MYBOTSHOP

ROBOTICS

# AgileX Hunter 2.0

## User Manual

CONTACT

info@mybotshop.de
support@mybotshop.de
mybotshop.de

# Contents

# 1 Attention

**Attention:** Read this document carefully. In case of doubt, it is essential to consult specialists, experts, or the manufacturer (AgileX Robotics) or distributor (MYBOTSHOP) 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 emergency stop procedure (Chapter 5, Safety) before switching the robot on.

**Caution:** The Hunter 2.0 is an Ackermann steering ground vehicle capable of speeds up to 1.5 m/s (4.8 m/s on the Hunter SE variant). Keep hands and body clear of the steering mechanism and wheels during operation, and maintain a safe distance during autonomous or remote-controlled driving.

# 2 Disclaimer

The AgileX Hunter 2.0 robot documented here is sold and supported by *MYBOTSHOP GmbH* as a distributor for AgileX Robotics. 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 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 and research applications only. Users should avoid making dangerous modifications to the electrical or mechanical systems, or using the robot in hazardous ways. This manual documents both the base Hunter 2.0 platform and the HUNTER RO research configuration (Chapter 9) — the HUNTER RO kit is an R&D device and does not carry CE marking. For terms, policies and compliance with local laws and regulations, refer to the AgileX 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 Hunter 2.0

## Overview

### AgileX Hunter 2.0

The AgileX Hunter 2.0 is specifically developed to excel in low-speed autonomous driving scenarios. Equipped with front-wheel Ackerman steering and rocker suspension, it can effectively navigate obstacles encountered on its path. The new-generation power system enhances payload capacity and extends endurance capabilities for superior performance.

*_images/agile_hunter_1.webp*

### Core Capabilities

- Payload Capacity: 150 kg
- Maximum Runtime: 240 minutes
- Maximum Speed: 1.5 m/s

### Features

Excellent Payload Capacity Hunter 2.0's high load capacity enhances productivity, adaptability, worker safety, accuracy, and system integration. It effortlessly handles heavy objects, boosting production, reducing costs, and improving working conditions.

High Endurance Scenarios Hunter 2.0 is ideal for high-endurance situations with its car-like chassis, reduced tire wear, front-wheel steering, and rocker suspension. It is perfect for navigating speed bumps and other obstacles.

Adaptable and Scalable The robot's modular design allows for easy reconfiguration and reassembly, making it adaptable to different task requirements. Hunter 2.0 can perform a wide variety of tasks.

Expansion and Secondary Development Hunter 2.0 provides aluminum T-slot rails for secure mounting of external sensors or kits. It supports CAN bus protocol and offers open-source SDK and software resources for expanded capabilities.

## Technical Specifications

Dimensions: 980 x 745 x 380 mm Platform Weight: 65–72 kg Maximum Payload: 150 kg Climbing Ability: ≤ 10° (with load) IP Rating: IP22 Communication: Standard CAN, 232 Serial port

Performance

- Maximum Speed: 1.5 m/s
- Maximum Travel Distance: 40 km
- Obstacle Clearance: 50 mm

## Power and Endurance

The new-generation power system allows Hunter 2.0 to achieve up to 240 minutes of continuous runtime, making it highly suitable for prolonged operational tasks in diverse environments.

## Navigation and Autonomy

With front-wheel Ackerman steering, rocker suspension, and robust mechanical design, Hunter 2.0 navigates obstacles and speed bumps efficiently while maintaining stability. Its modular architecture supports additional sensors and software modules for enhanced autonomy and task adaptability.

## Expansion and Integration

Hunter 2.0's aluminum T-slot rails and CAN bus support make it easy to integrate custom sensors, actuators, and software modules. Developers can leverage the open-source SDK for rapid secondary development.

> **Note:** Specifications may vary depending on payload, configuration, and operational environment. Maximum speed and climbing ability are tested under standard conditions. The robot is intended for civilian and research applications only; avoid unsafe modifications.

# 4 Quick Start

*_images/hunter_se_front.webp*

AgileX Hunter 2.0

This guide provides essential information for getting started with the AgileX Hunter 2.0.

## Pre-requisites

Before operating the Hunter 2.0, ensure you have:

- Completed safety training (see General Safety )
- Access to the robot's CAN interface
- A compatible RC transmitter (if using remote control)
- Adequate workspace with proper clearance

## Robot App

The Hunter 2.0 supports development via CAN bus interface. AgileX provides:

- Open-source SDK
- C++ and ROS packages
- CAN bus protocol documentation

## Robot Webserver

If configured with an onboard computer, the robot may have a web interface accessible via the robot's IP address.

## Charging Robot

The Hunter 2.0 is equipped with a 10A charger by default.

Charging Procedure:

1. Ensure the Hunter 2.0 chassis is powered off (Q3 switch in OFF position)
2. Connect the charger plug to the Q6 charging interface on the rear panel
3. Plug the charger into a power outlet and turn on the charger switch

Charging Times:

- 30Ah Battery : Approximately 3-4 hours
- 60Ah Battery : Approximately 6-7 hours

Battery Voltage Levels:

- Fully charged: ~26.8 V
- Normal operating: 24-26.8 V
- Low battery warning: < 24.5 V
- Critical (auto-shutdown): < 24.0 V

> **Warning:** Always monitor the charger and battery during charging for safety. Do not charge in temperatures outside -10°C to 40°C range.

## Charging Controller

The battery charging status should be monitored via the charger's own status LED. The chassis does not have a dedicated charging indicator light.

## Emergency Stop

The Hunter 2.0 has an emergency stop button located on top of the chassis.

Activating Emergency Stop:

- Press the red emergency stop button to immediately halt all movement
- The robot will engage the parking brake automatically

Resetting Emergency Stop:

1. Twist the emergency stop button clockwise to release it
2. Reset the SWA switch on the RC transmitter before movement is possible

## Power On

Pre-Startup Checks:

- Inspect the robot for any visible damage
- Ensure both emergency stop buttons are released
- Check that the Q3 drive power switch on the rear panel is not pressed down

Startup Procedure:

1. Press the Q3 button to turn on the chassis
2. The voltmeter should display the battery voltage (normal: 24-26.8 V)
3. Both front and rear lights should illuminate
4. If the beeper emits continuous warnings, the battery is low - charge before use

## Power Off

1. Press Q3 to cut off power supply
2. Wait for all lights to extinguish
3. For extended storage, ensure the battery is charged to at least 50%

## Basic Operation

## Remote Control

Setup:

1. Ensure the chassis is powered on
2. Turn on the RC transmitter (press and hold POWER button)
3. Move SWB switch to remote control mode (middle position)
4. Release parking with SWA switch (up position)

Controller Mapping:

| Control | Function |
|---|---|
| SWA | Parking mode (Up: released, Down: engaged) |
| SWB | Control mode (Up: command mode, Middle: RC mode) |
| S1 | Throttle (forward/backward movement) |
| S2 | Steering (left/right) |
| KEY1 | Clear error messages |
| KEY2 | Settings menu |

Display Information:

- Vol : Battery voltage
- Car : Chassis operating status
- Batt : Remaining battery percentage
- P : Parking status

## Network Interface

CAN Bus Connection:

The Hunter 2.0 uses CAN bus for communication. Connect via the USB-to-CAN module:

1. Connect the USB-to-CAN adapter to your computer
2. Connect the CAN interface to the robot's expansion port
3. Configure CAN interface: sudo ip link set can0 up type can bitrate 500000
4. Verify connection: candump can0

SSH Connection (if onboard computer is installed):

1. Connect to the robot's network
2. SSH to the robot: ssh user@<robot_ip>

## First Test

1. Power on the robot following the Power On procedure
2. Turn on the RC transmitter
3. Set SWB to remote control mode
4. Release parking (SWA up)
5. Gently push S1 forward to move the robot
6. Test steering with S2
7. Engage parking (SWA down) when finished

# 5 Safety

## General Safety

The AgileX Hunter 2.0 is an Ackermann steering mobile robot platform. Following proper safety procedures is essential to prevent injury and equipment damage.

**Danger:** Critical Hazards: Pinch/Crush Risk : Keep hands and body clear of the steering mechanism and wheels during operation. The Ackermann steering can cause pinch points. Impact Risk : The Hunter 2.0 can reach speeds up to 1.5 m/s (Hunter 2.0) or 4.8 m/s (Hunter SE). Maintain safe distance during autonomous operation. Battery Risk : Lithium batteries can be hazardous if damaged, short-circuited, or exposed to extreme temperatures. Handle with care.

## Emergency Stop System

The Hunter 2.0 features an emergency stop button on top of the chassis.

Physical Emergency Stop:

- Press the red mushroom button to immediately halt all movement
- The robot automatically engages the electromagnetic parking brake
- To reset: twist the button clockwise until it pops out

Parking Brake:

- Power-off electromagnetic band-type brake
- Engages automatically when emergency stop is pressed
- Can be manually released using the Q2 knob (use only in emergencies)

**Warning:** After releasing the emergency stop, the SWA switch on the RC transmitter must be reset before the robot can move again.

## Operating Limits

| Parameter | Safe Limit |
|---|---|
| Operating Temperature | -10°C to 40°C (14°F to 104°F) |
| Maximum Speed | 1.5 m/s (Hunter 2.0) / 4.8 m/s (Hunter SE) |
| Maximum Payload | 150 kg (Hunter 2.0) / 50 kg (Hunter SE) |
| Maximum Climb Grade | 10° (Hunter 2.0) / 30° (Hunter SE) |
| IP Rating | IP22 (not waterproof) |

## Safe Operating Practices

- Always release parking mode (SWA up) before attempting to drive
- Start with slow speeds while learning the controls
- Never cross the robot's path while it is moving
- Keep clear of wheels and steering mechanism during operation
- Use the emergency stop immediately if abnormal behavior occurs
- Monitor battery voltage regularly - charge when below 24.5V
- Do not operate in wet conditions (IP22 rating only)

## Battery Safety

- Use only the provided charger (10A default)
- Do not charge batteries outside the temperature range of -10°C to 40°C
- Charge to at least 50% before extended storage
- Monitor charging - do not leave unattended
- Replace batteries showing signs of damage or swelling

## 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.

# 6 Hardware Overview & Specification

## Introduction

The HUNTER 2.0 is a programmable unmanned ground vehicle (UGV) platform built on an Ackermann steering chassis. It offers car-like handling while providing superior performance on Portland cement and asphalt surfaces.

Compared to traditional four-wheel differential drive chassis, HUNTER 2.0 delivers:

- Enhanced payload capacity
- Higher top speed
- Reduced wear on tires and structure for long-term use

While not intended for extreme off-road terrain, the chassis features a rocker arm suspension that allows it to overcome common obstacles such as speed bumps.

Optional modules such as stereo cameras, LiDAR, GPS, IMU, and robotic arms can be integrated for advanced navigation, perception, and automation tasks.

HUNTER 2.0 is ideal for:

- Autonomous driving research and education
- Indoor and outdoor security patrols
- Environmental monitoring and sensing
- General logistics and material transportation

## Component List

| Name | Quantity |
|---|---|
| HUNTER 2.0 Robot body | x1 |
| Battery charger (AC 220V) | x1 |
| Aviation plug (male, 4-pin) | x1 |
| Remote control transmitter (optional) | x1 |
| USB to CAN communication module | x1 |

## Specifications

### Mechanical

| Parameter | Value |
|---|---|
| Dimensions (L × W × H mm) | 980 × 745 × 370 |

| | |
|---|---|
| Wheelbase (mm) | 650 |
| Front / rear wheelbase (mm) | 605 |
| Weight of chassis body (kg) | 65 / 70 |
| Maximum steering angle | 33° |
| Suspension type | Front wheel non-independent |
| IP rating | IP22 |

## Electrical System

| Parameter | Value |
|---|---|
| Battery type | Lithium battery |
| Battery capacity | 24V 30Ah / 60Ah |
| Charging time | 3 h (30Ah) / 6 h (60Ah) |
| Maximum battery life | Up to 8 hours |
| Working temperature | -10 °C to 40 °C |

## Drive System

| Parameter | Value |
|---|---|
| Drive motor type | DC brushless |
| Drive motor power | 2 × 400 W |
| Drive motor reduction ratio | 1 : 40 |
| Drive motor sensor | Magnetic encoder(2500 P/R) |
| Steering motor type | DC brushless 400 W |
| Steering type | Front-wheel Ackermann |
| Parking brake | Power-off electromagnetic band-type brake |

## Control System

| Parameter | Value |
|---|---|
| Control mode | Remote control / Command mode |
| RC transmitter | 2.4 GHz, up to 200 m range |
| System interface | CAN |

## Performance

| Parameter | Value |
|---|---|
| Maximum speed | 1.5 m/s |
| Minimum turning radius | 330 mm (approx.) |
| Maximum gradeability | 10° |
| Ground clearance | 100 mm |
| Maximum travel distance | 40 km (24V 60Ah battery) |
| | 20 km (24V 30Ah battery) |

**Note:** Performance values are based on standard test conditions. Actual performance may vary depending on payload, terrain, and battery condition.

## Safety Notes

**Note:** Intended for research, educational, and civilian applications only. Do not modify electrical or mechanical systems without technical approval. Always follow safety guidelines when operating the robot.

## Environment

**Note:** IP22 protection rating safeguards against limited dust and vertical water droplets. Not suitable for heavy rain or submerged operation.

## Development

HUNTER 2.0 supports optional FS RC transmitter control, allowing manual operation of movement and steering.

The CAN interface enables custom development and expansion of the platform.

**Note:** Remote control requires the optional FS RC transmitter. CAN ports are designed for advanced user development and integration.

## The Basics

This section introduces the HUNTER 2.0 mobile robot platform (see Figure 2.1 and 2.2).

_images/hunter_se_isometric.webp

Front View Components :

1. Standard Aluminum T-Slot Profile – Provides modular mounting support for sensors and accessories.
2. Top Cabin Panel – Main protective cover for electronic and mechanical components.
3. Top Extension Interface – Connectors for CAN, power, and optional device integration.
4. Emergency Stop Switch – Immediately cuts power to the robot in case of emergency.
5. Ackermann Front-Wheel Steering – Provides precise steering similar to automotive Ackermann geometry.

_images/hunter_se_side.webp

Rear View Components :

1. Emergency Stop Switch – Safety switch for immediate power cutoff.
2. Battery Replacement Panel – Access panel for battery installation or replacement.
3. Rear Panel – Houses power switches, CAN and 24V extension interfaces, and status indicators.

HUNTER 2.0 is designed as a versatile autonomous chassis that combines inflatable rubber wheels with an independent suspension system . Coupled with powerful DC brushless servo motors , the robot can traverse a variety of ground surfaces with high adaptability and obstacle passability.

An emergency stop switch is installed at the rear of the chassis to immediately cut power in case of abnormal behavior or emergency situations.

The robot is equipped with waterproof connectors for both DC power and communication interfaces on the top and rear panels. These connectors allow safe integration of external devices while protecting internal electronics even in harsh or humid environments.

> **Note:** Keep the emergency stop switch easily accessible at all times. Only use waterproof connectors in environments with potential water exposure or high humidity.

## Status Indication

The operational state of HUNTER 2.0 can be monitored via:

- Voltmeter – shows battery voltage
- Beeper – alerts for warnings or abnormal conditions
- Indicator lights – display chassis status

These components are mounted on the chassis body. A summary of the indicators is provided below:

### Status Indicators

| Status | Voltage / Condition | Description |
|---|---|---|
| Battery voltage | Normal operating voltage | The current battery voltage can be read from the voltmeter on the rear electrical panel. |
| Low battery alarm | < 24.5 V (SOC < 15% with BMS) | The chassis emits a repeated “beep-beep-beep” warning when battery voltage is low. |
| Replace battery | < 24.0 V (SOC < 10% with BMS) | The system cuts off power to external extensions and drive motors to prevent battery damage. Movement control and external commands are disabled. |
| Robot powered on | Power ON state | Rear indicator lights are switched on. |

> **Note:** All voltage thresholds assume connection to the Battery Management System (BMS). Regularly monitor the voltmeter and beeper to ensure safe operation.

## Electrical Interface

## Top Interface

The HUNTER 2.0 platform is fitted with two 4-pin aviation connectors and one DB9 (RS-232) interface on the top panel. In the current hardware version, the RS-232 port is used only for firmware updates and does not support command communication.

The physical locations of the aviation connectors and the DB9 interface are illustrated in Figure 2.3 .

*_images/hunter_se_front.webp*

Expansion ports are available on both the top surface and the rear of the robot. Each expansion port includes:

- A dedicated power supply output
- A CAN communication interface

These connectors allow external devices to receive power and exchange data with the main controller. Detailed pin assignments are shown in Figure 2.4 .

*_images/agile_hunter_1.webp*

> **Note:** The auxiliary power output is controlled internally by the system. When the battery voltage drops below a predefined safety threshold, the power to external devices is automatically switched off. A low-voltage warning is issued before shutdown occurs. Users should monitor battery levels and recharge the battery in time during operation.

## Rear Interface

The layout of the rear electrical panel is shown in Figure 2.6 . The panel contains the following elements:

_images/hunter_se_isometric.webp

- Q1 – Power status indicator
- Q2 – Manual parking release switch
- Q3 – Main power switch
- Q4 – Buzzer
- Q5 – CAN and 24 V power expansion interface
- Q6 – Charging connector

The detailed pinout for the Q5 interface is provided in Figure 2.7 .

_images/hunter_se_side.webp

The rear expansion interface provides the same CAN communication and 24 V power output as the top interface. Internally, the top and rear interfaces are electrically interconnected and share the same power and communication lines.

## Remote Control

An optional FS remote control transmitter is available for the HUNTER 2.0 platform. This model uses a left-hand throttle layout. The layout of controls and their functions are shown in Figure 2.8 .

## Buttons and Switches

The functions of the switches and buttons are described below:

- SWA – Parking mode switch - Up position: releases parking mode - Down position: activates parking mode - The robot can only be controlled when parking mode is released.
- SWB – Control mode selector - Up position: command (external) control mode - Middle position: manual remote control mode
- SWC / SWD – Currently not used
- S1 – Throttle control for forward and backward movement
- S2 – Steering control for the front wheels
- POWER – Power button - Press and hold to switch the remote controller on
- KEY1 – Clears vehicle error messages
- KEY2 – Opens the remote controller settings menu

> **Note:** The remote controller button mapping is preconfigured at the factory. Do not modify these settings, as doing so may cause incorrect or unsafe operation.

## Display Information

The remote control screen displays the following status information:

- Hunter – Device model name
- Vol – Battery voltage of the chassis
- Car – Chassis operating status
- Batt – Remaining battery percentage of the chassis
- P – Parking status
- Remoter – Remote controller battery level
- Fault Code – Error code information (corresponds to byte [5] in the 211 CAN frame)

## Control Commands

## Coordinate System

A fixed reference coordinate system can be defined on the HUNTER 2.0 chassis in accordance with ISO 8855, as shown in Figure 2.9 . The chassis is aligned with the X-axis of this system.

- Remote Control Mode: Push S1 forward → move in the positive X direction Push S1 backward → move in the negative X direction Maximum stick deflection corresponds to maximum speed in the respective X direction S2 controls the front wheel steering: Push left → chassis turns left (maximum left steering at full deflection) Push right → chassis turns right (maximum right steering at full deflection)
- Command Control Mode: - Positive linear velocity → movement along the positive X-axis - Negative linear velocity → movement along the negative X-axis - Steering angle corresponds to the inner wheel steering angle

> **Note:** The reference coordinate system ensures consistent mapping between control commands and robot movement.

—

## Getting Started

This section introduces basic operation and development of the HUNTER 2.0 platform using the CAN bus interface.

## Operation

### Pre-Startup Checks

Before powering on the robot:

- Inspect the robot for any visible damage or anomalies. Contact after-sales support if needed.
- Ensure both emergency stop buttons are released.
- For first-time use, check that the Q3 drive power switch on the rear panel is not pressed down. Release it if necessary to ensure the drive is powered off.

### Startup Procedure

- Press the Q3 button to turn on the chassis.
- The voltmeter should display the battery voltage, and both front and rear lights should illuminate.
- Normal battery voltage: 24–26.8 V
- If the beeper emits continuous warnings, the battery is low and should be charged.

### Shutdown

- Press Q3 to cut off power supply.

### Emergency Stop

- Press the emergency stop button located on top of the chassis to immediately halt all movement.

### RC Operation

1. Ensure the chassis is powered on.
2. Turn on the RC transmitter.
3. Move SWB to remote control mode.
4. The HUNTER 2.0 chassis can now be maneuvered via the RC transmitter.

### Parking Function

- The chassis uses a power-off electromagnetic brake for parking.
- Remote Control Mode: SWA controls parking: Top → parking released, movement allowed Bottom → parking activated; if moving, chassis decelerates to zero before engaging
- Command Control Mode: Parking is active by default at power-on Speed commands are ignored until parking release is sent To park after motion, send a parking command
- Emergency Stop: Triggers automatic parking After release, SWA must be reset before movement is possible
- Manual Unlock (Q2 knob): Can override parking in special cases (highest priority) Should be returned to normal after use
- Ramp Parking: When stopped on a slope, the system monitors current draw If it exceeds a threshold for a set duration, ramp parking engages automatically Movement commands release ramp parking automatically

> **Note:** Always release parking mode before attempting to drive Use emergency stop and manual unlock functions responsibly Ramp parking ensures safety on slopes but requires attention when re-engaging movement

## Charging and Battery

The HUNTER 2.0 is equipped with a 10A charger by default, which is sufficient for standard charging requirements. During charging, the chassis does not have a dedicated indicator light; charging status should be monitored via the charger's own status LED.

Follow these steps for proper charging:

1. Ensure the HUNTER 2.0 chassis is powered off. Confirm that the rear power switch (Q3) is in the OFF position.
2. Connect the charger plug to the Q6 charging interface on the rear panel.
3. Plug the charger into a power outlet and turn on the charger switch. The robot will enter the charging state.

> **Note:** A fully depleted battery (~21 V) requires approximately 4 hours to charge. Fully charged battery voltage is around 26.8 V . Always monitor the charger and battery during charging for safety.

To safely replace the battery:

1. Turn off the main power switch of the chassis.
2. Press the battery panel lock and open the battery compartment.
3. Disconnect the battery connectors: - XT60 power connector - BMS connector
4. Remove the battery carefully. Avoid dropping or colliding the battery.
5. Install the replacement battery and reconnect the power and BMS connectors.
6. Close the battery panel and secure the lock.
7. Turn on the power if required.

**Note:** Handle batteries with care; avoid impacts or short circuits. Always ensure proper connection of both XT60 and BMS connectors before powering the chassis.

# 7 ROS 2 Software – Operation

## Pre-requisites

- Safety Guidelines General Safety .
- Power On Power On .
- Network Interface Network Interface .
- Installation Robot Installation .

## Autostart

Autostart Configuration

Create a systemd service for automatic startup:

```
sudo nano /etc/systemd/system/hunter-ros2.service
```

Add:

```
[Unit]
Description=Hunter ROS 2 Base Driver
After=network.target

[Service]
Type=simple
User=your_username
ExecStart=/bin/bash -c "source /opt/ros/humble/setup.bash && source
~/hunter_ws/install/setup.bash && ros2 launch hunter_base hunter_base.launch.py"
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Enable:

```
sudo systemctl daemon-reload
sudo systemctl enable hunter-ros2.service
sudo systemctl start hunter-ros2.service
```

Check status:

```
sudo systemctl status hunter-ros2.service
```

Autostart Configuration

Configure the Hunter ROS 2 nodes to start automatically on boot.

### MYBOTSHOP Platform Service

If using the MYBOTSHOP configuration, services are managed via systemd:

Start Platform Service:

```
sudo service hunter-platform start
```

Check Service Status:

```
sudo service hunter-webserver status
sudo service hunter-platform status
```

Service Status Indicators:

- Green : Service is running correctly
- Red : Service has failed - restart needed
- Grey : Service has not started yet

Restart Service:

```
sudo service hunter-platform restart
```

Update Startup Job:

After modifying launch files, update the startup job:

```
ros2 run hunter_bringup startup_installer.py
```

> **Note:** If the service is green, do not launch hunter_bringup manually as it's already running in the background.

### Manual Launch

Launch the hunter ROS driver manually:

```
ros2 launch hunter_bringup system.launch.py
```

### Custom systemd Service

Create a custom service file:

```
sudo nano /etc/systemd/system/hunter-ros2.service
```

Add the following content:

```
[Unit]
Description=Hunter ROS 2 Base Driver
After=network.target

[Service]
Type=simple
User=your_username
Environment="ROS_DOMAIN_ID=0"
ExecStart=/bin/bash -c "source /opt/ros/jazzy/setup.bash && source
~/hunter_ws/install/setup.bash && ros2 launch hunter_base hunter_base.launch.py"
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Enable Service:

```
sudo systemctl daemon-reload
sudo systemctl enable hunter-ros2.service
sudo systemctl start hunter-ros2.service
```

Check Status:

```
sudo systemctl status hunter-ros2.service
```

View Logs:

```
journalctl -u hunter-ros2.service -f
```

CAN Interface Autostart

Create a script to bring up CAN on boot:

```
sudo nano /etc/network/interfaces.d/can0
```

Add:

```
auto can0
iface can0 inet manual
    pre-up /sbin/ip link set can0 type can bitrate 500000
    up /sbin/ip link set can0 up
    down /sbin/ip link set can0 down
```

## Webserver

Web Interface

ROS 2 Web Bridge:

```
sudo apt install ros-humble-rosbridge-server
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

This enables WebSocket communication with ROS 2 topics and services.

Foxglove Studio:

```
sudo apt install ros-humble-foxglove-bridge
ros2 launch foxglove_bridge foxglove_bridge_launch.xml
```

Connect using Foxglove Studio for visualization and debugging.

Web Interface

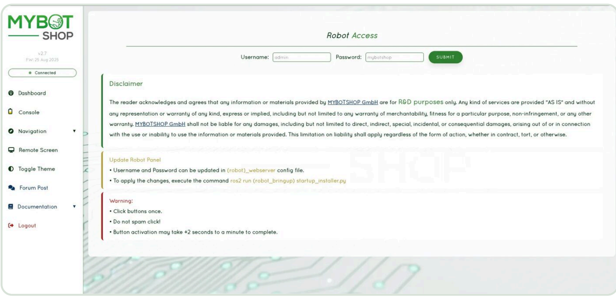
If the Hunter is configured with an onboard computer, a web interface may be available.

Accessing the Web Interface:

1. Connect to the robot's network
2. Open a web browser
3. Navigate to http://192.168.131.1:9000/

Login Credentials:

- Username: admin
- Password: mybotshop



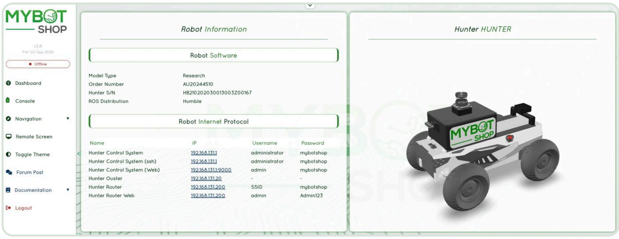

_images/web_login.webp

Web Login Interface

Dashboard

The dashboard provides system overview:

- View IP Address of the HUNTER
- View System load and status



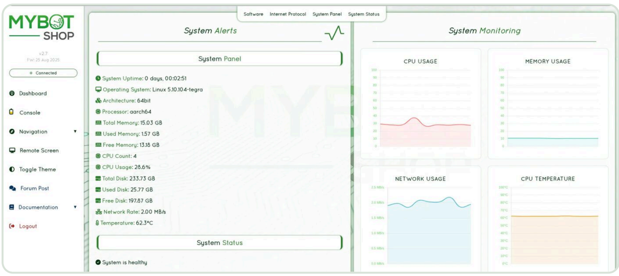

*_images/web_dashboard.webp*

Web Dashboard

System Monitoring

Monitor system resources in real-time:

- CPU, Memory, and Network usage
- Temperature monitoring
- System uptime and OS information



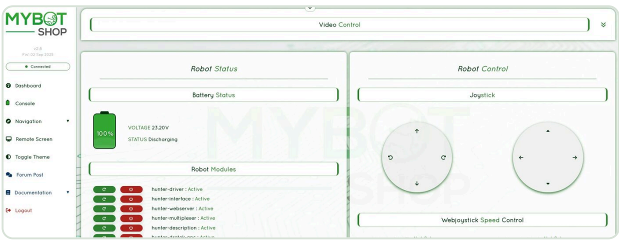

*_images/web_system.webp*

Web System Panel

Console

The console provides robot control features:

- Enable/Disable HUNTER ROS2 Services
- Battery status monitoring
- Web Joystick control
- Pre-configured action buttons




*_images/web_console.webp*

Web Console

Remote Desktop

Access the on-board NVIDIA computer desktop remotely:


*_images/web_remote_desktop.webp*

Web Remote Desktop

ROS 2 Web Bridge

For web-based ROS 2 interaction, install rosbridge:

```
sudo apt install ros-jazzy-rosbridge-server
```

Launch the bridge:

```
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

This enables WebSocket communication with ROS 2 topics and services.

Foxglove Studio

For modern web-based visualization:

1. Install Foxglove Bridge: sudo apt install ros-jazzy-foxglove-bridge
2. Launch the bridge: ros2 launch foxglove_bridge foxglove_bridge_launch.xml
3. Open Foxglove Studio and connect to your robot.

## Teleoperation

Keyboard Teleoperation

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Keyboard Controls:

- i : Forward
- k : Stop
- j : Turn left
- l : Turn right
- , : Backward

Joystick Teleoperation

```
sudo apt install ros-humble-joy ros-humble-teleop-twist-joy
ros2 launch teleop_twist_joy teleop-launch.py
```

Custom Velocity Commands

```
ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.5}, angular: {z: 0.0}}"
```

Keyboard Teleoperation

Control the Hunter using keyboard:

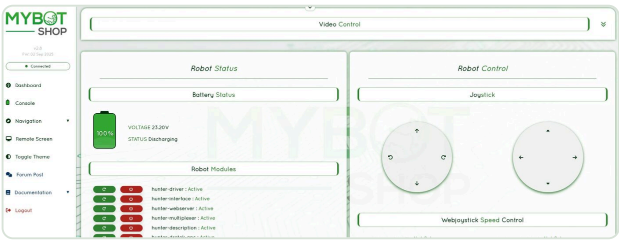
```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/hunter_0001/cmd_vel
```

Keyboard Controls:

- i : Forward
- k : Stop
- j : Turn left
- l : Turn right
- , : Backward
- u : Forward-left curve
- o : Forward-right curve

Web Joystick

If using the MYBOTSHOP webserver, a web-based joystick is available:




*_images/web_console.webp*

Web console with joystick control

Joystick Teleoperation

Install joystick packages:

```
sudo apt install ros-jazzy-joy ros-jazzy-teleop-twist-joy
```

Launch joystick teleoperation:

```
ros2 launch teleop_twist_joy teleop-launch.py
```

RC Transmitter

The Hunter can also be controlled via the optional FS RC transmitter:

- SWA : Parking mode switch (Up: released, Down: parking)
- SWB : Control mode selector (Up: command mode, Middle: RC mode)
- S1 : Throttle control (forward/backward)
- S2 : Steering control (left/right)

> **Note:** For PC control via ROS2, set SWB to the bottom 3rd position.

Custom Velocity Commands

Send velocity commands directly:

```
ros2 topic pub /hunter_0001/cmd_vel geometry_msgs/msg/Twist \
  "{linear: {x: 0.5}, angular: {z: 0.0}}"
```

> **Note:** The Hunter uses Ackermann steering. Angular velocity is converted to steering angle internally. Recommended teleoperation speed: 0.2 m/s for mapping.

## Visualization

RViz2 Visualization

Launch RViz2 with Hunter model:

```
ros2 launch hunter_description display.launch.py
```

View TF Tree

```
ros2 run tf2_tools view_frames
```

RViz2 Visualization

Launch RViz2 with Hunter model:

```
ros2 launch hunter_viz view_robot.launch.py
```

This displays:

- Robot URDF model
- TF frames
- Coordinate axes
- Sensor data (if available)

View Robot Model Only

To view just the robot model without running the base driver:

```
ros2 launch hunter_description display.launch.py use_rviz:=true
```

Web-Based Visualization

For remote visualization, use the web interface:

*_images/web_remote_desktop.webp*

Remote desktop access for visualization

Custom RViz Configuration

Save your RViz configuration:

1. Configure displays as desired
2. File > Save Config As > my_hunter_config.rviz

Load custom configuration:

```
rviz2 -d my_hunter_config.rviz
```

TF Tree

View the transform tree:

```
ros2 run tf2_tools view_frames
```

This generates a PDF showing the complete TF tree structure.

Foxglove Studio

For modern web-based 3D visualization:

1. Install Foxglove Bridge: sudo apt install ros-jazzy-foxglove-bridge
2. Launch the bridge: ros2 launch foxglove_bridge foxglove_bridge_launch.xml
3. Open Foxglove Studio and connect to your robot

## Rigs

Sensor Mounting

The Hunter features aluminum T-slot rails for mounting sensors and equipment.

T-Slot Rail Specifications:

- Standard aluminum profile
- Compatible with M5/M6 T-nuts
- Located on top panel

Power Output:

- 24V from expansion interfaces
- Maximum 15A continuous
- Auto-cutoff on low battery

Weight Distribution

- Maximum payload: 150 kg (Hunter 2.0) / 50 kg (Hunter SE)
- Center of gravity should remain low
- Distribute weight evenly for stability

Sensor Mounting

The Hunter 2.0 features aluminum T-slot rails for mounting sensors and equipment.

T-Slot Rail Specifications:

- Standard aluminum T-slot profile
- Compatible with M5/M6 T-nuts
- Located on top panel

Common Configurations

LiDAR Mount:

- Mount 2D/3D LiDAR on top center for optimal FOV
- Ensure clear line of sight (360° for spinning LiDAR)
- Typical height: 20-30cm above chassis

Camera Mount:

- Front-facing for navigation
- Multiple cameras for surround view
- Consider vibration dampening for sensitive cameras

Compute Box:

- Mount securely to prevent movement
- Ensure adequate ventilation
- Route cables cleanly to prevent interference

Power Considerations

The Hunter provides:

- 24V power output from expansion interfaces
- Maximum 15A continuous
- Auto-cutoff on low battery

Plan power budget for all sensors and compute equipment.

Weight Distribution

- Maximum payload: 150 kg (Hunter 2.0) / 50 kg (Hunter SE)
- Center of gravity should remain low
- Distribute weight evenly for stability

## Manipulation

Robotic Arm Integration

The Hunter 2.0's payload capacity supports integration with robotic manipulators.

Supported Arms:

- Lightweight collaborative arms (< 10 kg)
- Research-grade manipulators
- Custom end effectors

ROS 2 Integration:

```
sudo apt install ros-humble-moveit
```

Configure MoveIt for your specific arm model.

Safety Considerations

- Implement collision detection
- Define safe workspace boundaries
- Emergency stop should halt both base and arm
- Consider arm weight in payload calculations

> **Note:** Maximum payload: 150 kg (Hunter 2.0) / 50 kg (Hunter SE). Arm integration affects the robot's center of gravity.

Robotic Arm Integration

The Hunter 2.0's payload capacity supports integration with robotic manipulators.

Supported Arms:

- Lightweight collaborative arms (< 10 kg)
- Research-grade manipulators
- Custom end effectors

Power Requirements

- Arms typically require 24V or 48V power
- Use the Hunter's 24V expansion output for compatible arms
- Consider external power supply for high-power arms

Communication

Common Interfaces:

- CAN bus (shared with Hunter)
- Ethernet
- USB
- RS-485

ROS 2 Integration

For arm integration with ROS 2:

```
sudo apt install ros-jazzy-moveit
```

Configure MoveIt for your specific arm model.

Safety Considerations

- Implement collision detection
- Define safe workspace boundaries
- Emergency stop should halt both base and arm
- Consider arm weight in payload calculations

> **Note:** Arm integration significantly affects the robot's center of gravity. Recalculate stability margins after adding a manipulator.

## Navigation

Simultaneous Localization and Mapping

Install Nav2 and SLAM Toolbox:

```
sudo apt install ros-humble-navigation2 ros-humble-nav2-bringup
sudo apt install ros-humble-slam-toolbox
```

Launch SLAM:

```
ros2 launch slam_toolbox online_async_launch.py
```

Save the map:

```
ros2 run nav2_map_server map_saver_cli -f ~/maps/my_map
```

Odometry Navigation

Navigate using odometry without a map:

```
ros2 launch nav2_bringup navigation_launch.py
```

Map Navigation

Navigate with a pre-built map:

```
ros2 launch nav2_bringup bringup_launch.py map:=/path/to/map.yaml
```

Use RViz2 to set goals with “2D Goal Pose”.

GPS Navigation

For outdoor GPS-based navigation, integrate a GPS receiver and use robot_localization for sensor fusion.

Navigation Stack

The Hunter 2.0 supports Nav2 navigation stack for autonomous navigation.

Install Nav2:

```
sudo apt install ros-jazzy-navigation2 ros-jazzy-nav2-bringup
```

SLAM (Mapping)

Create a map using SLAM Toolbox:

```
sudo apt install ros-jazzy-slam-toolbox
```

Launch SLAM:

```
ros2 launch hunter_navigation slam.launch.py
```

Drive the robot around to build the map. Save the map:

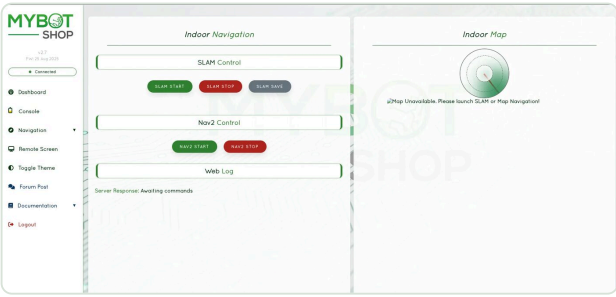
```
ros2 run nav2_map_server map_saver_cli -f \
  /opt/mybotshop/src/mybotshop/hunter_nav2/maps/custom_map \
  --ros-args --remap map:=/hunter_0001/map
```

Rebuild to make the map available:

```
cd /opt/mybotshop/ && colcon build --symlink-install
source /opt/mybotshop/install/setup.bash
```

Web-Based Navigation

If using the MYBOTSHOP webserver, navigation can be controlled via the web interface:




*_images/web_navigation.webp*

Web Navigation Interface with SLAM and Nav2 controls

Odometric Navigation

Navigate without creating a map:

```
ros2 launch hunter_navigation odom_navi.launch.py
```

Map Navigation

Navigate using a pre-built map:

```
ros2 launch hunter_navigation map_navi.launch.py
```

Localization

For navigation with a pre-built map, use AMCL:

```
ros2 launch nav2_bringup localization_launch.py map:=/path/to/map.yaml
```

Navigation

Launch the full navigation stack:

```
ros2 launch nav2_bringup navigation_launch.py
```

Use RViz2 to:

1. Set initial pose with “2D Pose Estimate”
2. Set goals with “2D Goal Pose”

**Note:** The Hunter uses Ackermann steering which requires appropriate Nav2 controller configuration. Use the Regulated Pure Pursuit controller or similar for best results.

## Sensors

Depth Cameras

Supported depth cameras:

- Intel RealSense D435/D455
- ZED/ZED2 Stereo Camera

```
sudo apt install ros-humble-realsense2-camera
ros2 launch realsense2_camera rs_launch.py
```

Lidars

Supported LiDAR sensors:

- Velodyne VLP-16
- Ouster OS1
- SICK LMS series
- RPLidar

Connect to Hunter's 24V expansion interface.

GPS

Supported GPS receivers:

- u-blox series
- Emlid Reach
- Fixposition

Auxiliary Sensors

- IMU (Xsens, Microstrain, Phidgets)
- Ultrasonic sensors
- Encoders

Connect sensors to the expansion ports providing 24V power and CAN communication.

Built-in Sensors

The Hunter 2.0 includes:

- Wheel Encoders : For odometry calculation
- Steering Angle Sensor : Magnetic encoder (2500 P/R)

### Supported External Sensors

The Hunter's aluminum T-slot rails and expansion interfaces support various sensors:

LiDAR:

- Velodyne VLP-16
- Ouster OS1
- SICK LMS series
- RPLidar

Cameras:

- Intel RealSense D435/D455
- ZED/ZED2 Stereo Camera
- USB webcams

IMU:

- Drotek IMU
- Xsens MTi series
- Microstrain
- Phidgets IMU

GPS:

- u-blox receivers
- Emlid Reach
- Fixposition

### Ouster LiDAR

The Ouster LiDAR provides a 3D point cloud of the environment.

Launch Ouster:

```
ros2 launch hunter_lidars ouster.launch.py
```

Configuration:

- IP Address: 192.168.131.20
- Provides 3D point cloud data

> **Note:** The Ouster is turned on by default when using the MYBOTSHOP platform service. Only launch manually when the auto service is turned off.

### Intel RealSense D435

The Intel RealSense D435 depth camera provides RGB and depth data.

Launch RealSense:

```
ros2 launch hunter_depth_camera realsense_d435.launch.py
```

Configuration file:

```
/opt/mybotshop/src/mybotshop/hunter_depth_camera/launch/realsense_d435i.launch.py
```

> **Note:** By default the RealSense D435 is off. The launch file is configured to enable continuous depth stream information without lag.

Adding Sensors

Connect sensors to the expansion ports:

- 24V Power : Available from expansion interface (max 15A continuous)
- CAN : For CAN-based sensors (500kbps)
- USB : Via onboard computer (if installed)

Sensor Topics

Common sensor topics:

- /scan : 2D LiDAR scan
- /ouster/points : 3D point cloud
- /camera/color/image_raw : RealSense color image
- /camera/depth/image_rect_raw : RealSense depth image
- /imu/data : IMU data
- /gps/fix : GPS fix

## Simulation

Gazebo Simulation

Install Gazebo:

```
sudo apt install ros-humble-gazebo-ros-pkgs
```

Launch Simulation:

```
ros2 launch hunter_gazebo hunter_gazebo.launch.py
```

The simulation provides the same topics as the real robot:

- /cmd_vel : Velocity commands
- /odom : Odometry

- /tf : Transforms

Isaac Sim

NVIDIA Isaac Sim support is available for advanced simulation scenarios. Refer to the NVIDIA Isaac Sim documentation for setup instructions.

Gazebo Simulation

Simulate the Hunter in Gazebo for development and testing.

Install Gazebo:

```
sudo apt install ros-jazzy-gazebo-ros-pkgs
```

Launch Simulation:

```
ros2 launch hunter_gazebo hunter_gazebo.launch.py
```

This starts Gazebo with a Hunter model and ROS 2 interfaces.

Simulated Topics

The simulation provides the same topics as the real robot:

- /cmd_vel : Velocity commands
- /odom : Odometry
- /tf : Transforms

Adding Sensors in Simulation

Modify the URDF/Xacro files to add simulated sensors:

- LiDAR: Add <gazebo> plugin for ray sensor
- Camera: Add camera plugin
- IMU: Add IMU plugin

Custom Worlds

Launch with a custom world:

```
ros2 launch hunter_gazebo hunter_gazebo.launch.py world:=/path/to/world.sdf
```

> **Note:** Simulation is useful for testing navigation and control algorithms before deploying on the real robot.

# 8 ROS 2 Package Reference & Installation

## Packages

Hunter ROS 2 Packages

ugv_sdk

Low-level SDK for AgileX robots providing CAN communication interface.

hunter_base

Base driver package for Hunter 2.0/SE communication.

```
ros2 launch hunter_base hunter_base.launch.py
```

hunter_description

URDF models and meshes for Hunter robots.

```
ros2 launch hunter_description display.launch.py
```

hunter_msgs

Custom ROS 2 message definitions for Hunter robots.

Key Topics

- /cmd_vel : Velocity commands
- /odom : Odometry data
- /hunter_status : Robot status
- /battery_state : Battery status

Hunter ROS 2 Packages

ugv_sdk

Low-level SDK for AgileX robots providing CAN communication interface.

hunter_base

Base driver package for Hunter 2.0/SE communication.

```
ros2 launch hunter_base hunter_base.launch.py
```

hunter_description

URDF models and meshes for Hunter robots.

```
ros2 launch hunter_description display.launch.py
```

hunter_msgs

Custom ROS 2 message definitions for Hunter robots.

Key Topics

| Topic | Description |
|---|---|
| /cmd_vel | Velocity commands (geometry_msgs/Twist) |
| /odom | Odometry data |
| /hunter_status | Robot status information |
| /battery_state | Battery status |

Key Parameters

- can_device : CAN interface name (default: "can0")
- robot_model : Robot model type ("hunter2" or "hunter_se")
- base_frame : Base frame ID (default: "base_link")
- odom_frame : Odometry frame ID (default: "odom")

## Debugging

Debugging Commands

List Topics:

```
ros2 topic list
```

Check Topic Rate:

```
ros2 topic hz /odom
```

Echo Topic Data:

```
ros2 topic echo /hunter_status
```

Monitor CAN Traffic:

```
candump can0
```

Common Issues

- Check CAN interface: ip link show can0

- Verify CAN bitrate: 500000
- Ensure emergency stop is released
- Verify parking mode is released (SWA up)

Log Files

```
ros2 run rqt_console rqt_console
```

Common Debugging Commands

List Active Topics:

```
ros2 topic list
```

Check Topic Rate:

```
ros2 topic hz /odom
```

Echo Topic Data:

```
ros2 topic echo /hunter_status
```

View Node Info:

```
ros2 node list
ros2 node info /hunter_base_node
```

CAN Bus Debugging

Check CAN Interface:

```
ip link show can0
```

Monitor CAN Traffic:

```
candump can0
```

Send Test CAN Message:

```
cansend can0 123#DEADBEEF
```

Common Issues

Robot not responding:

1. Check CAN interface is up: ip link show can0
2. Verify CAN bitrate: 500000

3. Check emergency stop is released
4. Verify parking mode is released (SWA up)

No odometry:

1. Check wheel encoder connections
2. Verify hunter_base node is running
3. Check /odom topic is being published

Steering not working:

1. Check steering motor connections
2. Verify steering limits in parameters
3. Check for error codes on RC display

Log Files

View ROS 2 logs:

```
ros2 run rqt_console rqt_console
```

## Miscellanious

Updating Packages

```
cd ~/hunter_ws/src/hunter_ros2
git pull
cd ~/hunter_ws
colcon build --symlink-install
```

CAN Protocol

The Hunter uses CAN bus at 500kbps. Refer to AgileX documentation for protocol details.

Network Configuration

```
sudo nmcli con mod "Wired connection 1" ipv4.addresses 192.168.1.100/24
sudo nmcli con mod "Wired connection 1" ipv4.method manual
sudo nmcli con up "Wired connection 1"
```

Time Synchronization

```
sudo apt install chrony
sudo systemctl enable chrony
sudo systemctl start chrony
```

Updating Packages

Update Hunter ROS 2 packages:

```
cd ~/hunter_ws/src/hunter_ros2
git pull
cd ~/hunter_ws
colcon build --symlink-install
```

### CAN Protocol

The Hunter uses CAN bus at 500kbps for communication.

Key CAN IDs:

- Motion control commands
- Status feedback
- Battery information
- Error codes

Refer to the AgileX CAN protocol documentation for detailed frame definitions.

### SDK Development

For custom development, use the ugv_sdk:

```
#include "ugv_sdk/mobile_robot/hunter_robot.hpp"

HunterRobot robot;
robot.Connect("can0");
robot.SetMotionCommand(0.5, 0.0);  // linear vel, steering angle
```

### Network Configuration

Static IP Setup:

```
sudo nmcli con mod "Wired connection 1" ipv4.addresses 192.168.1.100/24
sudo nmcli con mod "Wired connection 1" ipv4.method manual
sudo nmcli con up "Wired connection 1"
```

### Time Synchronization

For multi-machine setups, synchronize time:

```
sudo apt install chrony
sudo systemctl enable chrony
sudo systemctl start chrony
```

### Backup Configuration

Back up your ROS 2 workspace:

```
tar -czvf hunter_ws_backup.tar.gz ~/hunter_ws
```

## Installation

ROS 2 Humble Installation

Prerequisites:

- Ubuntu 22.04
- ROS 2 Humble installed
- USB-to-CAN adapter

Install ROS 2 Humble:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
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
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
sudo apt update
sudo apt install ros-humble-desktop
```

Install Hunter ROS 2 Packages:

```
mkdir -p ~/hunter_ws/src
cd ~/hunter_ws/src
git clone https://github.com/agilexrobotics/ugv_sdk.git
git clone https://github.com/agilexrobotics/hunter_ros2.git -b humble
```

Install Dependencies:

```
cd ~/hunter_ws
rosdep install --from-paths src --ignore-src -r -y
```

Build Workspace:

```
cd ~/hunter_ws
colcon build --symlink-install
source install/setup.bash
```

Add to bashrc:

```
echo "source ~/hunter_ws/install/setup.bash" >> ~/.bashrc
```

CAN Interface Setup

```
sudo ip link set can0 up type can bitrate 500000
```

## ROS 2 Jazzy Installation

Prerequisites:

- Ubuntu 24.04
- ROS 2 Jazzy installed
- USB-to-CAN adapter

Install ROS 2 Jazzy:

```
sudo apt install software-properties-common
sudo add-apt-repository universe
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
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
sudo apt update
sudo apt install ros-jazzy-desktop
```

Install Hunter ROS 2 Packages:

```
mkdir -p ~/hunter_ws/src
cd ~/hunter_ws/src
git clone https://github.com/agilexrobotics/ugv_sdk.git
git clone https://github.com/agilexrobotics/hunter_ros2.git
```

Install Dependencies:

```
cd ~/hunter_ws
rosdep install --from-paths src --ignore-src -r -y
```

Build Workspace:

```
cd ~/hunter_ws
colcon build --symlink-install
source install/setup.bash
```

Add to bashrc:

```
echo "source ~/hunter_ws/install/setup.bash" >> ~/.bashrc
```

## CAN Interface Setup

Configure the CAN interface for communication with the Hunter:

```
sudo ip link set can0 up type can bitrate 500000
```

To make CAN interface persistent, create a systemd service or add to rc.local.

# 9 HUNTER RO – Research Configuration

## HUNTER RO

The HUNTER RO is a research-oriented configuration of the AgileX Hunter 2.0 platform, integrated with sensors from Drotek, Intel RealSense D435, and Ouster LiDAR. This kit is designed for R&D applications in autonomous navigation, perception, and robotics research.


*../_images/hunter_ro_cover.webp*

**Note:** This manual is for the RnD kit and contains additional information about the HUNTER robot. For powering on and usage of the Hunter base platform, refer to the main Hunter documentation.

**Warning:** The robot is an R&D device and does not have CE Marking. Basic ROS understanding is required. If you are not familiar with ROS, check the ROS2 Wiki first.

## Quick Start

For control of the robot via PC:

1. Power on the robot and the remote according to the Hunter manual
2. Place the remote's SWB switch in the bottom 3rd position to control the robot via ROS2

### Webserver

The webserver module comes pre-installed for heavy integration projects. It is accessible at:

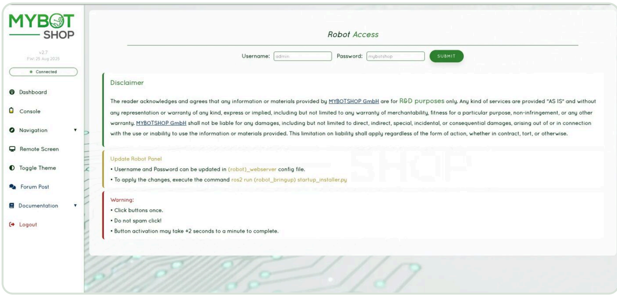
- URL: http://192.168.131.1:9000/
- WiFi: Connect to the robot's WiFi network first

The webserver can be configured via the config file:

```
/opt/mybotshop/src/mybotshop/hunter_webserver/config/robot_webserver.yaml
```

Login Credentials:

- Username: admin
- Password: mybotshop

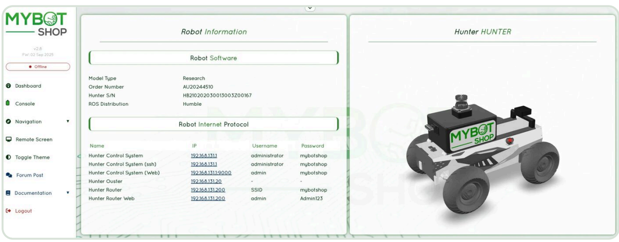



../_images/web_login.webp

Web Login Interface

Dashboard Features:

- View IP Address of the HUNTER
- View System load

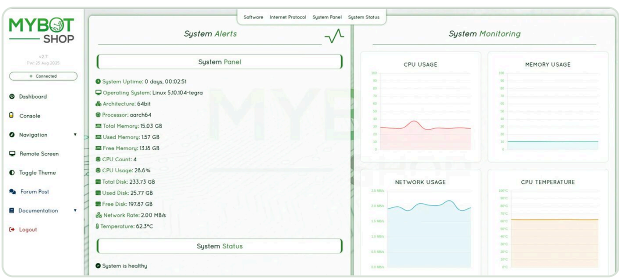



../_images/web_dashboard.webp

Web Dashboard

System Panel:

- System uptime and OS information
- CPU, Memory, Network, and Temperature monitoring



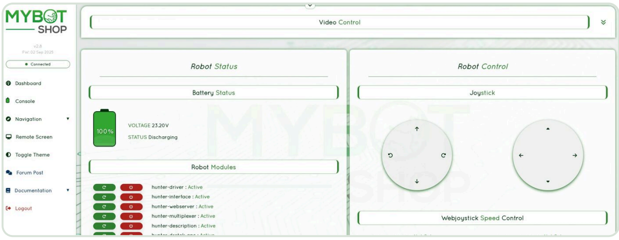

../_images/web_system.webp

Web System Panel

Console Features:

- Enable/Disable HUNTER ROS2 Services

- Record System HUNTER logs
- Battery status monitoring
- Pre-configured action buttons
- Web Joystick control
- Online image stream




*../_images/web_console.webp*

Web Console

Remote Desktop:

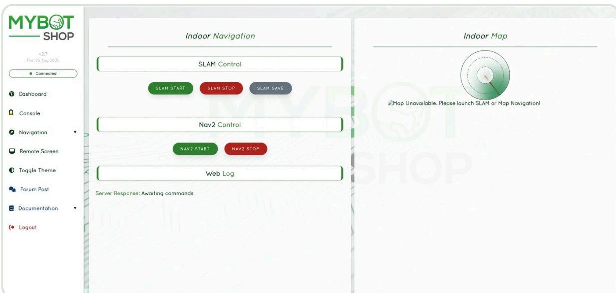
- On-board screen of the HUNTER's NVIDIA computer


*../_images/web_remote_desktop.webp*

Web Remote Desktop

Navigation Interface:

- SLAM Control (Start/Stop/Save)
- Nav2 Control




*../_images/web_navigation.webp*

Web Navigation Indoor

## Robot Interface

Instructions for interfacing with the robot using Ubuntu 22.04 and ROS2 Humble .

Static Network Connection:

For the first time, connect via WiFi or LAN cable to configure the robot's network.

To create a static connection on your PC (not the robot):

1. Go to Settings → Network, click + to create a new connection
2. Change the connection to Manual in IPv4 settings
3. Set Address IP as 192.168.131.51 and Netmask as 24
4. Save and restart your network

After successful connection, check the host's local IP:

```
ifconfig
```

Ping the robot:

```
ping 192.168.131.1
```

Access the robot via SSH:

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

The default password is: 123

HUNTER IP Addresses:

| Robot | IP Addresses | Username | Password |
|---|---|---|---|
| HUNTER MCU | 192.168.131.1 | administrator | 123 |
| HUNTER MCU Web | **http://192.168.131.1:9000 \| admin** | | mybotshop |
| Ouster | 192.168.131.20 | | |
| Router SSID | 192.168.131.200 | HUNTERXXXXX-5G | mybotshop |
| Router Web | 192.168.131.200 | admin | Admin123 |

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

Interfacing:

1. Connect to the HUNTER as described in the Network section
2. Open several SSH sessions: ssh -X administrator@192.168.131.1

3. Verify the robot is processing information: ros2 topic list If data is displayed, the drivers are running.

### Visualization

View the HUNTER's current state:

```
ros2 launch hunter_viz view_robot.launch.py
```

### Teleoperation

Teleoperate the HUNTER (change the id to match your robot):

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/hunter_0001/cmd_vel
```

## Navigation

### SLAM

To create a map for map-based navigation:

```
ros2 launch hunter_navigation slam.launch.py
```

Map using teleop at 0.2 m/s with keyboard or Logitech controller. Export the map:

```
ros2 run nav2_map_server map_saver_cli -f \
  /opt/mybotshop/src/mybotshop/hunter_nav2/maps/custom_map \
  --ros-args --remap map:=/hunter_0001/map
```

Rebuild so maps can be found:

```
cd /opt/mybotshop/ && colcon build --symlink-install
source /opt/mybotshop/install/setup.bash
```

### Odometric Navigation

Navigate without creating a map:

```
ros2 launch hunter_navigation odom_navi.launch.py
```

### Map Navigation

Navigate using a pre-built map:

```
ros2 launch hunter_navigation map_navi.launch.py
```

# Core Packages

## Auto Drivers Startup

**Note:** Generally, all drivers are off except the webserver and can be activated via webserver or command.

Start the platform service:

```
sudo service hunter-platform start
```

Check service status:

```
sudo service hunter-webserver status
sudo service hunter-platform status
```

If the service is green, do not launch hunter_bringup as it's running in background.

Launch the hunter ROS driver manually:

```
ros2 launch hunter_bringup system.launch.py
```

Restart service if needed:

```
sudo service hunter-platform restart
```

Update startup job after modifications:

```
ros2 run hunter_bringup startup_installer.py
```

## Ouster LiDAR

The Ouster LiDAR provides a 3D point cloud of the environment.

```
ros2 launch hunter_lidars ouster.launch.py
```

**Note:** This is turned on by default. Only use this command when the auto service is turned off.

## Intel RealSense D435

Launch the depth camera:

```
ros2 launch hunter_depth_camera realsense_d435.launch.py
```

Configuration file location:

```
/opt/mybotshop/src/mybotshop/hunter_depth_camera/launch/realsense_d435i.launch.py
```

> **Note:** By default the RealSense D435 is off. The launch file is configured to enable continuous depth stream without lag.

## Installation

### NVIDIA Orin Setup

> **Note:** This repository should already be available and built on Hunter's NVIDIA board if configured by the MYBOTSHOP team.

1. Create directory: cd /opt/ sudo mkdir mybotshop sudo chown -R administrator:administrator /opt/mybotshop
2. Sync the repository: rsync -avP -t --delete -e ssh src \ administrator@192.168.131.1://opt/mybotshop
3. Build the kernel for NVIDIA: ./jetson-gs_usb-kernel-builder.sh sudo visudo -f /etc/sudoers.d/can-setup administrator ALL =( ALL ) \ NOPASSWD: /sbin/modprobe gs_usb, /sbin/ip link \ set can2 up type can bitrate 500000
4. Patch NVIDIA kernel for RealSense camera: sha256sum -c install-modules.tar.gz.sha256 tar -xzf install-modules.tar.gz cd install-modules sudo ./install-realsense-modules.sh

## Support

In case of any issues:

- MYBOTSHOP Forums: forum.mybotshop.de
- MYBOTSHOP Support: support @ mybotshop . de

Documentation resources:

- MYBOTSHOP Docs: docs.mybotshop.de
- QUADRUPED Docs: docs.quadruped.de

# 10 Getting Help, RMA & FAQ

## RMA / Support

### How to Raise a Return Merchandise Authorization (RMA) Case

At QUADRUPED Robotics, 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 @ quadruped . 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: QUADRUPED Robotics GmbH c / o 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. Do not ship goods to QUADRUPED Robotics GmbH Office in Leverkusen!

### Resources

For your convenience, we have compiled the most important resources related to the robot platform. Each link can be accessed directly or by scanning the corresponding QR code.

### Resources

For your convenience, we have compiled the most important resources related to the robot platform. Each link can be accessed directly or by scanning the corresponding QR code.

Documentation

Our official documentation contains detailed setup instructions, usage guidelines, and troubleshooting information. Access the latest version here: https://www.docs.quadruped.de/index.html

*_images/qr_docs.webp*

Forum

Stay informed about product updates, events, and special offers by subscribing here: https://www.mybotshop.de/Newsletter_1

*_images/qr_newsletter.webp*

## Contact Information

Forum

Connect with other developers and researchers in our community forum. Here you can ask questions, share experiences, and stay up to date: https://forum.mybotshop.de/

_images/qr_forum.webp

Rentals

If you would like to test or temporarily use a robot, we also offer rental services. Browse our available systems here: https://www.mybotshop.de/Vermietung_3

_images/qr_rentals.webp

Contact Information

- Sales 📞 +49 (0) 2271 588 93 10 | ✉ sales @ mybotshop . de
- After Sales 📞 +49 (0) 2271 588 93 20 | ✉ support @ mybotshop . de
- Accounting 📞 +49 (0) 2271 588 93 80 | ✉ accounting @ mybotshop . de
- General inquiries 📞 +49 (0) 2271 588 93 0 | ✉ info @ mybotshop . de

## Frequently Asked Questions

...