featuring the following components:

1. Movable Lift with Robotic Arm: A customizable, movable lift system is integrated into the Ridgeback, providing variable height adjustment to the attached robotic arm. This setup allows for a wide range of manipulation tasks, from picking and placing to complex assembly operations.
2. Ouster 3D Scanner: Mounted on top of the lift, the Ouster 3D scanner delivers a 360-degree, high-resolution scan of the environment. This sensor is vital for creating detailed 3D maps, detecting obstacles, and navigating complex spaces with precision.
3. Hokuyo Lidar Sensors: The Ridgeback is equipped with two Hokuyo lidar sensors, one positioned at the front and the other at the back. Each sensor provides a 360-degree coverage, offering comprehensive environmental sensing. This dual lidar setup enhances the robot's ability to detect obstacles, ensuring safe and efficient operation.
4. Intel RealSense D455 and D405:

The Ridgeback is outfitted with Intel RealSense D455 and D405 cameras, delivering advanced depth sensing capabilities. The D455 offers a longer range and higher accuracy for wide-area scanning, while the D405 provides precise short-range depth data. This combination enables robust 3D perception for various applications, from navigation to object recognition.

_images/D455.webp

_images/D405.webp

1. Control Joystick — Steamdeck Controller:

For intuitive manual control, the Ridgeback includes a Steamdeck controller. This versatile joystick allows operators to effortlessly maneuver the robot and manage its functions, offering a user-friendly interface for real-time adjustments and remote operation.

_images/ridgeback_backward.gif

In summary, the Clearpath Ridgeback stands out as a versatile and powerful robotic platform, offering advanced features and extensive customization options. Its ability to move omni-directionally, coupled with precise control and comprehensive environmental sensing, makes it an invaluable asset in both research and industrial applications. Whether navigating a cluttered laboratory or streamlining a production line, the Ridgeback is designed to meet the challenges of modern robotics with reliability and precision.

Robot Upgrade

Manipulators

In terms of manipulators that can be installed and integrated with your mobile platform, several recommended arms are available. Each has its own perks and limitations. The table below illustrates some of the key features of the recommended arms that can be installed with the mobile platform.

Comparative Analysis

	xARM6	PANDA	UR10E
Company	UFACTORY	FRANKA EMIKA	UNIVERSAL ROBOTS
Operation	Host PC	Host PC	Host PC
			Teach Pendant
Software Development	Plug & play	Config. Req.	Config. Req.
I/O Ports	Multi	None	Multi
Sensor Integration	Available	None	Available
ROS Control	Immediate	Config. Req.	Config. Req.
Pricing	Low	High	High

Mobile Platforms

For mobile platform, several recommended mobile platforms are available. A new and interesting mobile platform that can be combined with your robot arm is the ROVO2 . The table below illustrates some of the key features of the recommended mobile platforms that can be utilized.

Comparative Analysis

	ROVO2	RIDGEBACK	HUSKY	WARTHOG
Robot Weight	290 kg	135 kg	50 kg	280 kg
Payload	500 kg	100 kg	75 kg	272 kg
Operation time	8 hrs	8 hrs	8 hrs	2.5 hrs
Top Speed	20 kmhr	3.96 kmhr	3.6 kmhr	18 kmhr
Terrain Adaptability	Extreme	None	Good	Very good
Pricing	Medium	Medium	Low	High

4 Quick Start – Power On & Controllers

Powering on the Ridgeback

Power Button	E-reset Button	Release Switch

- Ensure the emergency buttons on all 4 corners are released. ◦ To check if the emergency switch is released, twist the switch in the indicated direction on it. If it pops up, then the switch has successfully been released.
- Press the power button on the rear of the ridgeback. It has the power symbol.
- Once the ridgeback is powered on and all the switches have been released, press the e-reset button . It is on the rear where the ridgeback has a yellow flashing light.

Powering on the UR5e Arm

_images/ur5e_hmi.webp

UR5e should be in, i.e. it is in remote mode and on power off state

- The UR5e should only be powered on when the ridgeback is fully powered on (i.e. the ridgeback has white lights infront and/or the yellow light is flashing for the e-reset).
- The UR5e can be powered on via the teach pendants power button.
- It may take upto a few minutes for the UR5e to startup.
- Ensure all the emergency switches are released (i.e. on the teach pendant as well as on the lift).

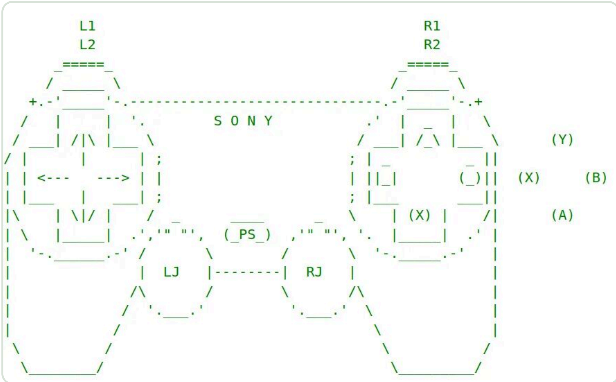
Teach pendant Switch Released	Custom Emergency Switch Released

> **Warning:** Do not start anything else in the UR5e as this is all done automatically via ROS. Powering on is only required.

Powering Aux. Components

On startup the ridgeback launches, the lift drivers, and the Hokuyos. The auxiliary components such as the ouster, dynamixels, realsense cameras, nvidia, etc. are not powered on startup automatically but can be added via the startup launch file if required.

PS4 Controller



_images/ps4_controller.webp

PS4 Controller

1. PS button is to power on.
2. X + Up arrow is for moving Lift up. (X is the square icon)
3. X + Down arrow is for moving Lift down. (X is the square icon)
4. B button for releasing Lift soft locks. (B is the circle icon)
5. L1 is the dead man's switch.
6. LJ is to move in x and y direction.
7. RJ is to move in the yaw orientation.

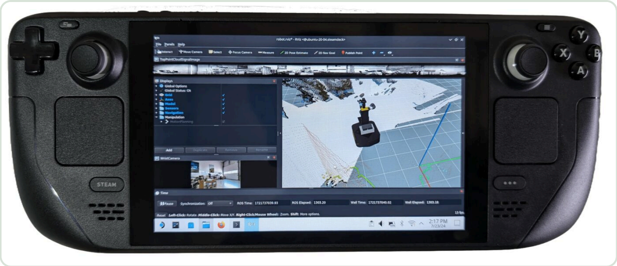
Steamdeck Controller

> **Warning:** The Steamdeck usage is in experimental phase. Loss of connection or (laggy connection) may lead to the robot moving and/or not stopping immediately. Caution should be taken when using the Steamdeck for the operation of the robot!

Steamdeck - Basic Controls

1. Power button is on the top of the Steamdeck with the power symbol.
2. R2 for mouse left-click
3. L2 for mouse right-click
4. Left trackpad for mouse scrolling
5. Right trackpad for mouse movement
6. L2+R2 for middle mouse button click
7. The 3 dashes button is located on the top right of the Steamdeck next to the Y button, Hold for 3 seconds to gamepadmode and hold for 3 seconds again to go to desktop mode(Switching to

desktop mode activates the cursor for the trackpad as well).



_images/steamdeck.webp

Steamdeck Controller

Steamdeck - Initialization with Robot

Before usage:

1. Ensure the Steamdeck is connected to the robot's WiFi access point.
2. Ensure the robot is powered on and operational.
3. Single tap the Joystick Launcher on the screen to activate the controller for the robot. (Do not open multiple instances of the joystick launchers, as this can cause zombie nodes).
4. Single tap the Rviz2 Launcher on the screen to visualize the robot.

Steamdeck - Robot Controls

1. L1 is the dead man's switch for slow speed.
2. R1 is the dead man's switch for high speed.
3. LJ is to move in x and y direction.
4. RJ is to move in yaw direction.
5. X+Up Arrow is to move lift up.
6. X+Down Arrow is to move lift down.
7. B is to release soft switches if the lift reaches maximum height.

Steamdeck - Rviz2 Controls

1. L2+R2 should be held when wanting to move the orbit in the xy plane.
2. R2 should be held to orbit around the orbits origin
3. Left trackpad can be moved up and down to zoom in and zoom out.

Manipulator Activation

Once the UR5e is powered on and its emergency switches are released.

- Then check if the UR5e is in remote mode (should be in remote mode by default) — Located in Teach pendant Top right .
- Then check if the UR5e is in Power off state with a icon — Located in Teach pendant Bottom left (Important) .

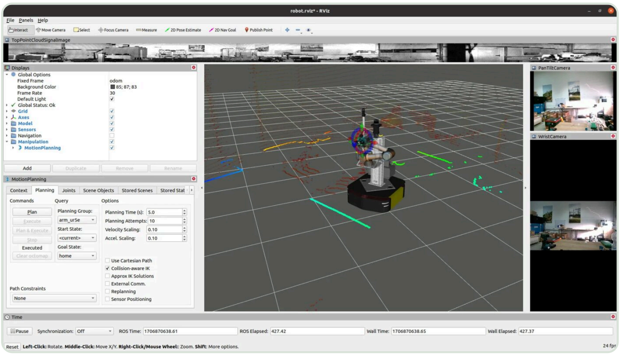
Auxiliary ROS Drivers

Once the manipulator is activated, next step is to launch the following driver:

```
roslaunch mmp_startup accessories.launch
```

This should turn on all the components including the UR5e, Robotiq 2F-85, Ouster, Realsense Cameras, and pan & tilt.

Digital Twin (RVIZ)



_images/mmp_mip_rviz.webp

Visualizing the MMP in Rviz

To visualize the mmp, you can run:

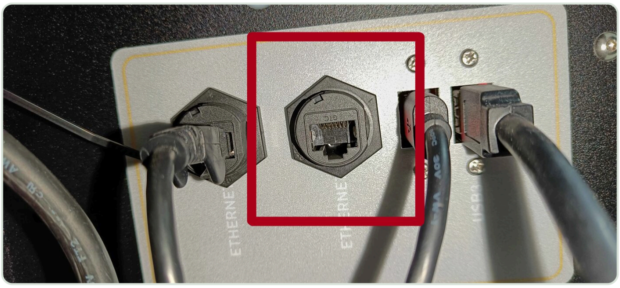
```
roslaunch mmp_viz view_robot.launch
```

Ideally, for latency-free streaming, please utilize ROS multimachine setup.

5 Network Setup

Robot Interface

Note: Instructions for interfacing with the robot using Ubuntu 20.04 and ROS Noetic .



_images/robot_interface.webp

The Ethernet port at the back of the Husky can be utilized for setting up communication via LAN.

Static IP Configuration

For the first time, one needs to connect through a LAN cable to configure the robot's WLAN network. To create a static connection in your PC (not the robots), in Ubuntu go to Settings → Network then click on + and create a new connection.

1. Change the connection to Manual in the IPv4 settings.
2. Set the Address IP as 192.168.131.51 and the Netmask as 24 .
3. Click save and restart your network.

After a successful connection, check the host's local IP by typing in the Host PC's terminal:

```
ifconfig
```

Now, ping the robot:

```
ping 192.168.131.1
```

Access the robot via SSH:

```
ssh -X administrator@192.168.131.1
```

The default password is:

```
mybotshop
```

Note: Sometimes other networks can cause disruptions when connecting to the MMP. It is best to have only your connection to the robot active and all others inactive.

Network Table

All the devices on the robot are network enabled. Table [robot_network] outlines all the devices and their network addresses.

Device	Network Address	Password
Ridgeback	192.168.131.1	mybotshop
Ridgeback MCU	192.168.131.2	
Nvidia IP	192.168.131.4	mybotshop
UR5e IP	192.168.131.5	
Hokuyo Front	192.168.131.20	
Hokuyo Rear	192.168.131.21	
Steamdeck	192.168.131.150	
Ouster	192.168.131.22	
Router	192.168.131.100	

Screen Connection

An alternative for connecting to the robot is by plugging in an HDMI cable as well as a mouse and keyboard. This allows you to connect the robot from your own local network.

6 MMP DAR – Full Configuration Reference

MMP DAR

_images/mmp_r100.png

This specifies the configuration for Mobile Manipulation Platform (MMP) Dual ARM Ridgeback (DAR) version running ROS2 Jazzy . It features dual xARM850 manipulators with Robotiq 2F-140 grippers, a custom 3-axis rig system, Ouster 3D lidar, dual Hokuyo lidars, Intel RealSense and ZED2i cameras.

Important: Carefully review this document. If there are uncertainties, it is crucial to seek guidance from specialists, experts, or the manufacturers of the assemblies employed. The robot should not be activated until clarification is obtained. If any doubts persist, refer to the provided guides or get in touch with a specialist, expert, or the manufacturer of the assemblies used.

Warning: The robot supplied by MYBOTSHOP GmbH is intended for research and development purposes and does not bear the CE Marking and/or Certificate of Incorporation. A fundamental understanding of ROS2 (Robot Operating System 2) is necessary. If you lack familiarity with ROS2, we recommend consulting the ROS2 Documentation as a first step. It is important to note that the mentioned robot is categorized as a partially completed machine according to the Machinery Directive 2006/42/EC and does not carry a CE marking.

Getting Started

For powering on and basic usage of the Ridgeback base, refer to the Clearpath Ridgeback documentation .

_images/ridgeback_clearpath_docs.webp

Data Sheets

- Ridgeback Data Sheet
- xARM850 DataSheet
- Robotiq 2F-140 Datasheet
- Ouster Data Sheets
- Realsense D455 & D405 Datasheet
- ZED2i Datasheet

Network Setup

Device	Network Address	Username	Password
R100 Computer	192.168.131.1	robot	mybotshop
R100 MCU	192.168.131.2	N/A	N/A
Webserver	192.168.131.1:9000	admin	mybotshop
Cockpit	192.168.131.1:9090	robot	mybotshop
Left xARM850	192.168.131.5:18333	N/A	N/A
Right xARM850	192.168.131.6:18333	N/A	N/A
Hokuyo Front	192.168.131.21	N/A	N/A
Hokuyo Rear	192.168.131.22	N/A	N/A
Ouster Lidar	192.168.131.30	N/A	N/A
Router	192.168.131.200	SSID	mybotshop

Quick Start ROS2

Power On

1. Release the emergency stop button by turning it clockwise.
2. Press the power button to turn on the robot.
3. Wait for the robot to boot up completely (approximately 1-2 minutes).
4. The Clearpath and MMP services will start automatically.
5. Verify the webserver is accessible at http://192.168.131.1:9000 .

Power Off

> **Warning:** Always follow the proper shutdown procedure to prevent data corruption and hardware damage.

1. Deactivate the rig system (if active):

```
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: false, left_plate: false, right_plate: false}"
```

1. Disable xARM850 arms via the web interface or stop the driver services:

```
sudo systemctl stop mmp-xarm-driver-left.service
sudo systemctl stop mmp-xarm-driver-right.service
```

1. Engage the emergency stop by pressing it down.
2. Disconnect battery from Arms by clicking the red button.
3. Shutdown the robot via SSH or Cockpit terminal:

```
sudo shutdown now
```

1. Turn off the main power switch on the back of the Ridgeback.

Network Connection

1. Connect to the robot's WiFi network or via Ethernet.
2. The robot IP address is 192.168.131.1 .
3. The WiFi password is mybotshop .
4. Access the webserver at 192.168.131.1:9000 .
5. The webserver username is admin and password is mybotshop .

SSH Connection

```
ssh robot@192.168.131.1
```

Check Service Status

```
sudo systemctl status clearpath-robot.service
sudo systemctl status mmp-webserver.service
```

Teleoperation

Teleoperate using the keyboard teleop package:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/r100_0118/cmd_vel \
  --ros-args -p stamped:=true
```

Note: Replace r100_0118 with your robot's namespace. Reduce speed by pressing z until 0.1 for safe indoor operation.

Visualization

View the robot state in RViz2:

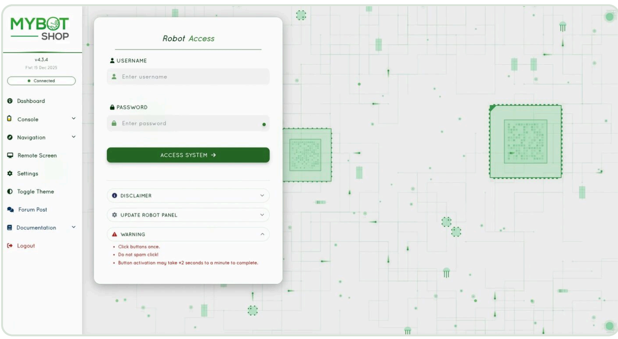
```
ros2 launch mmp_viz view_robot.launch.py
```

Webserver

The webserver module comes pre-installed and should be accessible directly at http://192.168.131.1:9000/ or via the WiFi IP to which the robot is connected.

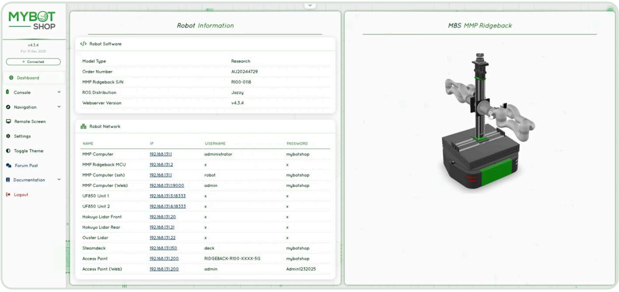
Login

- Username: admin
- Password: mybotshop



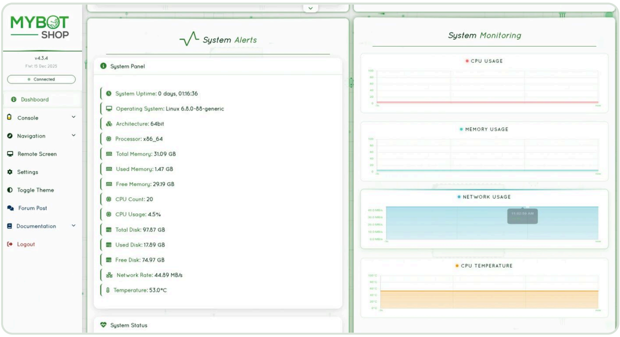
_images/web_login.webp

Dashboard



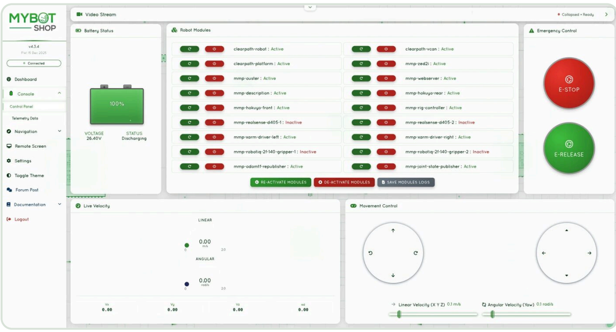
_images/web_dashboard.webp

- View IP Address of the R100
- View System load



_images/web_system.webp

Console

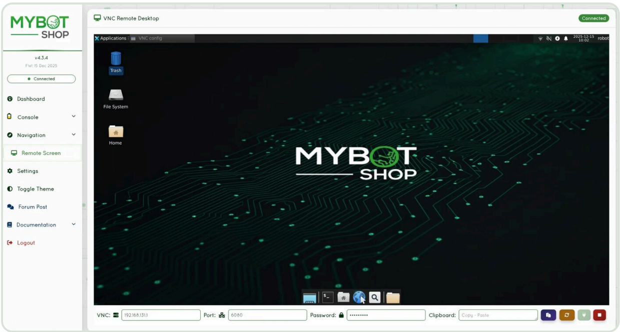


_images/web_console.webp

- Enable/Disable the R100 ROS2 Services
- Record System logs
- Battery status monitoring
- Pre-configured action buttons (adaptable to new ROS2 services)
- Web Joystick for teleoperation
- Online image stream

Remote Desktop (VNC)

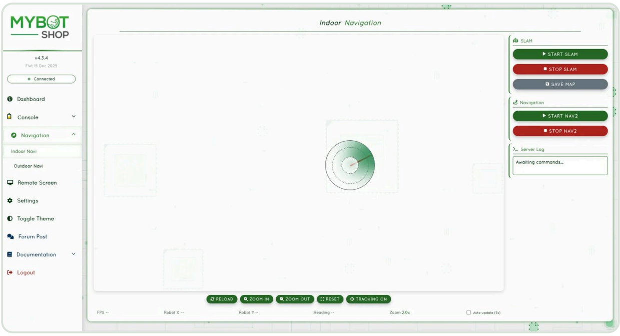
Access on-board screen of the R100’s computer:



_images/web_vnc.webp

Navigation Indoor

Map visualization and navigation control:

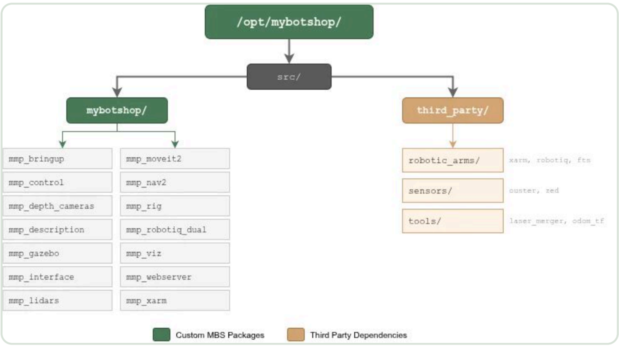


_images/web_nav_indoor.webp

ROS2 Package Reference

Workspace Structure

The MMP R100 workspace is organized into two main categories: mybotshop (custom MBS packages) and third_party (external dependencies).



_images/workspace_structure.webp

mybotshop Packages

Package	Description
mmp_bringup	System startup and service management

mmp_control	Base and manipulator controllers
mmp_depth_cameras	RealSense and ZED camera drivers
mmp_description	URDF/Xacro robot models
mmp_dual_uf850_moveit2	Dual arm MoveIt2 configuration
mmp_gazebo	Gazebo Fortress simulation
mmp_interface	Custom messages and services
mmp_lidars	Hokuyo and Ouster LiDAR drivers
mmp_nav2	Nav2 navigation stack configuration
mmp_rig	3-axis linear actuator control
mmp_robotiq_dual	Dual Robotiq 2F-140 gripper control
mmp_viz	RViz2 visualization configurations
mmp_webserver	Web dashboard and remote control
mmp_xarm	Dual xARM850 driver configuration

third_party Dependencies

Category	Package	Description
robotic_arms	xarm_ros2	UFactory xArm drivers
robotic_arms	robotiq_gripper_ros2	Robotiq gripper drivers
robotic_arms	rq_fts_ros2	Robotiq FT sensor drivers
sensors	ouster-ros	Ouster OS1 LiDAR driver
sensors	zed_ros2	Stereolabs ZED2i driver
tools	ros2_laser_scan_merger	Multi-LiDAR scan fusion
tools	odom_to_tf_ros2	Odometry to TF publisher

Emergency Stop (E-Stop)

Warning: The Emergency Stop (E-Stop) is a critical safety mechanism. When activated, it immediately halts all motion on ALL connected systems.

E-Stop Connected Systems

The E-Stop circuit is connected to and controls the following systems:

System	Description

Ridgeback Mobile Base	Clearpath Ridgeback omnidirectional platform (disables wheels)
Rig System - Base Plate	Horizontal translation (Forward/Rear, 0-400mm range)
Rig System - Left Plate	Vertical lift for left arm (Up/Down, 0-500mm range)
Rig System - Right Plate	Vertical lift for right arm (Up/Down, 0-500mm range)
Left xARM850	6-DOF robotic arm with Robotiq 2F-140 gripper
Right xARM850	6-DOF robotic arm with Robotiq 2F-140 gripper

E-Stop Behavior

When E-Stop is ENGAGED (pressed):

- All motors on the Ridgeback cease operation immediately
- All three rig axes stop and hold position (brakes engage)
- Both xARM850 arms enter emergency stop state (servos disabled)
- All motion commands are ignored until E-Stop is released

When E-Stop is RELEASED:

> **Important:** Systems do NOT automatically resume operation after E-Stop release.

Each system must be re-enabled manually:

1. Ridgeback: Automatically recovers when E-Stop is released
2. Rig System: Re-activate via service call:

```
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: true, left_plate: true, right_plate: true}"
```

1. xARM850 Arms: Re-enable via xARM web interface or driver restart

> **Warning:** If xARM850 arms fail to start after E-Stop release and you cannot hear clicking sounds, engage E-Stop again, wait 5 seconds, release, and retry enabling the arms.

Dual xARM850 Startup

1. Ensure the robot arms are receiving power (flip the 50A lever towards its black counterpart).
2. To enable, go to console in Dashboard (http://192.168.131.1:9000) and activate the xARM driver services.
3. If the xARM850 doesn't start and you can't hear clicking sound, engage the emergency stop and release it after 5 seconds. Then try again either with the driver or their web interface at: Left arm: http://192.168.131.5:18333 Right arm: http://192.168.131.6:18333

Rig Control

The 3-axis rig system provides positioning for the dual manipulator arms.

Rig Position Ranges

Axis	Min	Max	Direction
Base Plate	0	400	0=Front, 400=Rear
Left Plate	0	500	0=Down, 500=Up
Right Plate	0	500	0=Down, 500=Up

Activation

```
# Activate rig system
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: true, left_plate: true, right_plate: true}"

# Deactivate rig system (recommended when not in use)
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: false, left_plate: false, right_plate: false}"
```

Position Control

Warning: Ensure arms are not in collision path before moving rig!

```
# Move to middle + mid position
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 200.0, left_plate: 250.0, right_plate: 250.0}"

# Move to rear + highest position
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 400.0, left_plate: 500.0, right_plate: 500.0}"

# Move to front + lowest position
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 0.0, left_plate: 0.0, right_plate: 0.0}"
```

Velocity Control

For simple velocity-based movement:

```
# Stop all axes
ros2 topic pub /r100_0118/mmp_rig/cmd_vel \
  mmp_interface/msg/RigControlCmd \
  "{base_plate: 0, left_plate: 0, right_plate: 0}"

# Move base plate forward
ros2 topic pub /r100_0118/mmp_rig/cmd_vel \
  mmp_interface/msg/RigControlCmd \
  "{base_plate: 1, left_plate: 0, right_plate: 0}"
```

Dual xARM850 Control

Key Launch Files

- custom_dualarm_drivers.launch.py - Launch both arms with custom configuration
- include/xarm850_unit_1.launch.py - Left arm only
- include/xarm850_unit_2.launch.py - Right arm only

Configuration

- Left arm IP: 192.168.131.5 (Web interface: http://192.168.131.5:18333)
- Right arm IP: 192.168.131.6 (Web interface: http://192.168.131.6:18333)

MoveIt2 Configuration

MoveIt2 configuration for dual UF850 (xARM850) manipulator motion planning. Provides motion planning, collision avoidance, and coordinated dual-arm control.

Planning Groups:

- left_arm - Left xARM850 planning group
- right_arm - Right xARM850 planning group
- dual_arm - Coordinated dual-arm planning

Usage:

```
# Launch dual arm MoveIt
ros2 launch mmp_dual_uf850_moveit2 mbs_dual_arm.launch.py

# Launch visualizer
ros2 launch mmp_viz view_dualarm.launch.py
```

> **Note:** Configure planner with OMPL -> TRRT (default). Ensure to see trails of planned path before execution.

Test xARM Trajectory

```
# Make Left Robot Arm Stand
ros2 action send_goal
/r100_0118_uf1/uf850_ufactory_unit01_traj_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory "{
  trajectory: {
    joint_names: [uf_left_joint1, uf_left_joint2,
      uf_left_joint3, uf_left_joint4,
      uf_left_joint5, uf_left_joint6],
    points: [{
      positions: [-0.003, 0.424, -0.907,
        -0.005, 0.241, 0.1],
      time_from_start: {sec: 5, nanosec: 0}
    }]
  }
}"

# Make Right Robot Arm Stand
ros2 action send_goal
/r100_0118_uf2/uf850_ufactory_unit02_traj_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory "{
  trajectory: {
    joint_names: [uf_right_joint1, uf_right_joint2,
      uf_right_joint3, uf_right_joint4,
      uf_right_joint5, uf_right_joint6],
    points: [{
      positions: [-0.003, 0.424, -0.907,
        -0.005, 0.241, 0.1],
      time_from_start: {sec: 5, nanosec: 0}
    }]
  }
}"
```

Dual Gripper Control

Dual Robotiq 2F-140 gripper control via xArm RS485 tool port.

Key Launch Files

- gripper_left.launch.py - Left gripper controller
- gripper_right.launch.py - Right gripper controller

Services

- /r100_0118/gripper/left/open - Open left gripper
- /r100_0118/gripper/left/close - Close left gripper
- /r100_0118/gripper/left/set_position - Set left gripper position
- /r100_0118/gripper/left/reset - Reset left gripper
- (Same services available for right gripper)

Usage

```
# Launch gripper controllers
ros2 launch mmp_robotiq_dual gripper_left.launch.py
ros2 launch mmp_robotiq_dual gripper_right.launch.py

# Open/Close grippers via service
ros2 service call /r100_0118/gripper/left/open std_srvs/srv/Trigger
ros2 service call /r100_0118/gripper/left/close std_srvs/srv/Trigger

# Set gripper position (0=open, 255=closed)
ros2 service call /r100_0118/gripper/left/set_position \
  mmp_interface/srv/GripperCommand \
  "{position: 128, speed: 255, force: 100}"

# Reset gripper (use when gripper stops responding)
ros2 service call /r100_0118/gripper/left/reset std_srvs/srv/Trigger
```

MoveIt Action Interface

Position in radians: 0.0=open, 0.7=closed

```
# Open gripper fully (0.0 rad)
ros2 action send_goal \
  /r100_0118/left_gripper_controller/gripper_cmd \
  control_msgs/action/GripperCommand \
  "{command: {position: 0.0, max_effort: 50.0}}"

# Close gripper fully (0.7 rad)
ros2 action send_goal \
  /r100_0118/left_gripper_controller/gripper_cmd \
  control_msgs/action/GripperCommand \
  "{command: {position: 0.7, max_effort: 50.0}}"
```

> **Note:** MoveIt sends position as joint angle in radians (matching URDF finger joint limits 0-0.7 rad).

Monitor Gripper Joint States

```
ros2 topic echo /r100_0118/robotiq_2f_140/left/joint_states
ros2 topic echo /r100_0118/robotiq_2f_140/right/joint_states
```

LiDAR Sensors

Key Launch Files

- combined_scan.launch.py - Merges multiple LiDAR scans
- include/hokuyo_f.launch.py - Front Hokuyo UST-20LX
- include/hokuyo_r.launch.py - Rear Hokuyo UST-20LX
- include/ouster.launch.py - Ouster OS1 3D LiDAR

IP Configuration

Sensor	IP Address
Hokuyo Front	192.168.131.21
Hokuyo Rear	192.168.131.22
Ouster	192.168.131.30

Depth Cameras

Key Launch Files

- realsense_i.launch.py - First RealSense camera (D405)
- realsense_ii.launch.py - Second RealSense camera (D455)
- zed.launch.py - ZED2i stereo camera

> **Note:** Running Ouster and RealSense simultaneously may cause camera latency due to high bandwidth usage. Consider using one or the other for optimal performance.

Navigation

SLAM (2D Mapping)

Run SLAM to create a 2D occupancy grid map using the LiDAR sensor:

```
ros2 launch mmp_nav2 slam.launch.py
ros2 lifecycle set /r100_0118/slam_toolbox configure
ros2 lifecycle set /r100_0118/slam_toolbox activate
```

Drive the robot slowly (recommended 0.2 m/s) using the joystick controller or keyboard teleop to build the map. Once satisfied with the map, save it using:

```
ros2 run nav2_map_server map_saver_cli -f \
  /opt/mybotshop/src/mybotshop/mmp_nav2/maps/custom_map \
  --ros-args --remap map:=/r100_0118/map
```

> **Note:** Replace r100_0118 with your robot's namespace. Maps are saved to the mmp_nav2/maps/ directory. After saving, rebuild the workspace with colcon build --symlink-install so the new map can be found.

Odom Navigation

Run odometry-based navigation without a pre-built map:

```
ros2 launch mmp_nav2 odom_navi.launch.py
```

This is useful for relative navigation where you don't have a map of the environment yet.

Map-Based Navigation

Use a pre-built 2D map for autonomous navigation with AMCL localization:

```
ros2 launch mmp_nav2 map_navi.launch.py
```

This launches the full Nav2 stack including:

- Map Server - loads pre-built occupancy grid maps
- AMCL - Adaptive Monte Carlo Localization
- Controller Server - path following (DWB controller)
- Planner Server - global path planning (NavFn)
- Behavior Server - recovery behaviors
- Waypoint Follower - for multi-point navigation

Simulation

The R100 simulation package uses ROS2 Jazzy with Gazebo Fortress, featuring the complete MMP platform with dual xArm850 arms, Robotiq grippers, FT300 sensors, and adjustable rig on a Clearpath Ridgeback base.

Prerequisites

- ROS 2 Jazzy
- Gazebo Fortress
- gz_ros2_control
- Clearpath robot packages

Launch Simulation

```
export ROS_LOCALHOST_ONLY=1
source /opt/ros/jazzy/setup.bash
source /opt/mybotshop/install/setup.bash
ros2 launch mmp_gazebo simulation.launch.py
```

To kill Gazebo zombie processes:

```
ros2 run mmp_gazebo kill_gz.sh
```

Environment Variables

Configure the simulation using environment variables before launch:

Variable	Default	Description
MMP_RIDGEBACK_GAZEBO	1	Enable Gazebo simulation mode
MMP_RIDGEBACK_RIG	1	Include rig (tower + brackets)
MMP_RIG_PLATE_JOINTS	1	Enable rig joint control
MMP_RIDGEBACK_XARM850	1	Include dual xArm850 arms
MMP_RIDGEBACK_OUSTER	0	Include Ouster lidar
MMP_RIDGEBACK_ZED	0	Include ZED2i camera
MMP_RIDGEBACK_FT	1	Include Robotiq FT300 sensors
MMP_RIDGEBACK_GRIPPER	1	Include Robotiq 2F-140 grippers
MMP_RIDGEBACK_GRIPPER_D405	0	Include Intel D405 cameras

Available Controllers

Controller	Description
joint_state_broadcaster	Publishes joint states
platform_velocity_controller	Base movement (4 wheels)
uf_left_arm_controller	Left xArm850 (6 joints)
uf_right_arm_controller	Right xArm850 (6 joints)
rig_position_controller	Tower + brackets (3 joints)
uf_left_gripper_controller	Left Robotiq 2F-140
uf_right_gripper_controller	Right Robotiq 2F-140

Check controllers:

```
ros2 control list_controllers -c /r100_0118/controller_manager
```

RViz2 Visualization

```
ros2 launch mmp_viz view_robot.launch.py
```

Teleoperation (Keyboard)

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/r100_0118/cmd_vel \
  --ros-args -p stamped:=true
```

Move Arms in Simulation

```
# Move left arm
ros2 action send_goal \
  /r100_0118/uf_left_arm_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory "{
  trajectory: {
    joint_names: [uf_left_joint1, uf_left_joint2,
      uf_left_joint3, uf_left_joint4,
      uf_left_joint5, uf_left_joint6],
    points: [{
      positions: [-1.57, 1.57, 0.0, 0.5, 0.0, 0.0],
      time_from_start: {sec: 2}
    }]
  }
}"
```

Move Rig in Simulation

```
# Raise tower and extend brackets
ros2 action send_goal \
  /r100_0118/rig_position_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory "{
  trajectory: {
    joint_names: [ridgeback_tower_xarm850_joint,
      ridgeback_xarm850_bracket_left_joint,
      ridgeback_xarm850_bracket_right_joint],
    points: [{
      positions: [0.2, 0.3, 0.3],
      time_from_start: {sec: 2}
    }]
  }
}"
```

Control Grippers in Simulation

```
# Open left gripper (0.0 = open)
ros2 action send_goal \
  /r100_0118/uf_left_gripper_controller/gripper_cmd \
  control_msgs/action/GripperCommand \
  "{command: {position: 0.0, max_effort: 100.0}}"

# Close left gripper (0.695 = closed)
ros2 action send_goal \
  /r100_0118/uf_left_gripper_controller/gripper_cmd \
  control_msgs/action/GripperCommand \
  "{command: {position: 0.695, max_effort: 100.0}}"
```

Installation

ROS2 Jazzy Fresh Installation

1. Install ROS Jazzy and Clearpath packages:

```
sudo apt-get install ros-jazzy-clearpath-* -y
sudo apt update && sudo apt upgrade
```

1. Run the Clearpath computer installer script:

```
cd mbs_ridgeback/src/mybotshop/mmp_bringup/scripts
bash clearpath_computer_installer.sh
```

Add R100 Ridgeback when prompted.

1. Start the Clearpath service:

```
sudo systemctl daemon-reload
sudo systemctl start clearpath-robot.service
```

1. Create the mybotshop workspace directory:

```
sudo mkdir /opt/mybotshop
sudo chown -R robot:robot /opt/mybotshop
sudo chown -R robot:robot /etc/clearpath
```

1. Copy the workspace to /opt/mybotshop/ and run the dependency installer:

```
cd /opt/mybotshop/src
./ros2_dependencies
```

1. Build the workspace:

```
cd /opt/mybotshop
colcon build --symlink-install \
  --cmake-args -DCMAKE_BUILD_TYPE=Release
source install/setup.bash
```

1. Install custom sensor services:

```
ros2 run mmp_bringup startup_installer.py
```

1. Disable system suspend:

```
sudo systemctl mask sleep.target suspend.target \
  hibernate.target hybrid-sleep.target
```

Sync Host Computer to Robot

To sync your development workspace to the robot:

```
rsync -avP -t --delete -e ssh src \
  administrator@192.168.131.1://opt/mybotshop
```

Caution

- Always remain close to the emergency stop buttons.
- Power off the robot correctly via the proper shutdown procedure.
- Sudden jerks to the mobile robot can cause a disconnection. The velocity provided should always be smooth.
- Ensure arms are not in collision path before moving the rig system.

Known Issues

- xARM850 not starting: If the xARM850 doesn't start and you can't hear clicking sounds, engage the emergency stop and release it after 5 seconds. Then try again.
- Camera latency: Running Ouster and RealSense simultaneously may cause latency. Consider using one or the other for optimal performance.

Safety Guidelines

For comprehensive safety guidelines regarding robotic manipulators and autonomous mobile robots, please refer to the Robotic Manipulator Safety section.

7 Shared ROS Driver Reference – All Configurations

Overview

Component Activation

The ROS2 drivers mentioned below are all automatically started when launching the system services. These drivers further expand on the functionality for interfacing as well as for independent component checking.

Service Status

Check service status:

```
sudo systemctl status clearpath-robot.service
sudo systemctl status mmp-webserver.service
```

Webserver

The webserver is accessible at http://192.168.131.1:9000 with credentials:

- Username: admin
- Password: mybotshop

ROS2 Domain Setup

For ROS2 Jazzy, use the following environment setup:

```
source /opt/ros/jazzy/setup.bash
source /opt/mybotshop/install/setup.bash
```

Teleoperation

Teleoperate using the keyboard teleop package:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/r100_0118/cmd_vel \
  --ros-args -p stamped:=true
```

> **Note:** Replace r100_0118 with your robot's namespace. Reduce speed by pressing z until 0.1 for safe indoor operation.

Visualization

View the robot state in RViz2:

```
ros2 launch mmp_viz view_robot.launch.py
```

Component Activation

The ros drivers mentioned below are all automatically started when launching the accessories launch file. These drivers further expand on the functionality for interfacing as well as for independent component checking.

ROS Multimachine

To use Rviz on your pc for faster updates please setup the multimachine. By default, if you connect to the built-in router, you need only to follow the builtin router subsection.

Multimachine - Client - Builtin Router

For all the terminals in your pc, you may export these environment variables. The ros master should correspond to the robots ip where as the ros ip and ros hostname should correspond to your pc's ip that is connected to the robot in the same network.

```
export ROS_MASTER_URI = http://192.168.131.1:11311/
export ROS_IP =192.168.131. < your_assigned_ip_from_router >
export ROS_HOSTNAME =192.168.131. < your_assigned_ip_from_router >
```

Important: Please note that you should ideally have the mmp_description and mmp_viz rospackages built on your pc so that you may view the robot over the network.

Multimachine - Host - Custom Router

For all the terminals in the robot 4 — 5 recommended, you can export the following environment variables. The ip should correspond to the ip address of your robot which has been obtained from sudo nmtui .

```
export ROS_MASTER_URI =http://192.168.0.219:11311/
export ROS_IP =192.168.0.219
export ROS_HOSTNAME =192.168.0.219
```

Multimachine - Client - Custom Router

For all the terminals in your pc, you may export these environment variables. The ros master should correspond to the robots ip where as the ros ip and ros hostname should correspond to your pc's ip that is connected to the robot in the same network.

```
export ROS_MASTER_URI =http://192.168.0.219:11311/
export ROS_IP =192.168.0.225
export ROS_HOSTNAME =192.168.0.225
```

Important: Please note that you should ideally have the mmp_description and mmp_viz rospackages built on your pc so that you may view the robot over the network.

Multimachine Usage

Stop the mmp service via (On the robot).

```
sudo service mmp stop
```

On the robot launch.

```
roslaunch mmp_startup system_startup.launch
```

On the users pc launch.

```
roslaunch mmp_viz view_robot.launch
```

Robotic Arms

Dual xARM850 + Robotiq 2F-140

Please follow this checklist to interact with the dual xARM850 arms and grippers via ROS2.

- Ensure the robot arms are receiving power (flip the 50A lever towards its black counterpart)
- Release emergency switch
- Enable arms via webserver console or xARM web interface: Left arm: http://192.168.131.5:18333 Right arm: http://192.168.131.6:18333

Launch xARM drivers:

```
ros2 launch mmp_xarm custom_dualarm_drivers.launch.py
```

Or launch individual arms:

```
# Left arm only
ros2 launch mmp_xarm include/xarm850_unit_1.launch.py

# Right arm only
ros2 launch mmp_xarm include/xarm850_unit_2.launch.py
```

Gripper Control

Launch gripper controllers:

```
ros2 launch mmp_robotiq_dual gripper_left.launch.py
ros2 launch mmp_robotiq_dual gripper_right.launch.py
```

Open/Close grippers via service:

```
ros2 service call /r100_0118/gripper/left/open std_srvs/srv/Trigger
ros2 service call /r100_0118/gripper/left/close std_srvs/srv/Trigger
```

Set gripper position (0=open, 255=closed):

```
ros2 service call /r100_0118/gripper/left/set_position \
  mmp_interface/srv/GripperCommand \
  "{position: 128, speed: 255, force: 100}"
```

MoveIt2

To use MoveIt2 for dual arm planning:

```
ros2 launch mmp_dual_uf850_moveit2 mbs_dual_arm.launch.py
```

Launch visualizer:

```
ros2 launch mmp_viz view_dualarm.launch.py
```

Note: Configure planner with OMPL -> TRRT (default). Ensure to see trails of planned path before execution.

Test xARM Trajectory

```
# Make Left Robot Arm Stand
ros2 action send_goal
/r100_0118_uf1/uf850_ufactory_unit01_traj_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory "{
  trajectory: {
    joint_names: [uf_left_joint1, uf_left_joint2,
      uf_left_joint3, uf_left_joint4,
      uf_left_joint5, uf_left_joint6],
    points: [{
      positions: [-0.003, 0.424, -0.907, -0.005, 0.241, 0.1],
      time_from_start: {sec: 5, nanosec: 0}
    }]
  }
}"
```

UR5e + Robotiq

Please follow this checklist to interact with the UR5e and Gripper via ROS.

- Power on teach pendant — In Teach pendant Top Center Right
- Release emergency switch — In Teach pendant Top Center
- UR5e is in remote mode — In Teach pendant Top right
- UR5e is Power off state with a icon — In Teach pendant Bottom left (Important)

Launch ROS driver:

```
roslaunch mmp_arm ur5e_robotiq_driver.launch
```

To test the gripper run:

```
rosrun mmp_demo gripper.py
```

To test the UR5e and gripper use moveit you can use Moveit :

```
roslaunch mmp_ur_moveit_config mmp_moveit.launch
```

Auxiliary Rigs

3-Axis Rig System

The 3-axis rig system provides positioning for the dual manipulator arms with one Y-axis (base plate) and two Z-axes (lift plates).

Position Ranges

- Base Plate: 0-400 (0=Front, 400=Rear)
- Left Plate: 0-500 (0=Down, 500=Up)
- Right Plate: 0-500 (0=Down, 500=Up)

Activation

Activate rig system:

```
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: true, left_plate: true, right_plate: true}"
```

Deactivate rig system (recommended when not in use):

```
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: false, left_plate: false, right_plate: false}"
```

Position Control

Warning: Ensure arms are not in collision path before moving rig!

Move to middle position:

```
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 200.0, left_plate: 250.0, right_plate: 250.0}"
```

Move to rear + highest position:

```
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 400.0, left_plate: 500.0, right_plate: 500.0}"
```

Move to front + lowest position:

```
ros2 service call /r100_0118/mmp_rig/control \
  mmp_interface/srv/RigControl \
  "{base_plate: 0.0, left_plate: 0.0, right_plate: 0.0}"
```

Velocity Control

For simple velocity-based movement:

```
# Stop all axes
ros2 topic pub /r100_0118/mmp_rig/cmd_vel \
  mmp_interface/msg/RigControlCmd \
  "{base_plate: 0, left_plate: 0, right_plate: 0}"

# Move base plate forward
ros2 topic pub /r100_0118/mmp_rig/cmd_vel \
  mmp_interface/msg/RigControlCmd \
  "{base_plate: 1, left_plate: 0, right_plate: 0}"
```

Lift Actuation

Note: The lift driver is on by default from the startup job.

Start the driver via:

```
roslaunch mmp_lift base.launch
```

Move lift up 1 to -600 microsteps

```
rostopic pub -r 1 /cmd_lift std_msgs/Float32 "data : -600.0"
```

Move lift down 1 to 600 microsteps

```
rostopic pub -r 1 /cmd_lift std_msgs /Float32 "data : 600.0"
```

Activate Lifts softlock, for the lift to move

```
rosservice call /get_lift_manual_control "{}"
```

Sensors

Ouster + Hokuyo

> **Note:** The Hokuyos are on by default from the startup services.

To launch all LiDARs run:

```
ros2 launch mmp_lidars combined_scan.launch.py
```

Individual LiDAR launches:

```
# Front Hokuyo
ros2 launch mmp_lidars include/hokuyo_f.launch.py

# Rear Hokuyo
ros2 launch mmp_lidars include/hokuyo_r.launch.py

# Ouster OS1
ros2 launch mmp_lidars include/ouster.launch.py
```

LiDAR IP Addresses:

- Hokuyo Front: 192.168.131.21
- Hokuyo Rear: 192.168.131.22
- Ouster: 192.168.131.30

Depth Cameras

Start Intel RealSense D405 (wrist camera):

```
ros2 launch mmp_depth_cameras realsense_i.launch.py
```

Start Intel RealSense D455:

```
ros2 launch mmp_depth_cameras realsense_ii.launch.py
```

Start ZED2i stereo camera:

```
ros2 launch mmp_depth_cameras zed.launch.py
```

> **Note:** Running Ouster and RealSense simultaneously may cause camera latency due to high bandwidth usage. Consider using one or the other for optimal performance.

Navigation

SLAM (2D Mapping):

```
ros2 launch mmp_nav2 slam.launch.py
ros2 lifecycle set /r100_0118/slam_toolbox configure
ros2 lifecycle set /r100_0118/slam_toolbox activate
```

Save map:

```
ros2 run nav2_map_server map_saver_cli -f \
  /opt/mybotshop/src/mybotshop/mmp_nav2/maps/custom_map \
  --ros-args --remap map:=/r100_0118/map
```

Odometry-based navigation:

```
ros2 launch mmp_nav2 odom_navi.launch.py
```

Map-based navigation:

```
ros2 launch mmp_nav2 map_navi.launch.py
```

Ouster + Hokuyo

Note: The Hokuyos are on by default from the startup job.

To launch all three lidars run:

```
roslaunch mmp_lidars ouster_hokuyos.launch
```

To launch the hokuyos only (ideal when working with realsense):

```
roslaunch mmp_lidars hokuyos.launch
```

Realsense Camera

Start the pan and tilt camera D455:

```
roslaunch mmp_realsense_cameras d455.launch
```

Start the wrist camera:

```
roslaunch mmp_realsense_cameras d405.launch
```

Start both cameras:

```
roslaunch mmp_realsense_cameras realsense.launch
```

> **Note:** When both the Ouster and Realsense cameras work at the same time, the latency of the camera stream increases due large amount of data transfer. Ideally, Ouster should be running or Realsense. Both can work at the same time but then Realsense suffer from latency issues.

Pan & Tilt

Start the pan and tilt dynamixel motors via:

```
roslaunch mmp_pan_and_tilt controller.launch
```

Move pan and tilt with ranges of -1.5708 to +1.5708 in radian

```
rostopic pub /cmd_pan_and_tilt mmp_msgs/PanTilt " pan : 1.0" "tilt : 0.0"
```

Caution

Caution Notes

- Always follow the proper shutdown procedure to prevent data corruption and hardware damage.
- Always remain close to the emergency stop buttons.
- Deactivate the rig system before shutting down.
- Disable xARM850 arms before engaging E-Stop.
- Sudden jerks to the mobile robot can cause a disconnection. The velocity provided should always be smooth.
- Ensure arms are not in collision path before moving the rig system.

Known Issues

- xARM850 not starting If the xARM850 doesn't start and you can't hear clicking sounds, engage the emergency stop and release it after 5 seconds. Then try again either with the driver or their web interface.
- Realsense cameras lagging Running Ouster and RealSense simultaneously may cause latency due to high bandwidth usage. Consider using one or the other for optimal performance.
- E-Stop Recovery Ridgeback recovers automatically when E-Stop is released. Rig requires re-activation via service call. xARM850 requires re-enable via web interface or driver restart.

Emergency Stop

When E-Stop is engaged:

- All motors on the Ridgeback cease operation immediately
- All three rig axes stop and hold position (brakes engage)
- Both xARM850 arms enter emergency stop state (servos disabled)
- All motion commands are ignored until E-Stop is released

After releasing E-Stop:

```
# Re-activate rig system
ros2 service call /r100_0118/mmp_rig/activation \
  mmp_interface/srv/RigActivation \
  "{base_plate: true, left_plate: true, right_plate: true}"
```

Then re-enable xARM850 arms via web interface:

- Left arm: http://192.168.131.5:18333
- Right arm: http://192.168.131.6:18333

Caution Notes

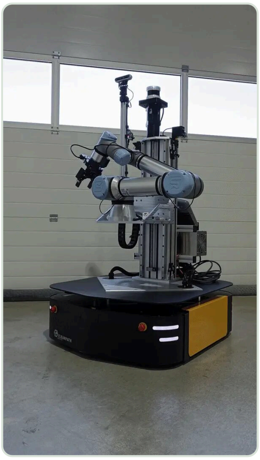
- Power off the UR5e correctly via the teach pendant.
- Always remain close to the emergency stop buttons.
- Sudden jerks to the mobile robot can cause a disconnection, the velocity that is provided should always have be smooth either through the joystick, keyboard, or program.

Known Issues

- Pan, tilt not working (Usually caused by the provision of wrong command values) ◦ Solution is to powercycle the dynamixels and restart their driver.
- Ridgeback shuts down sometimes when powered on. ◦ In rare cases the network does not startup in time causing the ridgeback to shut down.Powercycling should resolve the issue.
- Realsense cameras lagging ◦ Utilize multi-machine setup. ◦ Ensure ouster is not running concurrently with the realsense as it takes up alot of bandwith which may be causing the lag.
- Lift joystick not working ◦ This occasionally occurs if you try to move the lift while releasing the soft limit switches. Restarting the mmp service will correct this.

8 Configuration: MMP NÜ

MMP NÜ



_images/mmp_ru_front.webp

This specifies the configuration for Mobile Manipulation Platform (MMP) NÜ version. It has the following components Custom Lift System ,Stepper Motor, Rope Encoder, Ouster lidar, Nvidia Orin, Intel realsense D455 & D405 depth camera's, Universal Robot UR5e and Robotiq 2F-140 gripper.

Getting Started

To start-up the Ridgeback, the instructional video playlist can be used which details the opening and use of the Ridgeback.

Data Sheets

- Ridgeback Data Sheet
- Ouster Data Sheets
- Nvidia Orin AGX Datasheet
- Realsense D455 & D405 Datasheet
- UR5e DataSheet
- Robotiq 2F-140 Datasheet

Documentations

- Ridgeback ROS Docs
- Ouster ROS Docs

Repository

- Ridgeback
- Ouster
- Realsense D455 & D405
- UR5e
- Robotiq 2F-140 gripper

Auxiliary Sensor Startup

On ridgeback startup the following drivers are started:

1. Ridgeback base driver
2. Hokuyo

The Custom Lift System, Stepper Motor, Rope Encoder, Ouster lidar, Realsense D455 & D405 , UR5e and Robotiq 2F-140 are powered on when the Ridgeback robot starts. However, only the software drivers for Hokuyo and ridgeback are activated on start-up.

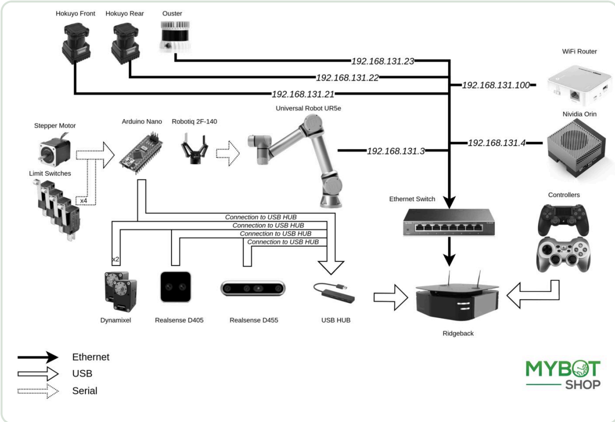
For UR5e, the arm has to be manually turned on by pressing the power button located on the top of the mounted screen. More on this in the UR5e section.

Network-Setup

The Ridgeback has multiple networks and vary slightly, the following table shows the networks.

Device	Network Address	Users	Password
Ridgeback	192.168.131.1	administrator	clearpath
Nvidia	192.168.131.2	administrator	clearpath
UR5e	192.168.131.3	N/A	N/A
Hokuyo Front	192.168.131.21	N/A	N/A
Hokuyo Rear	192.168.131.22	N/A	N/A
Ouster	192.168.131.23	N/A	N/A

Wiring Schematic



_images/mmp_wiring_schematic.webp

> **Note:** The PS4 controller is connected via bluetooth.

Operation

To use the connect and use the MMP Ridgeback, the following steps must be followed:

1. Connect to the Ridgeback LAN (Password is clearpath).
2. ssh into the ridgeback via:

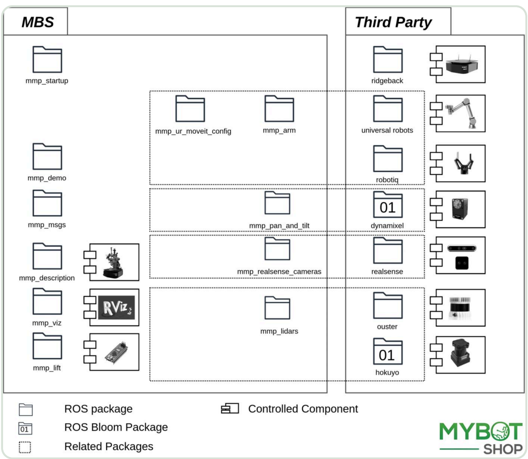
```
$ ssh -X administrator@192.168.131.1
$ clearpath
```

The following message will show up with quick instructions for startup.

```
-------------------------------------------------------------------------------

__  ____     ______   ____ _______ _____ _    _  ____  _____           _____
|  \/  \ \   / /  _ \ / __ \__   __/ ____| |  | |/ __ \|  __ \         / ____|
| \  / |\ \_/ /| |_) | |  | | | | | (___ | |__| | |  | | |__) |  _   _| |  __
| |\/| | \   / |  _ <| |  | | | |  \___ \|  __  | |  | |  ___/  | | | | | |_ |
| |  | |  | |  | |_) | |__| | | |  ____) | |  | | |__| | |      | |_| | |__| |
|_|  |_|  |_|  |____/ \____/  |_| |_____/|_|  |_|\____/|_|       \__,_|\_____|

Current PC: Ridgeback
-------------------------------------------------------------------------------

Ridgeback IP: 192.168.131.1
Pswd:         clearpath
SSH:          ssh -X administrator@192.168.131.1

Nvidia IP:    192.168.131.2
Pswd:         clearpath
SSH:          ssh -X administrator@192.168.131.2

UR5e IP:      192.168.131.3  (INFO: Default off)

Hokuyo F IP:  192.168.131.21 (INFO: Default on)
```

```
Hokuyo R IP:  192.168.131.22 (INFO: Default on)

Ouster IP:    192.168.131.23 (INFO: Default on)
--------------------------------------------------------------------------
MMP-RU Startup
--------------------------------------------------------------------------

MMP status:      sudo service mmp_ru status
MMP start:       sudo service mmp_ru start
MMP stop:        sudo service mmp_ru stop
MMP restart:     sudo service mmp_ru restart

Update startup:  rosrun mmp_ru startup_installer.py
--------------------------------------------------------------------------
MMP-RU Driver
--------------------------------------------------------------------------
UR5e Driver:     roslaunch mmp_arm arm_base.launch
UR5e MOveit:     roslaunch mmp_ur_moveit_config mmp_moveit.launch

Visualization:   roslaunch mmp_viz view_robot.launch
--------------------------------------------------------------------------
MMP-RU Services
--------------------------------------------------------------------------
Lift Movement: rostopic pub /cmd_lift std_msgs/Float32 "data: 0.0"
            (INFO: +ve for up, -ve down | -20,000 - +20,000 µsteps)

Lift Softlock: rosservice call /get_lift_manual_control "{}"
            (INFO: provides external control)

Pan and Tilt:  rostopic pub /cmd_pan_and_tilt mmp_msgs/PanTilt "pan: 0.0
            tilt: 0.0"
            (INFO: -1.5708 to +1.5708 in radian)
-------------------------------------------------------------------------------
```

Software Architecture



_images/mmp_software_architecture.webp

Important: The diagram illustrates the ROS package structure and interaction. Please note that the mmp_startup interacts will all others ros packages depending on the set environment variables.

This section will explain all the custom packages that we have written which are being used in this particular robot.

mmp_arm

This package is responsible for universal_robot UR5e & robotiq gripper control. The interface variables that are used to active UR5e & gripper software are RIDGEBACK_UR_ARM & RIDGEBACK_UR_HAND . These variables can be changed (activated/deactivated) according to your requirement by opening the setup.bash file located at mmp_startup/config changing the value to 0 or 1 . The provided code is package is used for controlling the Universal Robots UR5e robotic arm and Robotiq gripper. The launch file contains various arguments, which can be passed to it during execution to customize the behavior of the system. The launch file takes arguments for configuring the connection to the UR5e robot, such as the IP address of the robot and the ports used for communication. It also includes arguments for setting the tool communication parameters, such as the tool voltage, baud rate, parity configuration, and stop bits, among others. These parameters are only used when tool communication is enabled.

The launch file also includes arguments for setting the controller configurations, which determine the behavior of the robot arm. There are default controllers that are activated, such as the joint state controller, scaled position joint trajectory controller, and force-torque sensor controller. There are also controllers that are initially loaded but not started, such as the position joint trajectory controller and joint group vel controller.The launch file contains an argument for the robot description launch file, which is currently not used from UR as it is separately handled from Ridgeback. It also includes an argument for the kinematics configuration file used for calibration correction, which verifies that the robot's calibration matches the robot description.

Additionally, there are arguments for enabling the headless mode, which automatically sends URScript to the robot to execute, and for requiring the UR hardware interface node, which shuts down the ROS environment if the UR hardware interface node dies. Finally, the launch file includes an XML group that loads the UR common launch file and starts the UR remote control and relay arm joint states nodes. The UR common launch file sets up the connection to the robot and includes the ROS control controllers for the UR5e robot. The UR remote control node sends commands to the robot and listens for responses, while the relay arm joint states node relays the joint states between the UR5e robot and other nodes in the ROS environment.

mmp_lidars

This package is responsible for two Hokuyo Lidars (located at front and back of robot) & an Ouster lidar Sensor. The interface variables that are used to active the two Hokuyo & Ouster software are the RIDGEBACK_FRONT_HOKUYO_LASER , RIDGEBACK_REAR_HOKUYO_LASER & RIDGEBACK_OUSTER_LASER .These variables can be changed (activated/deactivated) by opening the setup.bash file located at mmp_startup/config changing the value to 0 or 1 .

The lidars.launch contains a launch file for the robot's lidar sensors. It includes three launch files for different types of lidars - hokuyo front lidar, hokuyo back lidar, and ouster. The ouster file contains launch file for the ouster lidar. It defines several arguments that can be customized when launching the node. These arguments include the IP address of the lidar, the UDP destination for the lidar data, the lidar and IMU ports, the lidar mode, the timestamp mode, and the visualization parameters. Additionally, the launch file conditionally launches the ouster sensor node based on whether the "RIDGEBACK_OUSTER_LASER" environment variable is set.

Lastly, the hokuyo lidar file contains a launch file for the front hokuyo lidar as well as back. It launches the "urg_node" node from the "urg_node" package, which communicates with the hokuyo lidars. The node's parameters include the IP address of the lidar, the baud rate, the frame ID, and the minimum and maximum angles for the sensor's range. The launch file conditionally launches the node based on whether the "RIDGEBACK_FRONT_HOKUYO_LASER" environment variable is set.

mmp_msgs

This package consists of custom messages for PanTilt and SoftSwitchesStatus which are used wile controlling different components of the robot.

mmp_startup

The mmp_ru package manages the robot startup files. Ridgeback needs ROS to be running at startup. Ridgeback driver needs to run at startup which communicates between the robot and its MCU which is on network address 192.168.131.1 .

> **Warning:** Make sure none of your devices are having 192.168.131.1 as well as 192.168.131.2 ip address this will crash the node and the robot will not function properly.

This package is dependent upon roswiki robot_upstart which converts roslaunch files to systemd service job in Ubuntu. This job runs at the startup of the robot. The one responsible for Ridgeback is mmp_startup . The status of this service can be checked by:

```
sudo service mmp_ru status
```

To change the setup files at startup of the robot include/remove nodes from launch/mbs_rp_system.launch. This is the file which is launched all the time.

To update this launch file at the robot startup, run:

```
rosrun mmp_startup startup_installer.py
```

> **Attention:** This command will remove the previous job and install a new one.

Note: Do not edit the files manually to systemd setup files or launch files (it won't work), always run the above mentioned command to reflect the changes.

This packages also includes UDEV rules for different devices on the robot. They are already set up but for a fresh installation they can be run by:

```
./setup_udev_rules.sh
```

The packages have some configuration files for the robot utilities. For example config/description.bash is responsble set up environment variables for URDF in order to include parts on the go. Similarly config/setup.bash is responsible for setting up environment variables for Ridgeback drivers.

mmp_demo

This package runs an example to demonstrate the working of all the robot component attached with the robot simultaneously. A special movement has been programed in this package that will allow you to see all the component workings.

mmp_pan_and_tilt

This package is responsible for two servo motors on which one of the camera is attached. The interface variable that is used to activate the motors is RIDGEBACK_PAN_AND_TILT_UNIT . These variables can be changed (activated/deactivated) according to your requirement by opening the setup.bash file located at mmp_startup/config changing the value to 0 or 1

mmp_ur_moveit_config

This package is responsible for the moveit configuration for all the sensors and components that are attached to the robot.

mmp_description

This is a ros package that contains the 3D models of the Ridgeback. Additionally, the seperate 3D models are stitched together into a single entity via the universal robot description format (URDF). It contains the models for auxiliary components such as lidars as well. Additions to the robot's 3D model can be made in the xacro/accessories.urdf.xacro . To view the robot with simulated dummy drivers.

```
roslaunch mmp_viz view_model.launch
```

If you want to add more accessories to the robot please make sure that you add those accessories to accessories.urdf.xacro file located in mmp_description/xacro/ so that the new accessories can be stitchedwith the old ones.

mmp_lift

This package is responsible for lift control. The interface variable that is used to activate the lift software is RIDGEBACK_LIFT_PLATE . This variables can be changed (activated/deactivated) according to your requirement by opening the setup.bash file located at mmp_startup/config changing the value to 0 or 1 . This package uses the rope encoder to provide jointstate of the lift hub for transforms between the different links.The lift itself communicates via an Arduino nano.

mmp_realsense_cameras

This package is responsible for two realsense cameras(D455 & D405) control. The interface variables that are used to active both cameras software are RIDGEBACK_PAN_AND_TILT_CAMERA & RIDGEBACK_WRIST_CAMERA . These variables can be changed (activated/deactivated) according to your requirement by opening the setup.bash file located at mmp_startup/config changing the value to 0 or 1 .

> **Note:** The package for Intel Realsense D405 has been ported from ROS2 to ROS1 and is custom, the default package will not work with mmp.

Custom Startup Job

The launch file mmp_ru_system.launch & bash file config/setup.bash``in the catkin package of ``mmp_startup includes everything which is being launched at the startup. The setup.bash file consists of all environment variables which are controlling the different components of the robot individually.By ultering the bash file we can set which individual component software is to be activated on startup.For example

```
# All the configuration variables
export RIDGEBACK_ACCESSORIES=0
export RIDGEBACK_OUSTER_LASER=0

export RIDGEBACK_FRONT_HOKUYO_LASER=1
export RIDGEBACK_REAR_HOKUYO_LASER=1

export RIDGEBACK_LIFT=1
export RIDGEBACK_PAN_AND_TILT_UNIT=1
export RIDGEBACK_PAN_AND_TILT_CAMERA=1

export RIDGEBACK_UR_ARM=1
export RIDGEBACK_WRIST_CAMERA=1

export RIDGEBACK_LIFT_PLATE=1
export RIDGEBACK_UR_HAND=1
```

All environment variables are used in a heirarichal manner, for parts in URDF and launch files to run the driver. Most of them are used for both visualization and for running the driver. Note that RIDGEBACK_UR_ARM_URDF is used for URDF to load properly.

Important: RIDGEBACK_ACCESSORIES is responsible for controlling all the accessories on lift i.e. Pan and Tilt Unit, UR Arm and Gripper, Lift Controller and all Realsense cameras.This does not control the lidars because they can be needed in cases where only Ridgeback operation is desired, so that's why Ouster is seperately set to 0 by default because it is part of lift. First RIDGEBACK_ACCESSORIES should be 1 and than all other lift accessories will be functional based on their environmet variable value.

Warning: Do not create multiple startup jobs with multiple launch files this will break the system. Only one launch file should be created and everything else should be included in this.

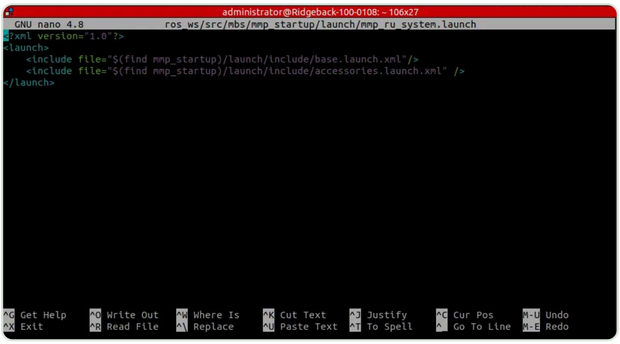
If the systems breaks at startup you can start debugging by:

```
roslaunch mmp_startup mmp_ru_system.launch
```

This will show which part of the launch file is having errors or you can start running everything in the launch file one by one in separate terminals.

Adding ROS-Nodes & Launch Files to Robot Startup

1. Navigate and open mmp_startup/launch/mmp_ru_system.launch .



_images/mbs_ridgeback_launch.webp

1. Add your custom node or launch file.
2. Run the following script:

```
rosrun mmp_startup startup_installer.sh
```

1. Verify if it is working correctly via:

```
sudo service mmp_ru status
```

Important: Once you add your custom node or launch files in the mmp_ru_system.launch , you will need to run rosrun mmp_startup startup_script.sh so that it updates the start-up job on the Ridgeback's PC otherwise it will not be updated.

Lidars

_images/ouster_s.webp

_images/hokuyo.webp

To manually launch the all three lidar driver, the following command can be used:

```
roslaunch mmp_lidars lidars.launch
```

If you want to run all the different lidars seperetly then insted of lidars.launch you can use ouster.launch , hokuyo_front_lidar.launch or hokuyo_back_lidar.launch .

Intel Realsense D455 & D405

_images/D455.webp

_images/D405.webp

On the robot start-up the Ridgeback starts, however, the Intel realsense D455 & D405 has to be manually launched.

- ssh into the Ridgeback via:

```
ssh -X administrator@192.168.131.1
```

- Password

```
clearpath
```

- To launch the Intel realsense D405 run the command

```
roslaunch mmp_realsense_cameras wrist_camera.launch
```

- To launch the Intel realsense D455 run the command

```
roslaunch mmp_realsense_cameras pan_and_tilt_camera.launch
```

Customization of the camera position can be done here via the following command

```
rostopic pub /cmd_pan_and_tilt mmp_msgs/PanTilt "pan: 0.0 tilt: 0.0"
```

Important: The Intel realsense D455 & D405 requires USB 3.0 port so please make sure to use 3.0 USB port.

UR5e

_images/Ur5e.webp

- You can manually launch Universal Robot UR5e, please run the each individual commands separately in a new terminal.

```
roslaunch mmp_arm arm_base.launch
roslaunch mmp_ur_moveit_config mmp_moveit.launch
roslaunch mmp_viz view_robot.launch
```

An emergency power cut-off for the UR5e is available in the auxiliary frame of the Ridgeback.

Important: make sure that the UR5e is manually powered on by pressing the power button located on “Teach Pendant”(tablet) that is mounted on the robot and make sure that the Emergency Stop button is released.

_images/power_button.webp

Robotiq 2F-140

_images/Gripper.webp

- To launch the Robotiq 2f-140 hardware driver, run the above command mentioned above:

```
roslaunch mmp_arm arm_base.launch
roslaunch mmp_ur_moveit_config mmp_moveit.launch
roslaunch mmp_viz view_robot.launch
```

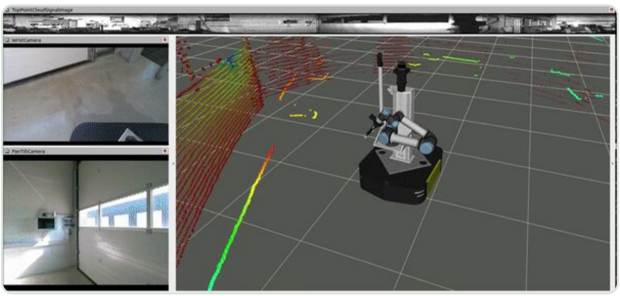
On Rviz window you will first select Manipulations from the left display section which in return will open a Display Box from which we can select Planning Group and Goal State to controll the gripper.

> **Important:** The robtiq is controlled via ros action client. You can run the following command to control the action of the gripper rostopic pub /ur_arm/command_robotiq_action/goal robotiq_2f_gripper_msgs/CommandRobotiqGripperActionGoal "header: seq: 0 stamp: secs: 0 nsecs: 0 frame_id: '' goal_id: stamp: secs: 0 nsecs: 0 id: '' goal: {emergency_release: false, emergency_release_dir: 0, stop: false, position: 1.0, speed: 0.0, force: 0.0}"

Visualization

- To visualize the robot model.

```
roslaunch mmp_viz view_robot.launch
```



_images/mmp_ru_viz.gif

Lift Control

- Movement of the lift is controlled by publishing topic mentioned below:

```
rostopic pub /cmd_lift std_msgs/Float32 "data: 0.0" -r1
```

where data: 0.0 value controlls the movement speed of the lift which is between 0 to 20,000 µsteps, where as the signs indicate the direction lift will be moving i.e. +ve for up, -ve down

9 Configuration: MMP DRE

MMP DRE

_images/mmp_ur5e.webp

This specifies the configuration for Mobile Manipulation Platform (MMP) Dresden version. It has the following components Custom Lift System ,Stepper Motor, Rope Encoder, Ouster lidar, Nvidia Orin, Intel realsense D455 & D405 depth camera's, Universal Robot UR5e and Robotiq 2F-140 gripper.

Important: Carefully review this document. If there are uncertainties, it is crucial to seek guidance from specialists, experts, or the manufacturers of the assemblies employed. The robot should not be activated until clarification is obtained. If any doubts persist, refer to the provided guides or get in touch with a specialist, expert, or the manufacturer of the assemblies used. Operational commencement of the robot should only occur after obtaining clarity. Exercise caution and do not permit individuals unfamiliar with robotics or these instructions to operate the robot. Untrained users pose a risk when handling robots.

Warning: The robot supplied by MYBOTSHOP GmbH is intended for research and development purposes and does not bear the CE Marking and/or Certificate of Incorporation. A fundamental understanding of ROS (Robot Operating System) is necessary. If you lack familiarity with ROS, we recommend consulting the ROS Wiki as a first step. It is important to note that the mentioned robot is categorized as a partially completed machine according to the Machinery Directive 2006/42/EC and does not carry a CE marking. The client acknowledges and agrees that any information or materials provided by MYBOTSHOP GmbH are exclusively for research and development purposes. All services are provided on an "AS IS" basis without any representation or warranty, whether express or implied, including but not limited to merchantability, fitness for a particular purpose, non-infringement, or any other warranty. MYBOTSHOP GmbH shall not be held liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising from or in connection with the use or inability to use the provided information or materials. This limitation on liability applies irrespective of the form of action, whether in contract, tort, or otherwise.

Getting Started

To start-up the Ridgeback, the instructional video playlist can be used which details the opening and use of the Ridgeback.

Data Sheets

- Ridgeback Data Sheet
- Ouster Data Sheets
- Nvidia Orin AGX Datasheet
- Realsense D455 & D405 Datasheet
- UR5e DataSheet
- Robotiq 2F-140 Datasheet

Documentations

- Ridgeback ROS Docs
- Ouster ROS Docs

Repository

- Ridgeback
- Ouster
- Realsense D455 & D405
- UR5e
- Robotiq 2F-140 gripper

Auxiliary Sensor Startup

On Ridgeback startup, the following drivers are initiated

- Ridgeback base driver
- Hokuyo Lidar

The Custom Lift System, Stepper Motor, Rope Encoder, Ouster Lidar, Realsense D455 & D405, UR5e, and Robotiq 2F-140 are powered on when the Ridgeback robot starts. However, only the software drivers for Hokuyo and Ridgeback are activated during startup.

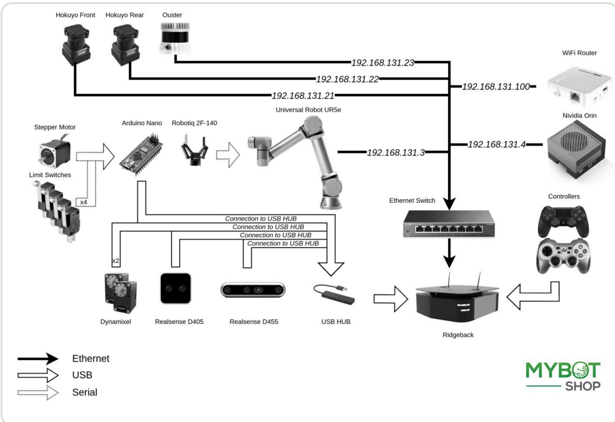
For UR5e, the arm has to be manually turned on by pressing the power button located on the top of the mounted screen. More details on this can be found in the UR5e section.

Network-Setup

The Ridgeback has multiple networks and vary slightly, the following table shows the networks.

Device	Network Address	Users	Password
Ridgeback	192.168.131.1	administrator	mybotshop
Nvidia	192.168.131.4	administrator	mybotshop
UR5e	192.168.131.5	N/A	N/A
Hokuyo Front	192.168.131.20	N/A	N/A
Hokuyo Rear	192.168.131.21	N/A	N/A
Ouster	192.168.131.22	N/A	N/A

Wiring Schematic



_images/mmp_wiring_schematic.webp

Note: The PS4 controller is connected via bluetooth.

Operation

_images/ethernet.webp

Initially, connect to the robot's WLAN network by using a LAN cable for the first setup. To establish a static connection on your PC (not the robot) in Ubuntu, navigate to Settings → Network, then select + to create a new connection and follow these steps

1. Switch the connection to Manual in the IPv4 settings.
2. Configure the IP address as 192.168.131.51 and the Netmask as 24 .
3. Save the settings and restart your network.

Once the connection is successful, identify the host's local IP by entering the following command in the Host PC's terminal:

```
ifconfig
```

Subsequently, ping the robot:

```
ping 192.168.131.1
```

Access the robot through SSH:

```
ssh -X administrator@192.168.131.1
```

The default password is:

```
mybotshop
```

The following message will show up with quick instructions for startup.

```
-------------------------------------------------------------------------------
      _  _  _  _  ____   __  ____  ____  _  _   __ ____
     ( \/ )( \/ )(  _ \ /  \(_  _)/ ___)/ )( \ /  \(  _ \
     / \/ \ )  /  ) _ ((  0 ) )(  \___ \) __ ((  0 )) __/
     \_)(_/(__/  (____/ \__/ (__) (____/\_)(_/ \__/(__)

Current PC: Ridgeback
-------------------------------------------------------------------------------
Ridgeback IP:           192.168.131.1
Pswd:                   mybotshop
SSH:                    ssh -X administrator@192.168.131.1

Ridgeback MCU:          192.168.131.2
e
Nvidia IP:              192.168.131.4
Pswd:                   mybotshop
SSH:                    ssh -X administrator@192.168.131.4

UR5e IP:                192.168.131.5  (? Default off)

Hokuyo F IP:            192.168.131.20
Hokuyo R IP:            192.168.131.21
Ouster IP:              192.168.131.22
-------------------------------------------------------------------------------
MMP Startup
-------------------------------------------------------------------------------
```

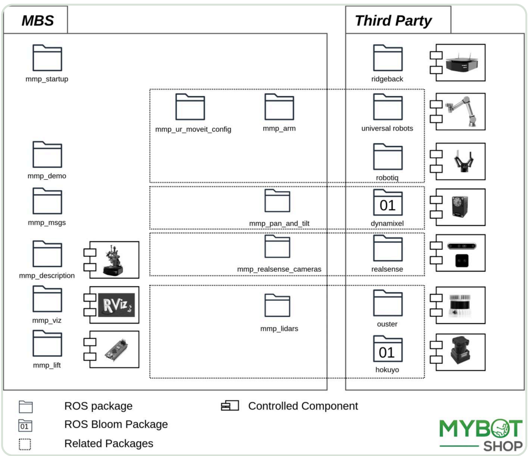
```
MMP status:          sudo service mmp status
MMP start:           sudo service mmp start
MMP stop:            sudo service mmp stop
MMP restart:         sudo service mmp restart

All Augments:        roslaunch mmp_startup accessories.launch
UR5e Moveit:         roslaunch mmp_ur_moveit_config mmp_moveit.launch
Visualization:       roslaunch mmp_viz view_robot.launch

Reinstall Startup:   rosrun mmp_startup startup_installer.py

Rebuild package:     catkin build -DCMAKE_BUILD_TYPE=Release
--------------------------------------------------------------------------------
MMP Services
--------------------------------------------------------------------------------
Lift Movement: rostopic pub /cmd_lift std_msgs/Float32 "data: 0.0"
               (? +ve for up/ -ve down | -9000 to 9000 microstep)

Lift Softlock: rosservice call /get_lift_manual_control "{}"
               (? provides external control)
--------------------------------------------------------------------------------
```

Note: Occasionally, disruptions may occur when connecting to the MMP due to other active networks. It is recommended to have only the connection to the robot active, with all others inactive.

Screen connection

As an alternative method, connect to the robot by plugging in an HDMI cable along with a mouse and keyboard. This allows you to establish a connection to the robot from your local network.

Software Architecture



_images/mmp_software_architecture.webp

Important: The diagram illustrates the ROS package structure and interaction. Please note that the mmp_startup interacts will all others ros packages depending on the set environment variables.

This section will explain all the custom packages that we have written which are being used in this particular robot.

mmp_arm

_images/arm2.webp

This package facilitates control of the Universal Robots UR5e robotic arm and Robotiq gripper through a customizable launch file. The launch file includes arguments for robot connection, tool communication parameters, and controller configurations. It also supports headless mode, URScript automation, and hardware interface node safety. Additionally, the launch file incorporates XML for UR common functionalities, connecting to the robot, and managing joint states in the ROS environment.

mmp_lidars

This package manages front and back Hokuyo Lidars, and an Ouster Lidar Sensor, with a dedicated lidars.launch file. The launch file includes configurations for each lidar type, allowing customization of parameters like IP address, ports, and visualization.

mmp_msgs

This package consists of custom messages for PanTilt and SoftSwitchesStatus which are used wile controlling different components of the robot.

mmp_startup

The mmp package manages the robot startup files. Ridgeback needs ROS to be running at startup. Ridgeback driver needs to run at startup which communicates between the robot and its MCU which is on network address 192.168.131.1 .

> **Warning:** Make sure none of your devices are having 192.168.131.1 as well as 192.168.131.2 ip address this will crash the node and the robot will not function properly.

This package is dependent upon roswiki robot_upstart which converts roslaunch files to systemd service job in Ubuntu. This job runs at the startup of the robot. The one responsible for Ridgeback is mmp_startup . The status of this service can be checked by:

```
sudo service mmp status
```

To update this launch file at the robot startup, run:

```
rosrun mmp_startup startup_installer.py
```

Attention: This command will remove the previous job and install a new one.

Note: Do not edit the files manually to systemd setup files or launch files (it won't work), always run the above mentioned command to reflect the changes.

This packages also includes UDEV rules for different devices on the robot. They are already set up but for a fresh installation they can be run by:

```
./setup_udev_rules.sh
```

The packages have some configuration files for the robot utilities. For example config/description.bash is responsible set up environment variables for URDF in order to include parts on the go. Similarly config/setup.bash is responsible for setting up environment variables for Ridgeback drivers.

mmp_demo

This package runs an example to demonstrate the working of all the robot component attached with the robot simultaneously. A special movement has been programed in this package that will allow you to see all the component workings.

mmp_pan_and_tilt

This package is responsible for two servo motors on which one of the camera is attached.

mmp_ur_moveit_config

This package is responsible for the moveit configuration for all the sensors and components that are attached to the robot.

mmp_description

This is a ros package that contains the 3D models of the Ridgeback. Additionally, the seperate 3D models are stitched together into a single entity via the universal robot description format (URDF). It contains the models for auxiliary components such as lidars as well. Additions to the robot's 3D model can be made in the xacro/accessories.urdf.xacro . To view the robot with simulated dummy drivers.

```
roslaunch mmp_viz view_model.launch
```

If you want to add more accessories to the robot please make sure that you add those accessories to accessories.urdf.xacro file located in mmp_description/xacro/ so that the new accessories can be stitchedwith the old ones.

mmp_lift

This package is responsible for lift control. This package uses the rope encoder to provide joint_state of the lift hub for transforms between the different links.The lift itself communicates via an Arduino nano.

mmp_realsense_cameras

This package is responsible for two realsense cameras(D455 & D405) control.

> **Note:** The package for Intel Realsense D405 has been ported from ROS2 to ROS1 and is custom, the default package will not work with mmp.

Quick-Start

Powering on the Ridgeback

_images/power.webp

Confirm that the emergency buttons on all four corners are released.

1. To verify the release of the emergency switch, twist it in the indicated direction. If it pops up, the switch has been successfully released.
2. Press the power button on the rear of the Ridgeback, identified by the power symbol.
3. Once the Ridgeback is powered on, and all emergency switches have been released, press the e-reset button located at the rear where the Ridgeback has a yellow flashing light.

Powering on the UR5e

_images/tp.webp

1. The UR5e should only be powered on when the Ridgeback is fully powered on (indicated by white lights in front and/or a flashing yellow light for the e-reset).
2. Power on the UR5e using the power button on the teach pendant.
3. Allow a few minutes for the UR5e to complete its startup.
4. Confirm that all emergency switches are released, both on the teach pendant and on the lift. Avoid initiating any actions on the UR5e, as ROS handles these automatically. Powering on is the only required action.

Powering Auxiliary Components

Auxiliary components such as the lift, Ouster, Hokuyo lidars, Dynamixels, RealSense cameras, Nvidia, etc., are powered on automatically during startup.

Joystick Teleoperation for Ridgeback

_images/ps4.webp

1. Utilize the PS button to power on the joystick.
2. L1 serves as the dead man's switch.
3. Use Left-Joystick for movement in the x and y directions.
4. Utilize Right-Joystick for movement in the yaw orientation.
5. Square button serves as lift and gripper dead man's switch .
6. Top and Bottom arrow key on PS4 controller is used to controll lift up/down.

7. Left and Right arrow key on PS4 controller is used to gripper open/close.

Manipulator Activation

Once the UR5e is powered on and its emergency switches are released:

- Confirm that the UR5e is in remote mode (should be in remote mode by default) — Located in the top right of the Teach pendant.
- Check if the UR5e is in a Power off state with an icon — Located in the bottom left of the Teach pendant (Important).

> **Important:** The arm must always be in the Power off state on the Teach pendant for the ROS driver to execute the automated script that boots and configures the UR5e.

Auxiliary ROS Drivers

After activating the manipulator, the next step is to launch the following driver:

```
roslaunch mmp_startup accessories.launch
```

This command should activate all components, including the arm, gripper, lift, lidars, camera, and pan & tilt.

Digital Twin Visualization (RVIZ)

To observe the MMP digitally, execute the following command:

```
roslaunch mmp_viz view_robot.launch
```

For optimal and low-latency streaming, it is recommended to employ the ROS multimachine setup outlined in ROS Multimachine Setup section.

Individual Component Activation

The ROS drivers mentioned below are automatically initiated when launching the accessories launch file. These drivers enhance functionality for interfacing and independent component checking.

ROS Driver — UR5e + Robotiq 2F-85

_images/arm.webp

Follow this checklist to interact with the UR5e and Gripper via ROS:

1. Power on the teach pendant — Located in the Teach pendant Top Center Right.
2. Release the emergency switch — Located in the Teach pendant Top Center.
3. Confirm the UR5e is in remote mode — Located in the Teach pendant Top Right.
4. Ensure the UR5e is in the Power off state with an icon — Located in the Teach pendant Bottom Left (Important).

> **Note:** The arm must always be in a powered-off state for the ROS driver to execute, as the script boots the UR5e.

Launch the ROS driver:

```
roslaunch mmp_arm ur5e_robotiq_driver.launch
```

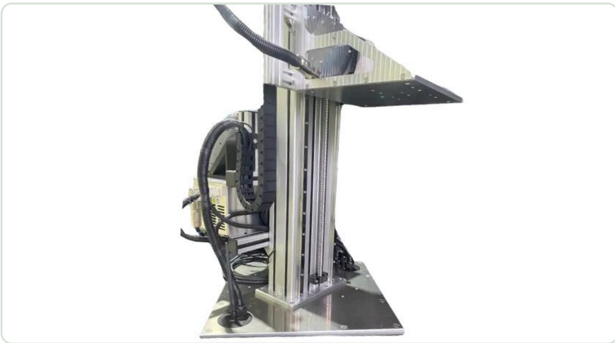
To test the gripper, run:

```
rosrun mmp_demo gripper.py
```

For testing the UR5e and gripper using Moveit, you can use Moveit:

```
roslaunch mmp_ur_moveit_config mmp_moveit.launch
```

ROS Driver — Lift



_images/lift2.webp

Start the driver via:

```
roslaunch mmp_lift base.launch
```

Move the lift up 1 to 9000 microsteps:

```
rostopic pub -r 1 /cmd_lift std_msgs/Float32 "data: 9000.0"
```

Move the lift down -1 to -9000 microsteps:

```
rostopic pub -r 1 /cmd_lift std_msgs/Float32 "data: -9000.0"
```

Activate Lift's softlock for the lift to move:

```
rosservice call /get_lift_manual_control "{}"
```

ROS Driver — Ouster + Hokuyo Front + Hokuyo Rear

_images/oh.webp

To launch all three lidars, execute the following command:

```
roslaunch mmp_lidars ouster_hokuyos.launch
```

To launch only the Hokuyos (ideal when working with RealSense):

```
roslaunch mmp_lidars hokuyos.launch
```

ROS Driver — RealSense Camera

_images/rs.webp

Start the pan and tilt camera D455:

```
roslaunch mmp_realsense_cameras d455.launch
```

Start the wrist camera:

```
roslaunch mmp_realsense_cameras d405.launch
```

Start both cameras:

```
roslaunch mmp_realsense_cameras realsense.launch
```

Note: When both the Ouster and RealSense cameras operate simultaneously, the camera stream's latency increases due to a large amount of data transfer. Ideally, Ouster or RealSense should be running. Both can work simultaneously, but then RealSense may experience latency issues.

ROS Driver — Pan & Tilt

_images/pt.webp

Start the pan and tilt Dynamixel motors via:

```
roslaunch mmp_pan_and_tilt controller.launch
```

Move pan and tilt within ranges of -1.5708 to +1.5708 in radians:

```
rostopic pub /cmd_pan_and_tilt mmp_msgs/Panlt "pan: 1.0" tilt: 0.0"
```

ROS Multimachine Setup

Multi-Machine - Host

For all terminals on the robot (4-5 terminals recommended), export the following environment variables. The IP should correspond to the robot's IP obtained from sudo nmtui .

```
export ROS_MASTER_URI=http://192.168.0.219:11311/
export ROS_IP=192.168.0.219
export ROS_HOSTNAME=192.168.0.219
```

Multimachine - Client

For all terminals on your PC, export these environment variables. The ROS master should correspond to the robot's IP, while ROS IP and ROS hostname s hould match your PC's IP connected to the robot in the same network.

```
export ROS_MASTER_URI=http://192.168.0.219:11311/
export ROS_IP=192.168.0.225
export ROS_HOSTNAME=192.168.0.225
```

Please ensure that the mmp_description and mmp viz ROS packages are built on your PC to view the robot over the network.

Multimachine Usage

Stop the mmp service on the robot:

```
sudo service mmp stop
```

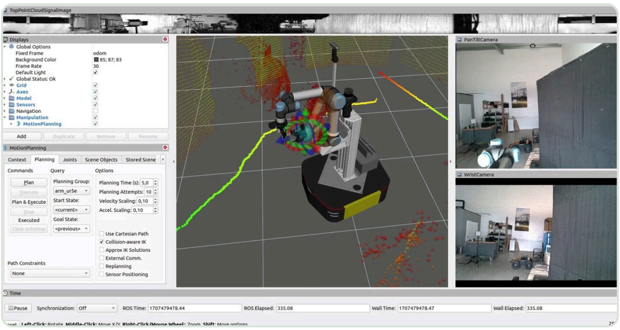
Launch on the robot:

```
roslaunch mmp_startup system_startup.launch
```

On the user's PC, launch:

```
roslaunch mmp_viz view_robot.launch
```

Demo



_images/ur5e_viz.webp

Launch all accessories:

```
roslaunch mmp_startup accessories.launch
```

Note: Ensure the UR5e setup is as mentioned before, i.e., the UR5e is powered on, emergency switch released, and is in remote mode.

Run MoveIt configuration for the robot:

```
roslaunch mmp_ur_moveit_config mmp_moveit.launch
```

Run the demo:

```
rosrun mmp_demo demo.py
```

Caution

- Power off the UR5e correctly via the teach pendant.
- Always remain close to the emergency stop buttons.
- Sudden jerks to the mobile robot can cause a disconnection. The velocity provided should always be smooth, either through a joystick, keyboard, or program.

Known Issues

- Pan and tilt not working (Usually caused by the provision of wrong command values). Power cycle the dynamixels and restart their driver. A button is available for quick power cycle.
- Ridgeback shuts down sometimes when powered on. In rare cases, the network may not startup in time, causing the Ridgeback to shut down. Power cycling should resolve the issue.
- Realsense cameras lagging. Utilize a multi-machine setup. Ensure Ouster is not running concurrently with the Realsense, as it takes up a lot of bandwidth, which may be causing the lag.

Robotic Manipulator Safety Guidelines:

When deploying robotic manipulators, it is crucial to prioritize safety procedures to mitigate risks and ensure secure operations. The following guidelines outline key safety measures when working with robotic manipulators:

Work Area Safety

- Maintain a clean and well-organized work area. Cluttered or inadequately lit spaces can impede the proper functioning of sensors and hinder precise manipulation.
- Avoid operating robotic manipulators in hazardous environments, such as those containing corrosive substances, extreme temperatures, or sharp objects that may damage the manipulator.

- Ensure that only authorized personnel are present in the vicinity during manipulator operation to prevent interference and ensure a safe working environment.

Electrical Safety

- Ensure the manipulator's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement safeguards to protect the manipulator from adverse environmental conditions, such as exposure to moisture or extreme temperatures.
- Regularly inspect power cables and connections, promptly replacing damaged components to minimize the risk of electrical issues.

Manipulation Safety

- Implement collision detection systems to prevent unintended contact with objects, humans, or other equipment during manipulation tasks.
- Define and enforce safety zones around the manipulator's workspace to minimize the risk of unintended interactions with personnel or other objects.
- Regularly calibrate and test the manipulator's sensors and systems to ensure precise and reliable performance during manipulation tasks.

Emergency Response

- Install an emergency stop mechanism to swiftly halt manipulator operation in unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations within the manipulator's operational area.

Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations during manipulator operation.

Data Security and Privacy

- Implement robust cybersecurity measures to safeguard the manipulator's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the manipulator's sensors.
- Regularly update software and firmware to address security vulnerabilities and enhance overall system integrity.

Human Interaction Safety:

- Integrate sensors and communication systems to detect and respond to the presence of humans in the manipulator's vicinity.
- Clearly communicate the manipulator's operational status and intentions using visual and audible signals to alert nearby individuals.

- Establish protocols for safe human-robot collaboration, particularly in shared workspaces where manipulators are in operation.

Residual Risks

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions during complex manipulation tasks.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

Robotic manipulators are sophisticated technologies that demand correct usage to avoid accidents and ensure a secure environment. Learn and adhere to proper procedures diligently, prioritizing both precision and safety. By following these safety guidelines, you contribute to maintaining a secure environment when deploying robotic manipulators.

Autonomous Mobile Robot Safety Guidelines

When deploying autonomous mobile robots, prioritizing safety procedures is essential to prevent accidents and ensure secure operations. The following guidelines outline key safety measures when working with an autonomous mobile robot.

Work Area Safety

- Maintain a clean and well-lit work area. Cluttered or poorly illuminated spaces can impede the proper functioning of sensors and navigation systems.
- Avoid operating autonomous mobile robots in explosive atmospheres, such as areas with flammable liquids, gases, or dust. The robot's components may pose a risk in such environments.
- Keep bystanders and unauthorized personnel at a safe distance during robot operation to prevent interference with autonomous navigation.

Electrical Safety

- Ensure the robot's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement mechanisms to protect the robot from adverse weather conditions, such as rain or wet environments.
- Regularly inspect power cables and connections and replace damaged components promptly to minimize the risk of electrical issues.

Navigation Safety

- Implement obstacle detection and avoidance systems to prevent collisions with objects, people, or other robots.

- Define and enforce safety zones within the robot's operational area to minimize the risk of unintended interactions with personnel or other equipment.
- Regularly calibrate and test the robot's navigation sensors and systems to ensure accurate and reliable performance.

Emergency Response:

- Install an emergency stop mechanism to quickly halt the robot's operation in case of unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations throughout the robot's operational area.
- Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations.

Data Security and Privacy:

- Implement robust cybersecurity measures to protect the robot's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the robot's sensors.
- Regularly update software and firmware to address security vulnerabilities and enhance overall system integrity.

Human Interaction Safety

- Integrate sensors and communication systems to detect and respond to the presence of humans in the robot's vicinity.
- Clearly communicate the robot's operational status and intentions using visual and audible signals to alert nearby individuals.
- Establish protocols for safe human-robot collaboration, especially in shared workspaces.

Residual Risks:

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions in crowded or dynamic environments.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

These autonomous mobile robots are advanced technologies that require correct usage to avoid accidents and ensure a secure environment. Learn and follow the proper procedures diligently; prioritize both quality and safety. By adhering to these safety guidelines, you contribute to ensuring a safe environment while deploying an autonomous mobile robot.

10 Configuration: MMP MPI

MMP MPI

Network Setup

_images/rp_banner.webp

A WLAN router is installed on Ridgeback and has a network address in 192.168.131.* range. As Clearpath is also on range 192.168.131.* for this reason router is also set on the same network. SSID of the Wifi network is MBS - MAXPLANCK - RP and the password is mybotshop2022! .

The ip address of the router is 192.168.131.100 (password for the router network is also mybotshop2022). By connecting to the Wifi network one can access Ridgeback on ip address 192.168.131.1 . Moreover if a static connection is desired than setting for the host PC in the same network will also work. For example by setting 192.168.131.51 in the ip and 255.255.255.0 in netmask for the static connection will connect to the Ridgeback network as well. To SSH into Ridgeback run the following command:

```
ssh administrator@192.168.131.1
```

The password for the robot is clearpath .

Network tables

All the devices on the robot are network enabled. Table [robot_network] outlines all the devices and their network addresses.

Device	Network Address
Ridgeback	192.168.131.1
Ridgeback MCU	192.168.131.2
Front Lidar	192.168.131.20
Rear Lidar	192.168.131.21

Panda arm	192.168.131.22
Ouster	192.168.131.23
Router	192.168.131.100

Heavy Duty Lift

lift

Starting the Lift

The sensors node is one of the most important node as it controls the communication with three control boards.

```
rosrun mbs_rp_lift sensors
```

The node offers the following functionalities for the corresponding boards:

1. Arm power control : board offers rosservice set_arm_power , a call to this service would switch on/off the Panda arm. Please keep in mind if the arm is switched on and the node is turned down, this will switch off the arm as well.
2. Soft switches control : board publishes the status of soft switches on topic soft_switches_status . The topic is subscribed by the driver node and the status is used for the lift control.
3. Lift encoder : The external rope encoders status is published on lift_joint_states topic.

For convenience to launch both lift_utils and driver node, the following launch file can be run:

```
roslaunch mbs_rp_lift rp_lift.launch
```

Attention: In case of some failure and the soft switches are bypassed. Hard switches will cut the power connection to the lift. The normal driver will stop working when hard switches are in failure state. So to bypass all safety checks of driver node, manual control node needs to be used.

Hard Switch Override

The manual_control ros node is used to override the hardswitches. With this node brakes needs to be release and applied manually. Additionally, it sends the movements command accordingly. Depending upon which switches are pressed all the information on how to get out of failure state is printed in the console.

To release the brakes:

```
rosservice call /apply_brake 'data: False'
```

To apply the brakes:

```
rosservice call /apply_brake 'data: True'
```

Lift Velocity

The driver node is the main control node to move the lift:

```
rosrun rp_lift driver
```

It offers a ROS service to move the lift up and down.

To move the lift up:

```
rosservice call /set_lift_velocity 'data: 2047'
```

To move the lift down:

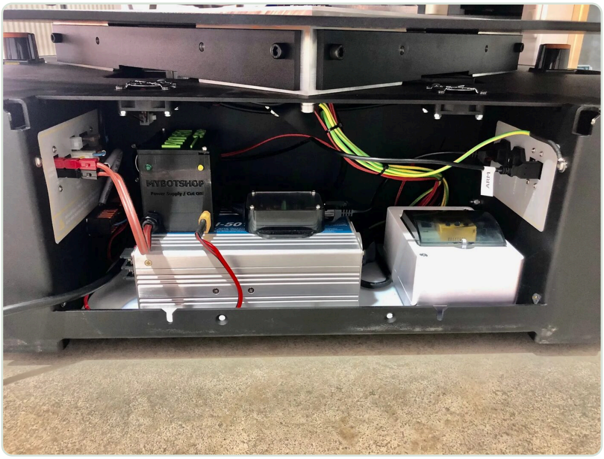
```
rosservice call /set_lift_velocity 'data: -2047'
```

Positive values are to move the lift upwards and negative to move downwards. The driver take a value between -2047 and 2047.

> **Important:** These values are micro-steps not meters per second as offered by the motor driver for the lift. The driver checks the status of soft and hard switches all the time and provides corresponding control and advice for each state.

Safety-Controller

DC-AC Converter



Panda

The Panda is connected to a DC-AC converter which powers it from the external battery. Specific features of it are:

1. Conversions of 24V DC - 230V AC
2. User can supply power to the arm asl well as cut off power to the arm
3. Safety feature based on the Bender Isometer
4. LED status of the converter on top of the Ridgeback
5. Power cut-off in emergency situation

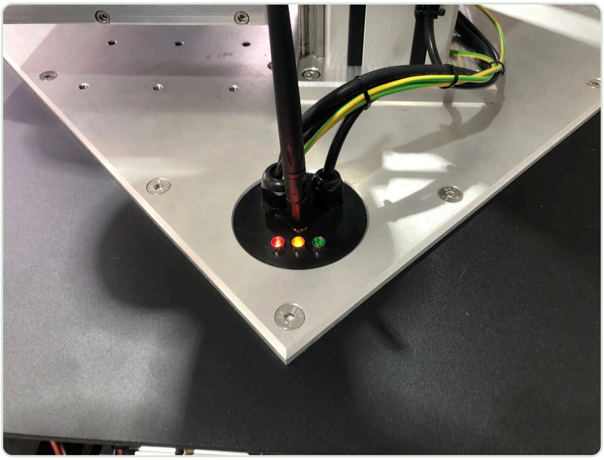
Bender Isometer

Panda

1. Safety Check on Startup
2. Checking internal resistance
3. Pushing Test-Button
4. No change in resistace was recognized
5. Power will be supplied

> **Note:** More information on the Bender Isometer .

LED Indicator



Panda

- Green: 24V DC was supplied to Converter
- Yellow: A failure has happened, operator must stop working with the robot, and must check for a broken cable etc.
- Red: Power was cut off

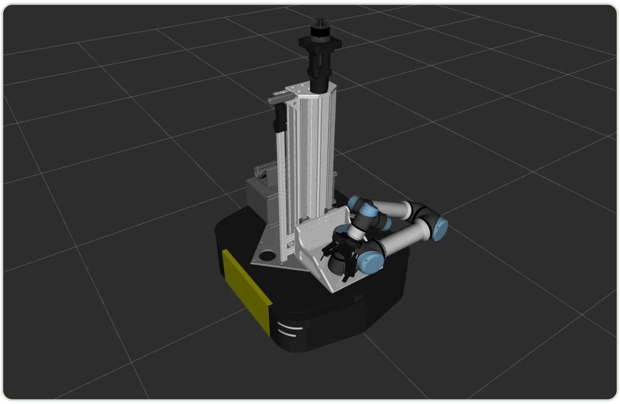
MMP Viz

Important: This is applicable to the MMP Ridgeback Panda v1.2 (released prior to 2023).

This mmp_viz package is a configuration package used for visualizations in RVIZ.

To view the URDF model of the robot:

```
roslaunch rp_viz view_model.launch
```

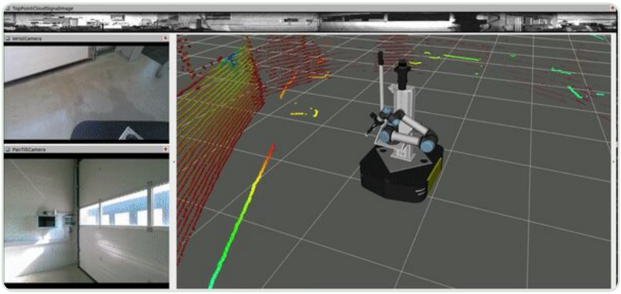


_images/mmp_viz_view_model.webp

Ridgeback UR5E Rviz

To view the live sensor data from the robot:

```
roslaunch rp_viz view_robot.launch
```



_images/mmp_ru_viz.gif

Rviz with Ridgeback's whole body control demo

MMP Startup

> **Important:** This is applicable to the MMP Ridgeback Panda v1.2 (released prior to 2023).

The mbs_rp_startup package manages the robot startup files. Ridgeback needs ROS to be running at startup. Ridgeback driver needs to run at startup which communicates between the robot and its MCU which is on network address 192.168.131.2.

> **Warning:** Make sure none of your devices are having 192.168.131.2 ip address this will crash the node and the robot will not function properly.

This package is dependent upon roswiki robot_upstart which converts roslaunch files to systemd service job in Ubuntu. This job runs at the startup of the robot. The one responsible for Ridgeback is mbs_ridgeback . The status of this service can be checked by:

```
sudo service mbs_rp status
```

To change the setup files at startup of the robot include/remove nodes from launch/mbs_rp_system.launch. This is the file which is launched all the time.

To update this launch file at the robot startup, run:

```
rosrun rp_startup startup_installer.py
```

> **Attention:** This command will remove the previous job and install a new one.

> **Note:** Do not edit the files manually to systemd setup files or launch files (it won't work), always run the above mentioned command to reflect the changes.

This packages also includes UDEV rules for different devices on the robot. They are already set up but for a fresh installation they can be run by:

```
rosrun rp_startup create_udev_rules.sh
```

The packages have some configuration files for the robot utilities. For example config/description.bash is responsble set up environment variables for URDF in order to include parts on the go. Similarly config/setup.bash is responsible to setup environment variables for Ridgeback drivers. Lastly config/can-udp-bridge* are responsible for setting up can bus and is used in startup script of the robot.

MMP Manipulation

Important: This is applicable to the MMP Ridgeback Panda v1.2 (released prior to 2023).

Franka-Emika-Panda

Panda

Note: As the MYBOTSHOP Ridgeback Panda is a multi-robot system consisting of Franka Emika's Panda, hence, the guide provided by Franka Emika for Franka Emika Guide is applicable.

MBS Panda

This mbs_rp_franka package is responsible for to launch Franka control software from franka control package. Before using the arm it needs to switched on by the rosrun rp_lift sensors .

To switch on the arm:

```
rosservice call /set_arm_power 'data: True'
```

Note: The arm is switched on indicated by solid yellow lights on the robot.

To switch off the arm:

```
rosservice call /set_arm_power 'data: False'
```

Panda Setup

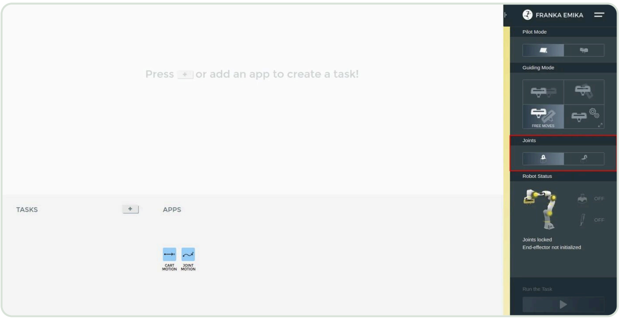
Once arm is powered on, one can access the Panda desk at 192.168.131.22 . The desk will prompt for user name and password which is:

User: admin

Password: panda_admin

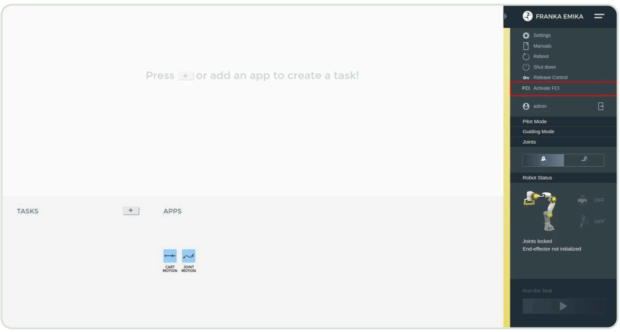
> **Tip:** Getting started with Panda Desk

First the arm joints should be unlocked as shown.



Panda

After the arm's joints are unlocked, FCI needs to be enabled so that the arm exposes its functionalities to ROS control i.e. to be externally controlled.



Panda

Finally, the push button connected to X2 port of the robot needs to be pressed as well.

MMP Navigation

> **Important:** This is applicable to the MMP Ridgeback Panda v1.2 (released prior to 2023).

RIDGEBACK NAVIGATION

ridgeback_navigation

> **Note:** As the MYBOTSHOP Ridgeback Panda is a multi-robot system consisting of Clearpath's Ridgeback, hence, the guide provided by Clearpath for Ridgeback Navigation is applicable. For ease of view the guide is available below as well.

Below are the example launch files for three different configurations for navigating Ridgeback:

- Navigation in an odometric frame without a map, using only roswiki move_base .
- Generating a map using roswiki gmapping .
- Localization with a known map using roswiki amcl .

If you're working with a real Ridgeback, it's suggested to connect via SSH and launch the roswiki ridgeback_navigation launchfiles from on board the robot. You'll need to have bidirectional communication with the robot's roscore in order to launch roswiki rviz on your workstation.

Navigation Without a Map

In the odometry navigation demo Ridgeback attempts to reach a given goal in the world within a user-specified tolerance. The 2D navigation, generated by move_base, takes in information from odometry, laser scanner, and a goal pose and outputs safe velocity commands. In this demo the configuration of move_base is set for navigation without a map in an odometric frame (that is, without reference to a map).

To get all Navigation related files for Ridgeback, run:

```
sudo apt-get install ros-melodic-ridgeback-navigation
```

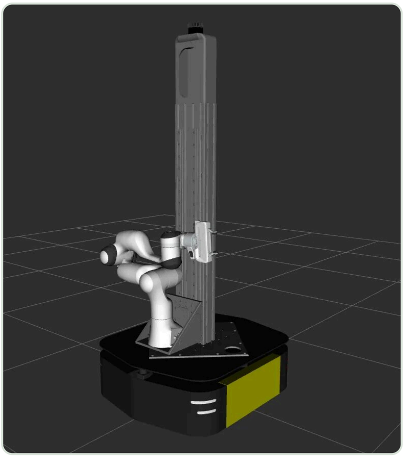
To launch the navigation demo, run:

```
roslaunch ridgeback_navigation odom_navigation_demo.launch
```

To visualize with the suggested rviz configuration launch:

```
roslaunch rp_viz view_model.launch
```

Note: Add the topic of /points to visualize the point clouds as well.



rviz with Ridgeback's odom navigation configuration.

To send goals to the robot, select the 2D Nav Goal tool from the top toolbar, and then click anywhere in the rviz view to set the position. Alternatively, click and drag slightly to set the goal position and orientation.

If you wish to customize the parameters of move_base, local costmap, global costmap and base_local_planner, clone roswiki ridgeback_navigation into your own workspace and modify the corresponding files in the params subfolder.

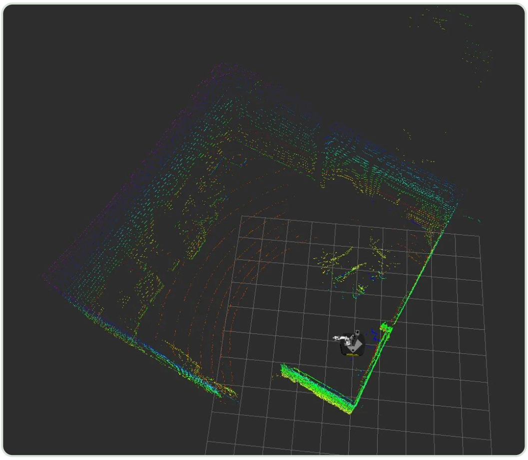
Making a Map

In this demonstration, Ridgeback generates a map using gmapping. Begin by launch the gmapping launch file on the robot:

```
roslaunch ridgeback_navigation gmapping_demo.launch
```

And on your workstation, launch rviz with the suggested configuration:

```
roslaunch ridgeback_viz view_robot.launch config:=gmapping
```



Rviz with Ridgeback's gmapping configuration.

You must slowly drive Ridgeback around to build the map. As obstacles come into view of the laser scanner, they will be added to the map, which is shown in rviz. You can either drive manually using the interactive markers, or semi-autonomously by sending navigation goals (as above).

When you're satisfied, you can save the produced map using map_saver :

```
rosrun map_server map_saver -f mymap
```

This will create a mymap.yaml and mymap.pgm file in your current directory.

Navigation With a Map

Using roswiki amcl , Ridgeback is able to globally localize itself in a known map. AMCL takes in information from odometry, laser scanner and an existing map and estimates the robot's pose.

To start the AMCL demo:

```
roslaunch ridgeback_navigation amcl_demo.launch [map_file:=/path/to/my/map.yaml]
```

If you don't specify map_file , it defaults to an included pre-made map of the default "Ridgeback Race" environment which Ridgeback's simulator spawns in. If you're using a real Ridgeback in your own environment, you'll definitely want to override this with the map created using the gmapping demo.

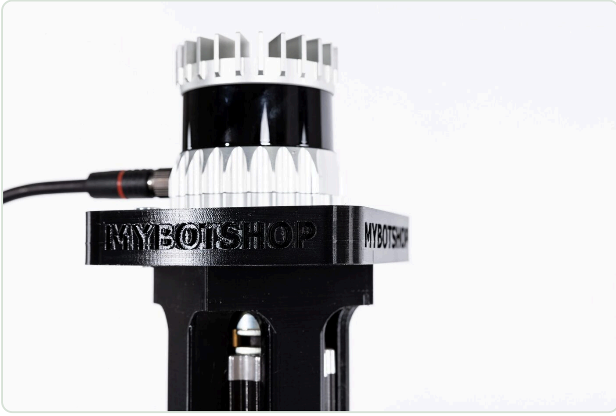
Before navigating, you need to initialize the localization system by setting the pose of the robot in the map. This can be done using 2D Pose Estimate in rviz or by setting the amcl initial_pose parameters. To visualize with the suggested rviz configuration launch:

```
roslaunch ridgeback_viz view_robot.launch config:=localization
```

When rviz appears, select the Set 2D Pose tool from the toolbar, and click on the map to indicate to the robot approximately where it is.

MMP Lidars

Important: This is applicable to the MMP Ridgeback Panda v1.2 (released prior to 2023).

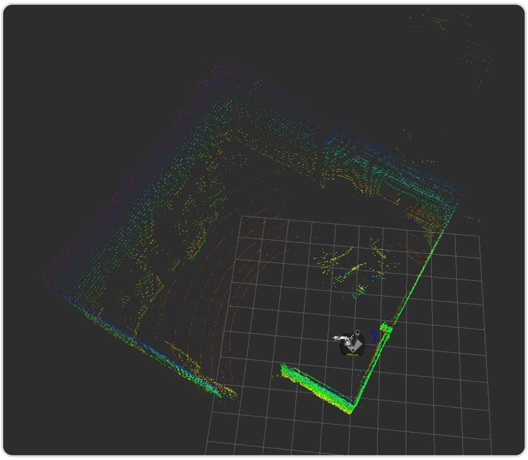


lift

MBS Ouster Package

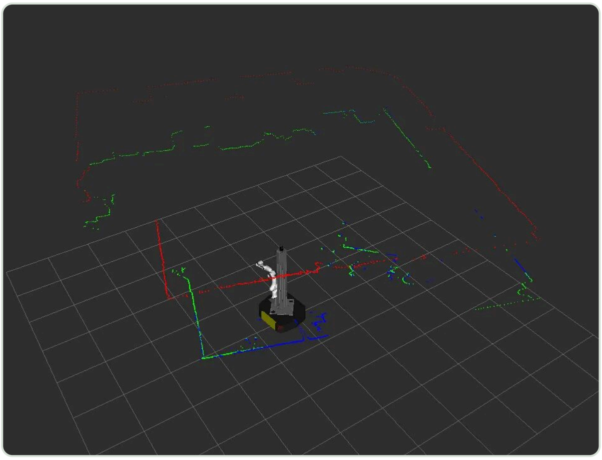
The mbs_rp_ouster package cleanly manage the nodes/data/logs of the ouster sensor. Ouster is mounted on the top and provides 3d pointcloud of surroundings. So this packages provides two types of data:

3D pointcould: is provided on /points topic.



lift

2D pointcloud: is the down-sampled data of the pointcloud topic and is provided on /top/scan` topic.



lift

11 Configuration: MMP BI

MMP BI

_images/mmp_bi_2.webp

This specifies the configuration for Mobile Manipulation Platform (MMP) BI version. It has the following components Custom Lift System ,Stepper Motor, Rope Encoder, Ouster lidar, Intel realsense D455 and D405 depth cameras, Universal Robot UR5e and Robotiq 2F-140 gripper.

Data Sheets

- Ridgeback Data Sheet
- MP1-11TGS (Main PC) Data Sheet
- Telefonica RUT240 (WiFi Router) Data Sheet
- Fixposition Data Sheet
- Ouster Data Sheets
- Hokuyo Data Sheets
- RealSense Data Sheet
- Robotiq Force Torque Sensor Data Sheet
- Robotiq gripper Data Sheet
- Universal Robots UR5e Data Sheet

Documentations

- Ouster ROS Docs
- RealSense SDK Docs
- Robotiq Force Torque sensor Docs
- Robotiq gripper Docs
- Universal Robots UR5e Docs

Powering on the Ridgeback

Power Button	E-reset Button	Release Switch

- Ensure the emergency buttons on all 4 corners are released. ◦ To check if the emergency switch is released, twist the switch in the indicated direction on it. If it pops up, then the switch has successfully been released.
- Press the power button on the rear of the ridgeback. It has the power symbol.
- Once the ridgeback is powered on and all the switches have been released, press the e-reset button . It is on the rear where the ridgeback has a yellow flashing light.

Powering on the UR5e

_images/ur5e_hmi.webp

UR5e should be in, i.e. it is in remote mode and on power off state

- The UR5e should only be powered on when the ridgeback is fully powered on (i.e. the ridgeback has white lights infront and/or the yellow light is flashing for the e-reset).
- The UR5e can be powered on via the teach pendants power button.
- It may take upto a few minutes for the UR5e to startup.
- Ensure all the emergency switches are released (i.e. on the teach pendant as well as on the lift).

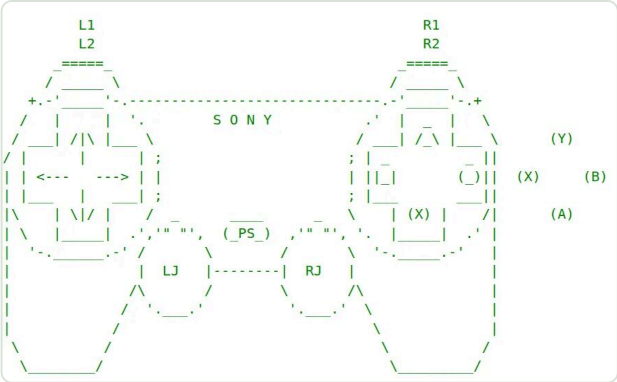
Teach pendant Switch Released	Custom Emergency Switch Released

Warning: Do not start anything else in the UR5e as this is all done automatically via ROS. Powering on is only required.

Powering Auxiliary Components

On startup the ridgeback launches, the lift drivers, and the Hokuyos. The auxiliary components such as the ouster, dynamixels, realsense cameras, nvidia, etc. are not powered on startup automatically but can be added via the startup launch file if required.

Teleop Control Joystick — PS4 Controller



_images/ps4_controller.webp

PS4 Controller

1. PS button is to power on.
2. X + Up arrow is for moving Lift up. (X is the square icon)
3. X + Down arrow is for moving Lift down. (X is the square icon)
4. B button for releasing Lift soft locks. (B is the circle icon)
5. L1 is the dead man's switch.
6. LJ is to move in x and y direction.
7. RJ is to move in the yaw orientation.

Teleop Control Joystick — Steamdeck Controller

> **Warning:** The Steamdeck usage is in experimental phase. Loss of connection or (laggy connection) may lead to the robot moving and/or not stopping immediately. Caution should be taken when using the Steamdeck for the operation of the robot!

Steamdeck - Basic Controls

1. Power button is on the top of the Steamdeck with the power symbol.
2. R2 for mouse left-click
3. L2 for mouse right-click
4. Left trackpad for mouse scrolling
5. Right trackpad for mouse movement
6. L2+R2 for middle mouse button click
7. The 3 dashes button is located on the top right of the Steamdeck next to the Y button, Hold for 3 seconds to gamepadmode and hold for 3 seconds again to go to desktop mode(Switching to desktop mode activates the cursor for the trackpad as well).

_images/steamdeck.webp

Steamdeck Controller

Steamdeck - Initialization with Robot

Before usage:

1. Ensure the Steamdeck is connected to the robot's WiFi access point.
2. Ensure the robot is powered on and operational.
3. Single tap the Joystick Launcher on the screen to activate the controller for the robot. (Do not open multiple instances of the joystick launchers, as this can cause zombie nodes).
4. Single tap the Rviz2 Launcher on the screen to visualize the robot.

Steamdeck - Robot Controls

1. L1 is the dead man's switch for slow speed.
2. R1 is the dead man's switch for high speed.
3. LJ is to move in x and y direction.
4. RJ is to move in yaw direction.
5. X+Up Arrow is to move lift up.
6. X+Down Arrow is to move lift down.
7. B is to release soft switches if the lift reaches maximum height.

Steamdeck - Rviz2 Controls

1. L2+R2 should be held when wanting to move the orbit in the xy plane.
2. R2 should be held to orbit around the orbits origin
3. Left trackpad can be moved up and down to zoom in and zoom out.

Manipulator Activation

Once the UR5e is powered on and its emergency switches are released.

- Then check if the UR5e is in remote mode (should be in remote mode by default) — Located in Teach pendant Top right .
- Then check if the UR5e is in Power off state with a icon — Located in Teach pendant Bottom left (Important) .

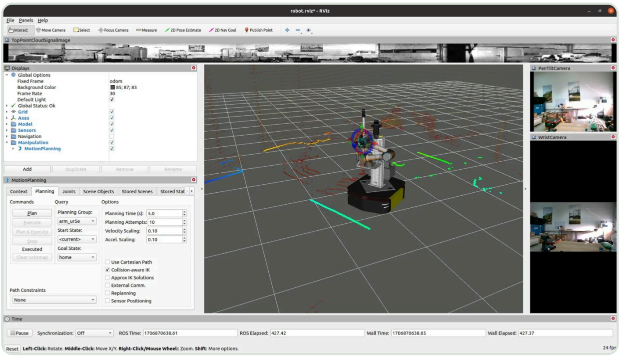
Auxiliary ROS Drivers

Once the manipulator is activated, next step is to launch the following driver:

```
roslaunch mmp_startup accessories.launch
```

This should turn on all the components including the UR5e, Robotiq 2F-85, Ouster, Realsense Cameras, and pan & tilt.

Digitial Twin (RVIZ)



_images/mmp_mip_rviz.webp

Visualizing the MMP in Rviz

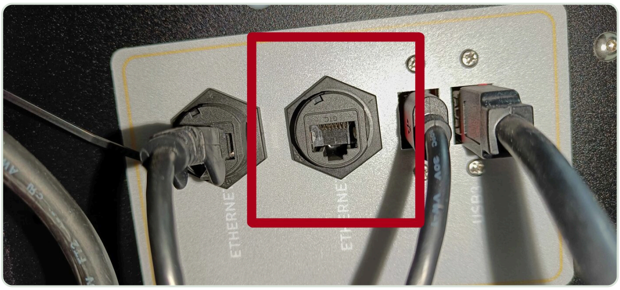
To visualize the mmp, you can run:

```
roslaunch mmp_viz view_robot.launch
```

Ideally, for latency-free streaming, please utilize ROS multimachine setup.

Robot Interface

Note: Instructions for interfacing with the robot using Ubuntu 20.04 and ROS Noetic .



_images/robot_interface.webp

The Ethernet port at the back of the Husky can be utilized for setting up communication via LAN.

Static Network connection

For the first time, one needs to connect through a LAN cable to configure the robot's WLAN network. To create a static connection in your PC (not the robots), in Ubuntu go to Settings → Network then click on + and create a new connection.

1. Change the connection to Manual in the IPv4 settings.
2. Set the Address IP as 192.168.131.51 and the Netmask as 24 .
3. Click save and restart your network.

After a successful connection, check the host's local IP by typing in the Host PC's terminal:

```
ifconfig
```

Now, ping the robot:

```
ping 192.168.131.1
```

Access the robot via SSH:

```
ssh -X administrator@192.168.131.1
```

The default password is:

```
mybotshop
```

Note: Sometimes other networks can cause disruptions when connecting to the MMP. It is best to have only your connection to the robot active and all others inactive.

Network table

All the devices on the robot are network enabled. Table [robot_network] outlines all the devices and their network addresses.

Device	Network Address	Password
Ridgeback	192.168.131.1	mybotshop
Ridgeback MCU	192.168.131.2	
Nvidia IP	192.168.131.4	mybotshop
UR5e IP	192.168.131.5	
Hokuyo Front	192.168.131.20	
Hokuyo Rear	192.168.131.21	
Steamdeck	192.168.131.150	

Ouster	192.168.131.22	
Router	192.168.131.100	

Screen Connection

An alternative for connecting to the robot is by plugging in an HDMI cable as well as a mouse and keyboard. This allows you to connect the robot from your own local network.

ROS Multimachine

To use Rviz on your pc for faster updates please setup the multimachine. By default, if you connect to the built-in router, you need only to follow the builtin router subsection.

Multimachine - Client - Builtin Router

For all the terminals in your pc, you may export these environment variables. The ros master should correspond to the robots ip where as the ros ip and ros hostname should correspond to your pc's ip that is connected to the robot in the same network.

```
export ROS_MASTER_URI = http://192.168.131.1:11311/
export ROS_IP =192.168.131. < your_assigned_ip_from_router >
export ROS_HOSTNAME =192.168.131. < your_assigned_ip_from_router >
```

Important: Please note that you should ideally have the mmp_description and mmp_viz rospackages built on your pc so that you may view the robot over the network.

Multi Machine - Host - Custom Router

For all the terminals in the robot 4 — 5 recommended, you can export the following environment variables. The ip should correspond to the ip address of your robot which has been obtained from sudo nmtui .

```
export ROS_MASTER_URI =http://192.168.0.219:11311/
export ROS_IP =192.168.0.219
export ROS_HOSTNAME =192.168.0.219
```

Multimachine - Client - Custom Router

For all the terminals in your pc, you may export these environment variables. The ros master should correspond to the robots ip where as the ros ip and ros hostname should correspond to your pc's ip that is connected to the robot in the same network.

```
export ROS_MASTER_URI =http://192.168.0.219:11311/
export ROS_IP =192.168.0.225
```

```
export ROS_HOSTNAME =192.168.0.225
```

Important: Please note that you should ideally have the mmp_description and mmp_viz rospackages built on your pc so that you may view the robot over the network.

Multimachine Usage

Stop the mmp service via (On the robot).

```
sudo service mmp stop
```

On the robot launch.

```
roslaunch mmp_startup system_startup.launch
```

On the users pc launch.

```
roslaunch mmp_viz view_robot.launch
```

Individual Component Activation

The ros drivers mentioned below are all automatically started when launching the accessories launch file. These drivers further expand on the functionality for interfacing as well as for independent component checking.

ROS Driver — UR5e + Robotiq 2F-85

Please follow this checklist to interact with the UR5e and Gripper via ROS.

- Power on teach pendant — In Teach pendant Top Center Right
- Release emergency switch — In Teach pendant Top Center
- UR5e is in remote mode — In Teach pendant Top right

Launch ROS driver:

```
roslaunch mmp_arm ur5e_robotiq_driver.launch
```

To test the gripper run:

```
rosrun mmp_demo gripper.py
```

To test the UR5e and gripper use moveit you can use Moveit :

```
roslaunch mmp_ur_moveit_config mmp_moveit.launch
```

ROS Driver — Lift

Note: The lift driver is on by default from the startup job.

Start the driver via:

```
roslaunch mmp_lift base.launch
```

Move lift up 1 to -600 microsteps

```
rostopic pub -r 1 /cmd_lift std_msgs/Float32 "data : -600.0"
```

Move lift down 1 to 600 microsteps

```
rostopic pub -r 1 /cmd_lift std_msgs /Float32 "data : 600.0"
```

Activate Lifts softlock, for the lift to move

```
rosservice call /get_lift_manual_control "{}"
```

ROS Driver — Ouster + Hokuyo Front + Hokuyo Rear

Note: The Hokuyos are on by default from the startup job.

To launch all three lidars run:

```
roslaunch mmp_lidars ouster_hokuyos.launch
```

To launch the hokuyos only (ideal when working with realsense):

```
roslaunch mmp_lidars hokuyos.launch
```

ROS Driver — Realsense Camera

Start the wrist camera:

```
roslaunch mmp_realsense_cameras d405.launch
```

Start both cameras:

```
roslaunch mmp_realsense_cameras realsense.launch
```

Note: When both the Ouster and Realsense cameras work at the same time, the latency of the camera stream increases due large amount of data transfer. Ideally, Ouster should be running or Realsense. Both can work at the same time but then Realsense suffer from latency issues.

Caution

- Power off the UR5e correctly via the teach pendant.
- Always remain close to the emergency stop buttons.
- Sudden jerks to the mobile robot can cause a disconnection, the velocity that is provided should always have be smooth either through the joystick, keyboard, or program.

Known Issues

- Ridgeback shuts down sometimes when powered on. ◦ In rare cases the network does not startup in time causing the ridgeback to shut down.Powercycling should resolve the issue.
- Realsense cameras lagging ◦ Utilize multi-machine setup. ◦ Ensure ouster is not running concurrently with the realsense as it takes up alot of bandwith which may be causing the lag.
- Lift joystick not working ◦ This occasionally occurs if you try to move the lift while releasing the soft limit switches. Restarting the mmp service will correct this.

12 Safety Guidelines

Robotic Manipulator Safety

When deploying robotic manipulators, it is imperative to prioritize safety procedures to mitigate risks and ensure secure operations. The following guidelines delineate key safety measures when working with robotic manipulators.

Work Area Safety

- Maintain a clean and well-organized work area. Cluttered or inadequately lit spaces can impede the proper functioning of sensors and hinder precise manipulation.
- Avoid operating robotic manipulators in hazardous environments, such as those containing corrosive substances, extreme temperatures, or sharp objects that may damage the manipulator.
- Ensure that only authorized personnel are present in the vicinity during manipulator operation to prevent interference and ensure a safe working environment.

Electrical Safety

- Ensure the manipulator's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement safeguards to protect the manipulator from adverse environmental conditions, such as exposure to moisture or extreme temperatures.
- Regularly inspect power cables and connections, promptly replacing damaged components to minimize the risk of electrical issues.

Manipulation Safety

- Implement collision detection systems to prevent unintended contact with objects, humans, or other equipment during manipulation tasks.
- Define and enforce safety zones around the manipulator's workspace to minimize the risk of unintended interactions with personnel or other objects.
- Regularly calibrate and test the manipulator's sensors and systems to ensure precise and reliable performance during manipulation tasks.

Emergency Response

- Install an emergency stop mechanism to swiftly halt manipulator operation in unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations within the manipulator's operational area.

- Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations during manipulator operation.

Data Security and Privacy

- Implement robust cybersecurity measures to safeguard the manipulator's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the manipulator's sensors.

Human Interaction Safety

- Integrate sensors and communication systems to detect and respond to the presence of humans in the manipulator's vicinity.
- Clearly communicate the manipulator's operational status and intentions using visual and audible signals to alert nearby individuals.
- Establish protocols for safe human-robot collaboration, particularly in shared workspaces where manipulators are in operation.

Residual Risks

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions during complex manipulation tasks.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

Robotic manipulators are sophisticated technologies that demand correct usage to avoid accidents and ensure a secure environment. Please adhere to proper procedures diligently, prioritizing both precision and safety.

Autonomous Robot Safety

When deploying autonomous mobile robots, prioritizing safety procedures is essential to prevent accidents and ensure secure operations. The following guidelines outline key safety measures when working with an autonomous mobile robot:

Work Area Safety

Maintain a clean and well-lit work area. Cluttered or poorly illuminated spaces can impede the proper functioning of sensors and navigation systems.

- Avoid operating autonomous mobile robots in explosive atmospheres, such as areas with flammable liquids, gases, or dust. The robot's components may pose a risk in such environments.

- Keep bystanders and unauthorized personnel at a safe distance during robot operation to prevent interference with autonomous navigation.

Electrical Safety

- Ensure the robot's power system adheres to electrical safety standards. Regularly inspect and maintain power components to prevent malfunctions.
- Implement mechanisms to protect the robot from adverse weather conditions, such as rain or wet environments.
- Regularly inspect power cables and connections and replace damaged components promptly to minimize the risk of electrical issues.

Navigation Safety

- Implement obstacle detection and avoidance systems to prevent collisions with objects, people, or other robots.
- Define and enforce safety zones within the robot's operational area to minimize the risk of unintended interactions with personnel or other equipment.
- Regularly calibrate and test the robot's navigation sensors and systems to ensure accurate and reliable performance.

Emergency Response

- Install an emergency stop mechanism to quickly halt the robot's operation in case of unforeseen circumstances or emergencies.
- Clearly mark and communicate emergency stop locations throughout the robot's operational area.
- Conduct regular emergency response drills to ensure personnel are familiar with procedures for handling unexpected situations.

Data Security and Privacy

- Implement robust cybersecurity measures to protect the robot's control systems and data from unauthorized access or manipulation.
- Ensure compliance with privacy regulations when collecting, storing, or transmitting data captured by the robot's sensors.

Human Interaction Safety

- Integrate sensors and communication systems to detect and respond to the presence of humans in the robot's vicinity.

- Clearly communicate the robot's operational status and intentions using visual and audible signals to alert nearby individuals.
- Establish protocols for safe human-robot collaboration, especially in shared workspaces.

Residual Risks

Despite the implementation of safety measures, certain residual risks may persist. These include:

- Impairment of sensor functionality.
- Risk of collisions in crowded or dynamic environments.
- Cybersecurity vulnerabilities.
- Unintended human interactions due to unforeseen circumstances.

Autonomous Mobile Robots (AMR) are advanced technologies that require correct usage to avoid accidents and ensure a secure environment. Learn and follow the proper procedures diligently; prioritize both quality and safety.

13 Data Sheets & Vendor Documentation

Data Sheets

- Ridgeback Data Sheet
- MP1-11TGS (Main PC) Data Sheet
- Telefonica RUT240 (WiFi Router) Data Sheet
- Fixposition Data Sheet
- Ouster Data Sheets
- Hokuyo Data Sheets
- Nvidia Jetson Orin 2GB
- RealSense Data Sheet
- Franka Emika Panda
- Ufactory xarm6 Data Sheet
- Robotiq Force Torque Sensor Data Sheet
- Robotiq gripper Data Sheet
- Universal Robots UR5e Data Sheet

Documentations

- Husky Noetic ROS Docs
- Ouster ROS Docs
- Nvidia Jetson Orin SDK Resources
- RealSense SDK Docs
- Ufactory xarm6 Documentation
- Ufactory xarm6 ROS Docs
- Robotiq Force Torque sensor Docs
- Robotiq gripper Docs
- Universal Robots UR5e Docs

14 Getting Help & RMA

RMA / Support

How to Raise a Return Merchandise Authorization (RMA) Case

At MYBOTSHOP GmbH, we are committed to providing excellent support to ensure your issues are resolved promptly. If you encounter any problems with our products, please follow the steps below to raise a Return Merchandise Authorization (RMA) case:

Step 1: Open a Topic in Our Forum

1. Visit MYBOTSHOP Forum .
2. Open a new topic describing your issue in detail. Our support team and community members will attempt to assist you in resolving the problem remotely.

Step 2: Contact Support for RMA Number

If the issue cannot be resolved remotely through our forum:

1. Email us at support @ mybotshop . de .
2. Provide a detailed description of the issue and include any troubleshooting steps already taken.
3. Our support team will evaluate your case and, if necessary, issue a RMA number along with a return address or a shipping label.

Important Shipping Information

Note: Please note: All goods must be sent back to the following address: MYBOTSHOP GmbH Willy - Messerschmitt - Strasse 12 50126 Bergheim Germany Do not ship goods without obtaining a RMA number. Parcels sent without a RMA number will be declined and returned to the sender.

Help Resources

In case of any issues, help can be taken from any of the following forums/resources:

1. MYBOTSHOP Forum : The forum is actively monitored by support teams at MYBOTSHOP GmbH .
2. Github Issues : Issues can be reported in the repository.
3. Online QA Sites: Questions can be asked on any of the QA sites like StackOverflow and Answers ROS .

15 FAQ

FAQs

Coming soon ...