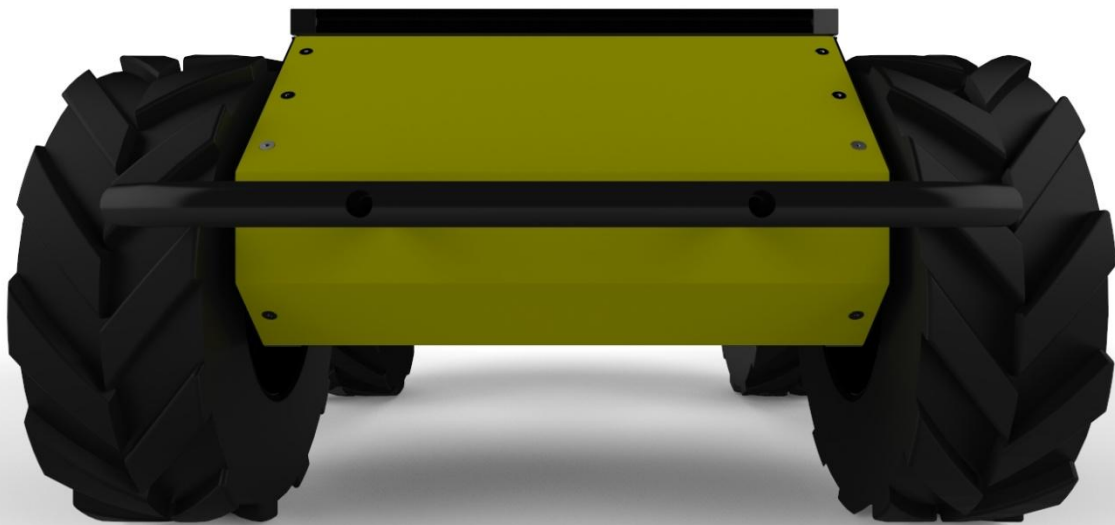


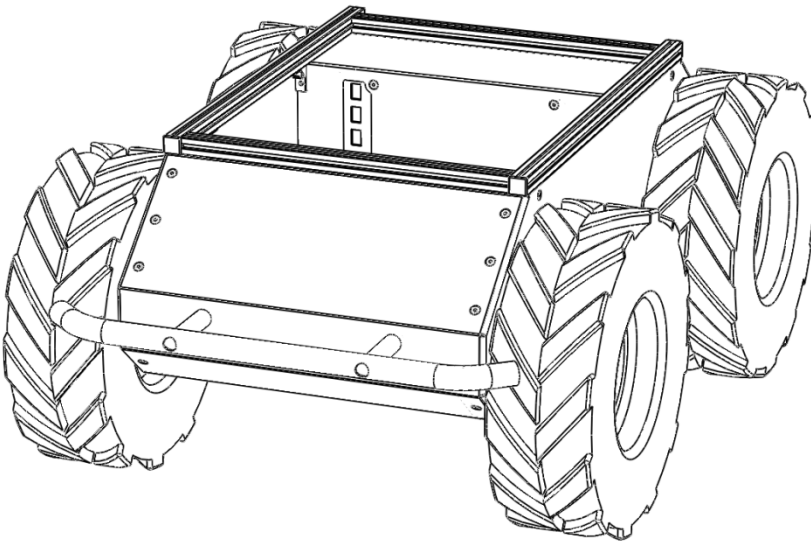


HUSKY A200™

UNMANNED GROUND VEHICLE



USER MANUAL



CONTENTS

Contents	2
Introduction	4
What's Included	4
What's Required	4
Expansions	4
The Basics	5
Status Panel.....	6
Orientation References.....	7
Pinout References.....	7
System Specifications.....	8
Vehicle Equations.....	9
Control Modes	9

Safety	10
General Warnings	10
E-Stop and Lockout	10
Performance Recommendations	11
Electrical System	11
Lifting and Transport.....	11
Getting Started	12
PC Setup	12
Connecting Computer	12
Connecting Power	12
Verification.....	14
ROS-Based Teleoperation	15
Operation	16
Control Protocol.....	16
Python	16
C++	17
Battery & Maintenance	19
Charging	19
Battery Care	19
Wheels	20
Chassis.....	20
Tips and Troubleshooting	21
Mechanical Tips	21
Troubleshooting.....	21
A. PC Setup	22
Python	22
ROS.....	23
C++	23
B. Product Dimensions	24
C. Service and Support	25

INTRODUCTION

Husky A200 is a rugged and easy-to-use unmanned ground vehicle for rapid prototyping applications. In this guide, you will find information about the setup, operation, and maintenance of your Husky A200.

What's Included

Included with your Husky A200 kit are the following:

- 1x Clearpath Robotics Husky A200
- 1x 24V Sealed Lead-Acid Battery Pack
- 1x Battery Door Cover
- 1x Battery Charger
- 2x Lockout Keys

What's Required

To use Husky A200, you will require a PC running Windows or Linux. Communication with Husky occurs over standard RS232 serial, using the Clearpath Communication Protocol (CCP). Open-source software interfaces for CCP are available in C++, Python, and National Instruments LabVIEW.

If you purchased a computer option from Clearpath Robotics with your Husky, then the laptop or minicomputer you received has already been configured for use with CCP.

Expansions

To expand the capabilities of Husky, consider the following additional sensor packages offered by Clearpath Robotics:

- LIDAR
- Network Camera
- GPS

Each sensor package ships from Clearpath Robotics with a custom mounting bracket and cabling, for easy attachment to Husky's extruded aluminum payload mounting rail.

THE BASICS

This section provides an overview of the key specifications of the Husky A200 platform. Figure 1 gives a tour of key Husky A200 components.

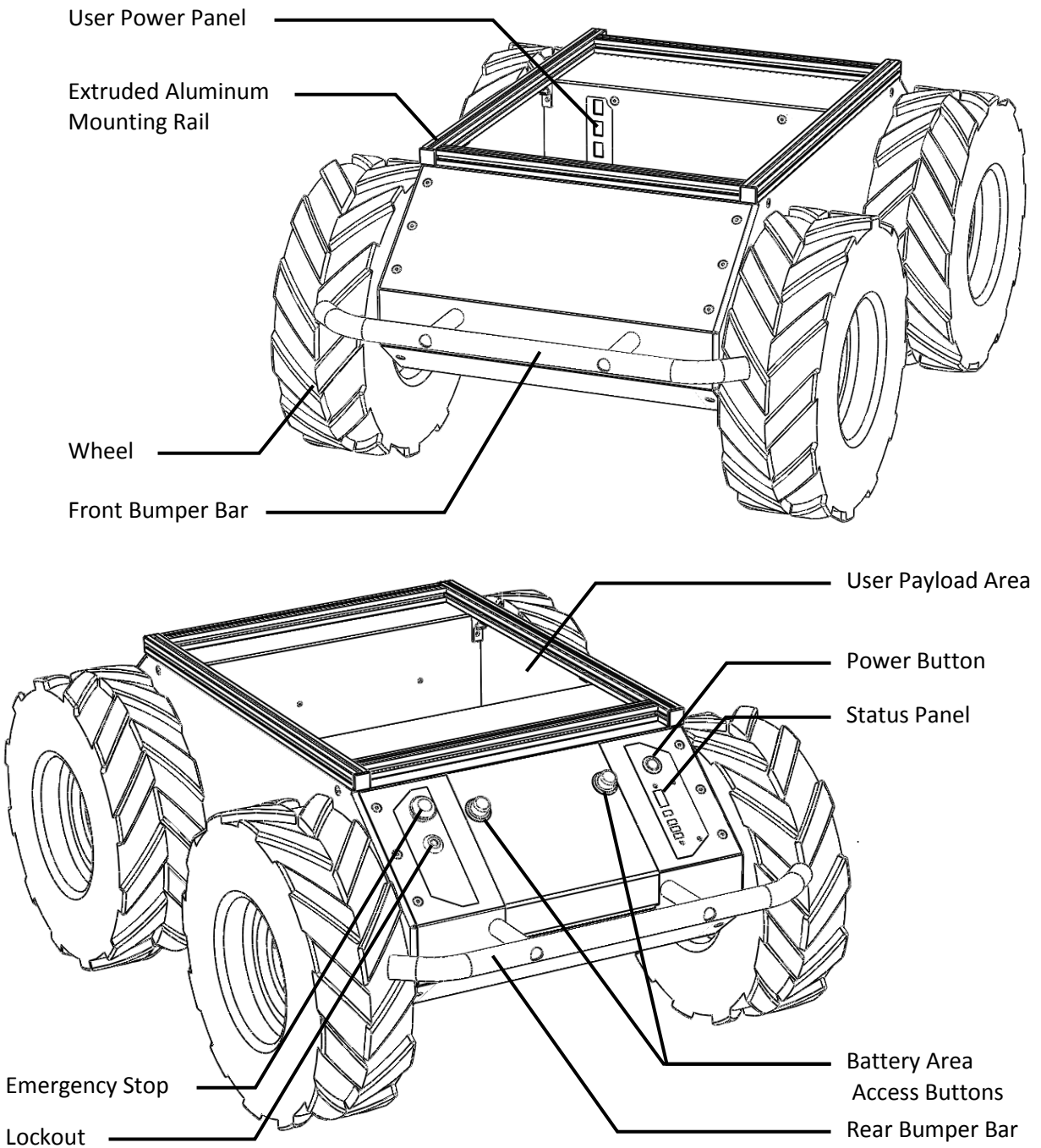


Figure 1: Husky A200 at a Glance

Status Panel

The status panel is an array of LED indicators which provide information about the current status of Husky, located on the rear of the chassis. The indicators are described in Table 1.






Icon	Description
	<p>Battery status. The four LED segments provide an approximate indication as to the relative lifetime remaining in the battery.</p>
	<p>Communications status. When green, Husky is receiving a stream of correctly-formatted motion commands, and is ready to drive. When yellow, Husky is receiving commands but will not drive due to emergency stop or another error. When red, serial communications are currently timed-out.</p>
	<p>General error status. Illuminates red when Husky will not drive due to an error state. Such states include emergency stop, insufficient battery power, or an unspecified software error.</p>
	<p>Emergency stop status. Illuminates red when Husky will not drive due to the emergency stop being activated, either onboard or wireless (if available)</p>
	<p>Charge indicator. Illuminates red when Husky user power is being supplied externally.</p>

Table 1: Husky A200 Status Panel Icons

Orientation References

The reference frame used by all Clearpath Robotics ground vehicles is based on ISO 8855, and is shown in Figure 2. Husky is being viewed from the front; 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.

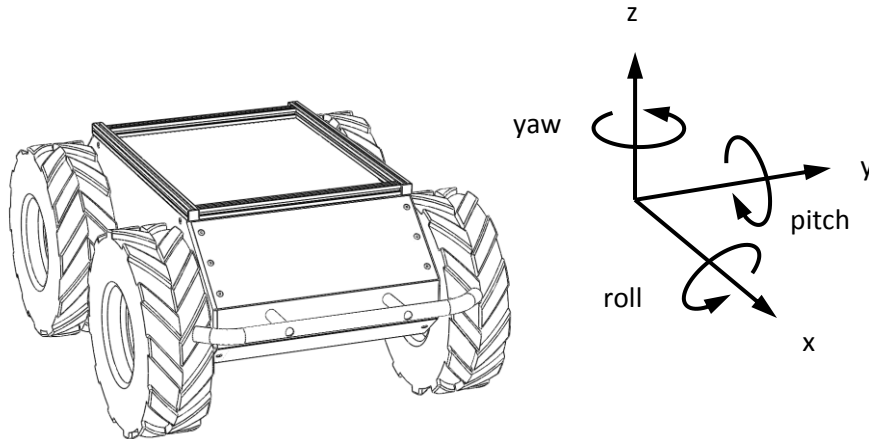
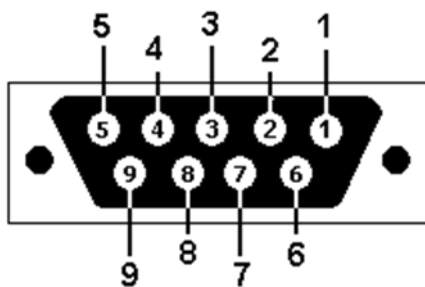


Figure 2: Husky A200 Reference Frame

Pinout References

Husky A200 provides a female DE-9 connector, for communication with a host device. The pinout of this connector is shown in Figure 3.



Pin	Name	Dir	Description
2	RX	IN	Data from Platform
3	TX	OUT	Data to Platform
5	GND	N/A	Common Ground

Figure 3: Husky A200 DE9 Pinout

System Specifications

Key specifications of Husky A200 are shown in Table 2.

Dimensions	990 mm length 670 mm width 390 mm height	39 in length 26.4 in width 14.6 in height
Track	555 mm	21.9 in
Wheelbase	512 mm	20.2 in
Weight	50 kg	110 lbs
Maximum payload ¹	75 kg	165 lbs
All-terrain payload ²	20 kg	44 lbs
Speed (max)	1.0 m/s	3.3 ft/s
Ground clearance	130 mm	5 in
Climb grade	45°	100% slope
Traversal grade	30°	58% slope
Operating time	3 hours typical 8 hours standby (no motion)	
Battery	24V 20 Ah Sealed Lead Acid	
Battery charger	Short-circuit, over-current, over-voltage, and reverse voltage protection.	
Charge time	10 hours	
User Power	5V / 12V / 24V Each fused at 5A	
Communication	RS-232 115200 Baud	
Wheel encoders	200,000 pulses/m	
Internal sensing	Battery status Wheel odometry Motor currents	

Table 2: Husky A200 System Specifications

¹: Continuous operation on relatively flat terrain with wide turns

²: Vehicle climbing 30° grade with high-mounted payload, or turning in place in high-friction conditions

Vehicle Equations

As a starting point, Clearpath Robotics recommends using the following relationship between wheel velocity and platform velocity:

$$v = \frac{v_r + v_l}{2}, \omega = \frac{v_r - v_l}{w}$$

v represents the instantaneous translational speed of the platform and ω the instantaneous rotational speed. v_r and v_l are the right and left wheel velocities, respectively. w is the effective track of the vehicle (0.555 m).

Control Modes

Husky A200 supports three distinct control modes: voltage control, speed control, and velocity control.

- **Voltage control**, accessed via CCP message **0x0202**, allows the direct control of motor voltage, as a percentage of 24V.
- **Speed Control**, accessed via CCP message **0x0200**, activates a PID feedback control loop, and allows the specifying of the left and right wheel speeds in m/s.
- **Current Control**, accessed via CCP message **0x0220**, activates a PID feedback control loop, and allows the specifying of the left and right wheel currents in amperes.
- **Velocity Control**, accessed via CCP message **0x0204**, activates feedback control, and a basic kinematic model, which allows the specifying of a target overall translational and rotational velocity of the chassis.

For more details, please refer to the Clearpath Control Protocol documentation, and the documentation specific to your chosen software API.

Navigate to <http://clearpathrobotics.com/downloads> to download the current release of the Clearpath Control Protocol.

SAFETY

Clearpath Robotics is committed to high standards of safety. Husky A200 contains several features which protect both the safety of users and the integrity of the vehicle.

General Warnings

Husky A200 is a rugged and high-performance vehicle. For the safety of yourself and others, always conduct initial experiments and software development with the vehicle raised off the ground. Place a wooden crate, a set of sawhorses, a sturdy storage tub, or any other solid flat structure having a height greater than 6 inches under Husky to keep the wheels clear of the ground (“up on blocks”).

When starting out, favor slower wheel speeds. Husky’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 Husky is operating, keep clear of the wheels, paying particular attention to the pinch hazard which exists between each wheel and the end of the corresponding bumper bar.

E-Stop and Lockout

The red emergency stop button (e-stop) and lockout are located on the back of Husky, opposite the status panel, shown in Figure 4. Power supply to Husky’s motor drivers is enabled by a normally-open relay, which is closed in series with the e-stop switch. When in e-stop mode, the status panel e-stop light will illuminate red, and Husky will not drive.

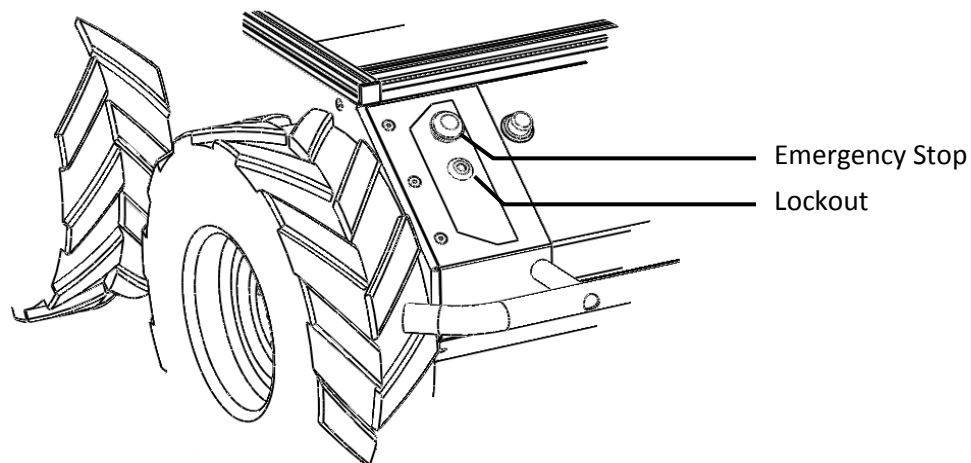


Figure 4: Husky A200 Emergency Stop and Lockout

Always ensure the e-stop button is accessible at all times. Avoid mounting payloads which extend over the rear of Husky and would occlude the e-stop.

The lockout provides a secondary e-stop for periods during which Husky will be left unattended. When in lockout mode, the robot will still power on, but the motors will not drive.

Performance Recommendations

Husky's motors are rated to draw 4A continuous, but they will spike to several times this, particularly when traversing rough terrain, and when turning on the spot. When you are ready to begin running Husky for continuous periods of longer than ten minutes, it is recommended that you monitor current.

In your software, use CCP message **0x4004** to request general system status. The three current fields represent total draw, left motor current, and right motor current, respectively. Monitoring these fields over longer periods of operation will allow you to ensure that you are not putting excessive wear on the motors.

To reduce current draw, consider commanding wider-radius turns from your control software.

Electrical System

Husky is powered by a single 24V sealed lead-acid battery, similar to the type found in electric wheelchairs, golf carts, and other devices. Husky's battery is capable of delivering 1800W—similar to wall mains. This gives Husky's motors their great performance, but it is also enough power to cause severe bodily harm. Please observe the following precautions:

- Do not tamper with the plug attached to the battery.
- Do not tamper with the fuse panel, except to check and change the fuses, and to connect and disconnect the battery plug.
- Do not operate Husky without the battery door in place. The battery is insufficiently restrained without the door, and will come loose, damaging the fuse panel.
- Charge the battery only with the charger provided by Clearpath Robotics.
- Return the battery to Clearpath Robotics for proper disposal.

Lifting and Transport

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

- Husky should be lifted by two persons, firmly gripping the front and rear bumper-bars.
- Ensure that Husky is e-stopped when transporting short distances and powered off when transporting longer distances.

GETTING STARTED

You are ready to go! This section details how to get the wheels rolling.

Before beginning, place Husky “up on blocks”, as described in the Safety section. You can use an inverted storage tub or a wooden crate—anything, really, so long as the wheels are clear of the ground.

PC Setup

If you purchased a computer from Clearpath Robotics with Husky, it is set up and ready to go. Provided on this machine are the officially-supported interfaces for Python and C++, as well as drivers for Player/Stage and ROS, if requested.

For details on how to set up additional computers, please refer to Appendix A for details.

Connecting Computer

Husky’s serial port is located in the user power panel accessible from the user area, shown in Figure 5.

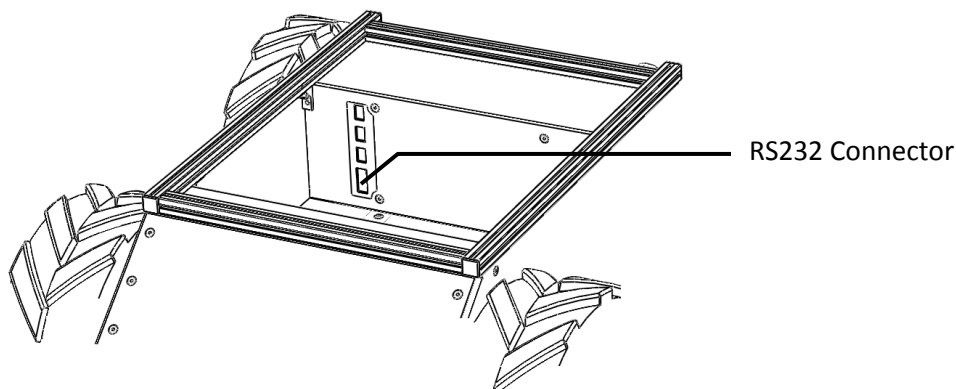


Figure 5: Husky A200 RS232 Connector

The connector is a female DE9, suitable for connection directly to a USB-serial adaptor, or to a PC serial port via a straight-through modem cable (*not* a null modem cable). How you connect the PC to Husky depends on whether the PC has a dedicated serial port:

- **If your PC contains a serial port**, connect the male end of the provided serial cable to the female DE9 on Husky, and the female end of the cable to the male DE9 on the PC.
- **If your PC contains only USB ports**, connect the USB end of the USB-serial adaptor to the PC, and connect the male DE9 to Husky’s serial port.

Connecting Power

Husky A200 comes with the battery fully charged and installed, but disconnected for safety during shipping. To reconnect the battery:

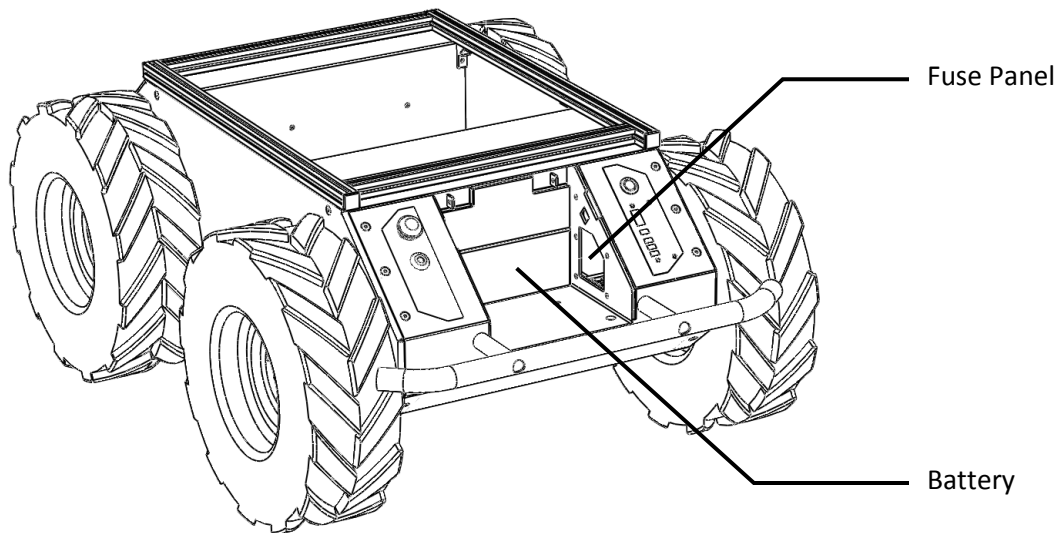


Figure 6: Husky A200 Battery Area

1. Ensure Husky's main power button is in the outer "off" position and the e-stop is activated.
2. Using curled index fingers to grip the battery door latches, press with thumbs on the latch buttons and lift the battery door free of the chassis.
3. Carefully connect the battery plug to the mating connector in the fuse panel, shown in Figure 6. Ensure that it is firmly seated and clicks into place. It will be difficult to connect it in the incorrect polarity.
4. Reseat the battery door cover, ensuring that it has clicked into place and is flush with the rear panel.

To power on Husky, firmly press the power button located above the status panel. It will illuminate blue, and the status panel lights will show a brief test pattern. The **comm** status light will show red, as the PC is not yet communicating with Husky.

The stop status light will be red if the e-stop is pressed. If it is not, press the e-stop button and verify that the stop light illuminates. Leave the platform in an e-stop state until it is successfully receiving commands.

Verification

On the PC, attempt to communicate with Husky using the Python command-line interface (CLI). From a terminal prompt, type:

```
$ python -m clearpath.cli
```

When the connection is made successfully, you should see a readout of information about your Husky A200 platform, followed by a prompt where you may directly issue Clearpath Communication Protocol commands. The comm light will still be illuminated red at this point, as the CLI does not issue commands unless instructed to do so, and thus the connection times out.

At the prompt, issue a command to drive at 0.5 m/s:

```
clearpath:/dev/ttyUSB0$ set_velocity 0.5 0 2.0
```

The comm status light should momentarily switch to green. If you haven't yet done so, release the e-stop, and run the command a second time. This time, the comm light will flash green and the wheels should turn also. After 200ms, Husky times out and reverts to the comm error status. To maintain the commanded speed, use the repeat command to send the velocity command continuously at 20 Hz:

```
clearpath:/dev/ttyUSB0$ repeat 20 set_velocity 0.5 0 2.0
```

With the wheels spinning continuously, try using `request_system_status` to return some key data from Husky:

```
clearpath:/dev/ttyUSB0$ request_system_status
```

You should see three voltages, representing the voltage level of the battery, and the input voltage of the left and right motor drivers, three currents, representing the draw from the battery and the output current at each motor driver, and two temperatures, representing the temperature of each motor driver.

When finished, cease the repeated command and exit:

```
clearpath:/dev/ttyUSB0$ repeat off  
clearpath:/dev/ttyUSB0$ exit
```

For a full list of all available commands, type `help` at the prompt. For help on a specific command, type `help <command>`, for example `help set_velocity`.

ROS-Based Teleoperation

Clearpath Robotics provides an officially-supported Robot Operating System (ROS) driver for Husky A200, based on the Python interface. ROS is a flexible, distributed architecture for controlling robotic systems, and is a recommend method for controlling Husky. Using ROS, you can quickly take joystick control of Husky.

A simple ROS-based tele-operation network is shown in Figure 7. The `joy_node` is a device driver which interfaces with the joystick and produces ROS **Joy** messages. The `clearpath_teleop` node consumes these **Joy** messages, applies scale factors and produces **Twist** messages. The **Twist** messages are received by `clearpath_base`, which uses CCP to forward them as velocity commands to Husky.

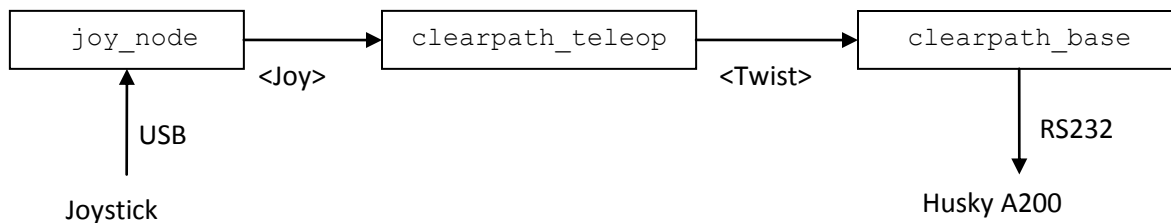


Figure 7: ROS Teleoperation Network

With a joystick connected directly to Husky's PC, and available on `/dev/js0`, launch the teleop script:

```
roslaunch clearpath_teleop teleop_a200.launch
```

You should now be able to drive the robot from the connected joystick.

To drive remotely, set up a second ROS PC as the pilot's computer. It is from here that you will launch `teleop_a200.launch`, but you will need to specify that the `clearpath_base` node run remotely on the Husky's PC rather than locally on the pilot's computer. To do this, you must specify in the launch file a `<machine>` tag giving the details of the Husky computer's ROS installation. A typical configuration is as follows:

```
<machine name="husky-pc" address="husky-pc" user="administrator"
  ros-root="/opt/ros/cturtle/ros"
  ros-package-path="/opt/ros/cturtle/stacks:
    /home/administrator/ros-clearpath" />
```

Then add `machine="husky-pc"` to the `<node>` tag which launches `clearpath_base`, and run `roslaunch teleop_a200.launch` from the pilot machine!

For more details, consult the following pages on the official ROS wiki:

- <http://www.ros.org/wiki/ROS/NetworkSetup>
- <http://www.ros.org/wiki/roslaunch/XML/node>
- <http://www.ros.org/wiki/roslaunch/XML/machine>

OPERATION

This section provides a guide to using the various available control methods on Husky.

Control Protocol

The Clearpath Control Protocol (CCP) is a simple way for users to interface with Clearpath hardware from a PC or other higher-level controller. All Clearpath Robotics hardware implements the subset of messages relevant to the specific chassis configuration present. For example, Husky A200 provides handlers for the differential drive messages, but not for the Ackermann messages.

CCP includes several features intended to increase communication reliability while keeping message overhead and implementation complexity low. It is not intended for multiple devices to be simultaneously connected to a single communication line, removing the need for addressing or bus negotiation.

CCP is a binary serial protocol, documented in full in the CCP Guide. Provided by Clearpath Robotics are implementations in Python, C++, and LabVIEW, as well as a driver for Robot Operating System (ROS) built on the Python implementation. This section provides an overview and introduction to the two low-level implementations.

Python

The below code requests and prints a 1 Hz system status update and ramps the motor driver outputs from 0 to 100% voltage.

```
#!/usr/bin/python
from clearpath import Interface
from clearpath.transports import Serial
import time

cpr = Interface(transport=Serial,
                transport_args={'port': '/dev/ttyUSB1'})
cpr.open()

def status_handler(code, payload, timestamp):
    print(payload)

cpr.add_handler(status_handler, request='request_system_status')
cpr.request_system_status(subscription=1)

for i in range(0, 10):
    left_percent = right_percent = i * 10
    cpr.set_differential_output(left_percent, right_percent)
    time.sleep(0.5)

time.sleep(1)
cpr.close()
```


To use feedback speed control instead of direct voltage control, consider the `set_differential_speed` or `set_velocity` messages. Also, if you would prefer to poll for received messages rather than use callbacks, you may use the `get_waiting` method to receive a list of unhandled messages. For more details on using the Python driver, see the included documentation and examples.

C++

This code is similar to the Python code, except that it uses the built in speed controller to implement a velocity ramp as opposed to the Python demo's use of the raw motor control functionality the protocol also provides.

```
#include <iostream>
#include <cstdio> // want getchar()
#include <cstdlib>
#include <unistd.h>
#include <typeinfo>
#include "clearpath.h"

using namespace std;
int main(int argc, char *argv[]) {
    /* Configure the serial port */
    const char* port = (argc == 2) ? argv[1] : "/dev/ttyUSB0";
    clearpath::Transport::instance().configure(port, 3 /* max retries*/);

    /* Subscribe to some interesting data */
    clearpath::DataSystemStatus::subscribe(1);
    clearpath::DataSystemStatus * cur_status = NULL;

    /* Ramp speed to 0.6m/s over 6 seconds.
     * We use the 1Hz system status message for timing, so we want to
     * begin by waiting for the first message, which is sent shortly
     * after the beginning of the subscription */
    cur_status = clearpath::DataSystemStatus::waitNext();
    cout << *cur_status << endl;
    delete cur_status;

    for(int i=1; i<=6; ++i) {
        // Set motor speed
        clearpath::SetVelocity(0.1*i, 0.0, 1.0).send();

        /* Wait for the next system status message to arrive */
        while(!(cur_status = clearpath::DataSystemStatus::popNext()));
        cout << *cur_status << endl;
        delete cur_status;
    }

    /* Terminate subscriptions */
    clearpath::DataSystemStatus::subscribe(0xffff);

    return 0;
}
```

Distributed with the C++ API are three simple test programs:

`botinfo.cpp`: Requests four pieces of information from Husky A200 and exits

`keyboard_ctrl.cpp/h`: Allows the user to control the velocity of Husky A200 via the keyboard

`dead_reckon.cpp/h`: Allows the user to control the position of Husky A200 via the keyboard. The position is estimated using the onboard encoders.

BATTERY & MAINTENANCE

Husky A200 is built for rugged, long-term use. However, there are steps that can be taken to maintain and extend the life of the platform even further.

Charging

The battery which ships with Husky A200 may be charged off-board, or while still inside of Husky. If charging while installed, you will need to open the access door to the battery area, and disconnect the battery from its connector in the fuse panel. From there:

5. Connect the DC output cable from the charger to the battery terminal connector.
6. Plug the charger power cord into into the charger, and then into a wall receptacle.
7. The POWER LED and CHARGING LED lights on the charger should both be illuminated.
8. When battery is fully charged, the CHARGING LED will turn off. Unplug the charger from the wall, and then disconnect it from the battery.

The battery charger which ships with Clearpath Robotics lead-acid battery-powered products uses a three-state charge cycle:

9. A fast, high current initial charge, until the battery voltage reaches 29.6 V.
10. A topping charge, during which the battery charges at 29.6 V constant voltage until the current decreases to 500 mA.
11. A precision float charge, during which the battery voltage is held at 27.6 V. This mode allows the charger to remain connected to the battery during periods of non-use, keeping it in a state of full charge. When in this mode, the CHARGING LED will be off.

Battery Care

Husky’s power supply is a sealed 24 V lead acid battery pack (VRLA), providing 20 ampere-hours of charge. To maximize the lifetime of the battery, recharge immediately after use, and keep charged to prevent loss in capacity. Battery life expectancy is shown in Table 3 and shelf charge retention in Table 4.

Cycles	Depth of Discharge
200	100%
225	80%
500	50%

Table 3: Battery Life Expectancy

Shelf Time	Battery Charge
None	100%
1 month	92%
3 month	90%
6 month	80%

Table 4: Battery Shelf Life

Husky's battery should never be used or stored in an environment exceeding 40 degrees Celsius (104 °F), and should always be charged at temperatures above freezing. At the optimal operating temperature of 25 degrees Celsius, the discharge time of the battery may be seen in Figure 8.

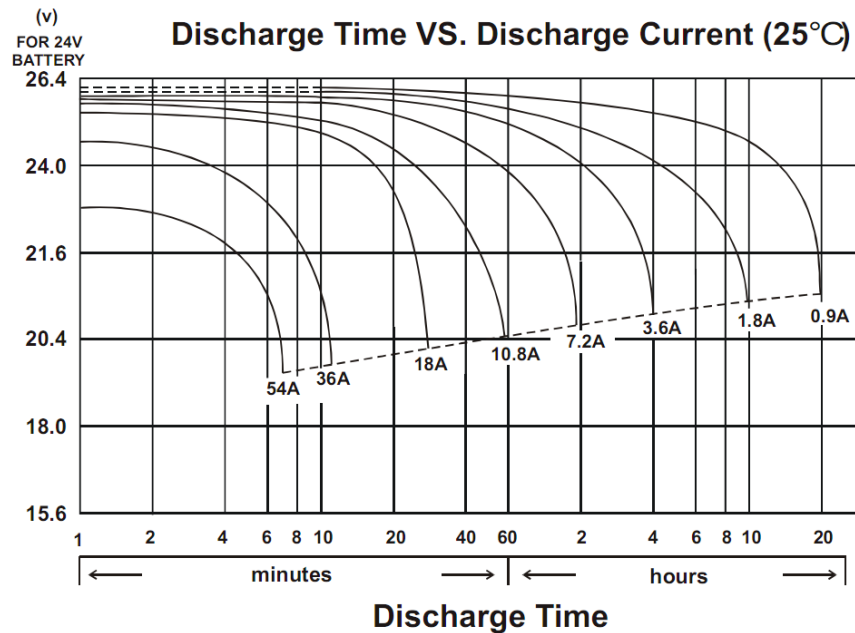


Figure 8: Husky A200 Battery Voltage Characteristics

Wheels

Tire pressure may change with temperature, and should be checked periodically with a pressure gauge. Checking, releasing pressure and inflating a tire are all done through the tire's inflation stem. Tire pressure should not exceed **20 psi**, and lower pressure may be desired based on terrain requirements.

If a tire must be removed, first unfasten the four M5 Socket Head Cap Screws that join the wheel to the axle hub, and slide it off the axle. When replacing, these screws should be tightened to 3.7 ft-lb [5 N-m] torque.

Chassis

Husky is an all-weather robot, but it is not waterproof. Care should be taken that no part of the main chassis is ever submerged in water. When the chassis becomes wet or dirty, wipe it down with a cloth and warm water, and dry with a towel.

If water is suspected to have entered the Husky A200 chassis, remove the battery and allow Husky to fully dry for a period of 24 hours.

TIPS AND TROUBLESHOOTING

Mechanical Tips

Husky's wheels can handle a range of different terrains, but work best when inflated to the proper pressure. Lower tire pressures, for example **10 psi**, ensure better traction in rough and varied terrain, where rocks or other obstacles may be encountered. This has the adverse effect of lowering drivetrain efficiency and decreasing battery life, so high pressures up to a maximum of **20 psi** should be used when driving on flat surfaces.

Troubleshooting

This section lists a few possible issues which may be encountered.

- **No blue light when pressing the power button.** Ensure that the battery is charged and correctly connected. Use a multi-meter to verify the voltage on the battery terminals.
- **Comm light stays red/flickers.** Your program may not be sending motion commands fast enough. If there is a poor quality connection, some commands may be lost, so it may be necessary to increase the frequency of commands sent.
- **E-stop light illuminated.** Twist the red e-stop to release it, and confirm that the robot is not locked out.
- **Battery indicator flashing.** Battery voltage is too low for Husky to drive the motors. Charge the battery and try again.

If you're having some trouble that you don't see here, or the suggested solution isn't working out, please get in touch so we can help you with it.

A. PC SETUP

This appendix details the software on a Clearpath-ready PC, and the setup necessary to prepare a computer for use with Husky.

Python

The recommended starting point for users of Clearpath Robotics products is the software interface written in Python, called `clearpath-py`. In order to use `clearpath-py`, you will need to install Python 2.6 or 3.0, and the `pyserial` module. Most Linux distributions come with a recent Python install, but Windows users will need to visit <http://python.org/download/> to download an installer.

To verify your Python version, open a command prompt and type `python`. You should see an output similar to the following:

```
$ python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:57:41)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If Python is not installed, or you see a version number less than 2.6, you will need to install a newer version from the Python website, or use your Linux distribution's package manager to upgrade to version 2.6.

To check for `pyserial`, type `import serial` at the python prompt:

```
>>> import serial
>>>
```

If you see an error message, you will need to install `pyserial`, found through package management (`python-serial` in the Ubuntu repositories), or from <http://pyserial.sourceforge.net/>.

Navigate to <http://clearpathrobotics.com/downloads> to download the current release of the Python API, and follow the directions in the README file.

ROS

To install ROS, follow the instructions listed at <http://www.ros.org/wiki/ROS/Installation>.

When your ROS installation is complete and verified, use Subversion to check out the Clearpath ROS packages:

```
$ svn co http://clearpath-ros-pkg.googlecode.com/svn/trunk/ clearpath-ros-pkg
```

Ensure that `clearpath-ros-pkg` is in your `ROS_PACKAGE_PATH` environment variable. You can test this using `rospack`:

```
$ rospack find clearpath_base
```

When `rospack` is correctly able to find the Clearpath packages, proceed to compile the message definitions:

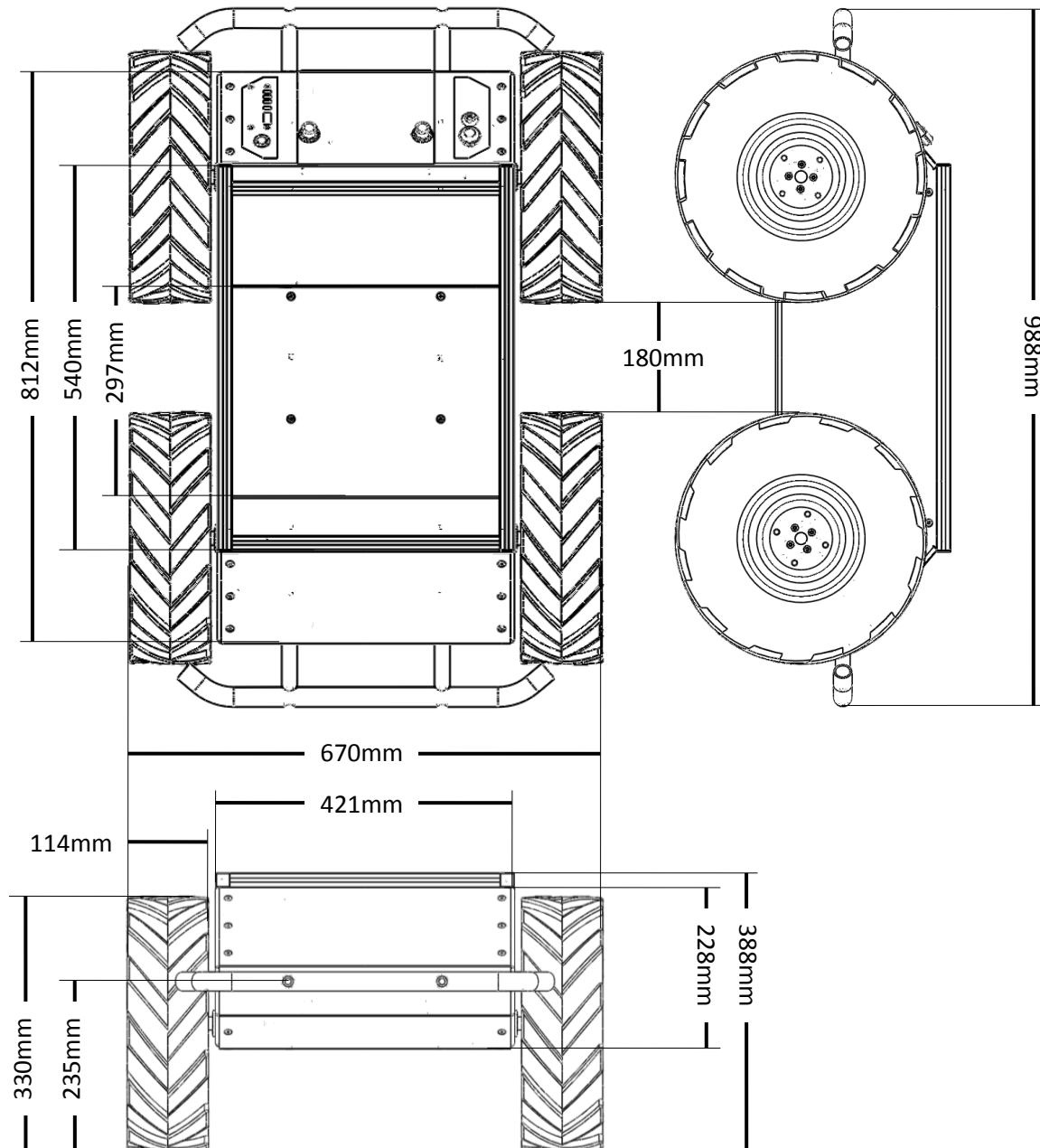
```
$ rosmake clearpath_base
```

When this is complete, you should be ready to go!

C++

Navigate to <http://clearpathrobotics.com/downloads> to access the current release of the C++ API.

B. PRODUCT DIMENSIONS



C. SERVICE AND SUPPORT

Clearpath Robotics is committed to your success and satisfaction. We are located in Waterloo, Ontario, and can accept phone calls from 9AM to 5PM Monday to Friday, at our toll-free number, or emails at any time.

1-800-301-3863
support@clearpathrobotics.com