MYBOTSHOP

ROBOTICS

Clearpath Jackal

User Manual

CONTACT

info@mybotshop.de

support@mybotshop.de

mybotshop.de

Contents

1 Attention

Attention: Read this document carefully before operating the robot. In case of doubt, consult the manufacturer (Clearpath Robotics) or the distributor (MYBOTSHOP GmbH) of the assemblies used. The robot must not be put into operation before clarification. Contact information is available in Chapter 10, Getting Help & RMA.

Do not allow persons who are not familiar with the robot or these instructions to operate it. Robots are dangerous in the hands of untrained users.

Do not operate the robot if you are intoxicated, under the influence of drugs, or unable to concentrate. Familiarize yourself with the battery, power and emergency-stop procedures in this manual before switching the robot on.

Caution: The Jackal is a wheeled, differential-drive ground vehicle intended for indoor and light outdoor research use. Keep a clear area around the robot while it is powered and under active control, and always confirm which ROS 2 distribution (Foxy, Humble or Jazzy) is installed on your unit before following a command in this manual — package names differ between distributions.

2 Disclaimer

The Clearpath Jackal documented here is sold and supported by *MYBOTSHOP GmbH* as the German distributor for Clearpath Robotics (Canada). Basic ROS 2 understanding is required to operate the software stack covered in this manual — if you are not familiar with ROS 2, we recommend checking the ROS 2 documentation first.

The client acknowledges and agrees that any information or materials provided by MYBOTSHOP GmbH are for R&D and development purposes only. Any services are provided “AS IS” and without any representation or warranty of any kind, express or implied, including but not limited to any warranty of merchantability, fitness for a particular purpose, non-infringement, or any other warranty.

This product is intended for civilian, indoor/light-outdoor research and prototyping use only. Users should avoid making dangerous modifications or using the robot in hazardous ways. This manual covers five hardware/software variants of the Jackal (BU, AAC, MGB, RS, SUL) across three ROS 2 distributions (Foxy, Humble, Jazzy) — always confirm which variant and distribution is installed on your unit before following a package name or launch command in this manual. For terms, policies and compliance with local laws and regulations, refer to the Clearpath Robotics website and the RMA/support process in Chapter 10.

MYBOTSHOP GmbH shall not be liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising out of or in connection with the use or inability to use the information or materials in this document. This limitation on liability shall apply regardless of the form of action, whether in contract, tort, or otherwise.

3 About the Clearpath Jackal

Clearpath Jackal

Attention: These tutorials assume that you are comfortable working with ROS2. We recommend starting with ROS2 tutorials if you are not familiar with ROS2 already.

The Jackal, a differential drive robot, is an indoor development platform created by Clearpath Robotics, a Canadian manufacturer. This documentation covers multiple ROS2 distributions to support various deployment scenarios.

Supported ROS2 Distributions:

- ROS2 Foxy (Ubuntu 20.04) - Legacy support
- ROS2 Humble (Ubuntu 22.04) - LTS release
- ROS2 Jazzy (Ubuntu 24.04) - Latest release (Recommended)

The Jackal robot is designed to serve as a versatile platform for indoor research and development. Its robust construction and flexible configuration options make it suitable for a wide range of applications, from academic research to industrial automation.

Key Features:

- Differential Drive: The Jackal's differential drive system allows for precise control and maneuverability, making it ideal for navigating complex indoor environments.
- Multi-ROS2 Support: Documentation includes support for Foxy, Humble, and Jazzy ROS2 distributions with synchronized tab navigation.
- Ouster LiDAR Integration: Equipped with an Ouster LiDAR sensor, the Jackal offers high-resolution 3D mapping and obstacle detection, crucial for accurate navigation and environment perception.
- ZED2i Stereo Camera: The addition of the ZED2i stereo camera enhances the Jackal's vision capabilities, providing depth perception and visual SLAM (Simultaneous Localization and Mapping) functionalities.
- Waypoint Navigation: The integrated waypoint navigation system allows the Jackal to autonomously navigate pre-defined paths, making it an excellent tool for autonomous indoor logistics and exploration.

Customization and Components:

The Jackal's current configuration is tailored to maximize its functionality and adaptability for advanced robotics applications. The key components include:

1. Ouster LiDAR: Mounted on the Jackal, the Ouster LiDAR provides detailed 3D point clouds, essential for real-time mapping and navigation in dynamic environments.

_images/ouster.webp

1. ZED2i Stereo Camera: This stereo camera system offers high-definition video and depth sensing, enhancing the robot's ability to perceive and interact with its surroundings.

_images/zed-2-front.webp

1. Enhanced Computing Power: The onboard PC has been upgraded to support the computational demands of ROS2, LiDAR processing, and visual SLAM algorithms.
2. Custom Mounts and Enclosures: Specialized mounts and enclosures have been designed to securely integrate the Ouster LiDAR and ZED2i camera, ensuring optimal performance and durability.

4 Quick Start – Powering On, Charging & the Controller

Powering on the Jackal

The Jackal is activated by pressing the designated power button. When initiating startup alongside the Ouster and Zed2i, it may require up to a minute to commence fully.

_images/jackal_power.webp

- The illuminated green light on the `M` signifies the successful initiation of the MCU, while the directional arrows indicate its readiness to receive commands from the onboard PC.
- This particular Jackal model is equipped with an Ouster LiDAR and Stereolab ZED2i camera. Instances, where the LiDAR or other supplementary components are integrated into the robot's startup procedure but encounter cable disconnection or damage, may result in startup failure.
- To enable software remote connection, initial access to the robot is necessary for configuring the WiFi network. Configuration can be accomplished through two methods: the first involves a static network connection, while the second, carried out via the screen interface, offers a quicker and more user-friendly setup for clients.

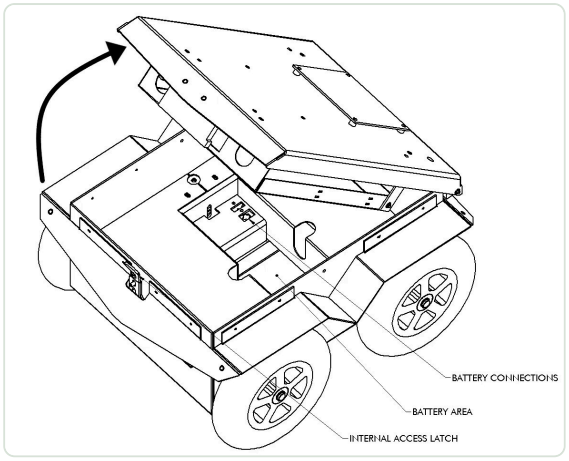
Charging the Jackal

Accessing the Battery

To access the battery, release the latches under the front end of Jackal's lid, opposite the HMI. Once the lid is lifted, you will see Jackal's onboard lithium battery and its two connectors:

- Anderson Power Pole Connector: Supplies power to Jackal and must be connected for operation.
- Molex Connector: Allows the battery to be charged internally while Jackal is powered off.

Anderson Power Pole Connector	Molex Connector

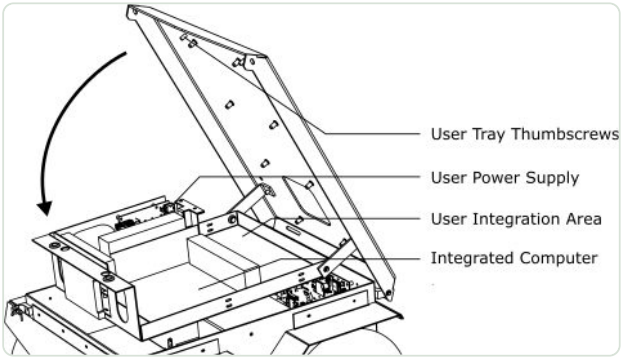


_images/battery-view.webp

Jackal's Battery Compartment

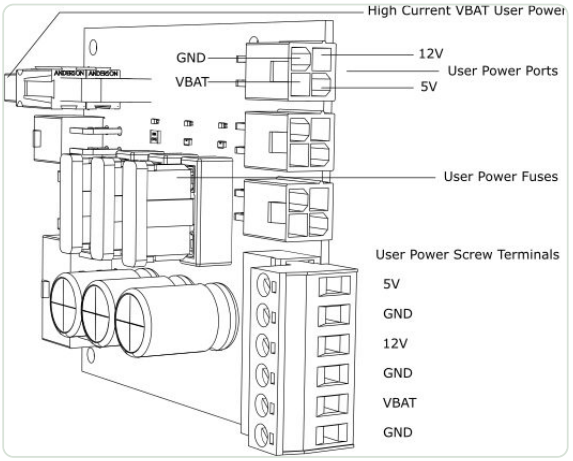
Charging Methods

1. Internal Charging: To charge the battery inside Jackal, plug the charger into the charge port located under the rear fender. Ensure Jackal is powered off during this process.
2. External Charging: If you have multiple batteries, you can remove the battery for external charging. When charging externally, disconnect the pigtail that adapts the charger to the platform's weather-sealed charge port.



_images/user-tray.webp

Computer and user tray



_images/user-power.webp

User power supply

Battery Safety and Maintenance

- The battery, manufactured by AllCell Technologies, includes safeguards against overcurrent, overdischarge, and short circuits.
- Do not operate or store the battery above +60°C or below -19°C.
- Do not puncture or disassemble the battery.
- Dispose of the battery at a local hazardous waste facility.

Logitech Controller

1. Logitech button is to power on the controller.
2. L1 is the dead man's switch for slow speed.
3. R1 is the dead man's switch for high speed.
4. LJ is to move in x and y direction.

5 Network & Robot Access

FOXY

Note: Instructions for interfacing with the robot using Ubuntu 20.04 LTS (Focal Fossa) and ROS 2 Foxy Fitzroy .

HUMBLE

Note: Instructions for interfacing with the robot using Ubuntu 22.04 LTS (Jammy Jellyfish) and ROS 2 Humble Hawksbill .

JAZZY

Note: Instructions for interfacing with the robot using Ubuntu 24.04 LTS (Noble Numbat) and ROS 2 Jazzy Jalisco .

Important: This procedure should be followed after reading the manuals for the Clearpath Jackal. Furthermore, all of the Jackal's functionality can be verified via the tutorials by Clearpath. Instructions for set up can be found at Jackal Docs .

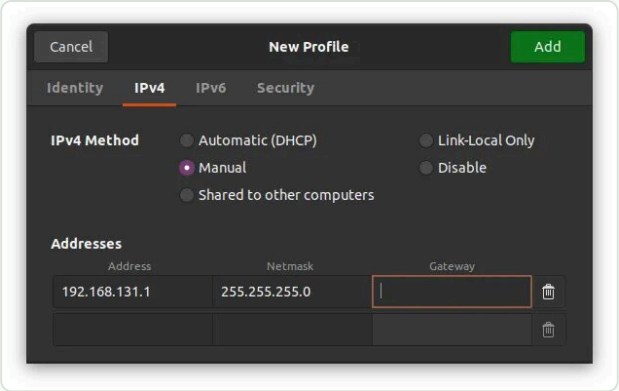
Network Interface

FOXY

Static Network Connection

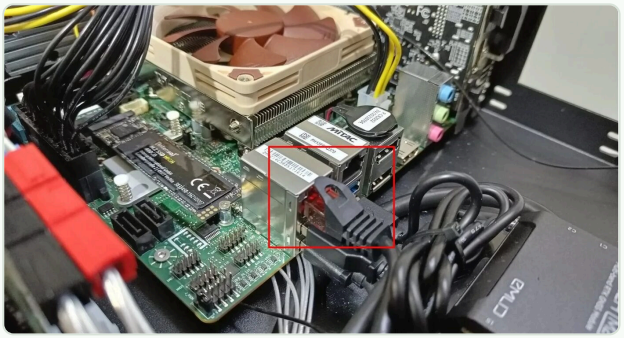
To create a static connection in your PC (not the robot), in Ubuntu go to Settings → Network then click on + and create a new connection.

1. Change the connection to Manual in the IPv4 settings.
2. Set the Address IP as 192.168.131.1 (note: this value might vary depending on the specific robot) and the Netmask as 24.
3. Click save and restart your network.



_images/jackal_network_1.webp

Once you've saved the settings and restarted your network, proceed by opening the Jackal and connecting the LAN cable to the robot.



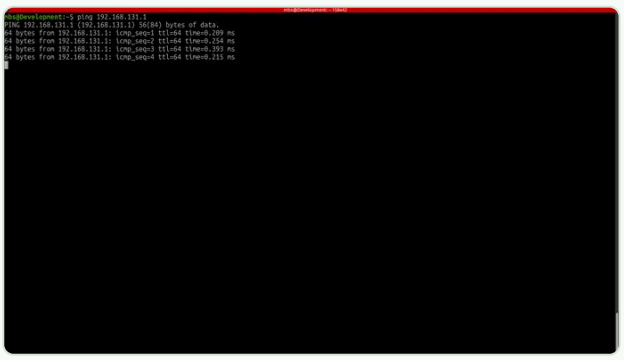
_images/jackal_lan.webp

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

```
ifconfig
```

This command will display the assigned host IP from the previous step. To check if we can communicate with the robot, attempt to ping it from your host PC by executing:

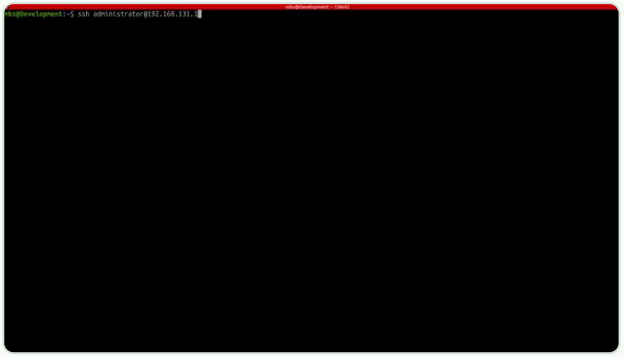
```
ping 192.168.131.1
```



_images/jackal_network_2.webp

After a successful ping, you can access the robot by using the following command:

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



_images/jackal_ssh.webp

The password required is:

```
mybotshop
```

Note: For Jackals operating on the default DHCP connections, there are a couple of approaches for connecting to the Jackal, one is described here. Alternative for Static Connection: You can choose not to set up a static connection and instead run the command nmap -sP 192.168.0.* . This command scans all available networks within the .0 range. After identifying the network associated with the Jackal, you can establish a connection using SSH. This alternative method allows you to locate the Jackal's network and subsequently connect to it via SSH without configuring a static connection on your system.

Network Configuration

The Jackal is equipped with an industrial-grade onboard computer. Specifically:

1. The MCU port functions on 192.168.131.1 .
2. The Ouster Lidar port operates on 192.168.132.20 .

Device	IP Address	Password
Jackal	192.168.131.1	
Ouster Lidar	192.168.132.20	

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

HUMBLE

Static Network Connection

For the first time, one needs to connect through a LAN cable to configure the robot's 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.

_images/jackal_lan.webp

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
```

Jackal IP Addresses

Device	IP Address	Password
Jackal Main PC	192.168.131.1	mybotshop
Jackal Ouster	192.168.131.20	X

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

JAZZY

Static Network Connection

For the first time, one needs to connect through a LAN cable to configure the robot's 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.

_images/jackal_lan.webp

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 robot@192.168.131.1
```

The default password is:

```
mybotshop
```

Jackal IP Addresses

Device	IP Address	Password
Jackal Main PC	192.168.131.1	mybotshop

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

Jackal Screen Connection

An alternative method to connect with the Jackal is by utilizing an HDMI cable along with a mouse and keyboard. This setup facilitates the connection of the Jackal to your local WiFi network, enabling subsequent connections via WiFi.

_images/jackal_hdmi.webp

Jackal PC Setup

To utilize this method:

1. Access your networks and add your WiFi connection.
2. Open the terminal and type:

```
ifconfig
```

Identify the IP that starts with wl (e.g., wlps0). This IP corresponds to your WiFi connection. Use this IP to SSH into your robot:

```
ssh -X administrator@192.168.0.228
```

Replace 192.168.0.228 with the actual IP obtained from the ifconfig command. This will establish an SSH connection to your robot.

> **Note:** The provided IP (192.168.0.228) is an example. The real IP is acquired from the ifconfig command output.

6 Hardware Variants – BU / AAC / MGB / RS / SUL

The Jackal ships as one of five hardware/software SKUs, sharing the same J100 base platform and Platform Overview below, but differing in the perception sensors fitted and the ROS 2 distribution installed. Confirm which variant you have before following a package name elsewhere in this manual.

Platform Overview: Jackal is a durable, lightweight, and fast unmanned ground robot designed for rapid prototyping and research purposes. It comes equipped with an onboard computer, as well as essential IMU and GPS capabilities. Standard perception tools are available, featuring URDF, simulator integration, and demonstration applications.

SKU	ROS 2 Distribution	Added Perception Hardware
Jackal BU	Humble	Base unit — no additional perception sensor
Jackal AAC	Humble	Robosense RS-LiDAR-M1, Fixposition Vision-RTK2 (GNSS/INS), Intel RealSense D455
Jackal MGB	Jazzy	ZED 2i stereo camera
Jackal RS	Humble	No separate datasheet published for this SKU on the docs site
Jackal SUL	Jazzy	Ouster LiDAR

Jackal SUL – Sensors & ROS 2 Package Reference

The Jackal SUL is the only variant with its own dedicated ROS 2 package reference and debugging-tools section in the source documentation, reproduced in full below.

Sensors

LiDAR - Ouster

The Ouster LiDAR is automatically started when launching the system launch file. The ROS2 driver expands functionality for interfacing and independent component checking.

_images/ouster.webp

Documentation: https://github.com/ouster-lidar/ouster-ros

Published Topics:

Topic	Type
/ouster/scan	sensor_msgs/LaserScan
/ouster/points	sensor_msgs/PointCloud2

Debugging Tools

Check Transforms (TF Tree)

To debug TF tree:

```
ros2 run rqt_gui rqt_gui tf:=/j100_0941/tf tf_static:=/j100_0941/tf_static
```

Visualize URDF

To view the robot's URDF model:

```
ros2 launch jkl_description view_robot.launch.py
```

Start-up Job Status

The Jackal uses Clearpath robot services for automatic launch of ROS2 nodes. To check service status:

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

Service Status Indicators:

- Red marker - The startup job has failed
- Green marker - Everything is working correctly
- Grey marker - The service has not started yet

ROS2 Package Reference

Detailed documentation for each package in the J100 workspace.

jkl_description

URDF and Xacro robot description files for the J100 Jackal platform.

Usage:

```
ros2 launch jkl_description view_robot.launch.py
```

jkl_gazebo

Gazebo simulation package for the J100.

Usage:

```
ros2 launch jkl_gazebo fortress.launch.py
```

jkl_navigation

Nav2-based navigation stack for the J100 platform.

Launch Files:

Launch File	Description
slam.launch.py	2D SLAM mapping mode
odom_navi.launch.py	Odometric navigation (no map)
localization.launch.py	Localization only (AMCL + Map Server)
map_navi.launch.py	Full autonomous navigation with map

jkl_3d_slam

3D LiDAR SLAM package using graph-based optimization with NDT scan matching and loop closure detection.

Components:

- scanmatcher_node - NDT-based scan matching for odometry
- graph_based_slam_node - Loop closure and graph optimization

Usage:

```
ros2 launch jkl_3d_slam system.launch.py
```

jkl_viz

RViz2 visualization package with pre-configured displays.

Usage:

```
ros2 launch jkl_viz view_robot.launch.py
```

Jackal MGB – Simulation, Teleoperation & Debugging Tools

Simulation & Teleoperation

Simulation

To launch the Jackal simulation environment:

```
ros2 launch jkl_gazebo fortress.launch.py
```

RViz2 Visualization

To visualize the robot in RViz2:

```
ros2 launch jkl_viz view_robot.launch.py
```

Teleoperation (Keyboard)

To teleoperate the Jackal using your keyboard:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap
cmd_vel:=/j100_0892/cmd_vel
```

To enable stamped messages:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap
cmd_vel:=/j100_0892/cmd_vel --ros-args -p stamped:=true
```

Debugging Tools

Check Transforms (TF Tree)

To debug TF tree:

```
ros2 run rqt_gui rqt_gui tf:=/j100_0892/tf tf_static:=/j100_0892/tf_static
```

Visualize URDF

To view the robot's URDF model:

```
ros2 launch jkl_description view_robot.launch.py
```

Restart Clearpath Drivers

To restart Clearpath robot services:

```
sudo systemctl restart clearpath-robot.service
```

7 ROS 2 Software Operation

_images/jackal_1.webp

The Jackal drivers, set up by MYBOTSHOP , are stored within the ros2_ws . Specifically, in the src folder, you'll find the third_party section housing drivers designed for the Jackal and additional sensors such as the Ouster Lidar and ZED2 camera.

For configuring these auxiliary sensors, the link provided offers valuable information, particularly regarding navigation. It's important to note that while the setup maintains a default configuration, users have the flexibility to define their parameters within their custom package, as per their specific requirements.

Start-up Job

FOXY

Start-up Job

The default startup configuration for the Jackal is managed by the Clearpath startup job, unless explicitly specified otherwise. If there are any concerns regarding the robot's startup, SSH into the Jackal allows you to verify the status of the startup job:

```
sudo service ros2 status
```

Here's what the different markers signify:

- Red Marker: Indicates a failed startup job.
- Green Marker: Indicates normal functioning.
- Grey Marker: Indicates the service hasn't started yet.

In case of a red or grey marker, you can attempt to restart the service using:

```
sudo service ros2 restart
```

HUMBLE

Start-up Job

The Jackal uses MBS startup services for automatic launch of ROS2 nodes. If there are any concerns regarding the robot's startup, SSH into the Jackal allows you to verify the status of the startup job:

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

Here's what the different markers signify:

- Red Marker: Indicates a failed startup job.
- Green Marker: Indicates normal functioning.
- Grey Marker: Indicates the service hasn't started yet.

In case of a red or grey marker, you can attempt to restart the service using:

```
sudo systemctl restart clearpath-robot.service
```

Ouster Service

The Ouster LiDAR is managed by its own service:

```
sudo service mbs-ouster status
```

JAZZY

Start-up Job

The Jackal uses Clearpath robot services for automatic launch of ROS2 nodes. If there are any concerns regarding the robot's startup, SSH into the Jackal allows you to verify the status of the startup job:

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

Here's what the different markers signify:

- Red Marker: Indicates a failed startup job.
- Green Marker: Indicates normal functioning.
- Grey Marker: Indicates the service hasn't started yet.

In case of a red or grey marker, you can attempt to restart the service using:

```
sudo systemctl restart clearpath-robot.service
```

Auxiliary Sensors

FOXY

Auxiliary Sensors

To start the ouster lidar as well as the zed2i, you can launch it via:

```
ros2 launch jkl_sensors system.launch.py
```

Ouster Lidar

_images/ouster.webp

Start the Ouster Lidar driver only using:

```
ros2 launch jkl_sensors ouster.launch.py
```

Configuration settings are available within the jkl_sensors/ouster.launch.py file.

ZED2i

_images/zed-2-front.webp

The ZED2i camera employs an Nvidia Graphics card integrated into the onboard PC.

_images/jackal_graphic_card.webp

To start the Zed2i camera only:

```
ros2 launch jkl_sensors zed2i.launch.py
```

Ensure the camera is connected via USB3.0 . If disconnected, reattach it; otherwise, the camera may not be detected.

For usage instructions, refer to the ZED2 ROS2 Tutorial .

Configuration settings are available within the jkl_sensors/zed2i.launch.py file.

HUMBLE

Auxiliary Sensors

The ROS2 drivers are automatically started when launching the system launch file. These drivers further expand on the functionality for interfacing as well as for independent component checking.

Ouster Lidar

_images/ouster.webp

The Ouster is on by default from the startup job and it corresponds to the jkl_lidars/launch/include/ouster.launch.py .

```
sudo service mbs-ouster status
```

ZED2i

_images/zed-2-front.webp

The ZED2i camera employs an Nvidia Graphics card integrated into the onboard PC.

For usage instructions, refer to the ZED2 ROS2 Tutorial .

JAZZY

Auxiliary Sensors

The ROS2 drivers are automatically started when launching the system launch file. These drivers further expand on the functionality for interfacing as well as for independent component checking.

Ouster Lidar

_images/ouster.webp

For Ouster documentation, see the Ouster ROS2 Docs .

ZED2i

_images/zed-2-front.webp

For usage instructions, refer to the ZED2 ROS2 Tutorial .

Autonomous Navigation

FOXY

Odom Navigation

One can easily initiate basic navigation using the built-in odometry of the Jackal by executing the command:

```
ros2 launch jkl_navigation odom_navi.launch.py
```

This command lets you navigate the robot in the global frame of map

Map Navigation

To initiate map-based navigation, you need to first create a map of the environment and then perform waypoint navigation. Follow these steps:

1. Create a Map Start by executing the SLAM navigation command: ros2 launch jkl_navigation slam.launch.py In a separate terminal, launch the visualization command to observe the map being constructed: ros2 launch jkl_viz view_robot.launch.py Use the joystick controller to navigate and build the map. Once you are satisfied with the map, export it with the following command: ros2 run nav2_map_server map_saver_cli -f /home/administrator/ros2_ws/src/mbs/jkl_navigation/maps/custom_map
2. Rebuild Packages After saving the map, rebuild the packages to incorporate the new map: colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE = Release
3. Load and Visualize the Map Once rebuilding is complete, load the saved map with the map navigation command: ros2 launch jkl_navigation map_navi.launch.py In a separate terminal, execute the visualization command to visualize and control the robot using rviz: ros2 launch jkl_viz view_robot.launch.py

Debugging Controller

If you encounter issues with the joystick, SSH into the robot and teleoperate it using:

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

This arrangement provides a clear breakdown of steps and commands for map navigation and joystick debugging, ensuring efficient operation of the Jackal.

HUMBLE

Odom Navigation

Navigate in the global frame of odom :

```
ros2 launch jkl_nav2 odom_navi.launch.py
```

Odom Based Navigation: After launching the drivers and commands, providing a 2D navigation goal in RViz initiates the navigation process. The brown arrow indicates the 2D navigation goal, while the small orange line represents the global planner and the green arrows depict the local planner. The green box surrounding the robot signifies the bounding box. In the map, blue areas represent the local cost map, and the grey area corresponds to the global cost map.

Mapping

1. Mapping can begin using the joystick controller :

```
ros2 launch jkl_nav2 slam.launch.py
```

1. Once you are satisfied with your map you can export it by running the following command in one of the Jackals terminals:

```
ros2 run nav2_map_server map_saver_cli -f
/home/administrator/ros2_ws/src/mybotshop/jkl_nav2/maps/custom_map
```

1. Rebuild the Jackal nav2 so that the maps can be installed in the install folder:

```
colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE=Release && source
install/setup.bash
```

Note: The rebuilding step is not required if the filename is not changed.

Map Navigation

```
ros2 launch jkl_nav2 map_navi.launch.py
```

Map Based Navigation: After launching the drivers and commands, providing a 2D pose estimate in RViz in the Map frame initiates map navigation.

JAZZY

SLAM

Run SLAM navigation:

```
ros2 launch jkl_nav2 slam.launch.py
```

Odom Navigation

Run odometry-based navigation:

```
ros2 launch jkl_nav2 odom_navi.launch.py
```

Map-Based Navigation

Use a pre-built map for navigation:

```
ros2 launch jkl_nav2 map_navi.launch.py
```

Visualization

FOXY

Digital Twin (RVIZ2)

To visualize the Jackal, you can run:

```
ros2 launch jkl_viz view_robot.launch.py
```

HUMBLE

Digital Twin (RVIZ2)

To visualize the Jackal, you can run:

```
ros2 launch jkl_viz view_robot.launch
```

ROS2 Information

1. ROS DOMAIN ID: 0
2. RMW Middleware: rmw fastrtps cpp

JAZZY

Digital Twin (RVIZ2)

To visualize the Jackal, you can run:

```
ros2 launch jkl_viz view_robot.launch.py
```

ROS2 Information

1. ROS Version: Jazzy
2. Simulation: Ignition Fortress
3. Robot Namespace: /j100_0892
4. Middleware: Default RMW (FastRTPS or CycloneDDS, depending on environment)

Simulation

FOXY

Coming Soon

Simulation support for ROS2 Foxy is available via Gazebo Classic.

For simulation instructions, refer to the Jackal Simulation documentation.

HUMBLE

Coming Soon

Simulation support for ROS2 Humble is available via Gazebo Classic.

For simulation instructions, refer to the Jackal Simulation documentation.

JAZZY

Gazebo Simulation

To launch the Jackal simulation environment:

```
ros2 launch jkl_gazebo fortress.launch.py
```

RViz2 Visualization

To visualize the robot in RViz2:

```
ros2 launch jkl_viz view_robot.launch.py
```

Teleoperation (Keyboard)

To teleoperate the Jackal using your keyboard:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap
cmd_vel:=/j100_0892/cmd_vel
```

To enable stamped messages:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap
cmd_vel:=/j100_0892/cmd_vel --ros-args -p stamped:=true
```

Debugging

FOXY

Debugging Controller

If you encounter issues with the joystick, SSH into the robot and teleoperate it using:

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

Restart Services

To restart the ROS2 service:

```
sudo service ros2 restart
```

HUMBLE

Debugging Controller

If you encounter issues with the joystick, SSH into the robot and teleoperate it using:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Restart Services

To restart Clearpath robot services:

```
sudo systemctl restart clearpath-robot.service
```

JAZZY

Check Transforms (TF Tree)

To debug TF tree:

```
ros2 run rqt_gui rqt_gui tf:=/j100_0892/tf tf_static:=/j100_0892/tf_static
```

Visualize URDF

To view the robot's URDF model:

```
ros2 launch jkl_description view_robot.launch.py
```

Restart Clearpath Drivers

To restart Clearpath robot services:

```
sudo systemctl restart clearpath-robot.service
```

8 Autonomous Mobile Robot Safety Guidelines

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.

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.

9 Reference – Data Sheets, Documentation & ROS 2 Installation

Data Sheets

- ROS2 Cheat Sheet
- Jackal Specification Sheet
- Ouster Data Sheets

Documentations

- Jackal User Manual
- Jackal Docs
- Ouster ROS Docs
- Jackal Github

FOXY

- Jackal ROS2 Foxy Tutorials
- Ouster ROS2 Foxy Docs
- Jackal Navigation Foxy
- Jackal Simulation Foxy

HUMBLE

- Jackal ROS2 Humble Tutorials
- Ouster ROS2 Humble Docs
- Jackal Navigation Humble
- Jackal Simulation Humble

JAZZY

- Jackal ROS2 Jazzy Tutorials
- Ouster ROS2 Jazzy Docs

ROS2 Installation

FOXY

ROS2 Foxy Installation

Warning: Care should be taken when installing ROS so that it matches with your CPU's architecture (i.e. armhf, amd64, arm6, etc.)

Note: All the ROS related software should be installed/run on remote PC/Raspberry Pi/Nvidia board, nothing needed to be installed on robots control board if bought with the ROS pkg.

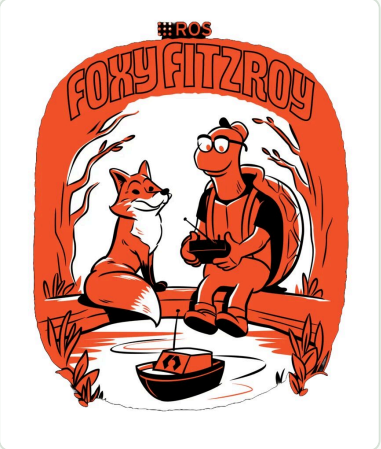
PC Setup

ROS2 Foxy installation requires Ubuntu 20.04 .

1. Download Ubuntu 20.04
2. Follow the Ubuntu 20.04 installation guide

ROS Foxy Installation

The ROS driver provided can be run either on a remote PC or on board robot computer. Typically, the on-board computer already has a pre-installed ROS distribution, so the instructions below will be applicable to a remote computer. You may execute the following commands to install ROS Foxy, or you can simply follow the instructions from the ROS Wiki.



_images/foxy.webp

Set Locale

Ensure that your system supports UTF-8 locale. Verify your locale settings with the following command:

```
locale
```

If necessary, set up a UTF-8 locale:

```
sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
```

Setup Sources

Add the ROS2 apt repository to your system. First, make sure that the Ubuntu Universe repository is enabled:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
```

Next, add the ROS2 GPG key with apt:

```
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

Then, add the repository to your sources list:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(./etc/os-release && echo
$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

Install ROS2 Packages

Update your apt repository caches after setting up the repositories:

```
sudo apt update
```

Ensure your system is up to date before installing new packages:

```
sudo apt upgrade
```

Desktop Install (Recommended)

Install ROS, RViz, demos, and tutorials:

```
sudo apt install ros-foxy-desktop python3-argcomplete
```

ROS-Base Install (Bare Bones)

Install communication libraries, message packages, and command line tools without GUI tools:

```
sudo apt install ros-foxy-ros-base python3-argcomplete
```

Development Tools

Install compilers and other tools to build ROS packages:

```
sudo apt install ros-dev-tools
```

Environment Setup

Source the setup script to set up your environment:

```
# Replace ".bash" with your shell if you're not using bash
# Possible values are: setup.bash, setup.sh, setup.zsh
source /opt/ros/foxy/setup.bash
```

Try Some Examples

If you installed ros-foxy-desktop above, try some examples:

Run a C++ talker in one terminal:

```
source /opt/ros/foxy/setup.bash
ros2 run demo_nodes_cpp talker
```

Run a Python listener in another terminal:

```
source /opt/ros/foxy/setup.bash
ros2 run demo_nodes_py listener
```

This verifies both the C++ and Python APIs are working properly.

Next Steps After Installing

Continue with the tutorials and demos to configure your environment, create your own workspace and packages, and learn ROS2 core concepts.

Using the ROS1 Bridge

Connect topics from ROS1 to ROS2 and vice-versa using the ROS1 bridge. See the dedicated documentation on how to build and use the ROS1 bridge.

Additional RMW Implementations (Optional)

ROS2 uses Fast-RTPS as the default middleware (RMW), but you can replace it at runtime. See the guide on how to work with multiple RMWs.

Troubleshooting

Refer to the troubleshooting techniques found here if you encounter any issues.

Uninstall

If you need to uninstall ROS2 or switch to a source-based install once you have already installed from binaries, run the following command:

```
sudo apt remove ~nros-foxy-* && sudo apt autoremove
```

You may also want to remove the repository:

```
sudo rm /etc/apt/sources.list.d/ros2.list
sudo apt update
sudo apt autoremove
# Consider upgrading for packages previously shadowed.
sudo apt upgrade
```

HUMBLE

ROS2 Humble Installation

Warning: Care should be taken when installing ROS so that it matches with your CPU's architecture (i.e. armhf, amd64, arm6, etc.)

Note: All the ROS related software should be installed/run on remote PC/Raspberry Pi/Nvidia board, nothing needed to be installed on robots control board if bought with the ROS pkg.

PC Setup

ROS2 Humble installation requires Ubuntu 22.04 .

1. Download Ubuntu 22.04
2. Follow the Ubuntu 22.04 installation guide

ROS Humble Installation

The ROS driver provided can be run either on a remote PC or on board robot computer. Typically, the on-board computer already has a pre-installed ROS distribution, so the instructions below will be applicable to a remote computer. You may execute the following commands to install ROS Humble, or you can simply follow the instructions from the ROS Wiki.

Set Locale

Ensure that your system supports UTF-8 locale. Verify your locale settings with the following command:

```
locale
```

If necessary, set up a UTF-8 locale:

```
sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
```

Setup Sources

Add the ROS2 apt repository to your system. First, make sure that the Ubuntu Universe repository is enabled:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
```

Next, add the ROS2 GPG key with apt:

```
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

Then, add the repository to your sources list:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo
$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

Install ROS2 Packages

Update your apt repository caches after setting up the repositories:

```
sudo apt update
```

Ensure your system is up to date before installing new packages:

```
sudo apt upgrade
```

Desktop Install (Recommended)

Install ROS, RViz, demos, and tutorials:

```
sudo apt install ros-humble-desktop python3-argcomplete
```

ROS-Base Install (Bare Bones)

Install communication libraries, message packages, and command line tools without GUI tools:

```
sudo apt install ros-humble-ros-base python3-argcomplete
```

Development Tools

Install compilers and other tools to build ROS packages:

```
sudo apt install ros-dev-tools
```

Environment Setup

Source the setup script to set up your environment:

```
# Replace ".bash" with your shell if you're not using bash
# Possible values are: setup.bash, setup.sh, setup.zsh
source /opt/ros/humble/setup.bash
```

Try Some Examples

Run a C++ talker in one terminal:

```
source /opt/ros/humble/setup.bash
ros2 run demo_nodes_cpp talker
```

Run a Python listener in another terminal:

```
source /opt/ros/humble/setup.bash
ros2 run demo_nodes_py listener
```

This verifies both the C++ and Python APIs are working properly.

Next Steps After Installing

Continue with the tutorials and demos to configure your environment, create your own workspace and packages, and learn ROS2 core concepts.

Troubleshooting

Refer to the troubleshooting techniques found here if you encounter any issues.

Uninstall

If you need to uninstall ROS2 or switch to a source-based install once you have already installed from binaries, run the following command:

```
sudo apt remove ~nros-humble-* && sudo apt autoremove
```

You may also want to remove the repository:

```
sudo rm /etc/apt/sources.list.d/ros2.list
sudo apt update
sudo apt autoremove
# Consider upgrading for packages previously shadowed.
sudo apt upgrade
```

JAZZY

ROS2 Jazzy Installation

Warning: Care should be taken when installing ROS so that it matches with your CPU's architecture (i.e. armhf, amd64, arm6, etc.)

Note: All the ROS related software should be installed/run on remote PC/Raspberry Pi/Nvidia board, nothing needed to be installed on robots control board if bought with the ROS pkg.

PC Setup

ROS2 Jazzy installation requires Ubuntu 24.04 .

1. Download Ubuntu 24.04
2. Follow the Ubuntu 24.04 installation guide

ROS Jazzy Installation

The ROS driver provided can be run either on a remote PC or on board robot computer. Typically, the on-board computer already has a pre-installed ROS distribution, so the instructions below will be applicable to a remote computer. You may execute the following commands to install ROS Jazzy, or you can simply follow the instructions from the ROS Wiki.

Set Locale

Ensure that your system supports UTF-8 locale. Verify your locale settings with the following command:

```
locale
```

If necessary, set up a UTF-8 locale:

```
sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
```

Setup Sources

Add the ROS2 apt repository to your system. First, make sure that the Ubuntu Universe repository is enabled:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
```

Next, add the ROS2 GPG key with apt:

```
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

Then, add the repository to your sources list:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo
$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

Install ROS2 Packages

Update your apt repository caches after setting up the repositories:

```
sudo apt update
```

Ensure your system is up to date before installing new packages:

```
sudo apt upgrade
```

Desktop Install (Recommended)

Install ROS, RViz, demos, and tutorials:

```
sudo apt install ros-jazzy-desktop python3-argcomplete
```

ROS-Base Install (Bare Bones)

Install communication libraries, message packages, and command line tools without GUI tools:

```
sudo apt install ros-jazzy-ros-base python3-argcomplete
```

Development Tools

Install compilers and other tools to build ROS packages:

```
sudo apt install ros-dev-tools
```

Environment Setup

Source the setup script to set up your environment:

```
# Replace ".bash" with your shell if you're not using bash
# Possible values are: setup.bash, setup.sh, setup.zsh
source /opt/ros/jazzy/setup.bash
```

Try Some Examples

Run a C++ talker in one terminal:

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

```
ros2 run demo_nodes_cpp talker
```

Run a Python listener in another terminal:

```
source /opt/ros/jazzy/setup.bash
ros2 run demo_nodes_py listener
```

This verifies both the C++ and Python APIs are working properly.

Next Steps After Installing

Continue with the tutorials and demos to configure your environment, create your own workspace and packages, and learn ROS2 core concepts.

Troubleshooting

Refer to the troubleshooting techniques found here if you encounter any issues.

Uninstall

If you need to uninstall ROS2 or switch to a source-based install once you have already installed from binaries, run the following command:

```
sudo apt remove ~nros-jazzy-* && sudo apt autoremove
```

You may also want to remove the repository:

```
sudo rm /etc/apt/sources.list.d/ros2.list
sudo apt update
sudo apt autoremove
# Consider upgrading for packages previously shadowed.
sudo apt upgrade
```

10 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 .