This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the latest version (ROS 2 Humble).

Version: ROS 1 Noetic

Warthog User Manual

Introduction

Clearpath Robotics Warthog is a rugged, all-terrain unmanned ground vehicle capable of brief periods of locomotion in water. Warthog fully supports the Robot Operating System (ROS) and can be equipped with a variety of payloads, including sensors and manipulators, to accommodate a wide range of robotics applications in mining, agriculture and environmental monitoring.

Shipment Contents

Your Warthog shipment contains the following:

- Warthog UGV
  - Onboard computer

- User Breakout Panel with power, Ethernet, Serial (RS232) and USB connectivity
- 48 V Lead-Acid Battery Pack or 48 V Lithium Battery Pack

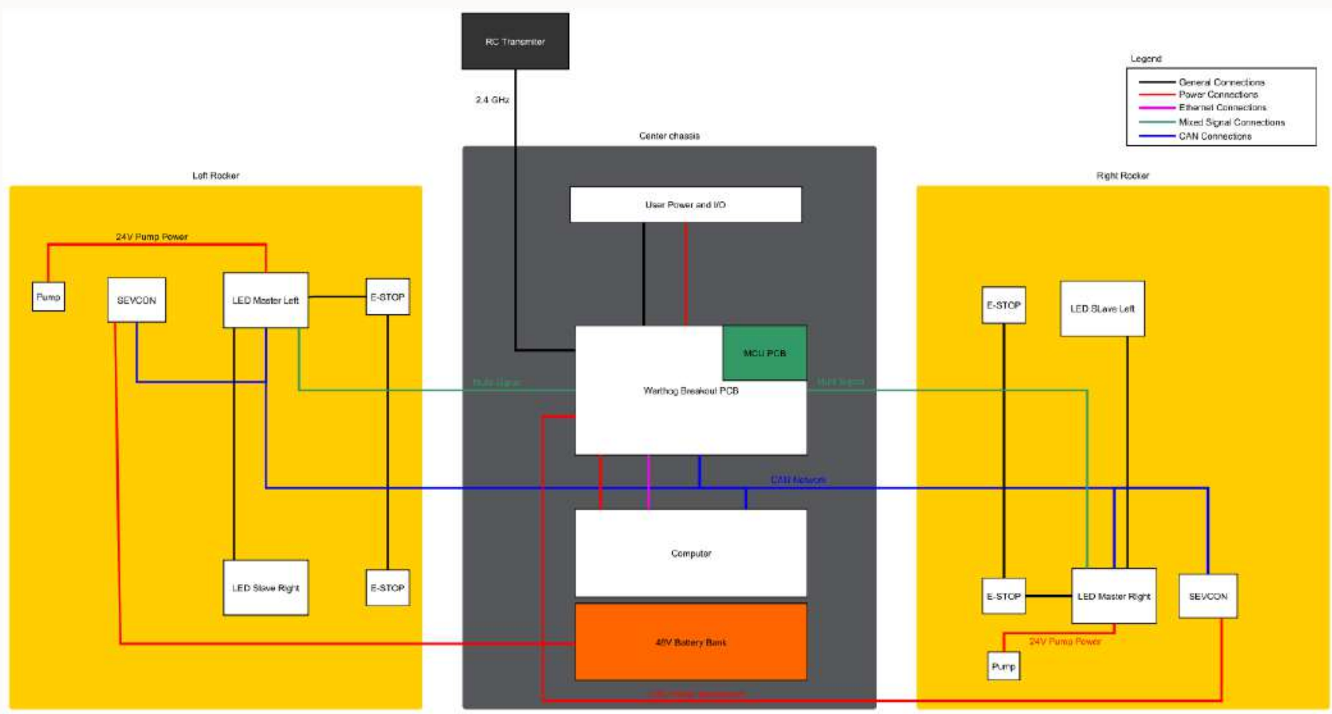
- One battery charger
- One Futaba remote control
- One Warthog quick start guide
- One Wireless Stop remote

If you purchased standard payload modules or custom integration services with Warthog, then additional equipment will be included per your specific configuration, plus further documentation as required.

Hardware Overview

System Architecture

Warthog is built around an onboard computer running Ubuntu (Intel-based computer or Jetson developer kit), paired with a 32-bit microcontroller MCU. The MCU handles IO, system and battery monitoring, and provides an interface to the CAN-controlled motor drivers. The communication channel between the MCU and computer is an Ethernet connection.



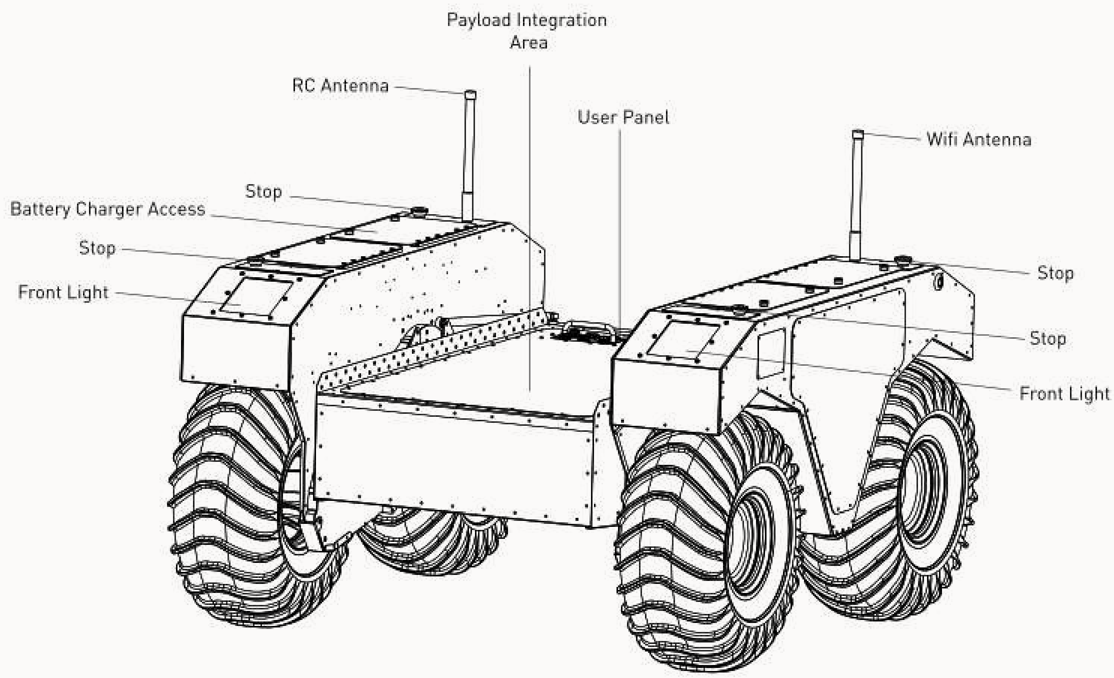
System Architecture

System Features

The Warthog is shown below and includes:

- Battery charger access

- Bilge pumps
- User breakouts
- Payload integration area
- Four Stop buttons (one on each corner)
- Four body lights (one on each corner)



Warthog hardware overview

Battery Charger Access

Please review this PDF for information about the provided charger.

Bilge Pumps

The Warthog has two bilge pumps, with one situated in each of the drive units underneath the motor. The bilge pumps are used to remove water from the main chassis and drive units during operation in water. At initial startup, an audible sound can be heard from the bilge pumps being initiated. These pumps are an automatic pumping system that check for water levels inside the drive units. The pumps automatically come on every two minutes and check for water levels. If the water level inside the drive unit exceeds a predetermined level, the pumps will remain on, otherwise they will shut off. During prolonged use in water, some water may appear in the drive unit. This is normal and acceptable. This outlet leads from the pump to the exterior of the Warthog. It is used to remove excess water from the chassis and drive units.

No obstructions should be placed in front of or around this area. Obstruction of water flow may result in damage to the internal electrical components and loss of function in the Warthog.

User Breakouts

The User Breakout Panel provides access to the user power panel, as well as USB, serial, and ethernet ports. The power panel can be used to power your payloads. The USB 3.0 and Ethernet ports are connected directly to the onboard PC. To connect a device to the onboard network, it's suggested to give it a static IP in the *192.168.131.xxx* subnet, avoiding IPs in use by the following pre-existing devices:

IP address	Description
192.168.131.1	Onboard PC (all ports, br0 network interface)
192.168.131.2	Ethernet-connected MCU

Typically addresses numbered 192.168.131.100 and above will not collide with any of Warthog's payloads. Please see Electrical Integration for more information.

Payload Integration Area

All payloads should be mounted to the central chassis when traversing through water to prevent rolling. The primary payload of the unit should be placed on the central chassis. If necessary loading can be placed on the drive units however payloads should not exceed 23 kg (50 lbs) on each drive unit.

For more information and guidance on mounting payload structures on top of Warthog, please refer to subsection Mechanical Mounting.

Body Lights

Warthog includes four RGB body lights, one on each corner of the chassis. These lights express system status according to the table below, but in the absence of one of the low-level conditions, they can be commanded from ROS to display indications from autonomy or other higher-level software. See http://wiki.ros.org/warthog_base for information on commanding the body lights.

Front Lights	Rear Lights	System State
Solid red	Solid red	The MCU is not in contact with the computer. That is, the `rosserial` connection is not active. This condition will be seen briefly on startup while

Front Lights	Rear Lights	System State
		Warthog's computer is booting up. If it persists, or is seen after initialization, either the base node on the computer has crashed, the network switch has failed, or a serious MCU error has occurred. If you suspect one of these conditions, please contact support.
Flashing red	Flashing red	Stop mode is engaged. To disengage, twist the mushroom buttons on the Warthog and on the Wireless Stop remote to ensure that they are unlatched. See also Using the Wireless Stop Remote.
Flashing orange	Flashing orange	Motor drivers not yet ready to drive. The motors have a brief initialization sequence which must complete after a stop condition clears before they are ready to drive. If this condition persists, please contact our support team.
Pulsing orange	Pulsing orange	Battery is low. Connect Warthog to the charger.
Solid white	Solid red	Normal operation; Warthog is ready to drive. This status may be overridden by publishing your own light patterns to the `/cmd_lights` ROS topic.

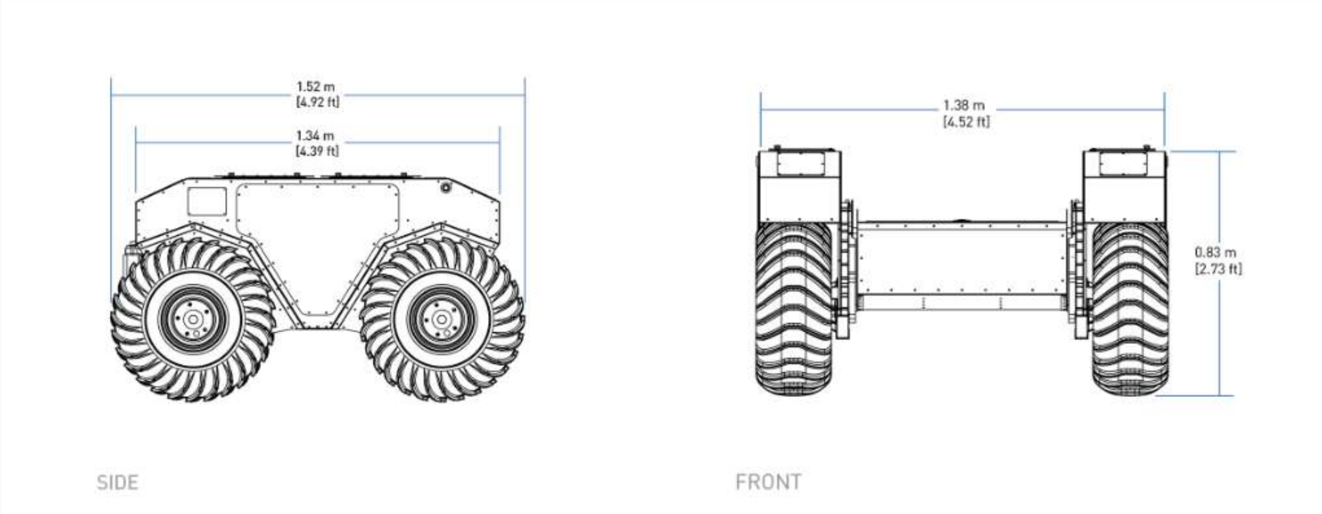
Orientation References

The reference frame used by all Clearpath Robotics ground robots is based on ISO 8855. When commanded with a positive translational velocity (forward), wheels travel in the positive X-direction. The direction of the axes differs from those used for roll, pitch, and yaw in aircraft, and care should be taken to ensure that data is interpreted correctly.

System Specifications

Specification	Value
External Dimensions (L x W x H)	1.52 X 1.38 X 0.83 metres (4.9' X 4.5' X 2.72')
Base Weight	280 kg (551 lbs)

Specification	Value
Ground Clearance	254 mm (10 in)
Max Payload	272 kg (600 lbs)
Max Incline	35 - 45°
Max Speed	18 km/h (11 mph)
Suspension	Geometric Passive Articulation
Drive Configuration	4 X 4 Skid Steer
Operating Environment	Outdoor
Traction	Ø610 mm (24") Argo Turf tire (default) or 300 mm (12") wide Quad Track System (optional)
Battery Chemistry	AGM sealed lead-acid (lithium optional)
Capacity	105 Ah at 48 V, expandable to 200 Ah with lithium option
Nominal Run Time	Lead-acid: 2.5 hrs, lithium: 6 hrs
Charge Time	4 Hours approx
User Power	12 V, 24 V and 48 V Fused at 10 A
Control Modes	Remote control, computer controlled velocity commands, indoor/outdoor autonomy packages
Feedback Battery	Voltage, motor currents, wheel odometry, control system status, temperature, safety status
Communication	Ethernet, USB, Remote Control, Wi-Fi
Drivers	Packaged with ROS Noetic (includes RViz, Gazebo support), Matlab API available



Warthog dimensions

Getting Started

You are ready to go! This section details how to get Warthog into action. To begin, place Warthog "up on blocks", making sure that the wheels are clear of the ground.

For most Warthog setups, there will be an Onboard Computer which is directly connected to the Warthog and a Remote Computer which is used to control Husky and gather data. These two computers must be set up to communicate with each other.

Onboard Computer Setup

Warthog ships with an Onboard Computer from Clearpath Robotics, which is already installed, connected, and powered. If you need to reinstall the software on it, refer to the Warthog Tutorials.

Powering Up

To power on your Warthog, twist the red power button on the back of Warthog. Once the body lights are flashing red, twist (to reset) all four Stop buttons (if necessary), and press go on the Wireless Stop remote (next section). In a moment, Warthog should go to solid red lights in back, and solid white in front. This indicates that ROS is up on the computer and has established communications with the MCU.

Using the Wireless Stop Remote

Included with the Warthog is a Wireless Stop remote. If for any reason the robot must be halted immediately, press the red STOP button. This will immediately cut power to the robot's motors.

Warthog's Wireless Stop Remote

It's capable of engaging a stop remotely. The system is designed to enter a stopped state when the remote is not communicating or loses communication with the base. The Wireless Stop remote operates as follows:

1. Pull the red STOP button outwards to disengage it from a stop condition.
2. Press and hold the green START button on the right side of the unit until the battery LED indicator above the buttons turns green. If the battery light is red, this means the STOP button is still engaged.
3. Press the button labelled RELEASE within three seconds followed by the START button again.
4. The battery LED should be blinking green quickly, this indicates the remote is paired with the receiver. To disengage the stop, press START once again.

The battery LED should now be blinking slower, this indicates the remote stop has been disengaged. The corner body lights of the Warthog will change from blinking red to solid red at the rear and white at the front assuming all other safety stop sources have been disengaged. The Warthog is now ready to drive.

> **NOTE**
>
> If you're not seeing any change in behaviour, please contact our support team.

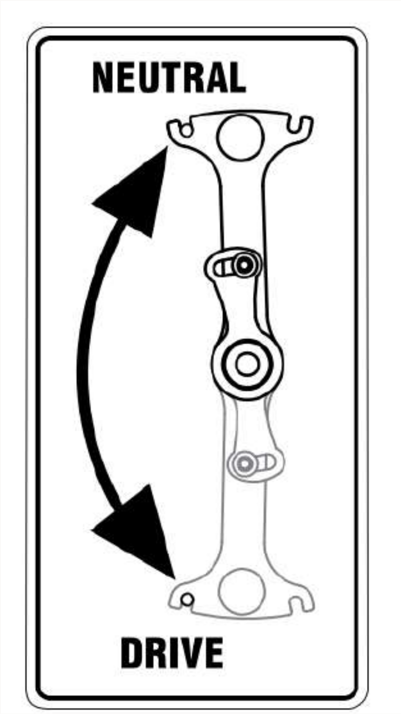
Using the Stop buttons

Warthog has four Stop buttons located on the four corners of the robot. Pressing any of these buttons will cut power to the motors, just like the STOP button on the Wireless Stop remote. To disengage the stop, simply twist the button in the direction indicated by the arrows.

Whenever you need to perform maintenance on Warthog, we recommend engaging the stop if the robot cannot be fully powered down.

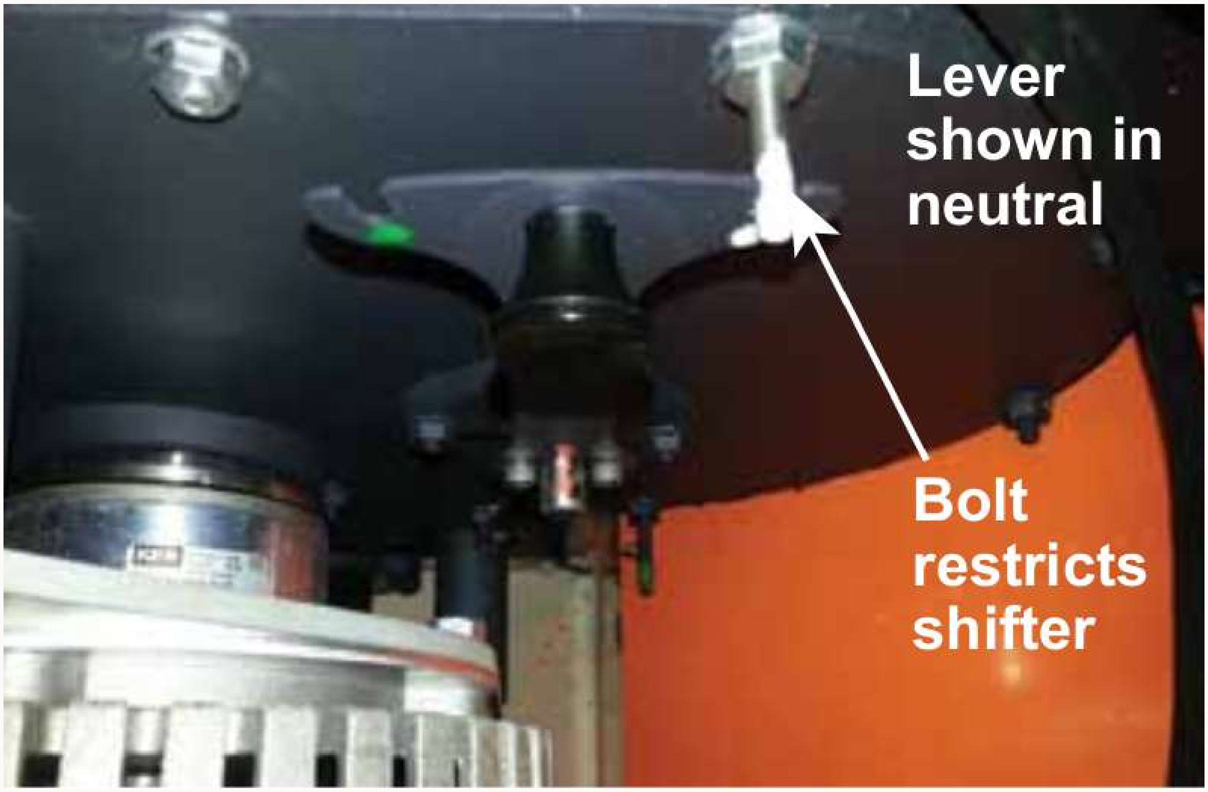
Drivetrain

The Warthog has the ability to be put into neutral which is controlled be the four levers (two per side) in the rockers.



Drivetrain lever

Use the bolts to restrict the lever from moving as down when the drivetrain is in neutral. The groove for the neutral position is marked with white and the bolt is also marked. This is shown below.



Drivetrain in neutral

Pull the handle on the lever to move it into gear which can be seen below. Ensure the lever is in the green groove. If the lever is difficult to move, rock the Warthog back and forth. Do not try to force the lever to move.

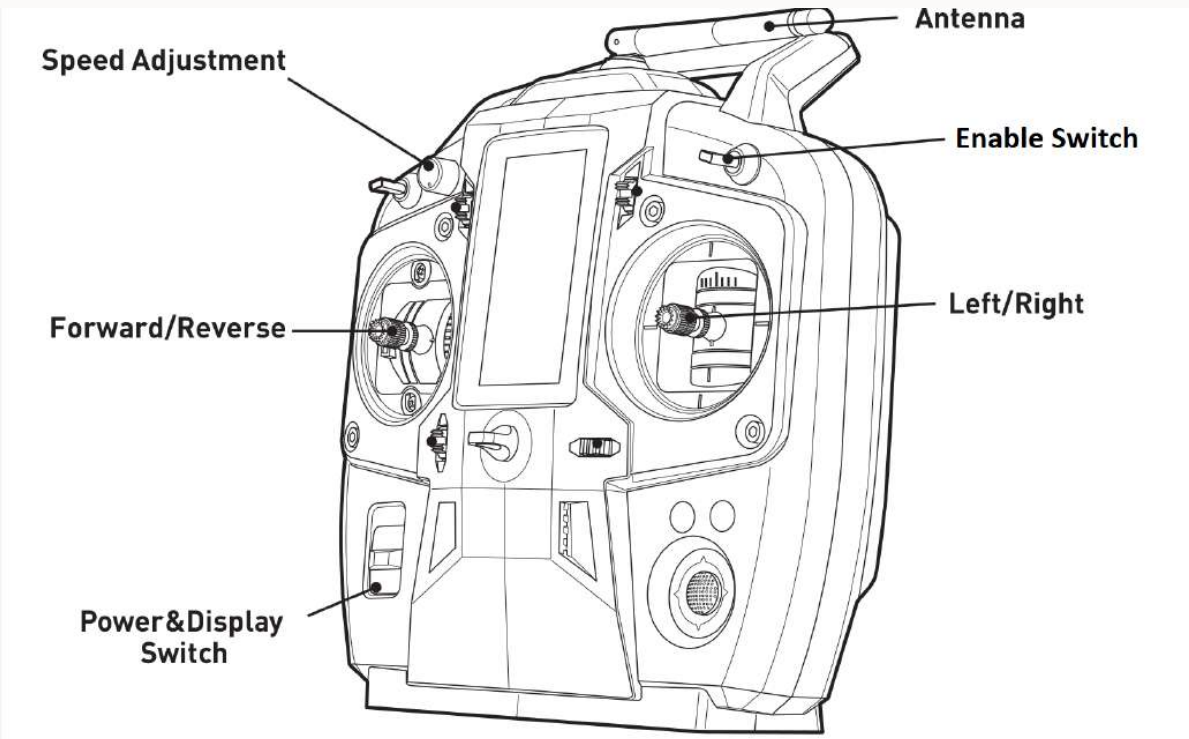
Drivetrain in gear

Using the Futaba Controller

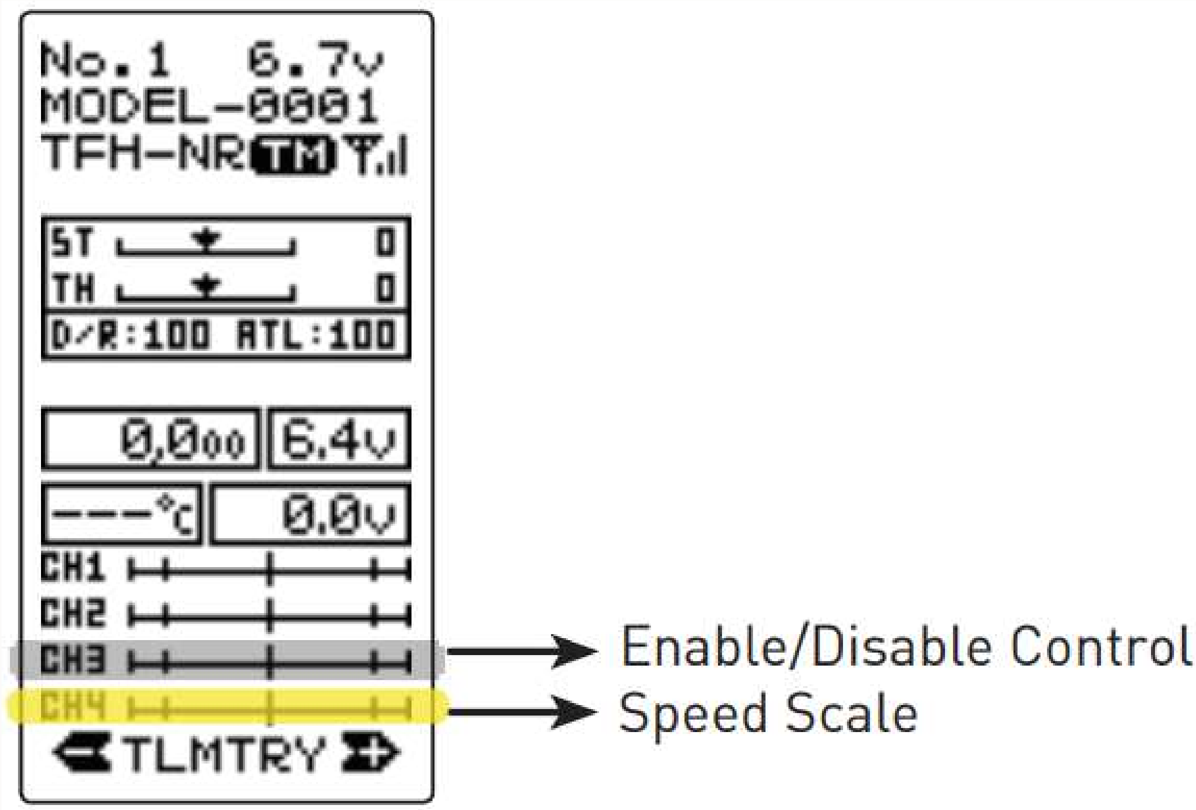
The long range remote control (RC) Futaba radio transmitter can be used to teleoperate the Warthog. To begin, slide the power switch to the ON position which is labelled in the figure below.

> **CAUTION**
>
> The speed adjustment knob should be turned initially all the way to the left while familiarizing yourself with the transmitter and slowly increasing it to get it moving.

The position of the speed adjustment scale is shown in the figure as *CH4*. The transmitter needs to be enabled which is done using the *Enable Switch*. The *Enable Switch* is a two position spring loaded switch. To enable the RC teleoperation, the switch needs to be held down. If released, the switch returns to the up position to disable the robot. The left joystick is used for the forward and reverse motion of the robot and the right joystick is used for turning.



Futaba radio transmitter



Futaba radio transmitter's screen

> **NOTE**
>
> If the Clearpath Robotics Outdoor Navigation software is installed on the Warthog, the Futaba remote controller (RC) must be turned OFF for autonomous missions to be properly executed by the navigation software. The Futaba RC will always publish messages to the `/rc_teleop/cmd_vel` topic when it is ON, and since it has higher priority than the navigation's `/cmd_vel`, the Warthog will not move if autonomous missions are sent and the RC is still ON.

Network Configuration

To get Warthog connected to your local Wi-Fi network, you must first access the Onboard Computer using a wired connection to the robot. Locate the User Breakout Panel on the rear of the chassis and connect to an Ethernet port with a standard Ethernet cable. See the Warthog Tutorials for details on completing the Wi-Fi setup.

Using ROS

Robot Operating System (ROS) is an extensible framework for controlling and working with robotic systems. Clearpath Robotics recommends using ROS with your robot. If this is your first time using ROS, it is strongly recommended to run through our series of ROS tutorials to learn the basics of ROS.

Safety Considerations

Clearpath Robotics is committed to high standards of safety. Warthog contains several features to protect the safety of users and the integrity of the robot.

> **DANGER**
>
> Warthog is a powerful, heavy, fast moving robotic platform. Please read the following safety information carefully.

General Warnings

For the safety of yourself and others, always conduct initial experiments and software development with the motors not engaged. Whenever the robot is not being operated and the motors are engaged, keep it in a stop state. Do not ride on the vehicle, it can accelerate and brake quickly.

When starting out, favour slower wheel speeds. Warthog's control loops can accurately maintain velocities as low as 0.1 m/s. Operating at such speeds will give you more time to react if things don't go quite as you expect.

When enabling the system using the GO button on the wireless remote, be sure to stand well back from the Warthog. User code running on the Warthog may still be trying to command the motors, and this can result in sudden and unexpected movement of the vehicle. Be prepared to stop the system again using the wireless remote.

Maneuverability in Water

Before entering the water it is important to ensure that:

- Bilge pumps are functioning properly
- The side panels and top cover are properly fastened down
- All access panels are fastened down

Pinch Points

When operating the Warthog it is important to maintain a safe distance away from the unit. The suspension has the ability to pivot. Do not place fingers anywhere along the suspension link as it can result in injury.

Warthog pinch point

Stop buttons

The Stop system on the Warthog has two major components: The hardwired Stop switches and the Wireless Stop remote.

Hardwired Stop

Pressing down one of the 4 red mushroom Stop buttons around the Warthog will disable power to the motor controller SEVCON devices (Key switch input on PIN 1). This disables the large contactors and also enables the brakes (passive, spring activated when not powered). The status indicator lights around the Warthog will flash red. To reset a Stop button, the top of the button should be twisted until the button pops out again. The GO button on Wireless remote must then pressed. The Warthog is fully enabled once a relay click is heard, and the front lights change to white.

Wireless Stop Remote

To operate the Warthog, the Wireless Stop remote also has to be powered on (by holding the Power button for at least 1 second). The remote STOP button toggles the stop status. It must be pressed once to enable stop, and pressed again to return to a stop reset ready state. The GO button on the remote will then reset the stop condition. Always ensure the STOP button is accessible at all times. Avoid mounting payloads that extend over the rear of Warthog and would occlude the Stop buttons.

Electrical System

The largest electrical safety consideration with the Warthog system is the VBAT connection. As it is pulled straight from the batteries, it may have a voltage of 48 V (depleted) to 62 V (Charging) and can be used to power large external devices. This voltage can cause electrical shock if directly contacted, and is fused internally with an inline fuse at 10 A. In general, take care to connect or disconnect devices preferably only when the entire system is powered off via the external switch on the rear (main power switch). Take note that triggering a stop condition only disables voltage to the motor controller SEVCON drivers and motors, not the rest of the system which includes the connectors. The labelled status LEDs on the user panels indicate status of the system voltages. If a LED is not lit, then it's most likely that a system fuse has blown. Contacting Clearpath support is the best option. To ensure safety, please also observe the following precautions:

- Do not tamper with the battery terminals or wiring.
- Consult Clearpath Robotics support if you need to service the battery pack.
- Do not lay tools or other objects on top of the battery.
- Do not move the robot while charging the battery.
- Charge the battery only with the charger provided by Clearpath Robotics.
- Please dispose of the batteries properly, or return the batteries to Clearpath Robotics to do so.

Lifting and Transport

For the safety of users and to maximize the lifetime of Warthog, please observe the following when manually transporting the robot:

- Ensure that Warthog's stop mode is engaged when transporting short distances and powered off when transporting longer distances.
- Do not push the robot at more than 0.5 m/s (1.6 ft/s) or damage to the motor controls may occur.

Performance Considerations

Included in Warthog are native software checks and limits to protect the robot. However, it is recommended to monitor the system's status during usage with the `/status` and `/diagnostics` topics. These topics provide useful information regarding voltages, currents, temperatures and general health of the system.

Recommended Safe Work Procedures

Common Safe Work Procedures

Clearpath Robotics recommends the following procedures be followed to reduce the likelihood of harm.

- Maintain a safe distance of at least 10 m from the robot whenever possible.
- Never assume that the robot can be reliably stopped on command.
- Assert a stop using the remote stop device (if present) before approaching the robot.
- Never place the remote stop device (if present) on or inside the robot.
- Always be vigilant around the robot, even when a stop is asserted.
- Perform no maintenance or troubleshooting on the robot without training and familiarisation.
- Follow electrical safety best practices when working with electrical systems and components.
- Disconnect battery using main disconnect of the robot before working inside robot.
- Wear personal protective equipment including high-visibility clothing, head and eye protection while operating, observing or working on the robot.
- Never climb on or ride the robot.
- Never lie on the ground under or near the robot.
- Avoid operating the robot on slopes and grades.
- Never stand downhill of the robot.
- Operate the robot at the lowest speed possible.
- Never attempt to defeat a safety function for any reason.
- Chock the robot's wheels when the robot will be stationary for more than a few seconds.
- Never work on the robot while it is on a slope.
- Replace fuses only with identical product from specified manufacturer.
- Discontinue use and contact Clearpath Robotics support at the first sign of strange robot behaviour.
- Keep tools and loose parts away from electrical components and out of the robot's bays.
- Check the functionality of stop systems on a regular basis, including before and after each use.

Warthog-Specific Safe Work Procedures

- Keep hands and fingers away from the suspension linkage components at the rear of the robot.
- Never approach the robot while it is operating in water.
- When working on or around the robot always disconnect the motors using the drivetrain levers as described in the user manual.
- Never disconnect the robot motors using drivetrain levers when on a slope.
- Do not attempt to move or operate the robot while charging the batteries.

Support

Clearpath is committed to your success. Please get in touch with us and we will do our best to get you rolling again quickly: support@clearpathrobotics.com.

To get in touch with a salesperson regarding Clearpath Robotics products, please email sales@clearpathrobotics.com.

If you have an issue that is specifically about ROS and is something which may be of interest to the broader community, consider asking it on https://answers.ros.org. If you do not get a satisfactory response, please ping us and include a link to your question as posted there. If appropriate, we will answer in the ROS Answers context for the benefit of the community.

Last updated on ***Jul 7, 2023***

Version: ROS 1 Noetic

Warthog Integration

To attach custom hardware to Warthog, you will have to take care of mechnical mounting, electrical supply, and software integration. This section aims to equip you with respect to these challenges.

Mechanical

When determining mechanical mounting, you have two options, both of which are described below.

- the "Standard" aluminum extrusion interface
- the "PACS™" mounting system and associated kits

Mechanical, Standard

A standard Warthog includes thru holes on the sides of the battery compartment. You can besign mounting brackets to interface with these holes.

Mechanical, PACS™

PACS™ is a Clearpath Robotics standard. We add a grid of M5×0.8 holes onto the top plate of the robot. This grid of holes has a 80 mm X 80 mm spacing. You can create your own brackets to interface with these holes, or can use an existing Clearpath Robotics designed bracket.

> **NOTE**
>
> Our Sensors and Accessories pages indicate the required bracket for the particular attachment.

Refer to the following pages for Warthog brackets that can simplify your integration.

- Extrusion Rails
- Extrusion Cross-Bar
- Top Plate

Common Mechanical Integration Guidelines

- 705 mm (27.75") is the maximum allowed width of any installed payload (this assumes that the payload is also centered across the width of the UGV chassis.
- No part of the payload may extend over the sheet metal housings of the drive units or into the small 50 mm (2") gaps between the chassis and drive units. Damage to both the UGV and the payload will result.
- The chassis has a removable access cover measuring 1175 mm X 667 mm (46.25" X 26.25"). This access cover is supported underneath by two adjustable cross members. Regardless of payload, it is imperative that both cross members remain installed (approximately evenly spaced) to provided required support to the access cover. Consider that any payload installed above the top deck will prevent access to the chassis through the access cover, without first removing the installed payload.
- The rotation of the suspension differential link in the horizontal plane will allow the payload to extend beyond the chassis top deck in both fore and aft locations. The amount of this payload extension (overhang) is dependent on several factors, including the weight and method of attachment of the payload as well as the terrain in which the UGV will operate. Ensure that the amount of overhanging payload allows the UGV to operate safely and does not contact the terrain, especially when crossing steep and/or deep gullies.
- The available internal chassis volume is:
  - Length of 445 mm (17.5")
  - Width of 660 mm (26")
  - Height of 241 mm (9.5"). This space is located at the center of the chassis between two battery packs. Consider that anything placed inside the chassis must be secured as to not move or shift during UGV operation.

> **CAUTION**
>
> Permanent damage resulting from custom modifications to the mounting plate is not covered under warranty and may not be supported by Clearpath Support. Please contact our Support team if you require assistance or have any questions relating to custom modifications.

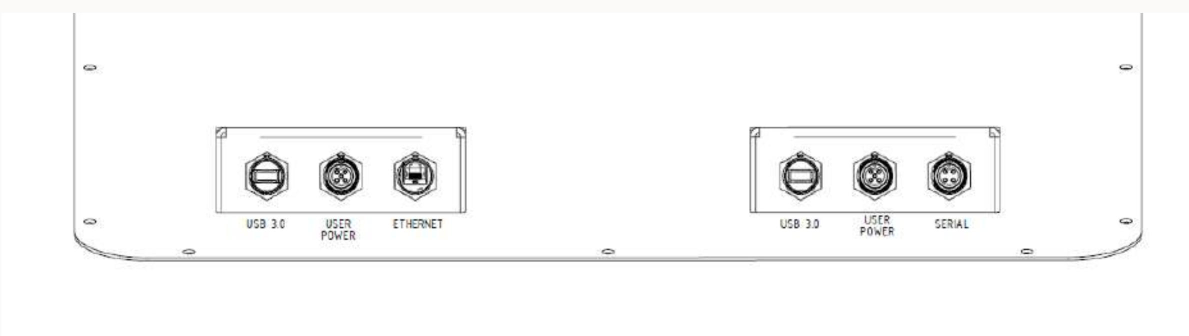
Electrical Integration

> **DANGER**
>
> For continued protection against risk of fire, only replace fuses with those of the same type and rating.

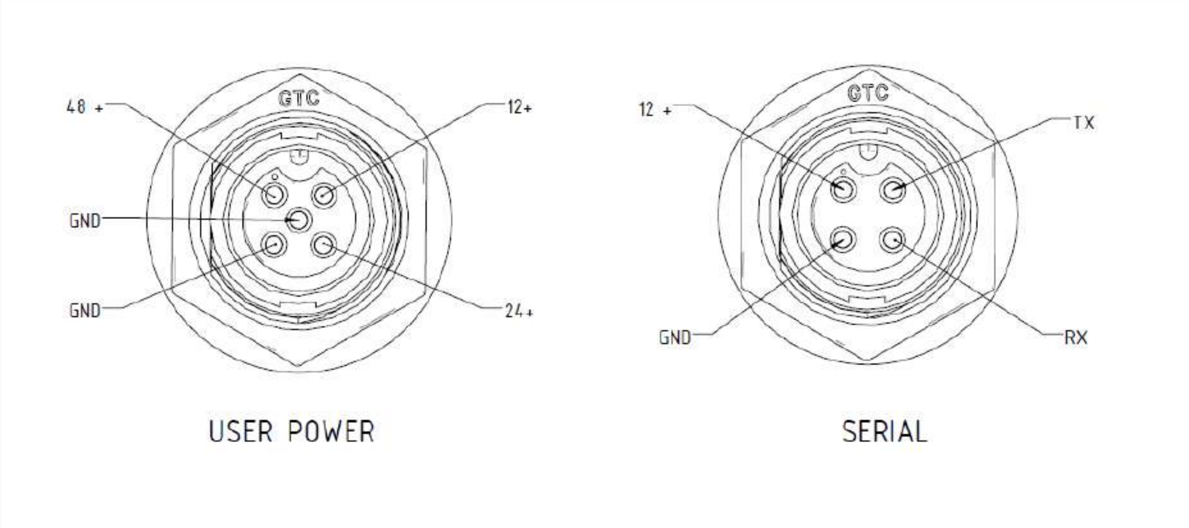
Except for bus-powered USB cameras, most payloads have separate leads for power and data.

Top Plate Power and Data Connections Connections

Description	Notes	Mating Connector (CPR Item)	Mating Connector (Manufacturer Item)
USER POWER	12 V, 24 V, 48 VBAT; each fused to 10 A	015016	RR-153274-23050
USB 3.0			Male USB type-A
ETHERNET RJ45			Male RJ45
SERIAL		015015	RR-153274-23040



Warthog user panel



Warthog connector pinouts

The user power receptacles located in the User Panel are capable of supplying 12 VDC, 24 VDC, and unregulated battery voltage (48 - 60 VDC) for powering Warthog's payloads. See the figures below for connector locations and pinouts. Ensure you select a contact that's appropriate for the gauge of wire used.

POWER

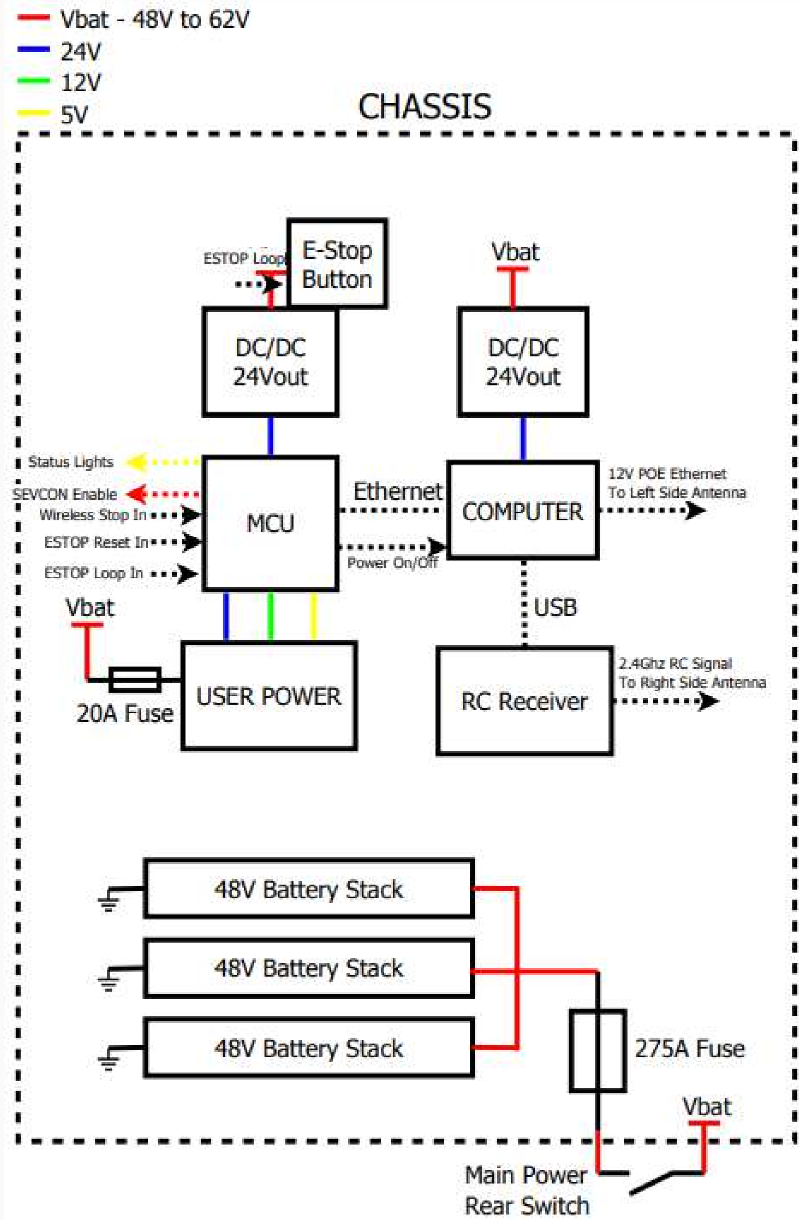
The user power breakouts are intended for small sensors, like lidar and cameras. Warthog's 48 V battery pack is fused at 275 A (13200 W). We often add DC-DC and DC-AC supplies inside the Warthog to power large devices like manipulators, lights, and other custom integration attachments.

Refer to the process described on our Power Supplies page to help you choose a power supply or inverter.

DANGER

Always use adequately sized fuses for your selected wire thickness to prevent electrical fires.

System Electrical Diagram



Warthog chassis electrical system

Software Integration

ROS has a large ecosystem of sensor drivers, some of which include pre-made URDF descriptions and even simulation configurations. Refer to Sensors supported by ROS.

For the best experience, consider purchasing supported accessories from Clearpath Robotics for your robot, which will include simulation, visualization, and driver support.

Refer to the following for more details:

- Computers

- Sensors
- Manipulators
- Accessories

Support

Clearpath is committed to your success. Please get in touch with us and we will do our best to get you rolling again quickly: support@clearpathrobotics.com.

To get in touch with a salesperson regarding Clearpath Robotics products, please email sales@clearpathrobotics.com.

If you have an issue that is specifically about ROS and is something which may be of interest to the broader community, consider asking it on https://answers.ros.org. If you do not get a satisfactory response, please ping us and include a link to your question as posted there. If appropriate, we will answer in the ROS Answers context for the benefit of the community.

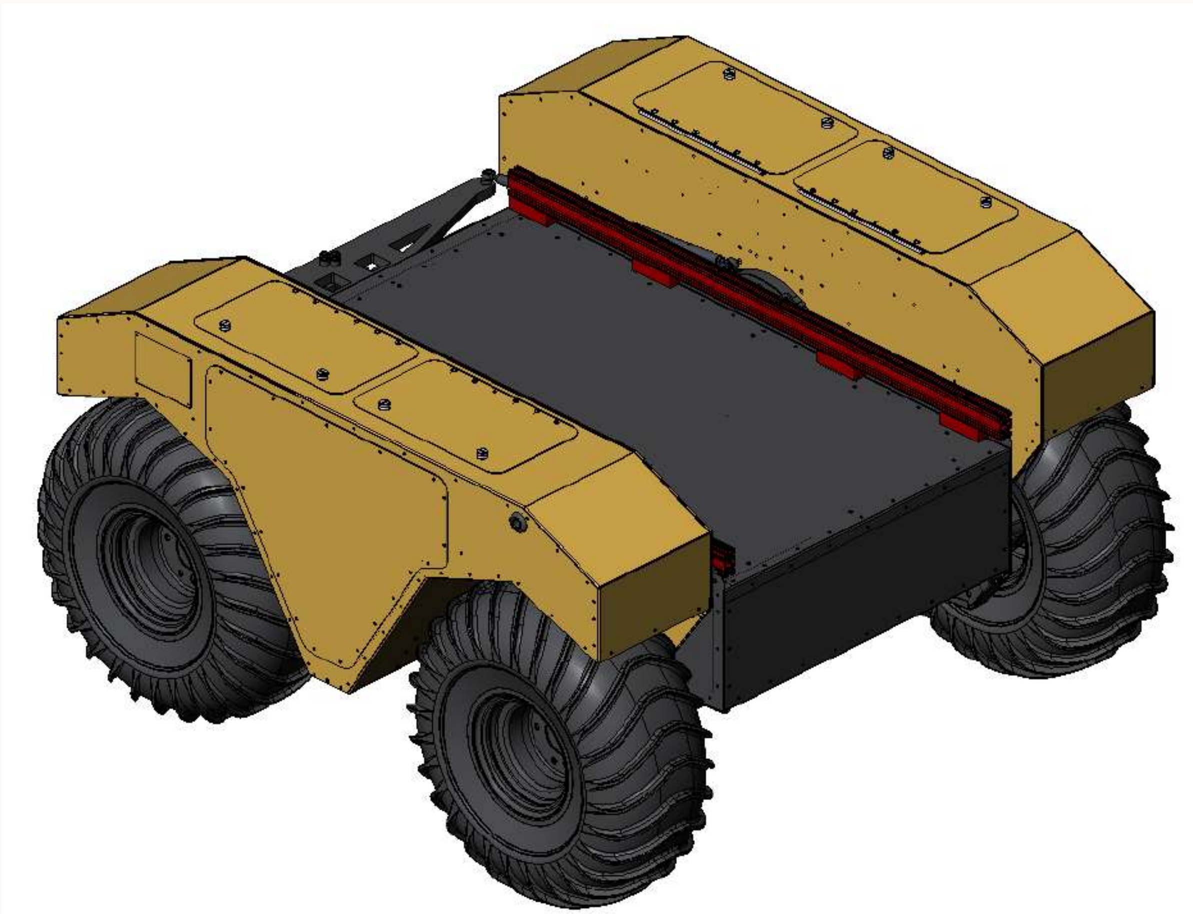
*Last updated on **Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Extrusion Rails



This kit adds 25 mm X 50 mm aluminum extrusions to both sides of the Warthog. The extrusions are attached to the Warthog using M6×1 stainless steel screws, and stainless steel rivets with a Ø5 mm shank.

You can attach your own custom brakets to the extrusions using M5×0.8 T-slot nuts, or these other useful off-the-shelf items:

Description	CPR Item	Manufacturer	Manufacturer Item	
Nut, T-Slot—M5×0.8 X 5.8 X 10, Steel	028463	Misumi	HNTTSS5-5	
Bracket, T-Slot, 90°—5.8, 20, Aluminum, Black	028464	Misumi	HBLFSSWB5	
Bracket, Flat, 90°—Stainless Steel		Misumi	SHPTUL5	
Beam, T-slot—25 X 25, Aluminum, Black	Configurable Lengths	Misumi	HFSB5-2525	
Beam, T-slot—25 X 50, Aluminum, Black	Configurable Lengths	Misumi	HFSB5-2550	

Misumi also offers a range of configurable products that can help with your integration.

Please email sales@clearpathrobotics.com if you want Clearpath's team to design a bracket structure for your application.

Sales Kits

Description	Sales Kit
Installation—Extrusion, 25 X 50, 1217	028454

Parts List

ID	Description	CPR item	Manufacturer Item	Quantity
1	Block—76.2 X 25.4 X 25.4 X M10×1.5	028452		8

ID	Description	CPR item	Manufacturer Item	Quantity
2	Insert, Thread, Adhesive—M6×1 X M10×1.5 X 10.5, Stainless Steel	028458	97120A230	8
3	Beam, T-slot—25 X 50, 1217, Aluminum, Black	028459		2
4	Rivet, Blind, Dome—Ø5 X 20 - 25, Stainless Steel	028460	97525A247	16
5	Screw, Cap, Socket Head—M6×1 X 35, Stainless Steel	023186	92290A334	8

Tools Required

- Drill, with Ø6 mm drill bit
- Rivet installation tool (Refer to our Physical Tools page for a suggested rivet installation tool)
- Large flat-head screwdriver, or a thread insert installation tool
- Hex Key, 5 mm
- Torque wrench *(optional, for safety applications)*
- Loctite 243 *(optional, for high vibration applications)*

Installation

Step 1

Use the drill to remove the 16 existing rivets on the top deck of the Warthog. You can use the Extrusion Beam *(ID3)* as a reference of what rivets to remove.

> **CAUTION**
>
> Just remove the rivet's head with the drill. We do not want to enlarge the holes in the Warthog's frame.

> **NOTE**

> This kit only includes 4 Blocks *(ID1)* per Beam. You can install up to 8 Blocks per beam if your application requires a stiffer base.

> **NOTE**
>
> Part of the rivet will fall into the Warthog's steel tube frame. We intend for you to leave these scrap bits of rivet within the Warthog's frame.

Step 2

Place the Blocks *(ID1)* above the holes where you had removed the rivets.

Step 3

Insert the new Rivets *(ID4)* through the Blocks *(ID1)* and into the Warthog's frame.

Step 4

Fasten the rivets using your rivet installation tool.

Step 5

Install a Thread Insert *(ID2)* into each of the Blocks *(ID1)* using your standard screwdriver or thread insert installation tool. Make sure the installed Thread Inserts are below the top surface of the Blocks.

Step 6

Place the two Beams *(ID3)* onto the Warthog's Blocks *(ID1)*.

> **NOTICE**
>
> Notice that the Beams have a small diameter hole on one side, and a larger diameter hole on the other side. The small holes face down, the large holes face up.

Step 7

Install a Screw *(ID5)* into each of the Blocks *(ID1)*. This will fasten the Beams *(ID3)* onto the Warthog. Torque these screws to 5 N·m.

NOTE

You may add a small amount of thread locker to these screws if you are concerned about them loosening from high vibration applications.

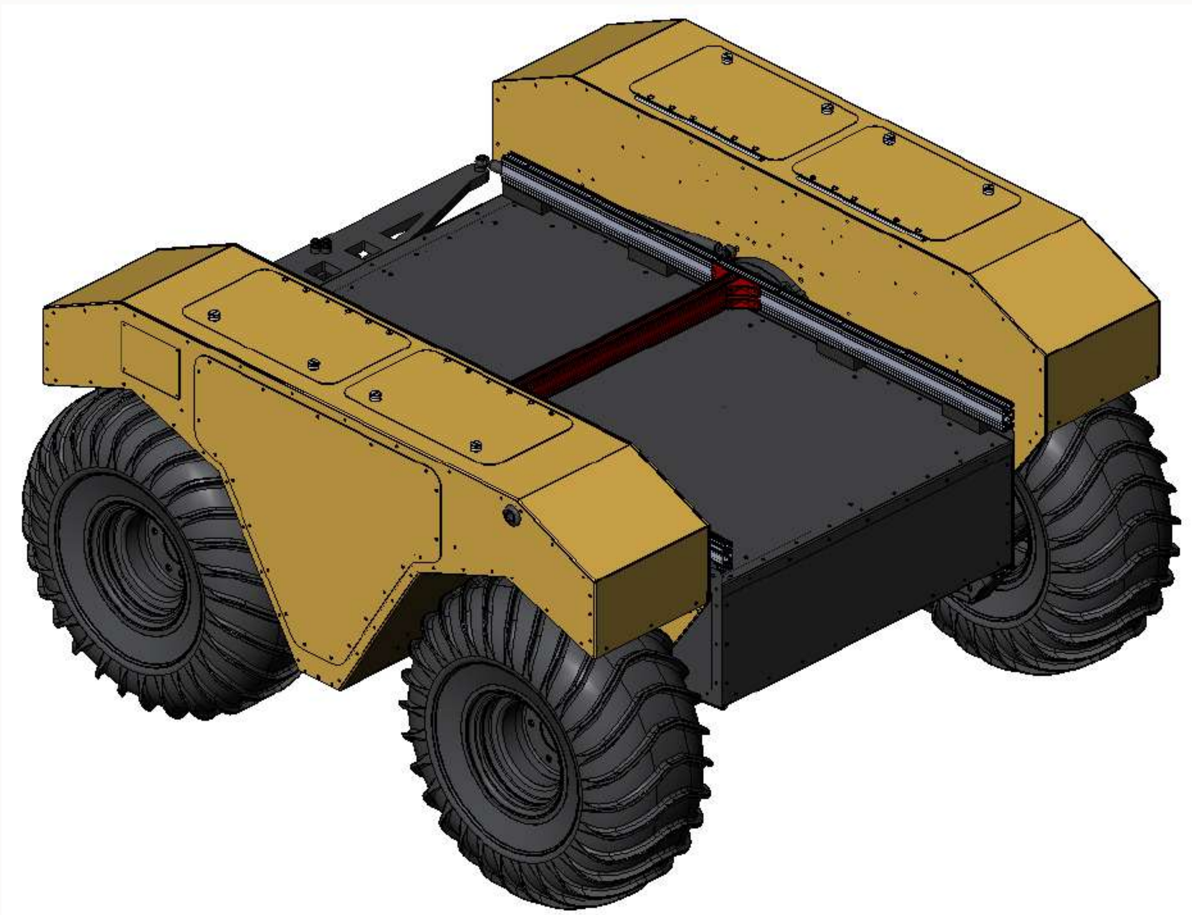
*Last updated on **Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Extrusion Cross-Bar



This is a 25 mm X 50 mm aluminum extrusion that mounts between the Warthog's Extrusion Rails. The Cross-Bar is installed with T-slot nuts and 90° brackets. The Cross-Bar can be attached anywhere along the length of the Warthog's Extrusion Rails.

Sales Kits

NOTE

This kit requires a Warthog with Extrusion Rails. Refer to the Extrusion Rails page for further details.

Description	Sales Kit
Installation—Extrusion Cross-Bar, 25 X 50, 655	028455

Parts List

ID	Description	CPR item	Quantity
1	Beam, T-slot—25 X 50, 655, Aluminum, Black	028465	1
2	Bracket, T-Slot, 90°—5.8, 20, Aluminum, Black	028464	8
3	Nut, T-Slot—M5×0.8 X 5.8 X 10, Steel	028463	32
4	Screw, Cap, Socket Head—M5×0.8 X 10, Stainless Steel	023150	32

Tools required

- Hex Key, 4 mm
- Torque wrench *(optional, for safety applications)*
- Loctite 243 *(optional, for high vibration applications)*

Installation

Step 1

Attach 8 Brackets *(ID2)* to the ends of the Beam *(ID1)*. Use 2 T-slot Nuts *(ID3)* and 2 Screws *(ID4)* per Bracket.

NOTE

> Torque these screws to 5 N·m if you are concerned about them loosening from high vibration applications.

> **NOTE**
>
> You may add a small amount of thread locker to these screws if you are concerned about them loosening from high vibration applications.

Step 2

Add a T-slot Nut *(ID3)* and Screw *(ID4)* to each of the remaining holes of the Brackets *(ID2)*. Leave these Screws loose, in anticipation of the next step.

> **NOTE**
>
> You may add a small amount of thread locker to these screws if you are concerned about them loosening from high vibration applications.

Step 3

Slide the Cross-Bar assembly between the Warthog's Extrusion Rails. Make sure all 16 of the loose T-slot nuts are guided into the Extrusion Rails.

Step 4

Slide the Cross-Bar assembly to your intended final position on the Warthog per your application.

Step 5

Tighten all 16 Screws.

> **NOTE**
>
> Torque these screws to 5 N·m if you are concerned about them loosening from high vibration applications.

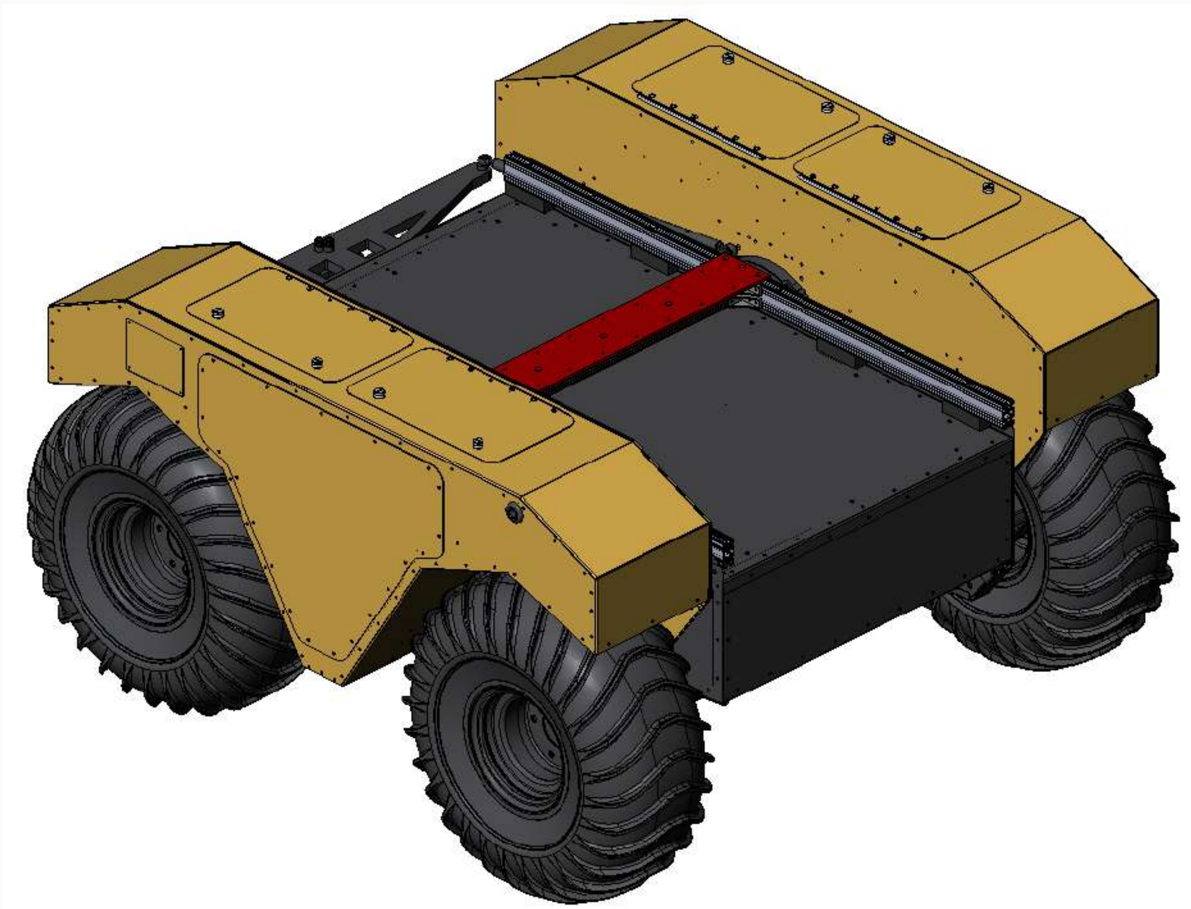
Last updated on ***Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Top Plate



The top plates for the Warthog include M5×0.8 threaded holes that match the Clearpath PACS™ standard.

> **NOTE**
>
> These top plates require extrusion rails to be mounted on the Warthog.

> **NOTE**
>
> You can install an extrusion cross-bar under the top plates to add stiffness.

Sales Kits

> **NOTE**
>
> This kit requires a Warthog with Extrusion Rails. Refer to the Extrusion Rails page for further details.

> **TIP**
>
> This Top Plate can be stiffened by adding a Crossbar underneath it. Refer to the Crossbar page for further details.

Description	Sales Kit
Installation, Kit, Attachment Top Plate—Warthog	028462

Parts List

ID	Description	CPR item	Quantity
1	Plate—110 X 704.5	028451	1
2	Nut, T-Slot—M5×0.8 X 5.8 X 10, Steel	028463	9
3	Screw, Flat Head—M5×0.8 X 14, Stainless Steel	023471	9

Tools required

- Hex Key, 3 mm
- Torque wrench *(optional, for safety applications)*
- Loctite 243 *(optional, for high vibration applications)*

Installation

Step 1

Insert T-slot nuts *(ID2)* into the Warthog's Extrusion Rail and Cross-Bar.

> **NOTE**
>
> You may need to loosen the Cross-Bar's brackets to get access to the end for inserting the T-slot nuts.

Step 2

Place the Plate *(ID1)* ontop of the Extrusion Rails and Crossbar. The Plate's countersink features should be facing up.

Step 3

Install Screws *(ID3)* into the Plate's holes; threading the Screws into the T-slop nuts from *Step 1*.

> **NOTE**
>
> Torque these screws to 5 N·m if you are concerned about them loosening from high vibration applications.

> **NOTE**
>
> You may add a small amount of thread locker to these screws if you are concerned about them loosening from high vibration applications.

*Last updated on **Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Warthog Tutorials

Warthog Overview

Introduction

Clearpath Robotics Warthog is a rugged, all-terrain unmanned ground vehicle. These tutorials will assist you with setting up and operating your Warthog. The tutorial topics are listed in the right column and presented in the suggested reading order.

For more information or to receive a quote, please visit us online.

NOTE

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

NOTE

These tutorials specifically target Warthog robots running Ubuntu 20.04 with ROS Noetic, as it is the standard OS environment for Warthog. If instead you have an older Warthog robot running Ubuntu 18.04 with ROS Melodic, please follow this tutorial to upgrade the robot OS environment to Ubuntu 20.04 with ROS Noetic.

Warthog ROS Packages provides the references for the software packages and key ROS topics.

Warthog Software Setup outlines the steps for setting up the software on your Warthog robot and optionally on a remote computer.

Using Warthog describes how to simulate and drive your Warthog. Simulation is a great way for most users to learn more about their Warthog; understanding how to effectively operate Warthog in simulation is valuable whether you are in the testing phase with software you intend to ultimately deploy on a physical Warthog or you do not have one and are simply exploring the platform's capabilities. Driving Warthog covers how to teleoperate Warthog using the remote control, as well as safety procedures for operating the physical robot. Anyone working with a physical robot should be familiar with this section.

Warthog Tests outlines how to validate that your physical Warthog is working correctly.

Advanced Topics covers items that are only required in atypical situations.

Warthog ROS Packages

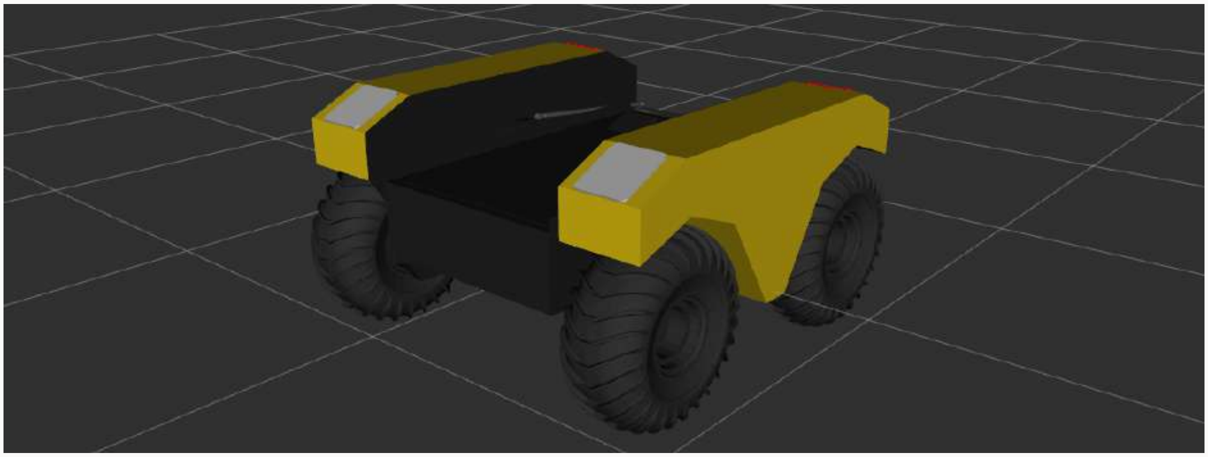
Warthog fully supports ROS; all of the packages are available in Warthog Github.

Description Package

The warthog_description repository provides a URDF model of Warthog.

Warthog's URDF model can be visualized in RViz. Once you have installed the desktop software in an upcoming tutorial, you will be able to run:

```
roslaunch warthog_viz view_model.launch
```



Warthog model

Environment Variables

Warthog can be customized and extended through the use of several environment variables. The details are in the warthog_description repository. Some of the most important ones are listed below.

Variable	Default	Description
`WARTHOG_ARM_MOUNT`	`0`	If enabled, a link called "arm_mount_link" will be created on the bulkhead
`WARTHOG_BULKHEAD`	`0`	Switch for enabling a large, rigid bulkhead on the front of the warthog for mount accessories like arms
`WARTHOG_IMU_RPY`	`0 0 0`	Compound rotations in radians of the IMU
`WARTHOG_IMU_XYX`	`0 0 0`	Position of the IMU in meters
`WARTHOG_JOY_TELEOP`	`0`	Switch to enable teleop control of the platform
`WARTHOG_NAVSAT_SMART6`	`0`	Is the platform equipped with a Smart7 GPS?
`WARTHOG_NAVSAT_SMART6_BAUD`	`57600`	Sets the baud rate for serial communication with the GPS module

Variable	Default	Description
`WARTHOG_NAVSAT_SMART6_MOUNT`	`navsat`	The mount on the robot model that the GPS antenna is mounted to. See the Warthog URDF and `WARTHOG_URDF_EXTRAS` for more details on mount points.
`WARTHOG_NAVSAT_SMART6_OFFSET`	`0 0 0`	Position of the Smart 6 GPS in meters
`WARTHOG_NAVSAT_SMART6_PORT`	`/dev/ttyS1`	The serial port that the GPS module communicates over
`WARTHOG_NAVSAT_SMART6_RPY`	`0 0 0`	Compound rotations in radians of the Smart 6 GPS
`WARTHOG_OFFBOARD_STOP`	`false`	Is a remote e-stop setup on the platform?
`WARTHOG_TRACKS`	`0`	Used to specify that Warthog is equipped with tracks instead of wheels
`WARTHOG_TWIST_MUX_EXTRAS`	`0`	Add additional sources to be controlled by the twise mux of the platform
`WARTHOG_URDF_EXTRAS`	`empty.urdf`	Path to a URDF file with additional modules connected to the robot

Configurations

As an alternative to individually specifying each accessory, some fixed configurations are provided in the package. These can be specified using the `config` arg to `description.launch`, and are intended especially as a convenience for simulation launch.

Config	Description
`arm_mount`	Includes mounting points for am arm payload
`base`	Base Warthog, includes IMU and GPS

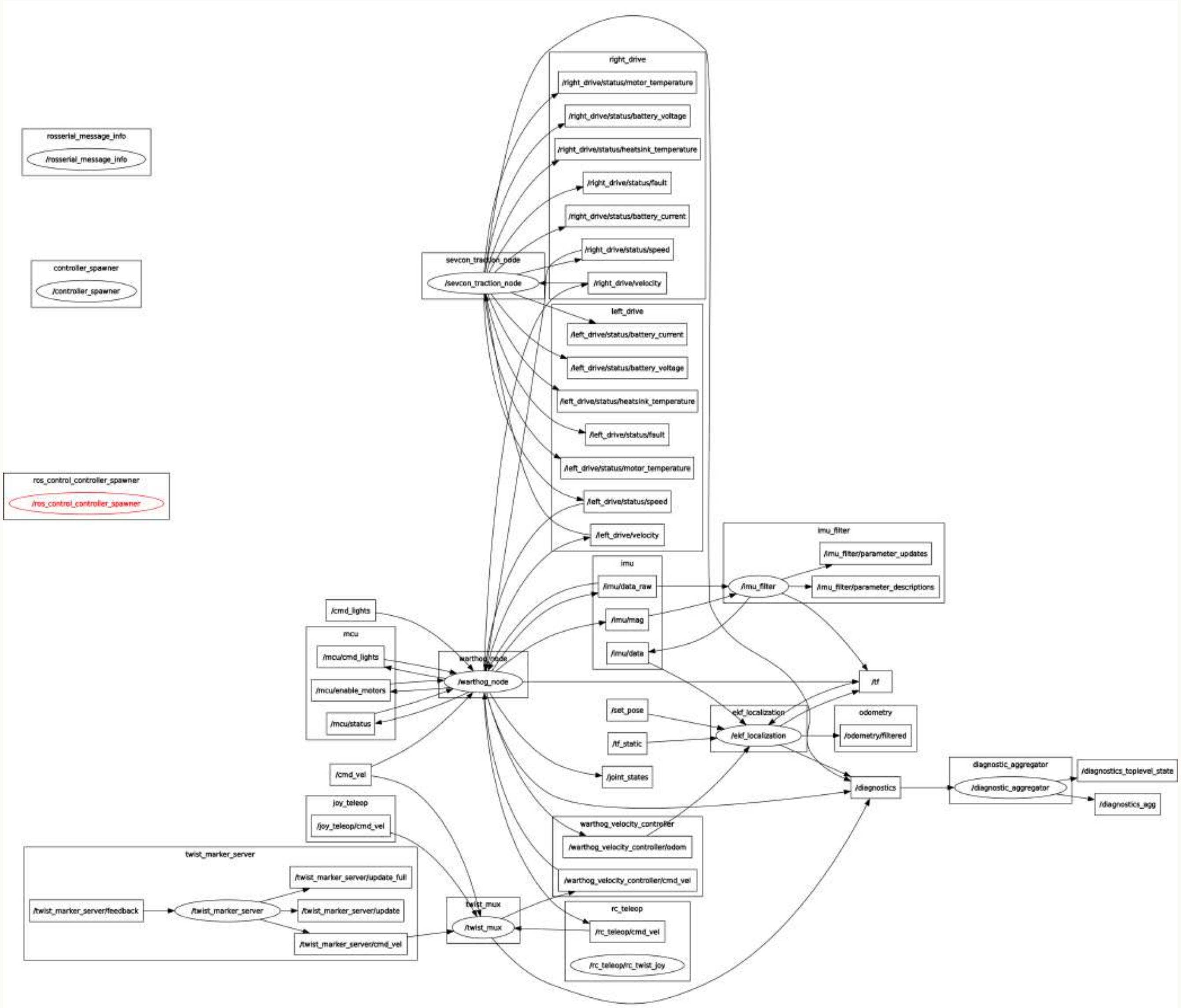
Config	Description
`bulkhead`	Includes a mounting point for additional payload options
`empty`	Includes no accessories at all

Key ROS Topics

You can view all topics that are active using `rostopic list`. The most important topics are summarized in the table below.

Topic	Message Type	Purpose
`/cmd_vel`	`geometry_msgs/Twist`	Input to Warthog's kinematic controller. Publish here to make Warthog go.
`/odometry/filtered`	`nav_msgs/Odometry`	Published by robot_localization, a filtered localization estimate based on wheel odometry (encoders) and integrated IMU.
`/imu/data`	`sensor_msgs/IMU`	Published by imu_filter_madgwick, an orientation estimate based on Warthog's internal gyroscope, accelerometer, and magnetometer.
`/status`	`warthog_msgs/Status`	Low-frequency status data for Warthog's systems. This information is republished in human readable form on the diagnostics topic and is best consumed with the Robot Monitor.
`/cmd_lights`	`warthog_msgs/Lights`	Input to controlling the Warthog's body lights when not in an error state.

Topic	Message Type	Purpose
`/SIDE/speed`	`std_msgs/Float64`	Input velocity for each motor where SIDE is either left or right. This should not be published to directly, commands from /cmd_vel will be converted to this.
`/SIDE/status/speed`	`std_msgs/Float64`	Reported velocity from each motor's encoder where SIDE is either left or right.
`/SIDE/status/fault`	`std_msgs/Bool`	Reported state from each motor controller where SIDE is either left or right.
`/SIDE/status/motor_temperature`	`std_msgs/Int32`	Reported temperature from each motor controller where SIDE is either left or right.



Warthog node and topic connections

Warthog Software Setup

Backing Up Robot Configuration

Upgrading your Clearpath Warthog to ROS Noetic from older ROS distributions is a straightforward process; however it's important to understand that each Warthog is different, having undergone customization to your specifications. For more complete upgrade instructions see this guide.

Please take the time to understand what these modifications are, and how to recreate them on your fresh install of Ubuntu Focal/ROS Noetic.

Performing a Backup

1. As a fail-safe, please make an image of your robot's hard drive. You should always be able to restore this image if you need to revert back to your previous configuration.
   - The easiest approach may be to either connect a removable (USB or similar) hard drive to the robot's computer, or to unplug the robot's hard drive and insert it into a computer or workstation.
   - You can then use a tool such as CloneZilla or `dd` to write a backup image of your robot's hard drive onto another hard drive.
   - Alternatively, you can simply replace the robot computer's hard drive, reserving the drive and installing a new one to use with Noetic.
2. There are several places in the filesystem you should specifically look for customizations for your robot:

Location	Description
`/etc/network/interfaces` or `/etc/netplan/*`	Your robot may have a custom network configuration in this file.
`/etc/ros/*/*-core.d/*.launch`	Will contain `base.launch` and `description.launch`, may contain custom launch files for your robot configuration. Replaced by `ros.d` in newer versions.
`/etc/ros/*/ros.d/*.launch`	Will contain `base.launch` and `description.launch`, may contain custom launch files for your robot configuration. Replaces `*-core.d` in newer versions.
`/etc/ros/setup.bash`	May contain environment variables for your configuration.

3. Please save all the files listed above and use them as a reference during Noetic configuration.

Installing and Configuring Robot Software

> **NOTE**
>
> If you are upgrading your Warthog from an older version of ROS, please refer to our upgrade instructions here and here.

Installing Warthog Software

> **NOTE**
>
> The physical Warthog robot comes pre-configured with ROS and the necessary Warthog packages already installed; therefore, you will only need to follow the instructions below if you are re-installing software on the Warthog.

There are three methods to install software on the physical robot.

The preferred method is using the Clearpath Robotics ISO image, which is covered in this section.

The second method is using Debian (.deb) packages, which is also covered in this section.

The final approach is installing from source by directly cloning Clearpath Robotics Github repositories and building them in your ROS (catkin) workspace; however, this method is not covered in this section.

Install from ISO Image

> **CAUTION**
>
> Installing with the Clearpath Robotics ISO image will completely wipe data on the robot's computer, since the ISO image will install Ubuntu 20.04 (Focal), ROS Noetic, and robot-specific packages.

> **NOTE**
>
> The Clearpath Robotics ISO image only targets Intel-family computers (`amd64` architecture). If your robot runs on an Nvidia Jetson computer, see Jetson Software for software installation details.

Clearpath provides a lightly customized installation image of Ubuntu 20.04 "Focal" that automatically pulls in all necessary dependencies for the robot software. To install the software on a physical robot through the Clearpath Robotics ISO image, you will first need a USB drive of at least 2 GB to create the installation media, an ethernet cable, a monitor, and a keyboard.

1. Download the appropriate Noetic ISO image for your platform.
2. Copy the image to a USB drive using `unetbootin`, `rufus`, `balena etcher`, or a similar program. For example:

   ```
   sudo unetbootin isofile="clearpath-universal-noetic-amd64-0.4.17.iso"
   ```

3. Connect your robot computer to internet access (via wired Ethernet), a keyboard, and a monitor. Make sure that the robot is connected to shore power (where applicable) or that the robot's battery is fully charged.

   > **CAUTION**
   >
   > The next step wipes your robot's hard drive, so make sure you have that image backed up.

4. Boot your robot computer from the USB drive and let the installer work its magic. If asked for a partitioning method choose `Guided - use entire disk and set up LVM`.

   > **NOTE**
   >
   > You may need to configure the computer's BIOS to prioritize booting from the USB drive. On most common motherboards, pressing `Delete` during the initial startup will open the BIOS for configuration.

5. The setup process will be automated and may take a long time depending on the speed of your internet connection.

6. Once the setup process is complete, the computer will turn off. Please unplug the USB drive and turn the computer back on.

7. On first boot, the username will be `administrator` and the password will be `clearpath`. You should use the `passwd` utility to change the `administrator` account password.

8. To set up a factory-standard robot, ensure all your peripherals are plugged in, and run the following command, which will configure a ros upstart service, that will bring up the base robot launch files on boot. The script will also detect any standard peripherals (IMU, GPS, etc.) you have installed and add them to the service.

   Husky | Jackal | Dingo | Ridgeback | Warthog | Boxer

   ```
   rosrun husky_bringup install
   sudo systemctl daemon-reload
   ```

9. Finally, start ROS for the first time. In terminal, run:

```
sudo systemctl start ros
```

Installing from Debian Packages

If you are installing software on a physical robot through Debian packages, you will first need to ensure that the robot's computer is running Ubuntu 20.04 (Focal) and ROS Noetic.

1. Before you can install the robot packages, you need to configure Ubuntu's APT package manager to add Clearpath's package server.

   i. Install the authentication key for the packages.clearpathrobotics.com repository. In terminal, run:

   ```
   wget https://packages.clearpathrobotics.com/public.key -O - | sudo apt-key
   add -
   ```

   ii. Add the debian sources for the repository. In terminal, run:

   ```
   sudo sh -c 'echo "deb https://packages.clearpathrobotics.com/stable/ubuntu
   $(lsb_release -cs) main" > /etc/apt/sources.list.d/clearpath-latest.list'
   ```

   iii. Update your computer's package cache. In terminal, run:

   ```
   sudo apt-get update
   ```

2. After the robot's computer is configured to use Clearpath's debian package repository, you can install the robot-specific packages.

   i. On a physical robot, you should only need the robot packages. In terminal, run:

   Husky Jackal Dingo Ridgeback Warthog Boxer

   `sudo apt-get install ros-noetic-husky-robot`

   ii. Install the `robot_upstart` job and configure the bringup service so that ROS will launch each time the robot starts. In terminal, run:

Husky Jackal Dingo Ridgeback Warthog Boxer

```
rosrun husky_bringup install
sudo systemctl daemon-reload
```

iii. Finally, start ROS for the first time. In terminal, run:

```
sudo systemctl start ros
```

Testing Base Configuration

You can check that the service has started correctly by checking the logs:

```
sudo journalctl -u ros
```

Your Warthog should now be accepting commands from your joystick (see next section). The service will automatically start each time you boot your Warthog's computer.

Setting up Warthog's Network Configuration

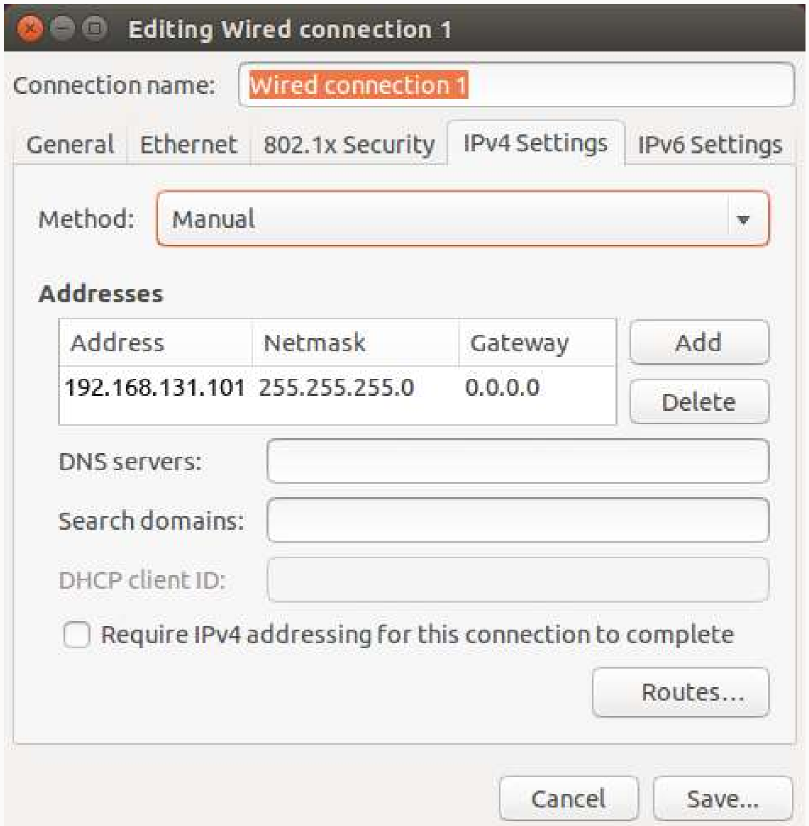
Warthog is normally equipped with a combination Wi-Fi + Bluetooth module. If this is your first unboxing, ensure that Warthog's wireless antennae are firmly screwed onto the chassis.

First Connection

By default, Warthog's Wi-Fi is in client mode, looking for the wireless network at the Clearpath factory.

Set your laptop's ethernet port to a static IP such as `192.168.131.101`. To do this in Ubuntu, follow the steps below:

1. Click on the Wi-Fi icon in the upper-right corner of your screen, and select Edit Connections.
2. In the Network Connections window, under Ethernet, select your wired connection and then click Edit.
3. Select the IPv4 Settings tab and then change the Method to Manual.
4. Click the Add button to add a new address.
5. Enter a `192.168.131.101` as the static IP under the Address column, and enter `255.255.255.0` under the Netmask column, and then select Save.



Static IP Configuration (Ubuntu)

The next step is to connect to your robot via SSH. To do so execute the following in a terminal window:

```
ssh administrator@192.168.131.1
```

You will be promoted to enter a password. The default password is `clearpath` and you set a new password on first connection.

Changing the Default Password

> **NOTE**
>
> All Clearpath robots ship from the factory with their login password set to `clearpath`. Upon receipt of your robot we recommend changing the password.

To change the password to log into your robot, run the following command:

```
passwd
```

This will prompt you to enter the current password, followed by the new password twice. While typing the passwords in the `passwd` prompt there will be no visual feedback (e.g. "*" characters).

To further restrict access to your robot you can reconfigure the robot's SSH service to disallow logging in with a password and require SSH certificates to log in. This tutorial covers how to configure SSH to disable password-based login.

Wi-Fi Setup

Now that you are connected via SSH over a wired connection using the steps above, you can set up your robot's computer (running Ubuntu 20.04) to connect to a local Wi-Fi network. (For legacy systems running Ubuntu 18.04, use `wicd-curses` instead.)

Clearpath robots running Ubuntu 20.04 and later use `netplan` for configuration of their wired and wireless interfaces. To connect your robot to your wireless network using `netplan`, create the file `/etc/netplan/60-wireless.yaml` and fill in the following:

```
network:
  wifis:
    # Replace WIRELESS_INTERFACE with the name of the wireless network device, e.g.
wlan0 or wlp3s0
    # Fill in the SSID and PASSWORD fields as appropriate.  The password may be
included as plain-text
    # or as a password hash.  To generate the hashed password, run
    #   echo -n 'WIFI_PASSWORD' | iconv -t UTF-16LE | openssl md4 -binary | xxd -p
    # If you have multiple wireless cards you may include a block for each device.
    # For more options, see https://netplan.io/reference/
    WIRELESS_INTERFACE:
      optional: true
      access-points:
        SSID_GOES_HERE:
          password: PASSWORD_GOES_HERE
      dhcp4: true
      dhcp4-overrides:
        send-hostname: true
```

Modify the following variables in the file:

1. Replace `WIRELESS_INTERFACE` with the name of the robot's Wi-Fi interface (e.g. wlan0, wlp2s0, or wlp3s0).
2. Replace `SSID_GOES_HERE` with the name of the local Wi-Fi network.
3. Replace `PASSWORD_GOES_HERE` with the password of the local Wi-Fi network.

Once you are done modifying the file, save it by pressing `CTRL + O`, then `ENTER`. Close the file by pressing `CTRL + X`.

Then, run the following to bring up the Wi-Fi connection:

```
sudo netplan apply
```

You can validate that the connection was successful and determine the IP address of the Wi-Fi interface by running:

```
ip a
```

A list of network connections will be displayed within the terminal. Locate the wireless network and make note of its IP address.

Now that you know robot's wireless IP address, you may now exit the Ethernet SSH session by executing `exit`.

Remove the Ethernet cable and close up your robot. Now you can SSH into your robot over the wireless network. To do so, execute:

`ssh administrator@<IP_OF_ROBOT>`

SSH sessions allow you to control your robot's internal computer. You can do various things such as download packages, run updates, add/remove files, transfer files etc.

Installing Remote Computer Software

> **NOTE**
>
> This step is optional.

It is often convenient to use a Remote Computer (eg. laptop) to command and observe your robot. To do this, your Remote Computer must be configured correctly.

1. Perform a basic ROS installation. See here for details.

2. Install the desktop packages:

Husky | Jackal | Dingo | Ridgeback | Warthog | Boxer

```
sudo apt-get install ros-noetic-husky-desktop
```

3. Configure Remote ROS Connectivity.

   Click to expand

4. From your Remote Computer, try launching RViz, the standard ROS robot visualization tool:

   Husky **Jackal** **Dingo** **Ridgeback** **Warthog** **Boxer**

   ```
   roslaunch husky_viz view_robot.launch
   ```

   From within RViz, you can use interactive markers to drive your robot, you can visualize its published localization estimate and you can visualize any attached sensors which have been added to its robot description XML *URDF*.

Adding a Source Workspace

Configuring non-standard peripherals requires a source workspace on the robot computer.

> **TIP**
>
> The instructions below use `cpr_noetic_ws` as the workspace name. You can choose a different workspace name and substitute it in the commands below.

1. Create a new workspace:

   ```
   mkdir -p ~/cpr_noetic_ws/src
   ```

2. Add any custom source packages to the `~/cpr_noetic_ws/src` directory.

3. After adding your packages, make sure any necessary dependencies are installed:

   ```
   cd ~/cpr_noetic_ws/
   rosdep install --from-paths src --ignore-src --rosdistro noetic -y
   ```

4. Build the workspace:

```
cd ~/cpr_noetic_ws/
catkin_make
```

5. Modify your robot-wide setup file (`/etc/ros/setup.bash`) to source your new workspace instead of the base noetic install:

```
source /home/administrator/cpr_noetic_ws/devel/setup.bash
```

6. Reinitialize your environment so that it picks up your new workspace:

```
source /etc/ros/setup.bash
```

Using Warthog

Simulating Warthog

Whether you actually have a Warthog robot or not, the Warthog simulator is a great way to get started with ROS robot development. In this tutorial, we will go through the basics of starting Gazebo and RViz and how to drive your Warthog around.

Installation

To get started with the Warthog simulation, make sure you have a working ROS installation set up on your Ubuntu desktop, and install the Warthog-specific metapackages for desktop and simulation:

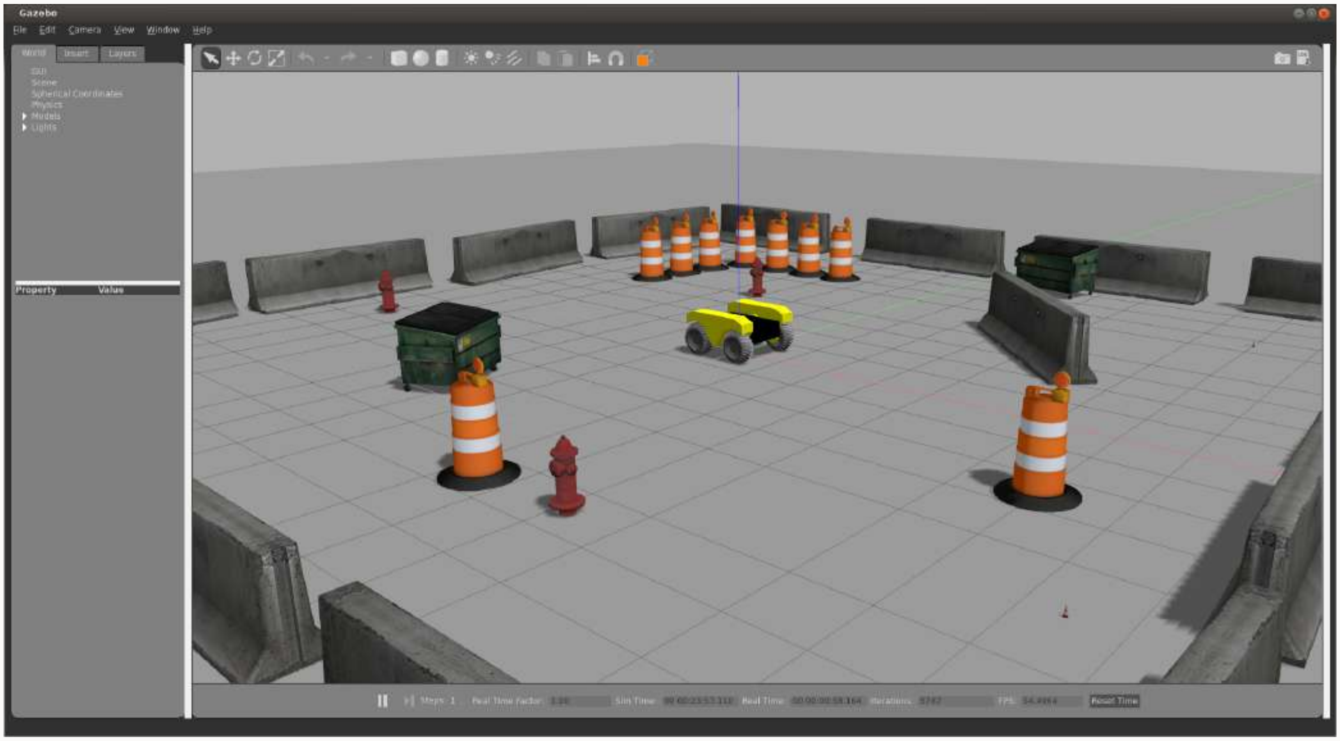
```
sudo apt-get install ros-noetic-warthog-simulator ros-noetic-warthog-desktop
```

Launching Gazebo

Gazebo is the most common simulation tool used in ROS. Warthog's model in Gazebo include reasonable approximations of its dynamics, including wheel slippage, skidding, and inertia. To launch simulated Warthog in a simple example world, run the following command:

```
roslaunch warthog_gazebo warthog_world.launch
```

You should see the following window appear, or something like it. You will see a base Warthog spawned with no additional sensors. You can adjust the camera angle by clicking and dragging while holding CTRL, ALT, or the Shift key.



Simulated Warthog in Example World

The window which you are looking at is the Gazebo Client. This window shows you the "true" state of the simulated world which the robot exists in. It communicates on the backend with the Gazebo Server, which is doing the heavy lifting of actually maintaining the simulated world. At the moment, you are running both the client and server locally on your own machine, but some advanced users may choose to run heavy duty simulations on separate hardware and connect to them over the network.

> **NOTE**
>
> When simulating, you must leave Gazebo running. Closing Gazebo will prevent other tools, such as RViz (see below) from working correctly.

> **NOTE**
>
> See also [Additional Simulation Worlds]().

Interfacing with Warthog

Both simulated and real Warthog robots expose the same ROS interface and can be interacted with in the same way.

> **NOTE**
>
> Please make sure that the desktop packages for Warthog are installed:
>
> ```
> sudo apt-get install ros-noetic-warthog-desktop
> ```

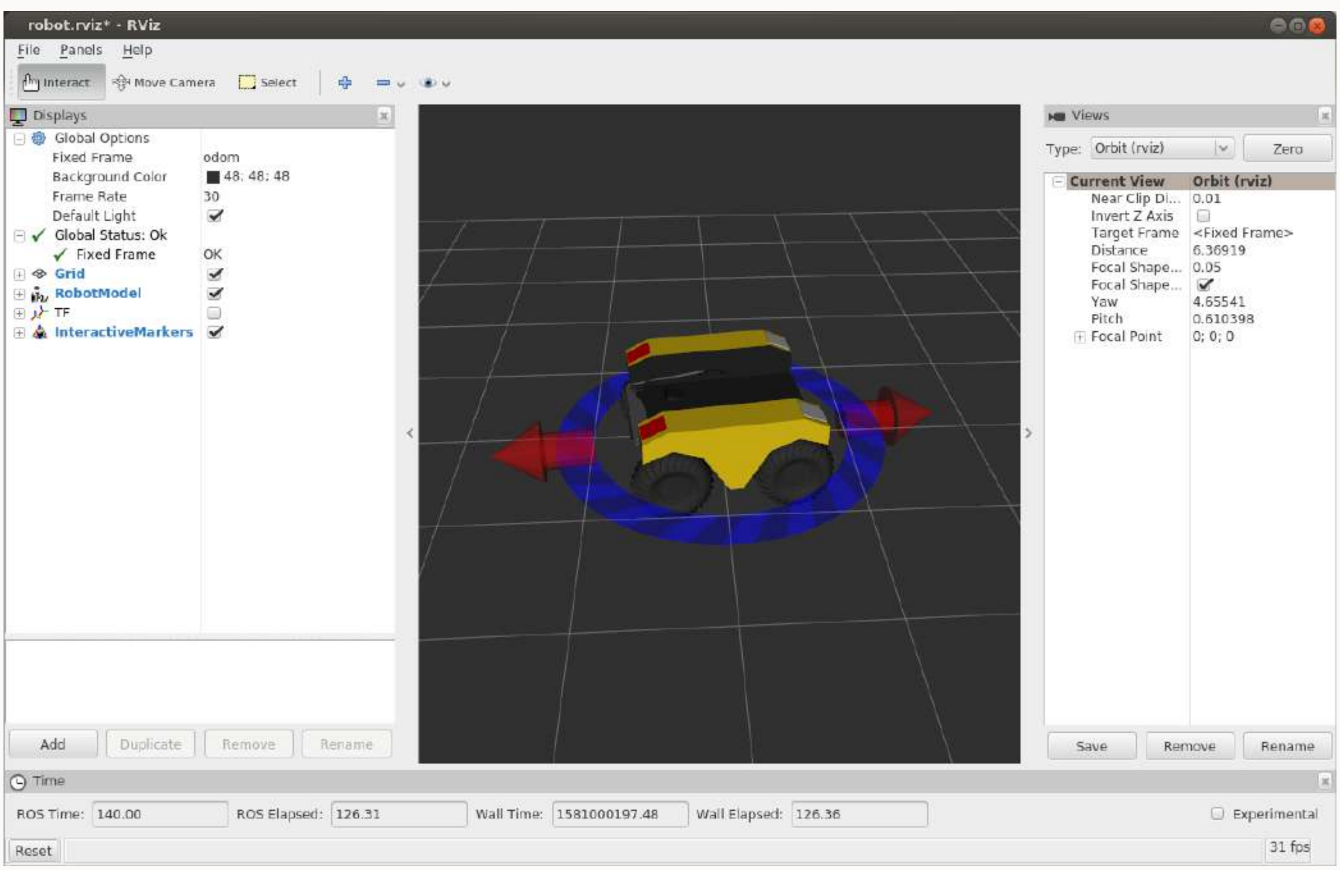
Launching RViz

The next tool we will encounter is RViz. Although superficially similar in appearance to Gazebo, RViz has a very different purpose. Unlike Gazebo, which shows the reality of the simulated world, RViz shows the robot's *perception* of its world, whether real or simulated. So while Gazebo won't be used with your real Warthog, RViz is used with both.

You can use the following launch invocation to start RViz with a predefined configuration suitable for visualizing any standard Warthog config.

```
roslaunch warthog_viz view_robot.launch
```

You should see RViz appear.



Warthog RViz

The RViz display only shows what the robot knows about its world, which presently, is nothing. Because the robot doesn't yet know about the barriers which exist in its Gazebo world, they are not shown here.

Driving with Interactive Controller

RViz will also show Warthog's interactive markers around your Warthog's model. These will appear as a blue ring and red arrows. Depending on your robot, there will also be green arrows. If you don't see them in your RViz display, select the Interact tool from the top toolbar and they should appear.

Drag the red arrows in RViz to move in the linear X direction, and the blue circle to move in the angular Z direction. If your robot supports lateral/sideways movement, you can drag the green arrows to move in the linear Y direction. RViz shows you Warthog moving relative to its odometric frame, but it is also moving relative to the simulated world supplied by Gazebo. If you click over to the Gazebo window, you will see Warthog moving within its simulated world. Or, if you drive real Warthog using this method, it will have moved in the real world.

Control

There are three ways to send your Warthog control commands:

1. Using the provided Fubata controller. Refer to the User Manual details on how to use the controller.

2. Using the RViz instance above. If you select the Interact option in the top toolbar, an interactive marker will appear around the Warthog and can be used to control speed.

3. The rqt_robot_steering plugin. Run the `rqt` command, and select **Plugins→Robot Tools→Robot Steering** from the top menu.

Warthog uses twist_mux to mix separate geometry_msgs\Twist control channels into the `warthog_velocity_controller/cmd_vel` topic.

Additional velocity channels can be defined in twist_mux.yaml, or can be piped into the lowest-priority `cmd_vel` topic.

Odometry

Warthog publishes odometry information on the `odometry/filtered` topic, as nav_msgs/Odometry messages. These are generated by ekf_localization_node, which processes data from several sensor sources using an Extended Kalman filter (EKF). This includes data from the wheel encoders and IMU (if available).

Diagnostics

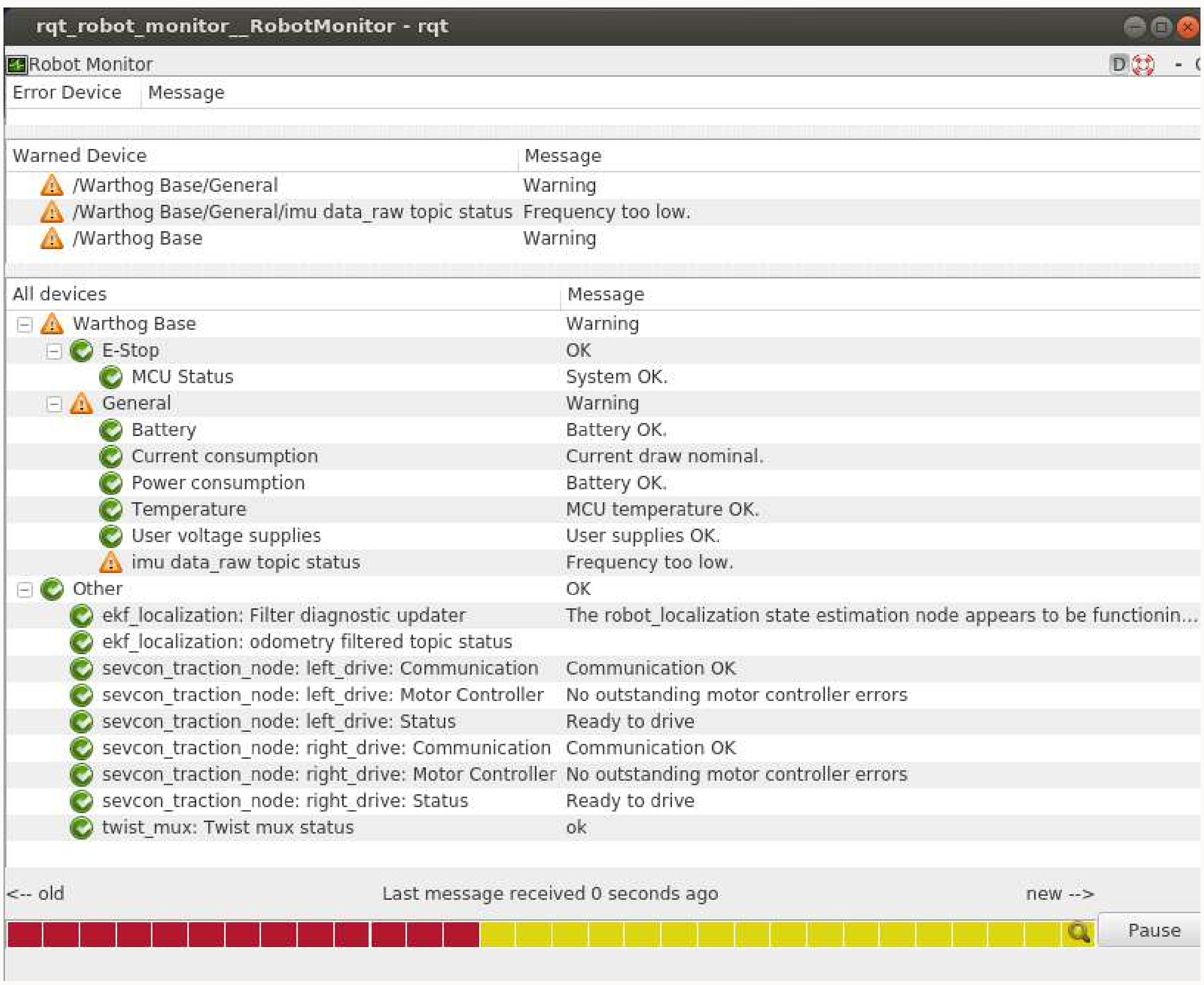
> **NOTE**
>
> Diagnostics are only applicable to real Warthog robots, not simulated ones.

Warthog provides hardware and software system diagnostics on the ROS standard `/diagnostics` topic. The best way to view these messages is using the rqt_runtime_monitor plugin. Run the `rqt` command, and select **Plugins→Robot Tools→Runtime Monitor** from the top menu.

The same information is also published as a warthog_msgs\Status message on the `Status` topic.

From your desktop, you can also launch the standard RQT Robot Monitor. This will report the diagnostic output from Warthog's self-monitoring capabilities, as shown in the figure below.

```
rosrun rqt_robot_monitor rqt_robot_monitor
```



Robot Monitor

Driving Warthog

There are three ways to drive Warthog and each way will work on a physical Warthog robot as well as on a simulated Warthog.

1. Using the interactive remote controller in RViz. See Simulating Warthog.
2. Using the controller for teleoperation. See below.
3. Publishing ROS messages. See below.

DANGER

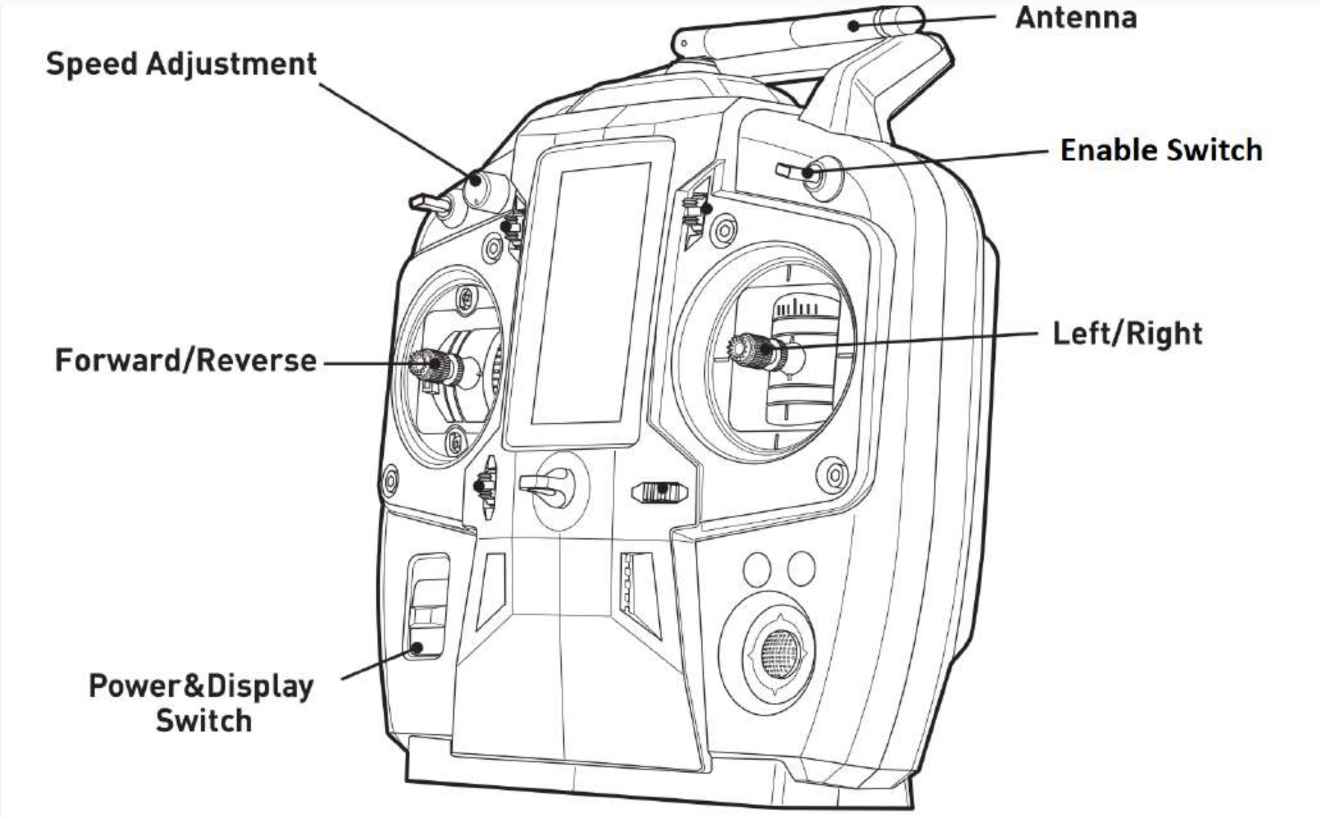
Warthog is capable of reaching high speeds. Careless driving can cause harm to the operator, bystanders, the robot, or other property. Always remain vigilant, ensure you have a clear line of sight to the robot, and operate the robot at safe speeds. The maximum speed is controllable through a dial on the Fubata controller; we strongly recommend setting the maximum speed to a low value for initial driving tests and only setting maximum speed to a high value in open areas that are free of people and obstacles.

PINCH POINTS

Refer to Pinch Points for pinch points on the Warthog that should be avoided during operation.

Driving with Remote Controller

Warthog uses a standard Futaba remote control radio transmitter. To operate the remote, first power it on by sliding the power switch to the ON position, as indicated in the image below.



Fubata Controller

To drive the Warthog, the spring-loaded deadman switch in the upper-right corner must be held down. Releasing this switch will prevent the robot from moving.

The lever on the left controls the robot's speed and the lever on the right controls the robot's rotation.

Driving with ROS Messages

You can manually publish `geometry_msgs/Twist` ROS messages to either the `/warthog_velocity_controller/cmd_vel` or the `/cmd_vel` ROS topics to drive Warthog.

For example, in terminal, run:

```
rostopic pub /warthog_velocity_controller/cmd_vel geometry_msgs/Twist '{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}'
```

The command above makes Warthog drive forward momentarily at 0.5 m/s without any rotation.

Extending Warthog Startup

Now that you've had Warthog for a while, you may be interested in how to extend it, perhaps add some more payloads, or augment the URDF.

Startup Launch Context

When ROS packages are grouped together in a directory and then built as one, the result is referred to as a workspace. Each workspace generates a `setup.bash` file which the user may source in order to correctly set up important environment variables such as `PATH`, `PYTHONPATH`, and `CMAKE_PREFIX_PATH`.

The standard system-wide setup file is in `/opt`:

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

When you run this command, you'll have access to `rosrun`, `roslaunch`, and all the other tools and packages installed on your system from Debian packages.

However, sometimes you want to add additional system-specific environment variables, or perhaps packages built from source. For this reason, Clearpath platforms use a wrapper setup file, located in `/etc/ros`:

```
source /etc/ros/setup.bash
```

This is the setup file which gets sourced by Warthog's background launch job, and in the default configuration, it is also sourced on your login session. For this reason it can be considered the "global" setup file for Warthog's ROS installation.

This file sets some environment variables and then sources a chosen ROS workspace, so it is one of your primary modification points for altering how Warthog launches.

Launch Files

The second major modification point is the `/etc/ros/noetic/ros.d` directory. This location contains the launch files associated with the `ros` background job. If you add launch files here, they will be launched with Warthog's startup.

However, it's important to note that in the default configuration, any launch files you add may only reference ROS software installed in `/opt/ros/noetic`. If you want to launch something from workspace in the home directory, you must change `/etc/ros/setup.bash` to source that workspace's setup file rather than the one from `/opt`.

Adding URDF

There are two possible approaches to augmenting Warthog's URDF. The first is that you may simply set the `Warthog_URDF_EXTRAS` environment variable in `/etc/ros/setup.bash`. By default, it points to an empty dummy file, but you can point it to a file of additional links and joints which you would like mixed into Warthog's URDF (via xacro) at runtime.

The second, more sophisticated way to modify the URDF is to create a *new* package for your own robot, and build your own URDF which wraps the one provided by warthog_description.

Keeping Warthog Updated

For details on updating Warthog software or firmware, refer to Software Maintenance.

Warthog Tests

Warthog robots come preinstalled with a set of test scripts as part of the `warthog_tests` ROS package, which can be run to verify robot functionality at the component and system levels.

If your Warthog does not have the `warthog_tests` ROS package installed already, you can manually install it by opening terminal and running:

```
sudo apt-get install ros-noetic-warthog-tests
```

ROS Tests

The `ros_tests` script exposes a set of interactive tests to verify the functionality of core features. These tests run at the ROS-level via ROS topics, and serve as a useful robot-level diagnostic tool for identifying the root cause of problems, or at the very least, narrowing down on where the root cause(s) may be.

Running ROS Tests

To run `ros_tests` on a Warthog robot, open terminal and run:

```
rosrun warthog_tests ros_tests
```

Upon running `ros_tests`, a list of available tests will be shown in a menu. From the menu, you can choose individual tests to run, or simply choose the option to automatically run all the tests.

The details of each test are shown below.

- **Lighting Test**

  The **Lighting Test** checks that the robot's lights are working properly.

  This test turns the lights off, red, green, and blue (in order) by publishing lighting commands to the `/cmd_lights` ROS topic. The user will be asked to verify that the lights change to the expected colours.

- **Motion Stop Test**

  The **Motion Stop Test** checks that the robot's motion-stop is working properly.

  This test subscribes to the `/mcu/status` ROS topic and checks that when the motion-stop is manually engaged by the user, the motion-stop state is correctly reported on the `/mcu/status` ROS topic. The user will be asked to verify that the lights flash red while the motion-stop is engaged.

- **ADC Test**

  The **ADC Test** checks that the robot's voltage and current values across its internal hardware components are within expected tolerances.

  This test subscribes to the `/mcu/status` ROS topic and checks that the voltage and current values across the internal hardware are within expected tolerances.

- **Rotate Test**

  The **Rotate Test** rotates the robot counter clockwise 2 full revolutions and checks that the motors, IMU, and EKF odometry are working properly.

  This test:

  - Subscribes to the `/imu/data` ROS topic to receive angular velocity measurements from the IMU's Gyroscope. These measurements are converted into angular displacement estimations, and the robot will rotate until 2 full revolutions are estimated.
  - Subscribes to the `/odometry/filtered` ROS topic to receive angular velocity estimations from the EKF odometry. These measurements are converted into angular displacement estimations, and are output as comparison to the angular displacement estimations from the IMU's Gyroscope.
  - Publishes to the `/cmd_vel` ROS topic to send drive commands to rotate the robot.

    - The user will be asked to verify that the robot rotates 2 full revolutions.

    > **NOTE**
    >
    > The **Rotate Test** rotates the robot using the IMU's Gyroscope data, which inherently will not be 100% accurate. Therefore, some undershoot/overshoot is to be expected.

- **Drive Test**

  The **Drive Test** drives the robot forward 1 metre and checks that the motors, encoders, and encoder-fused odometry are working properly.

  This test:

    - Subscribes to the `/warthog_velocity_controller/odom` ROS topic to receive linear displacement estimations from the encoder-fused odometry. The robot will drive forward until 1 metre is estimated.
    - Subscribes to the `/joint_state` ROS topic to receive linear displacement measurements from individual the encoders. These measurements are output as comparison to the linear displacement estimations from the encoder-fused odometry.
    - Publishes to the `/cmd_vel` ROS topic to send drive commands to drive the robot.
    - The user will be asked to verify that the robot drives forward 1 metre.

    > **NOTE**
    >
    > The **Drive Test** drives the robot using the Odometry data, which inherently will not be 100% accurate. Therefore, some undershoot/overshoot is to be expected.

CAN Bus Test

The `check_can_bus_interface` script checks that communication between the motors, encoders, robot's MCU, and robot's computer are working properly over the CAN bus interface.

This script verifies that the `can0` interface is detected and activated, then proceeds to check the output of `candump` to verify that good CAN packets are being transmitted.

Running CAN Bus Test

To run the `check_can_bus_interface` script on a Warthog robot, open terminal and run:

```
rosrun warthog_tests check_can_bus_interface
```

Advanced Topics

Configuring the Network Bridge

Your computer is configured to bridge its physical network ports together. This allows any network port to be used as a connection to the internal `192.168.131.1/24` network for connecting sensors, diagnostic equipment, or manipulators, or for connecting the your robot to the internet for the purposes of installing updates.

In the unlikely event you must modify your robot's Ethernet bridge, you can do so by editing the configuration file found at `/etc/netplan/50-clearpath-bridge.yaml`:

```
# Configure the wired ports to form a single bridge
# We assume wired ports are en* or eth*
# This host will have address 192.168.131.1

network:
version: 2
renderer: networkd
ethernets:
bridge_eth:
  dhcp4: no
  dhcp6: no
  match:
    name: eth*
bridge_en:
  dhcp4: no
  dhcp6: no
  match:
    name: en*
bridges:
br0:
  dhcp4: yes
  dhcp6: no
  interfaces: [bridge_en, bridge_eth]
  addresses:
    - 192.168.131.1/24
```

This file will create a bridged interface called `br0` that will have a static address of `192.168.131.1`, but will also be able to accept a DHCP lease when connected to a wired router. By default, all network ports named `en*` and `eth*` are added to the bridge. This includes all common wired port names, such as: `eth0`, `eno1`, `enx0123456789ab`, `enp3s0`, etc.

To include/exclude additional ports from the bridge, edit the match fields, or add additional `bridge_*` sections with their own match fields, and add those interfaces to the interfaces: `[bridge_en, bridge_eth]` line near the bottom of the file.

We do not recommend changing the static address of the bridge to be anything other than `192.168.131.1`; changing this may cause sensors that communicate over Ethernet (e.g. lidars, cameras, GPS arrays) from working properly.

> **NOTE**
>
> See also Network IP Addresses for common IP addresses on Clearpath robots.

Additional Simulation Worlds

In addtion to the default `warthog_world.launch` file, `warthog_gazebo` contains three additional launch files:

- `empty_world.launch`, which spawns Warthog in a featureless, infinite plane;
- `warthog_race.launch`, which spawns a world with lots of linear barriers; and
- `spawn_warthog.launch`, which is intended to be included in any custom world to add a Warthog simulation to it.

To add a Warthog to any of your own worlds, simply include the `spawn_warthog.launch` file in your own world's launch:

```
<include file="$(find warthog_gazebo)/launch/spawn_warthog.launch">
  <!-- Optionally configure the spawn position -->
  <arg name="x" value="$(arg x)"/>
  <arg name="y" value="$(arg y)"/>
  <arg name="z" value="$(arg z)"/>
  <arg name="yaw" value="$(arg yaw)"/>
</include>
```

Finally, Clearpath provides an additional suite of simulation environments that can be downloaded separately and used with Warthog, as described below.

Clearpath Gazebo Worlds

The Clearpath Gazebo Worlds collection contains 4 different simulation worlds, representative of different environments our robots are designed to operate in:

- Inspection World: a hilly outdoor world with water and a cave
- Agriculture World: a flat outdoor world with a barn, fences, and solar farm
- Office World: a flat indoor world with enclosed rooms and furniture
- Construction World: office world, under construction with small piles of debris and partial walls

Warthog is supported in Inspection and Agriculture Worlds.

Installation

To download the Clearpath Gazebo Worlds, clone the repository from github into the same workspace as your Warthog:

```
cd ~/catkin_ws/src
git clone https://github.com/clearpathrobotics/cpr_gazebo.git
```

Before you can build the package, make sure to install dependencies. Because Clearpath Gazebo Worlds depends on all of our robot's simulation packages, and some of these are currently only available as source code, installing dependencies with `rosdep install --from-paths [...]` will likely fail.

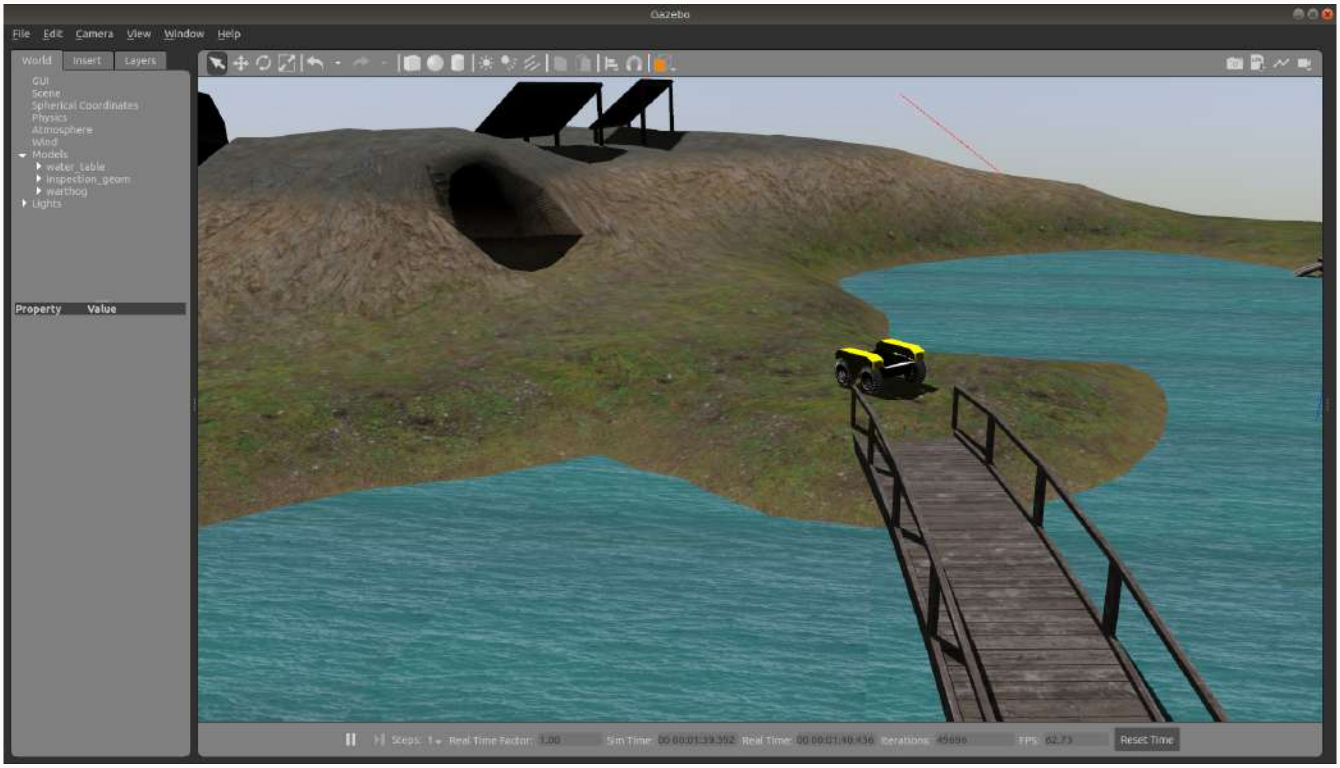
To simulate Warthog in the Office and Construction worlds the only additional dependency is the `gazebo_ros` package.

Once the dependencies are installed, you can build the package:

```
cd ~/catkin_ws
catkin_make
source devel/setup.bash
```

Running the Inspection Simulation

Inspection World is a hilly, outdoor world that includes a water feature, bridge, pipeline, small cave/mine, and a small solar farm. It is intended to simulate a variety of missions, including pipeline inspection, cave/underground navigation, and localization on non-planar terrain.



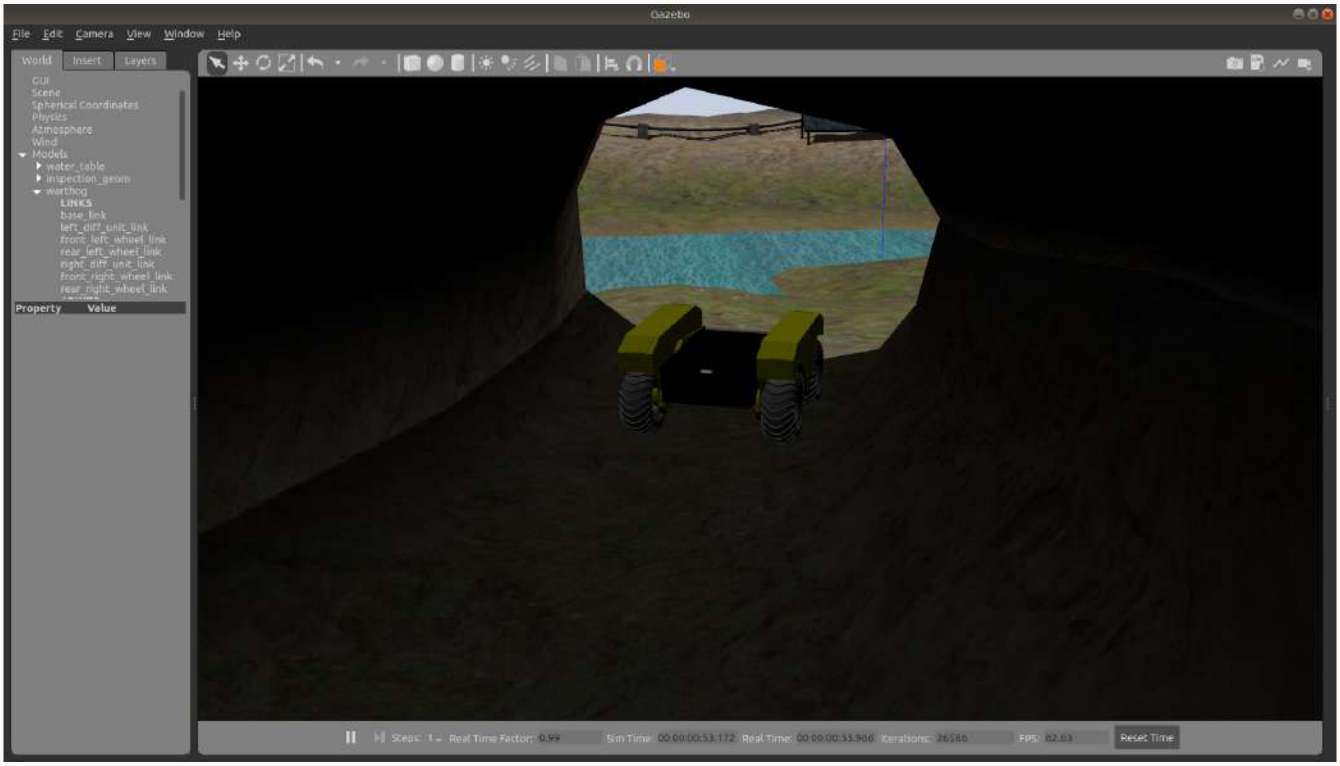
Warthog driving over the bridge in the inspection world

To launch the inspection simulation, run

```
roslaunch cpr_inspection_gazebo inspection_world.launch platform:=warthog
```

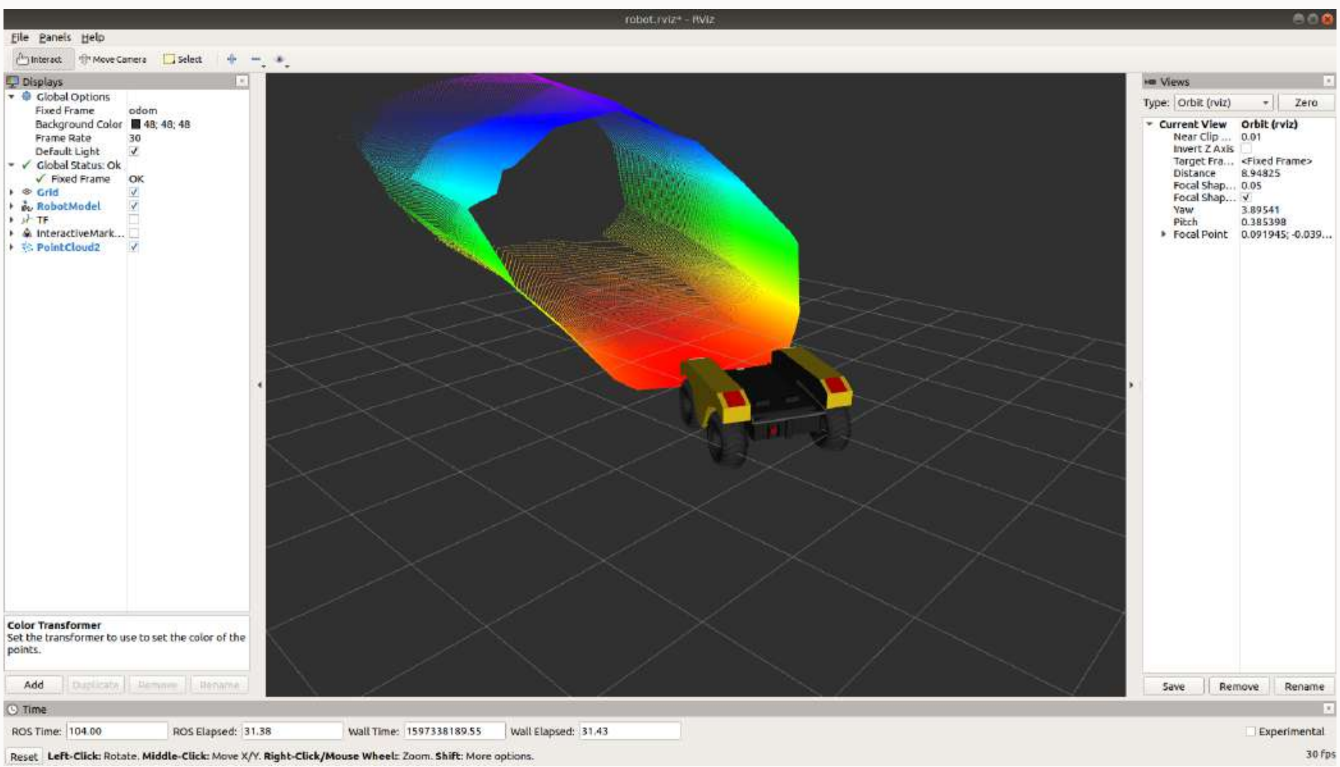
To customize Warthog's payload, for example to add additional sensors, see Customizing Warthog's Payload.

Once the simulation is running you can use RViz and other tools as described in Simulating Warthog to control and monitor the robot. For example, below we can see Warthog exploring the cave:



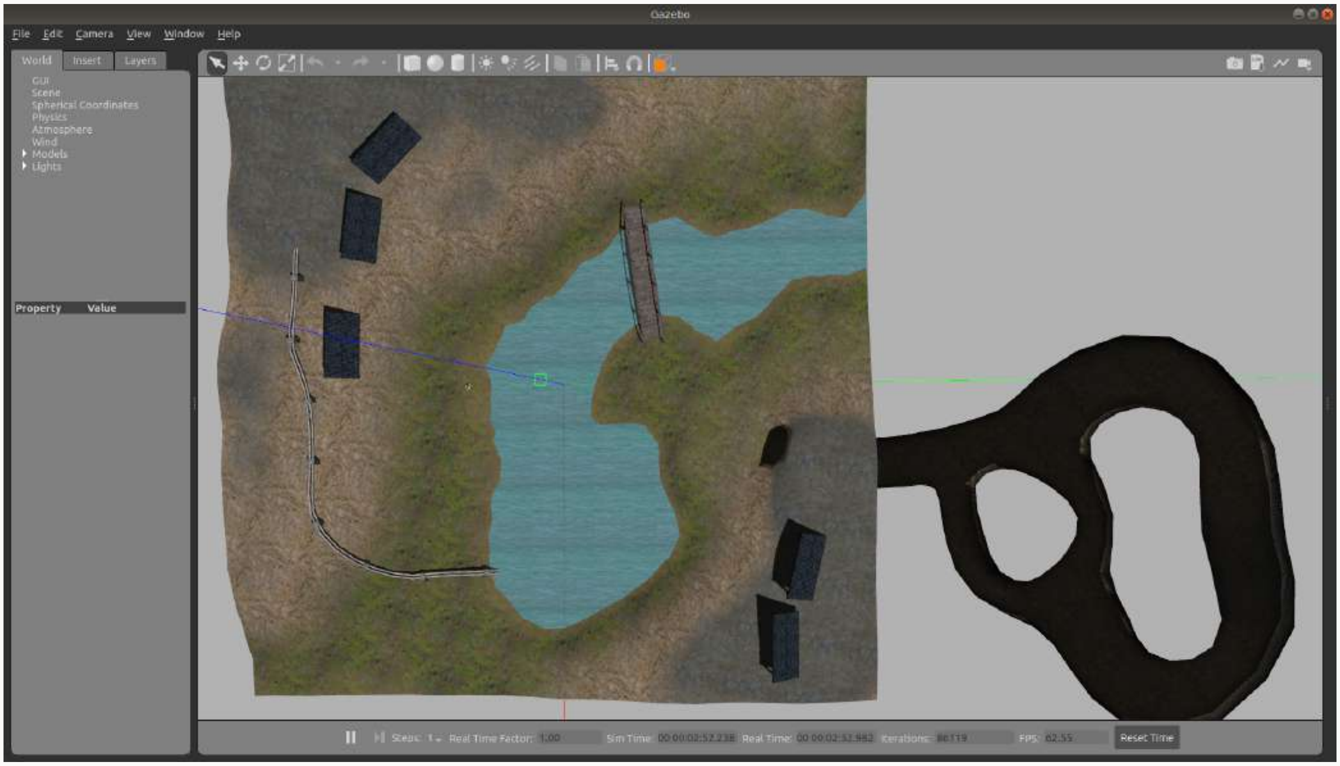
Warthog exploring the cave

and Warthog's perception of the inside of the cave as a 3D pointcloud in RViz:



Warthog in RViz exploring the cave

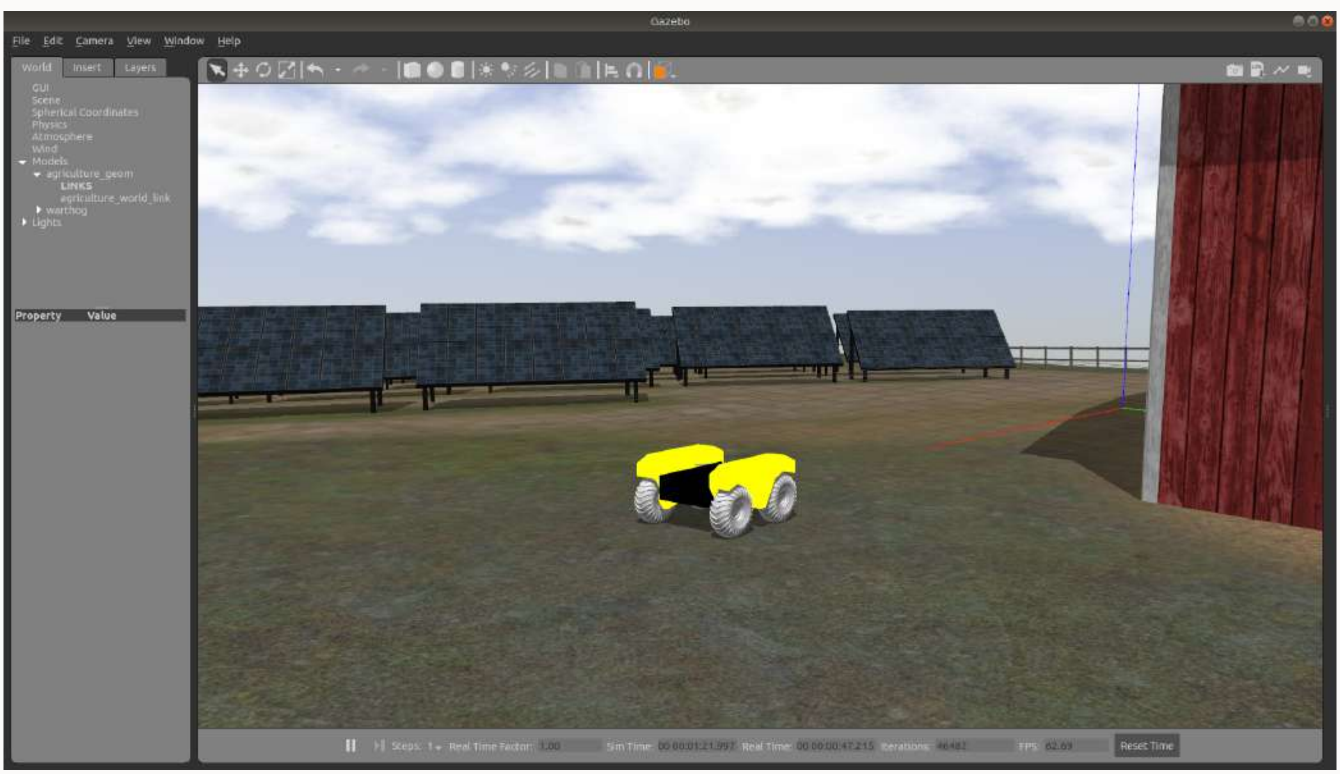
You can see the complete layout of the Inspection World below:



Inspection World

Running the Agriculture Simulation

Agriculture World is a flat, mixed indoor/outdoor world that include a large barn, open fields surrounded by fences, and a large solar farm. It is intended to simulate missions such as solar panel inspection and area coverage.



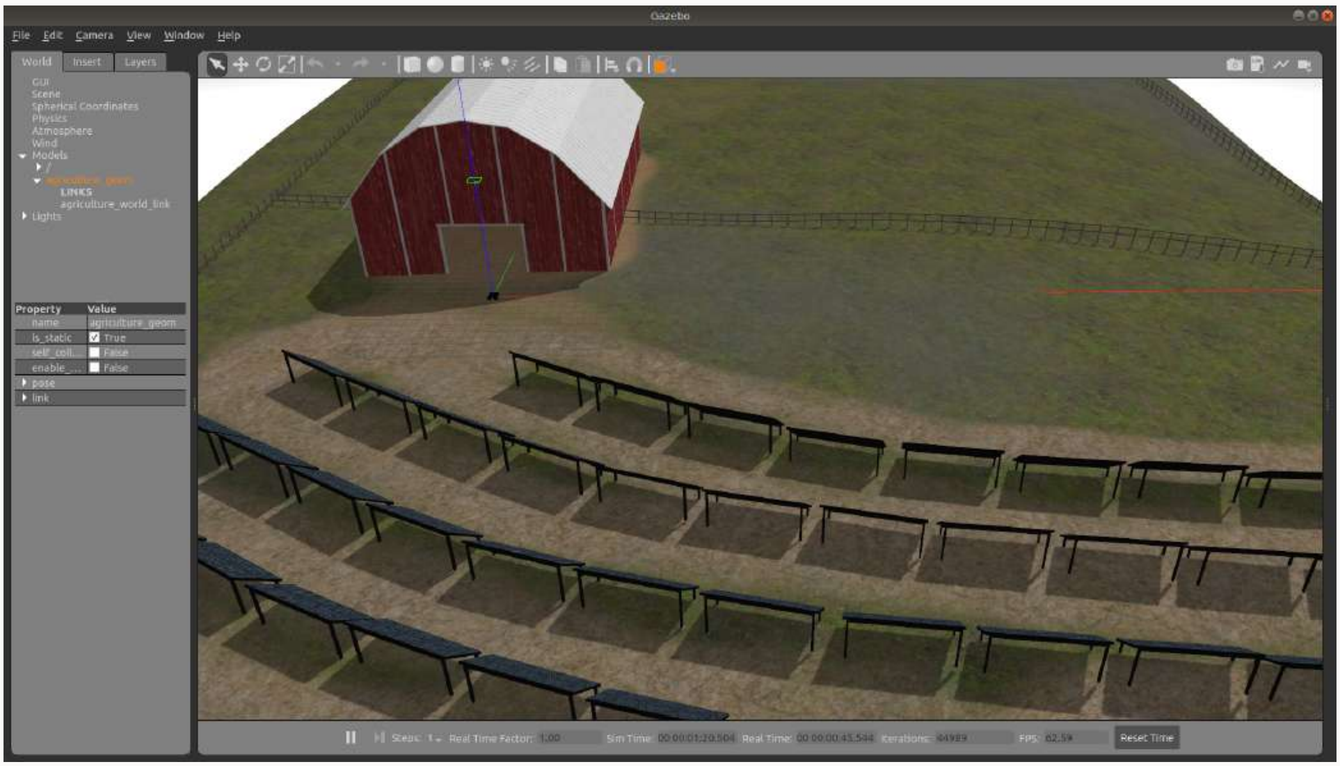
Warthog in the agriculture world

To launch the agriculture simulation, run

```
roslaunch cpr_agriculture_gazebo agriculture_world.launch platform:=warthog
```

To customize Warthog's payload, for example to add additional sensors, see Customizing Warthog's Payload.

Once the simulation is running you can use RViz and other tools as described in Simulating Warthog to control and monitor the robot.



Agriculture World

Customizing Warthog's Payload

To customize Warthog's payload you must use the environment variables described in Environment Variables.

You can add additional sensors by creating a customized URDF and setting the `WARTHOG_URDF_EXTRAS` environment variable to point to it.

For example, let's suppose you want to equip Warthog with an Intel RealSense D435 camera. First, install the `realsense2_camera` and `realsense2_description` packages, along with the gazebo plugins:

```
sudo apt-get install ros-$ROS_DISTRO-realsense2-camera ros-$ROS_DISTRO-realsense2-
```

```
description ros-$ROS_DISTRO-gazebo-plugins
```

Then create your customized URDF file, for example `$HOME/Desktop/realsense.urdf.xacro`. Put the following in it:

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://ros.org/wiki/xacro">

  <link name="front_realsense" />

  <!--
    The gazebo plugin aligns the depth data with the Z axis, with X=left and Y=up
    ROS expects the depth data along the X axis, with Y=left and Z=up
    This link only exists to give the gazebo plugin the correctly-oriented frame
  -->
  <link name="front_realsense_gazebo" />
  <joint name="front_realsense_gazebo_joint" type="fixed">
    <parent link="front_realsense"/>
    <child link="front_realsense_gazebo"/>
    <origin xyz="0.0 0 0" rpy="-1.5707963267948966 0 -1.5707963267948966"/>
  </joint>

  <gazebo reference="front_realsense">
    <turnGravityOff>true</turnGravityOff>
    <sensor type="depth" name="front_realsense_depth">
      <update_rate>30</update_rate>
      <camera>
        <!-- 75x65 degree FOV for the depth sensor -->
        <horizontal_fov>1.5184351666666667</horizontal_fov>
        <vertical_fov>1.0122901111111111</vertical_fov>

        <image>
          <width>640</width>
          <height>480</height>
          <format>RGB8</format>
        </image>
        <clip>
          <!-- give the color sensor a maximum range of 50m so that the simulation
renders nicely -->
          <near>0.01</near>
          <far>50.0</far>
        </clip>
      </camera>
      <plugin name="kinect_controller" filename="libgazebo_ros_openni_kinect.so">
        <baseline>0.2</baseline>
```

```
      <alwaysOn>true</alwaysOn>
      <updateRate>30</updateRate>
      <cameraName>realsense</cameraName>
      <imageTopicName>color/image_raw</imageTopicName>
      <cameraInfoTopicName>color/camera_info</cameraInfoTopicName>
      <depthImageTopicName>depth/image_rect_raw</depthImageTopicName>
      <depthImageInfoTopicName>depth/camera_info</depthImageInfoTopicName>
      <pointCloudTopicName>depth/color/points</pointCloudTopicName>
      <frameName>front_realsense_gazebo</frameName>
      <pointCloudCutoff>0.105</pointCloudCutoff>
      <pointCloudCutoffMax>8.0</pointCloudCutoffMax>
      <distortionK1>0.00000001</distortionK1>
      <distortionK2>0.00000001</distortionK2>
      <distortionK3>0.00000001</distortionK3>
      <distortionT1>0.00000001</distortionT1>
      <distortionT2>0.00000001</distortionT2>
      <CxPrime>0</CxPrime>
      <Cx>0</Cx>
      <Cy>0</Cy>
      <focalLength>0</focalLength>
      <hackBaseline>0</hackBaseline>
    </plugin>
  </sensor>
</gazebo>

<link name="front_realsense_lens">
  <visual>
    <origin xyz="0.02 0 0" rpy="${pi/2} 0 ${pi/2}" />
    <geometry>
      <mesh filename="package://realsense2_description/meshes/d435.dae" />
    </geometry>
    <material name="white" />
  </visual>
</link>

<joint type="fixed" name="front_realsense_lens_joint">
  <!-- Offset the camera to the front edge of the robot -->
  <origin xyz="0.60 0 0.01" rpy="0 0 0" />
  <parent link="top_chassis_link" />
  <child link="front_realsense_lens" />
</joint>
<joint type="fixed" name="front_realsense_joint">
  <origin xyz="0.025 0 0" rpy="0 0 0" />
  <parent link="front_realsense_lens" />
  <child link="front_realsense" />
```

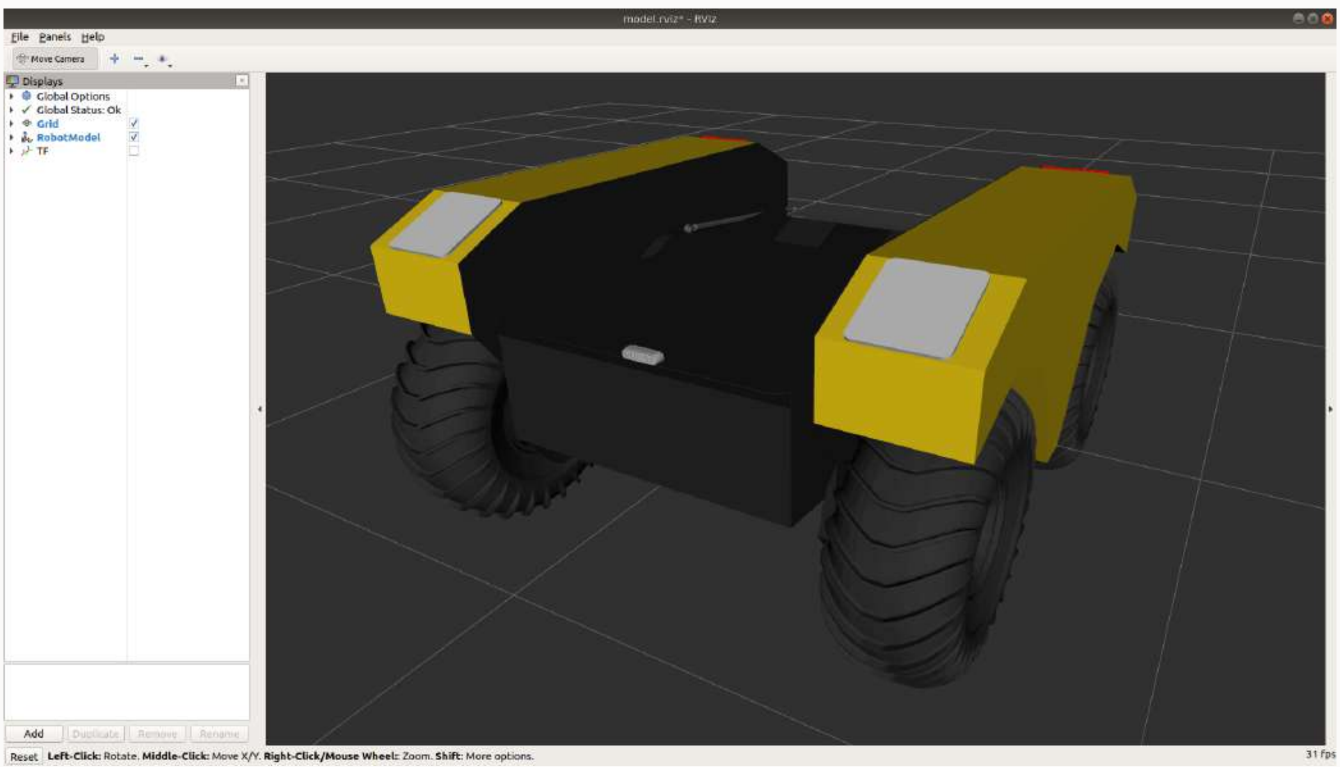
```
  </joint>
</robot>
```

This file defines the additional links for adding a RealSense camera to the robot, as well as configuring the `openni_kinect` plugin for Gazebo to simulate data from a depth camera. The camera itself will be connected to the Warthog's `top_chassis_link` link, offset 62.5 cm towards the front of the robot.

Now, set the `WARTHOG_URDF_EXTRAS` environment variable and try viewing the Warthog model:

```
export WARTHOG_URDF_EXTRAS=$HOME/Desktop/realsense.urdf.xacro
roslaunch warthog_viz view_model.launch
```

You should see the Warthog model in RViz, with the RealSense camera mounted to it:

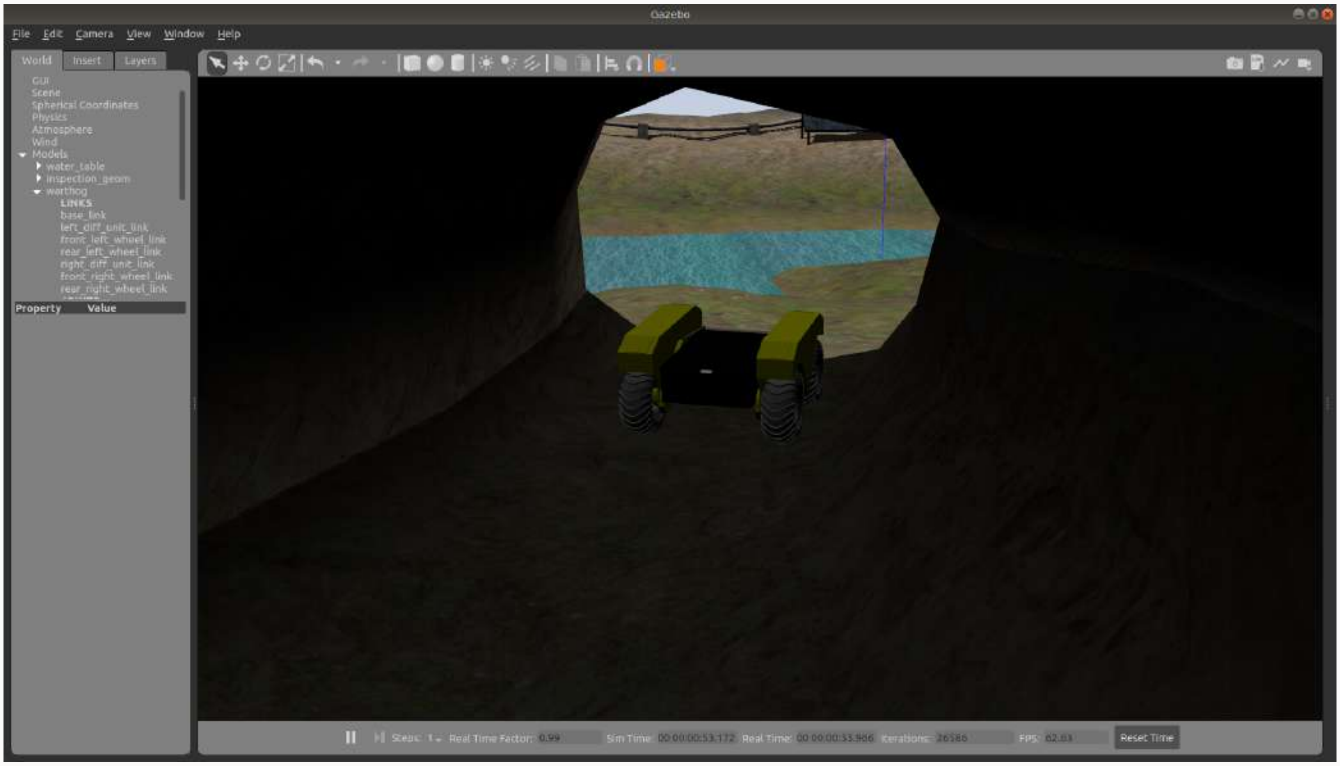


Warthog with a RealSense D435 connected to it

To launch the customized Warthog in any of the new simulation environments, similarly run:

```
export WARTHOG_URDF_EXTRAS=$HOME/Desktop/realsense.urdf.xacro
roslaunch cpr_inspection_gazebo inspection_world.launch platform:=warthog
```

You should see Warthog spawn in the office world with the RealSense camera:

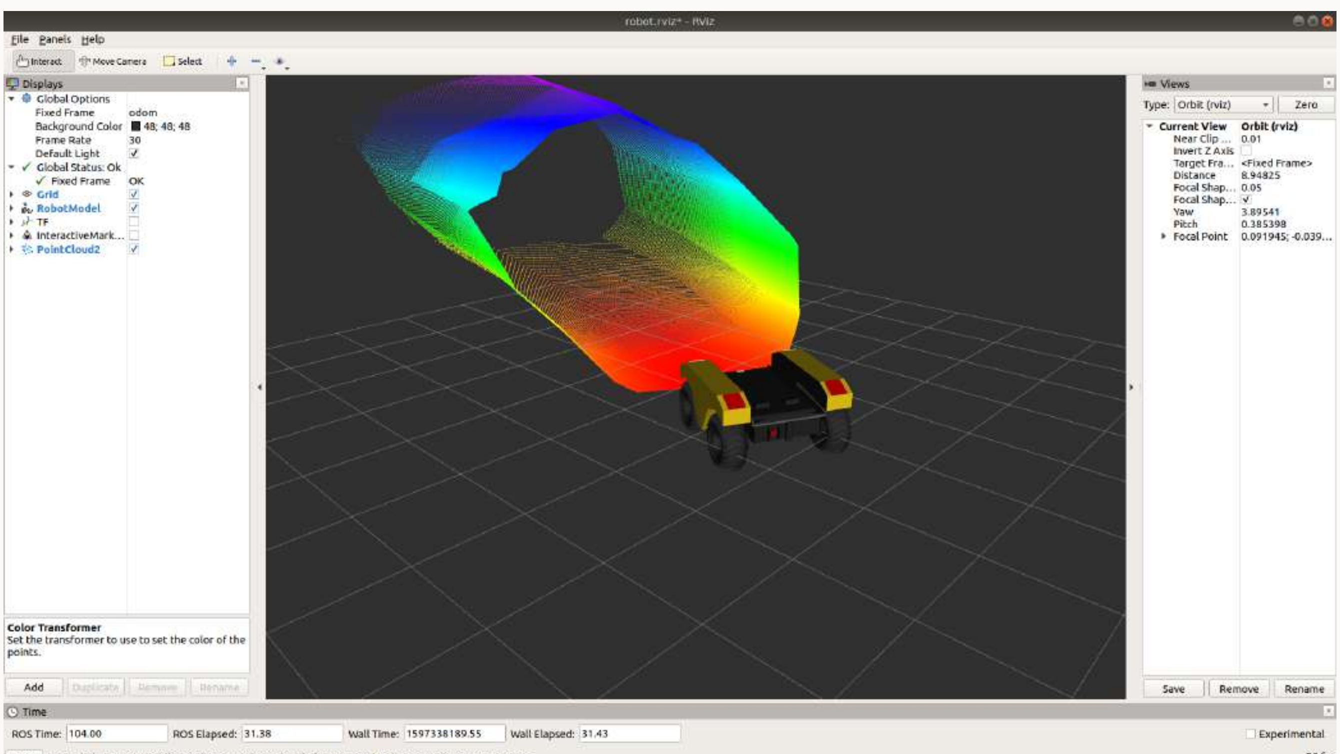


Warthog in Inspection cave

You can view the sensor data from the RealSense camera by running

```
roslaunch warthog_viz view_robot.launch
```

and adding the camera & pointcloud from the `/realsense/color/image_raw` and `/realsense/depth/color/points` topics:



Warthog with a RealSense in RViz showing pointcloud data

Support

Clearpath is committed to your success. Please get in touch with us and we will do our best to get you rolling again quickly: support@clearpathrobotics.com.

To get in touch with a salesperson regarding Clearpath Robotics products, please email sales@clearpathrobotics.com.

If you have an issue that is specifically about ROS and is something which may be of interest to the broader community, consider asking it on https://answers.ros.org. If you do not get a satisfactory response, please ping us and include a link to your question as posted there. If appropriate, we will answer in the ROS Answers context for the benefit of the community.

*Last updated on **Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Warthog Maintenance

Battery and Charging

Overview

Warthog contains a 48 V lead-acid battery pack, consisting of four 12 V lead-acid batteries, or a lithium battery pack, consisting of four 118 Ah 12.8 V batteries connected in a 4S1P configuration with a single Battery Management System (BMS) control unit. Battery configuration may vary with each unit. In order to maximize performance, it's important to ensure that the battery level across each set of lead-acid or lithium batteries is within 0.1-0.2 V. If the batteries exceed this tolerance, it's advised to charge them to within tolerance before wiring these packs in parallel. The overall battery life will vary depending upon the usage of the unit.

Battery Management System

> **NOTE**
>
> If your Warthog is equipped with a lead-acid battery pack, you can ignore this section on the Battery Management System.

If your Warthog is equipped with a lithium battery pack, it is also equipped with a single Battery Management System (BMS) control unit. The main function of the BMS is to:

- Protect the individual modules from being overcharged or excessively discharged.
- Protect the modules from operating outside of their acceptable temperature range.
- Provide status information to higher level systems via a CAN interface.
- Provide independent system management without a higher level control system.
- Provide status, warning and error messages from each of the modules connected to the battery power system.
- Control the charging operation.
- Perform module to module balancing of the system, the battery module will control cell balancing within itself.

There is a main power contactor which is controlled by the BMS unit and is placed in such a way that it controls all power to the system. Upon turning on the main disconnect switch:

The BMS powers up first and begins communicating with all of the battery modules in the system.

1. If they are all present and have no errors the BMS will close the main contactor and allow the Warthog to power up as normal, this process can take a few seconds before the main contactor closes.
    - If all the modules are not present, the system State of Charge (SOC) is too low, or there are errors being reported by any of the modules in the system, the contactor will not close, and the system will not power up.
2. Similarly, if a user is in discharge mode (the Warthog is in use) and any errors are detected, or if the SOC falls to a cut off point, the contactor will open and remove power from the system to protect the modules.
    - It is not good practice to run the state-of-charge (SOC) to 0%. This will decrease the life of the battery module. In general, keeping the SOC to a minimum of 30-50% can increase the lifetime of the batteries.
    - As a note, the system SOC is identical to the module with the lowest SOC in the system, so if three modules are at 45% SOC, and one module is at 25%, then the SOC of the system is 25%,

and the BMS will act accordingly cutting off power to protect the module with the lowest SOC. Cell balancing and proper charging techniques are important to maintaining good battery health.

3. The Warthog can be charged when the unit is powered and running, and can also be charged when the unit is not powered (main disconnect switch is open).
    - The charger has been configured and wired in a way that the BMS is actually controlling the charger and can turn it off, and on, depending on the status of the charging cycle.
    - Even though the unit can be charged while running, this process does not allow for proper cell balancing to occur. It will allow for bulk charge to occur to regain most of the charge in the system, however, repeatedly charging in this manner will allow each module to become out of balance with the other modules. If this process if followed long enough the SOC values for each module can vary by a substantial amount. The larger the variation, the longer it will take to fully balance the system. Therefore it is recommended (depending on use) that at least once every two to three weeks that **the system be completely charged while the main disconnect switch is in the off position.** Doing so will allow the BMS to go through an equalization phase where the charger is being turned on and off by the BMS to balance the voltages across each cell layer, in each module across the system. **This is a very important step in maintain battery health, and getting the maximum lifetime out of the battery system.**
    - Refer to BMS ROS Topics to view the necessary topics through the ROS API.
4. To allow for this level of control there are three contactors in the system:
    - The main contactor for discharge of the system.
    - The charge contactor to connect to the output of the charger.
    - The charge control relay to turn the charger outputs on and off and control the cell balancing or equalization phase of the charge cycle.

BMS ROS Topics

> **NOTE**
>
> If your Warthog is equipped with a lead-acid battery pack, you can ignore this section on the BMS ROS Topics.

The table below shows topics that will be very useful in monitoring battery health, as outlined in the section above.

Topic	Purpose
`/bms/system_status`	Primarily to view `state_of_charge` (%)

Topic	Purpose
`/bms/system_measurements`	Primarily to view `battery_system_current` (A)

Note that the `battery_system_current` topic will be positive when the batteries are charging, indicating that current is flowing into the batteries. The lights on the charger unit itself will also indicate when the device is charging.

To Connect the Charger

1. Plug in the charger DC output cable (heavier gauge cable).
2. Plug in the charger auxiliary control cable. (Note: Only relevant to Warthogs with lithium batteries.)
3. Plug the charger into AC power (directly to the wall outlet, no extension cords).
4. This can be done with or without power on Warthog being on (however the note above applies: at least once every 2-3 weeks, the charge process needs to be allowed to fully balance the system).

To Disconnect the Charger

1. Unplug the charger from AC power.
2. Unplug the charger auxiliary cable.
3. Unplug the charger DC output cable (heavier gauge cable).

Battery Pack Specifications

Key battery system specifications are listed in the table below.

Specification	Measurement
System Voltage (nominal)	51.2V (up to 59 V while charging)
System Capcity	118 Ah
System Energy	6.041 kWh
Continuous Discharge Current	150 A
Peak Discharge Current (30 sec)	300 A

Specification	Measurement
Discharge Temperature Range	-10 °C to 50 °C
Charge Temperature Range	0 °C to 45 °C

General Battery Safety and Maintenance

Always exercise caution and observe the following safety practices when connecting, disconnecting or handling batteries:

- Batteries are high voltage, high current.
- Batteries must be properly fastened down to ensure they do not move when the Warthog is in operation.
- Ensure that the batteries are evenly distributed throughout the Warthog to maximize stability.
- Battery levels on the unit should be checked on a regular basis. It's important to maintain the battery voltage at a suitable level for proper operation.

Long-term Storage

When storing Warthog for long periods of time, it's important to properly maintain the batteries to fully maximize their life. Consider the following procedure when placing Warthog in long-term storage:

- Fully charge Warthog, turn it off and put it into storage. Once a week, connect power to the charger and allow the charger to top up the battery for an hour or so.

Please contact Clearpath Robotics for additional information about Warthog's batteries.

Tread Upgrade

If your unit came equipped with the tread upgrade, please keep the following considerations in mind:

- The tread drive units pull considerably more current to operate. It is advised to operate a tread Warthog at reduced speeds, either through ROS commands or by reducing the speed scale when operating with the Futaba controller.

- Due to the increased drive current, expect lower battery life than a wheeled Warthog. It is recommend to drive at reduced speeds to increase battery life.
- It is possible to trigger an over current error in the motor controller by increasing speed drastically shortly after resetting a stop condition. The system's body lights will seem as if the unit is ready to drive, however it will not respond to drive commands. To clear this state, simply cycle another stop condition, and ramp up speeds slower.

Software Maintenance

> **NOTE**
>
> If you are upgrading your robot from an older version of ROS, please refer to our upgrade instructions for upgrading to Melodic, Noetic and ROS 2 Humble.

Getting New Packages

Clearpath Robotics robots are always being improved, both its own software and the many community ROS packages upon which it depends! You can use the `apt` package management system to receive new versions all software running on the platform.

Each robot leaves the factory already configured to pull packages from http://packages.ros.org as well as http://packages.clearpathrobotics.com. To update your package and download new package versions, simply run:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

MCU Firmware Update

> **CAUTION**
>
> Accessing Warthog's MCU requires access to several hard-to-get-to parts of the robot. Unless absolutely necessary, we do not recommend re-flashing the robot's MCU firmware.

You need to use an external computer to update Warthog's MCU firmware. You cannot use Warthog's Onboard Computer, as installing the firmware requires power-cycling the MCU. Warthog's MCU controls the power supply to the Onboard Computer. These instructions assume the external computer is running some flavour of Linux with access to Clearpath's ROS packages.

Follow the below procedure to flash the firmware to Warthog's MCU:

1. Place Warthog up on blocks or engage the emergency stop by pressing one of the red buttons located on each corner of the robot. Firmware loading does not usually result in unintended motion, but it's safest to ensure the robot cannot move accidentally.

2. Download the Warthog firmware package onto your computer:

   ```
   sudo apt-get install ros-noetic-warthog-firmware
   ```

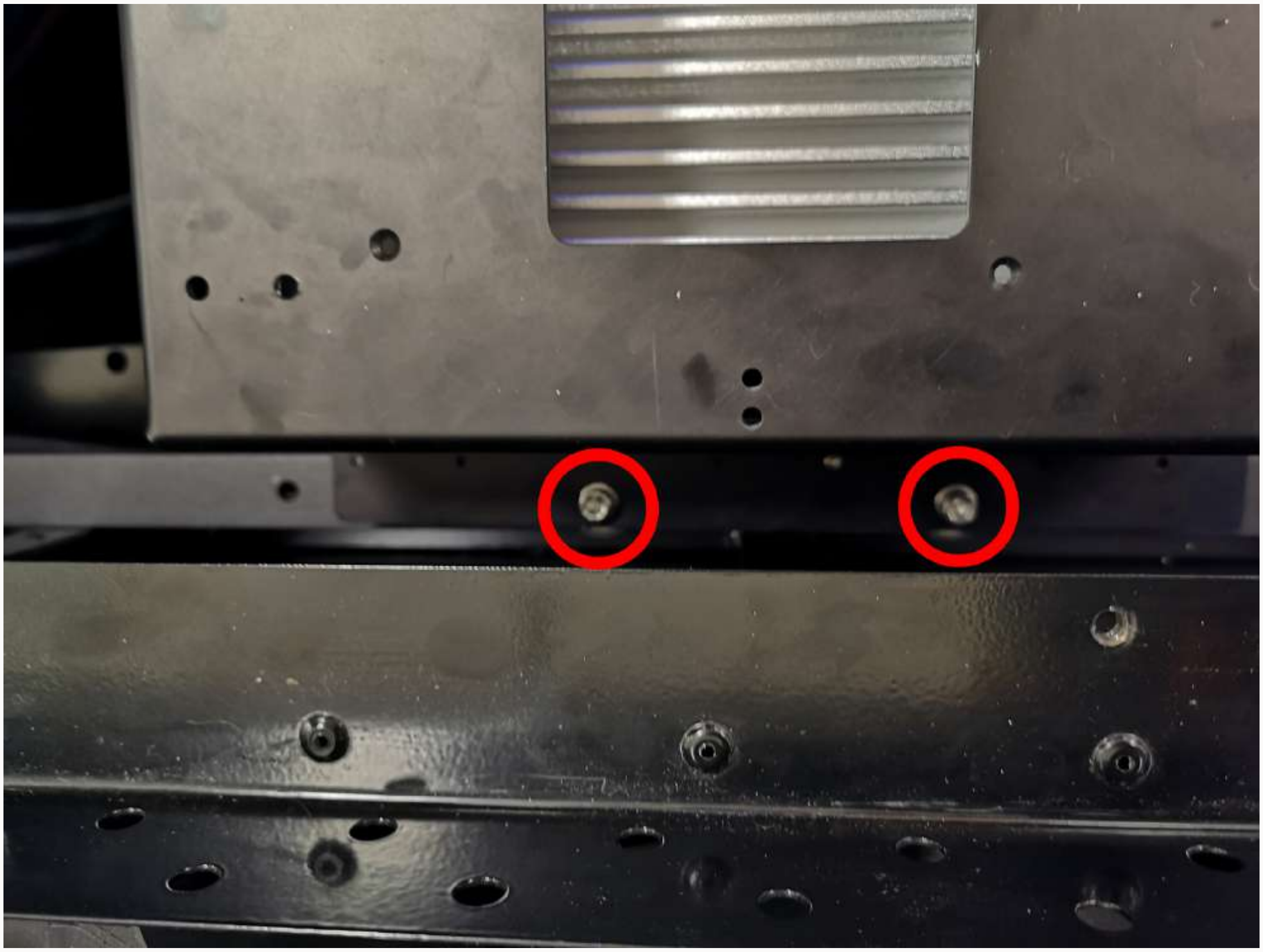
3. Remove the top panel from the Warthog. We recommend opening the panel to the left, as there are cables that run into the lid which can be strained.

The inside of Warthog's computer box

4. Warthog's MCU is located on the underside of the metal frame over the top of the computer. To access it you will need to disconnect all cables from the breakout on the sloped portion on the left and then undo 4 screws anchoring the frame:

The inside of Warthog's computer box

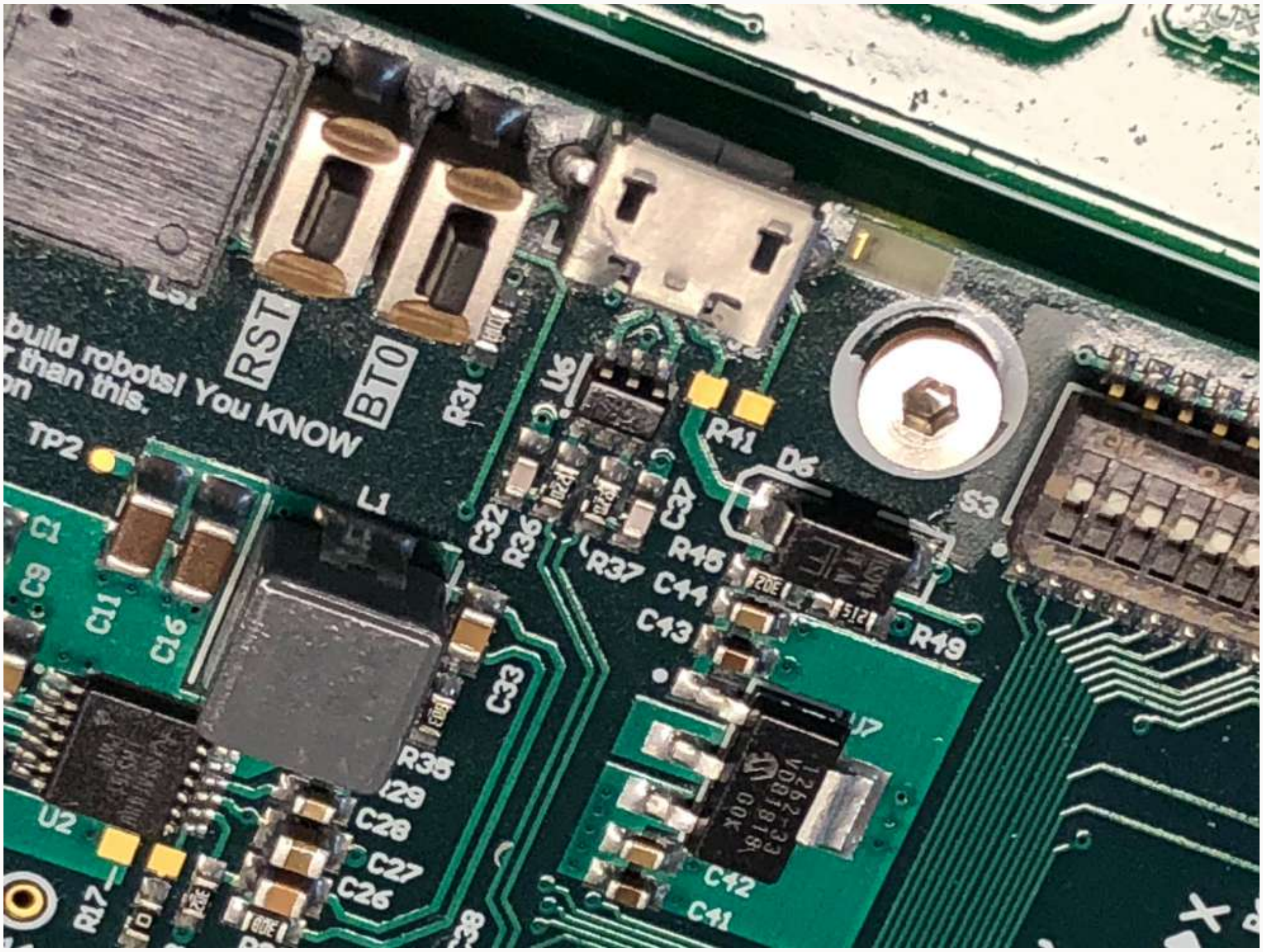
Screws on the left



Screws on the right

Once the screws are removed, carefully lift the center panel and turn it over to expose the MCU's micro USB port and buttons.

5. While pressing `BT0` on the MCU, connect the external computer to Warthog's MCU using a USB cable.



Warthog's MCU buttons

6. After connecting the computer you should see a device with a name similar to `Bus 001 Device 005: ID 0483:df11 STMicroelectronics STM Device in DFU Mode` in the output of `lsusb`.

> **NOTE**
>
> If the MCU does not show up as a DFU device in the output of `lsusb` after completing the above procedure, press and hold BT0 on the MCU press the RST button while holding BT0 down. This will reset the MCU and force it into DFU mode.

With the MCU in DFU mode, run the following command, replacing `001/005` with the value appropriate to the Bus and Device where Warthog's MCU is connected:

```
sudo chmod 666 /dev/bus/usb/001/005
```

Now run the following command to upload the firmware:

```
dfu-util -v -d 0483:df11 -a 0 -s 0x08000000 -D
```

```
/opt/ros/noetic/share/warthog_firmware/mcu.bin
```

You should see about 20 seconds worth of lines output beginning with "Download from image ...". When this is complete you may disconnect the computer from the MCU and power-cycle the robot.

Support

Clearpath is committed to your success. Please get in touch with us and we will do our best to get you rolling again quickly: support@clearpathrobotics.com.

To get in touch with a salesperson regarding Clearpath Robotics products, please email sales@clearpathrobotics.com.

If you have an issue that is specifically about ROS and is something which may be of interest to the broader community, consider asking it on https://answers.ros.org. If you do not get a satisfactory response, please ping us and include a link to your question as posted there. If appropriate, we will answer in the ROS Answers context for the benefit of the community.

*Last updated on **Jul 7, 2023***

This is documentation for Clearpath Robotics Documentation **ROS 1 Noetic**, which is no longer actively maintained.

For up-to-date documentation, see the **latest version** (ROS 2 Humble).

Version: ROS 1 Noetic

Warthog Troubleshooting

Robots are complex. Common issues are listed below. Contact our support team at support@clearpathrobotics.com if the issue persists.

INFO

Our Support team may ask you to create a *rosbag*. This is a tool that saves ROS data to a file, so we can review it after the robot has been turned off.

For example, we may ask you to record the */status* topic so we can review motor temperatures, and other diagnostic data. To create this rosbag, you would:

1. Log into the robot's computer.
2. Run the command `rosbag record /status`
3. Use your robot as you typically would, trying to recreate the issue you have been experiencing.
4. Once done, in the terminal, press `ctrl+c` to stop the command.

5. Enter `ls`, and find the new file that was created. It will have a file extension of *.bag*
6. Copy this file to your development computer, using a command like `scp`. Send this *.bag* file to Clearpath Robotics in your Support email request.

Networking

Not getting Wi-Fi internet connection

Refer to the Networking page.

Cannot connect to the robot's computer over a network cable

Reter to the Networking page.

Computer

The computer does not automatically start

Refer to the Mini ITX troubleshooting page.

The computer keeps reverting to old BIOS settings

Refer to the Mini ITX troubleshooting page.

ROS

ROS package is not starting

Verify the upstart logs, `/var/log/upstart/husky_core.log`, to see if there are any error during upstart.

Sensors are not turning on

1. Check that the sensor's User Power connector has the correct voltage.
2. If the voltage is correct, then review the Troubleshooting section for the related sensor.

If the Issue Persists

Clearpath is committed to your success. Please get in touch with us and we will do our best to get you rolling again quickly: support@clearpathrobotics.com.

To get in touch with a salesperson regarding Clearpath Robotics products, please email sales@clearpathrobotics.com.

If you have an issue that is specifically about ROS and is something which may be of interest to the broader community, consider asking it on https://answers.ros.org. If you do not get a satisfactory response, please ping us and include a link to your question as posted there. If appropriate, we will answer in the ROS Answers context for the benefit of the community.

Last updated on ***Jul 7, 2023***