adept mobilerobots

Seekur

SKR0001

Manual

Revision 3 ● August 29, 2013

Adept MobileRobots Seekur Manual revision 3

August 29, 2013

Copyright Notice:

10 Columbia Drive • Amherst, NH 03458 • USA • Phone +1 603.881.7960 • Fax+1 603.881.3818

www.mobilerobots.com www.adept.com sales@mobilerobots.com

Contents

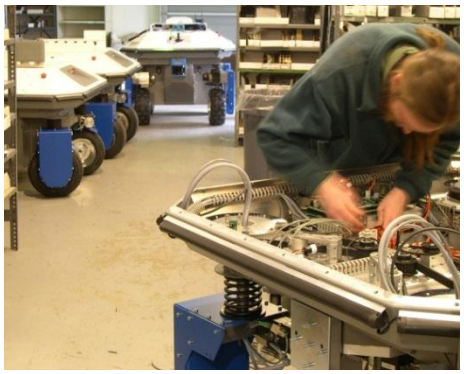
Introduction

Seekur is a four wheel, holonomic, rugged, weatherproof mobile robot system for research and development. Seekur is a complete robotic vehicle system that integrates components such as four traction motors for four independent wheels, four steering motors for the four independent wheels with automatically coordinated steering, power conversion and distribution system, batteries with integrated monitoring and recharging system, and a preprogrammed microcontroller with SeekurOS firmware to automatically achieve requested velocity control of the robotic vehicle, and to use the integrated wheel encoders and gyro or optional IMU sensors to automatically estimate robot position. The robot is programmed by writing software which runs on an onboard computer and communicates with the Seekur robot as well as additional sensors and other devices to control the motion and receive information as it operates. To help write this software, we provide ARIA, ARNL, MOGS and other software development tools with free lifetime upgrades, documentation, and example programs. Also available is access to Adept MobileRobots technical support, as well as the community of other robot users, for any help or troubleshooting required as you develop your projects.

This manual provides information on operating the robot, as well as details and specifications on the robot and its components. Following this introduction page is: a brief overview of safety considerations; information on obtaining support and help; how to unpack and assemble the robot; getting started operating the robot; details on robot components, ports and controls; maintenance; programming and configuring Seekur and SeekurOS; a brief troubleshooting guide; a partial list of available parts and accessories; diagram appendices; and finally a reference chart of physical and other specifications.

Please read this manual as well as supplementary documentation provided for any optional components included with your robot, and save them in a safe place for future reference.

If you have any questions, contact Adept MobileRobots Academic Support at http://robots.mobilerobots.com/wiki/Contact_Support or email support@mobilerobots.com.



Safety

Warning notation used in this manual is as follows:

***DANGER**: This indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury, or damage to the equipment.*

Notes and tips are also given as follows:

***NOTE:** Notes provide supplementary information, emphasizes a point or procedure, or gives a tip for easier operation.*

Qualification of Personnel

This manual assumes that all personnel have been trained and have a working knowledge of the system. The user must provide the necessary additional training for all personnel who will be working with the system. **Skilled persons** have technical knowledge or sufficient experience to enable them to avoid the dangers, electrical and/or mechanical. **Instructed persons** are adequately advised or supervised by skilled persons to enable them to avoid the dangers, electrical and/or mechanical.

All personnel must observe sound safety practices during the installation, operation, and testing of all equipment. To avoid injury or damage to equipment, always remove power by disconnecting AC power supply from the source before attempting any repair or upgrade activity, and power off the robot using the emergency-shutdown switches. Use appropriate lockout procedures to reduce the risk of power being restored by another person while you are working on the system.

Any person who programs, teaches, operates, maintains, or repairs the robot system must be trained and must demonstrate the competence to safely perform the assigned task.

The user must get confirmation from every entrusted person before they start working with the robot that the person:

- *Has received the user's guide.*
- *Has read the user's guide.*
- *Understands the user's guide.*
- *Will work in the manner specified by the user's guide.*

Safety Aspects While Performing Maintenance

Only skilled persons with the necessary knowledge about the safety and operating equipment are allowed to maintain the robot.

DANGER: *During maintenance and repair, the power to the charging station must be turned off. Unauthorized third parties must be prevented, through the use of lockout measures, from turning on power.*

DANGER*: Use caution when working inside the robot. Make sure that all cables are secured away from belts, pulleys, electronics and other components. Prevent all body parts, tools, cables, connectors and loose parts from making contact with circuit boards, exposed contacts, or other electrical components.*

Important Safety Instructions

- Read the installation and operation instructions before using the equipment.
- Do not ride on the robot.
- Monitor operation of the robot at all times
- Properly delineate the robot's operating area with posted signs and barriers. Exclude all unnecessary and unqualified personnel from the robot's operating areas.
- Do not exceed the maximum payload.
- Limit operation to a 35% slope (12% for payloads over 10 kg).
- Do not operate the robot in a reckless or irresponsible manner.
- Do not continue to run the robot after debris, clothing, wires, or any other items have become wound around the robot's axles or wheels.
- Never access the interior of the robot with the charger attached.
- Do not use parts not authorized by Adept.
- Do not use any charger or power supply not supplied by Adept.
- Do not submerge in water at or above bottom of robot.
- The installation and use of products must comply with all safety instructions and warnings in this manual. Installation and use must also comply with all applicable local and national requirements and safety.
- This equipment is not intended for use in any of the following situations:
    - In hazardous (explosive) atmospheres
    - Inside mobile, portable, marine, or aircraft systems
    - In life-support systems
    - In residential installations

Non-intended use can:

- Cause injury to personnel
- Damage the robot or other equipment
- Reduce system reliability and performance

All persons that install, operate, or maintain the robot must:

- Have the necessary qualifications
- Read and follow the instructions in this manual

If there is any doubt concerning the application, ask Adept to determine if it is an intended use or not.

What to Do in an Emergency Situation

Press one of the E-Stop/Emergency Shutdown buttons (red push-button on corner of robot). If a fire occurs, use CO_2 to extinguish the fire. In case of nickel-cadmium battery electrolyte leak, contact MobileRobots support and refer to battery material safety data (available at http://robots.mobilerobots.com or by contacting MobileRobots support).

Help, Documentation and Support

How Can I Get Help?

MobileRobots provides a customer support website at http://robots.mobilerobots.com. This website provides downloads of all manuals, software and device drivers, a searchable knowledge base of information, tips, links to more information and resources on the web, and answers to frequently asked questions.

For public questions and discussions on use of ARIA and other MobileRobots-provided software with users of MobileRobots platforms, MobileRobots provides the aria-users mailing list. See http://robots.mobilerobots.com/wiki/aria-users for archives of past discussions and instructions on joining the mailing list.

For public questions and discussions on robot hardware and general robotics topics with other users of MobileRobots platforms, MobileRobots provides the pioneer-users mailing list. See http://robots.mobilerobots.com/wiki/pioneer-users for archives of past discussions and instructions on joining the mailing list.

To contact MobileRobots' customer support specialists regarding any questions not answered in this documentation, or to troubleshoot problems with your robot, visit http://robots.mobilerobots.com/wiki/Contact_Support, or email support@mobilerobots.com describing your problem. **Include your robot's serial number** and a detailed description of your problem or question.

Factory Repairs

If after reading this manual, you are having hardware problems with your Adept MobileRobots system and are sure that it needs repair, contact us at:

support@mobilerobots.com

In the body of your e-mail message, provide your robot's serial number and describe the problem you are having in as much detail as possible.

Tell us when and how we can best contact you. We will assume e-mail is the best format, unless otherwise notified. We will try to resolve the problem through communication. *If the robot must be returned to the factory for repair, obtain a Repair Authorization Code and shipping instructions from us first.*

Unpacking and Assembly

Carefully inspect all shipping containers for evidence of damage during transit. If any damage is indicated, request that the carrier's agent be present at the time the container is unpacked.

Before signing the carrier's delivery sheet, compare the actual items received (not just the packing slip) with your equipment purchase order. Verify that all items are present and that the shipment is correct and free of visible damage. Your order may be shipped in several containers. If the items received do not match the packing slip, or are damaged, do not sign the receipt. Contact Adept MobileRobots as soon as possible. If the items received do not match your order, please contact Adept MobileRobots (sales@mobilerobots.com) immediately.

Retain all containers and packaging materials. These items may be necessary to store or relocate the equipment at a future date or ship back to Adept MobileRobots or repairs or upgrades.

Seekur comes secured inside a shipping crate. Additional accessories may be shipped separately.

Seekur is secured inside the shipping crate with tie-down straps. If these straps have been removed or broken or any damage or movement of the robot is visible contact Adept MobileRobots as soon as possible.

1. Unfasten the tie-down straps.
2. Release all four locking E-Stop buttons, located on each corner of the robot, by turning counter-clockwise. They will pop up when released.
3. Attach the joystick to the joystick port. Reduce the speed control on the joystick (rotate counter-clockwise).
4. Power on the robot, enable the motors and carefully drive Seekur out of the crate using the joystick controller.

How to perform these steps is described in more detail in the next chapter, **Operation**, on page 12.

Some accessory devices have been preattached to the robot, while others have been removed and packed separately to protect them during shipping. Refer to any supplemental documentation provided about these devices for more assembly and configuration information.

- The optional laser rangefinder, if ordered, is preattached to the front of the robot
- The optional GPS receiver, if ordered, is packed separately. The GPS should be mounted on its bracket in the center of the robot. The GPS data and power connections are through the multipurpose connector at the rear of the robot under the edge of the robot body. (Position of the GPS receiver antenna may be changed in ARIA parameters if moved.)
- The cameras and pan/tilt units, if ordered, are packed separately. Mount these to camera bracket or mount points on front panel of robot.
- Joysticks are packed separately
- Manipulator arms are packed separately
- WRAP wifi interface is packed separately. Attach the WRAP wifi interface bracket to any position on the top equipment mounting rails. Connect the WRAP to the powered ethernet ("PoE") port at the back of the robot under the edge of the robot with its cable.
- Onboard computers, optional IMU and other internal accessories are preinstalled and configured inside the robot.

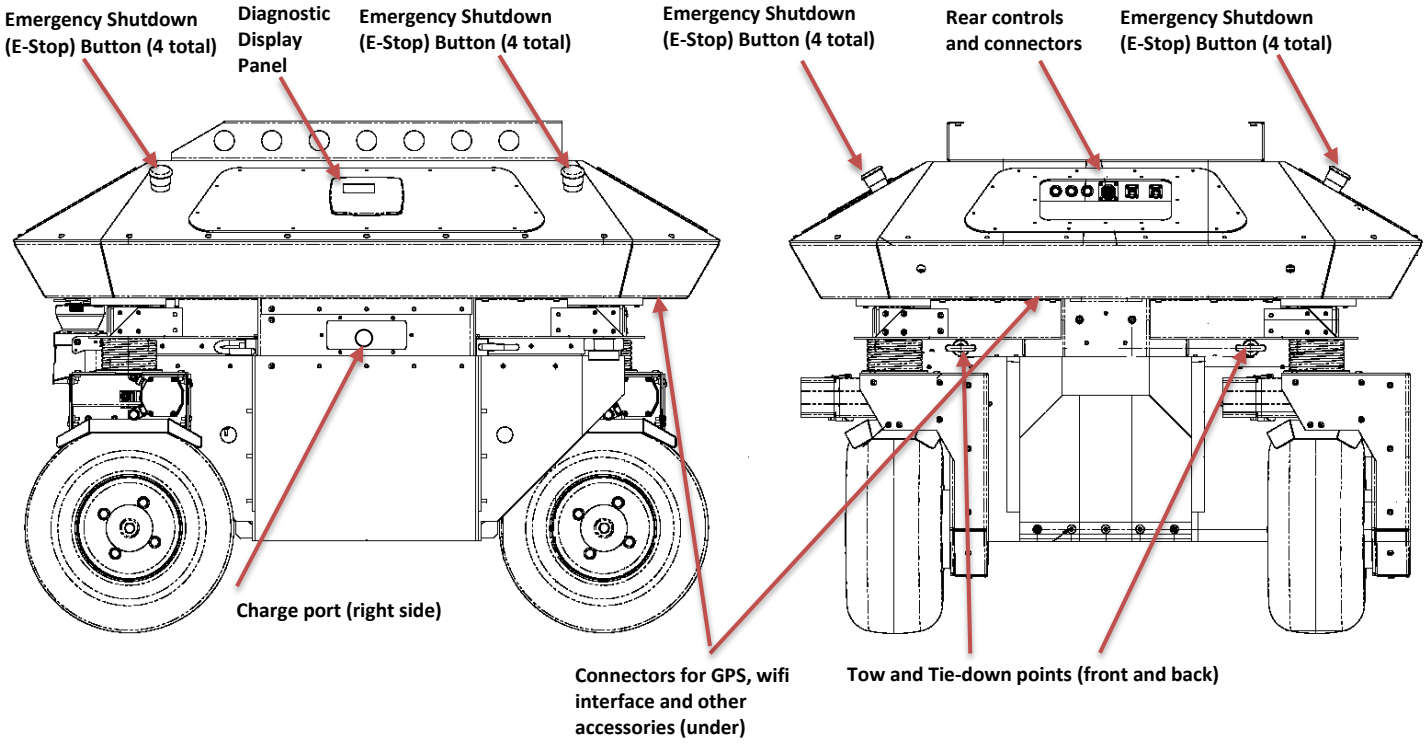
Refer to supplemental documentation on individual accessory devices for more information on assembly and operation. See also **Components, Specifications, Interfaces and Controls** on page 24 for more information on mounting equipment.

Repacking for Relocation

The robot must always be shipped in an upright orientation, fastened securely using the tie down points on the robot.

Power off the robot and depress each E-Stop button when packing.

Operation


Emergency Shutdown (E-Stop) Button (4 total)
Diagnostic Display Panel
Emergency Shutdown (E-Stop) Button (4 total)
Emergency Shutdown (E-Stop) Button (4 total)
Rear controls and connectors
Emergency Shutdown (E-Stop) Button (4 total)
Charge port (right side)
Connectors for GPS, wifi interface and other accessories (under)
Tow and Tie-down points (front and back)


Emergency Shutdown Button (4 total)
Diagnostic Display Panel
Emergency Shutdown Button (4 total)
Emergency Stop Bumper
DANGER
MOBILEROBOTS
Seekur

Seekur Rear Controls

Power On/Off

To power on the robot, first ensure that all four locking emergency shut-off (e-stop) buttons have been released by rotating each counter-clockwise (see discussion on bumpers and E-Stop buttons below below). Next, press the ON button located on the rear of the robot. During initial power on and self-tests, the ON button will blink. The LCD diagnostic panel will display "Initializing…". After successful power on and self-tests, the ON button will glow steady green, and the OFF button will glow steady red. When the Seekur has finished initializing, the LCD diagnostic screen is reset and the MOTORS button will blink twice repeatedly.

If the robot has been powered off for a long period of time, it may be necessary to press the ON button twice (once to re-enable the power monitoring circuit, again to power on the robot.)

To power off the robot, press the OFF button. The entire robot will shut down and all indicator lights will turn off.

Enable Motors

The robot starts up with motion disabled by default. The MOTORS button will blink blue twice repeatedly.

To enable the motors, press the MOTORS button. If this is the first time the motors were enabled since power-on, after a short delay the wheels will initialize and align. Once the wheel alignment procedure has finished, the MOTORS button will glow solid blue, the LCD diagnostic panel will show the MobileRobots logo, and the robot is ready to move.

While the robot controller is still initializing at initial power-on, the motors will not enable. The Motors button will blink rapidly if pressed. After the controller is finished initializing, the motors will enable and the MOTORS button will glow steady blue.

To disable motion, press the MOTORS button. It will blink blue twice repeatedly. Press again to re-enable.

When the robot is not in use, disable the motors for safety, and re-enable only when you are ready to use and monitor operation of the robot. When done with the robot, disable them again.

Wheel Initialization

When the motors are enabled for the first time after robot startup, Seekur will automatically rotate each wheel to unwrap internal cabling and reset them to initial positions, then align and calibrate the wheels. Each wheel will rotate until a predefined center point is reached (some wheels may perform more rotations than others), then all four will be moved to a straight-ahead alignment simultaneously. This alignment procedure may also be done manually from software (RECENTER command #120) or using the re-align button on the joystick. Driving is disabled during wheel alignment. During wheel initialization or if there is an error during wheel initialization, the MOTORS button will blink rapidly

Emergency shut-off (E-Stop) buttons and bumpers

A red emergency shut-off (E-Stop) button is located on each corner of the robot. Pressing any of these buttons down fully will trigger all power to the robot to be disconnected at the battery. The button will lock in place and must be released by rotating counter-clockwise before the robot can be powered on.

A rubber bumper strip is located on each outside edge of the robot. If any of these bumper strips detects a collision, robot motion will stop and be disabled. The LCD diagnostic panel will display “Bumper triggered!”. To re-enable motion after a bumper hit, stop any software, and press the enter (return) button on the LCD diagnostic panel to re-enable the robot.

Charging

A power supply has been provided with the robot to power the charging system in the robot. To recharge the robot, first turn power to the robot off (see above). Remove the weather cover from the robot’s charge port located on the right side. Attach the output cable of the power supply to the charge port on the right side of the robot (align key) and screw completely into place. Plug the power supply into a 20 amp AC outlet. 220-240 VAC is recommended (but charger will also work on 110-120 VAC if necessary) Turn on both switches on the power supply. The power supply should start, and its display should show approximately 0 values. Finally, press the Seekur’s ON button. After initializing and a short delay, Seekur will detect the presence of the power supply and enter charging mode.

Seekur communicates with the power supply and controls its operation. The adjustment knobs on the power supply are not used and have no effect. The display on the power supply shows power supply output. Seekur’s diagnostic LCD panel (see below) shows charge stage, input voltage and power (will show as a negative number for input), and other information. During the main charging stage (“Bulk”) it should show approximately 24-30 volts and approximately 17-22 amps input.

Motion is disabled in charging mode (the MOTORS button will remain off and pressing MOTORS will not enable the motors). The ventilation blower fan will run at high speed.

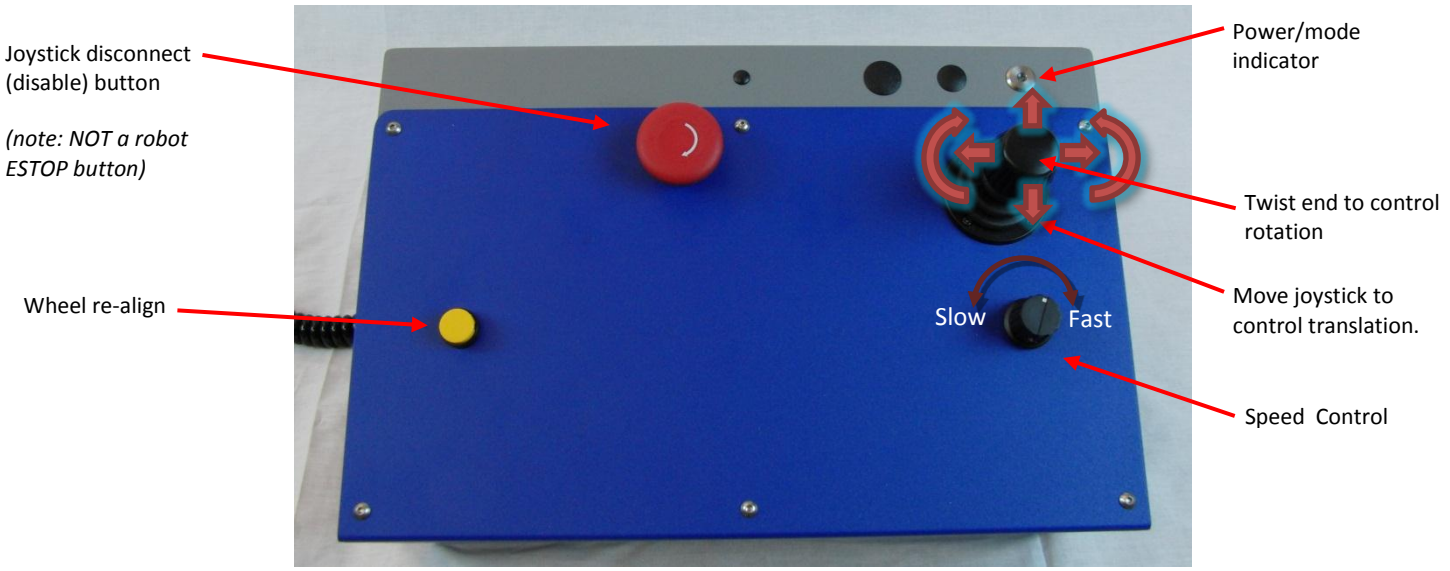
During operation, Seekur will automatically shut down once battery state of charge falls below 20 volts or 10% (this threshold may be changed in SeekurOS parameters, see **SeekurParamManager** on page 56 for details on changing parameters.) You can monitor the battery state of charge or voltage using ARIA, and in MobileEyes (icon in lower-right corner of window). ARIA will log warnings if charge approaches a warning threshold, which may be set in SeekurOS parameters as well.)

To maintain battery performance and lifetime, ensure that the batteries are not discharged below 10% or below 20V.

Seekur’s integrated recharging system will not function if the batteries are completely discharged. Contact support for help if this is the case.

Joystick Control (tethered joystick)

To operate Seekur with the tethered joystick, insert the joystick connector into the keyed joystick port on the rear panel of the robot, and enable motors with the MOTORS button.



The robot's speed is adjusted with the speed knob. Reset it to a low speed before operation. Speed can be carefully increased later as you drive.

To drive the robot forward or backward, press forward or backward on the joystick. To rotate the robot, twist the end of the joystick. To translate sideways, press the joystick left or right.

To manually re-align the wheels, press the yellow re-alignment button.

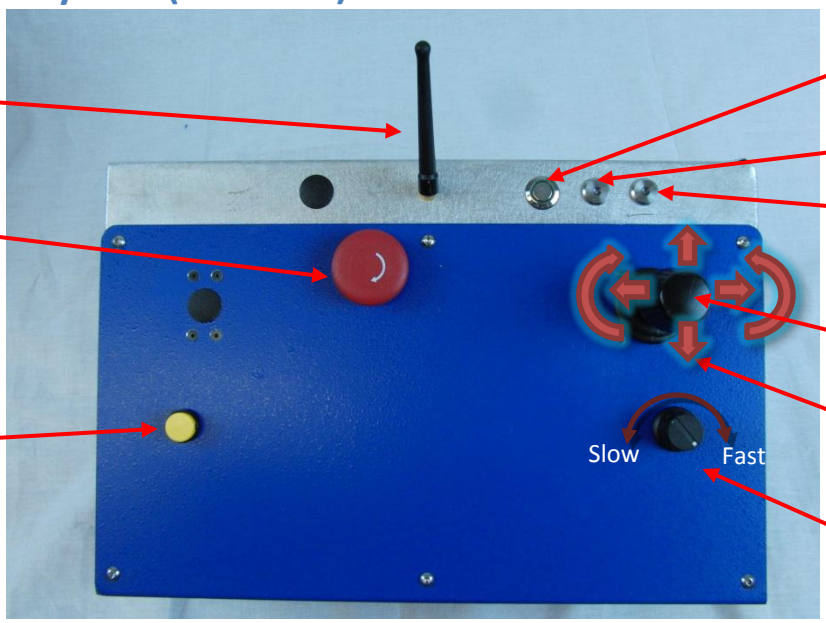
The latching red disconnect button on the joystick disables the joystick. Rotate counter-clockwise to release and re-enable the joystick. (This is *not* a robot e-stop button.)

> *When not actively using the joystick control to operate the robot, either disconnect it from the robot or depress the joystick's disconnect button to prevent accidental movement of the robot.*

Joystick Software Control Mode

The joystick can control the robot in two modes: direct control and software-mediated control. If no software is connected to the Seekur, the joystick controls robot motion directly. When software is running on the robot which has requested joystick control however (See **Joystick** on page 49), joystick speed requests are sent instead to the running software, which can then act further. For example, the ARIA demo, ARNL arnlServer and MOGS mogsServer example programs cause the robot to drive, limited based on configured speed and acceleration parameters, sensed obstacles, map position, and localization state.

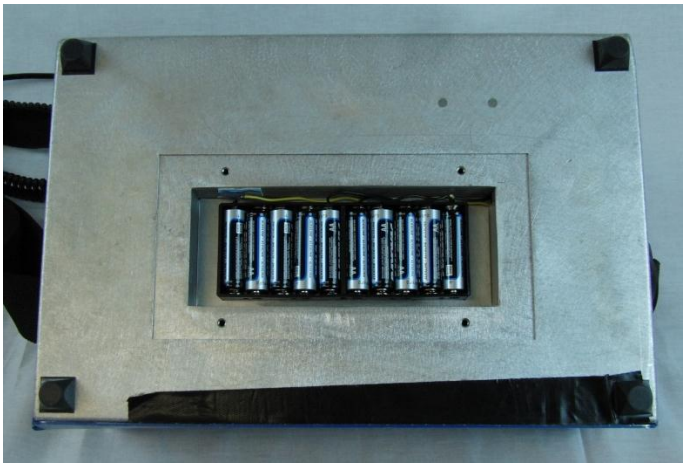
Optional Wireless Joystick (ACT0335)



The optional wireless joystick can be used to operate the Seekur without a cable tether. The wireless joystick consists of two components, the robot-side receiver module, and the joystick control. Attach the receiver module to the joystick port on the robot and switch it on. The indicator LED will glow blue when ready.

Next, switch on the joystick. The power indicator will illuminate. The connection/mode indicator will illuminate if successfully connected with the receiver *and* the robot's motors are also enabled. If connected with the receiver, *and* the robot's motors are enabled, *and* under software-mediated joystick control (see above), the connection/mode indicator will flash steadily.

The wireless joystick requires 10 AA batteries installed to operate. To replace batteries, turn off the joystick and remove the battery cover from the bottom of the joystick.



Diagnostic Display Panel

The display on the side of the robot provides some basic diagnostic information about Seekur. Press F2 to display battery and power information. Press F3 to display temperature monitoring information. Press F4 for firmware version information.

Onboard Computers, Internal Network (LAN) and Wireless (wifi) Network

WRAP wifi interface

One or more industrial extended-temperature-rated single-board computers (SBCs) are mounted internally to the robot. Either Debian Linux or Windows Embedded has been preinstalled on each computer by MobileRobots, along with all MobileRobots and accessory development libraries and required computer device drivers. The first computer's COM1 RS-232 serial connection is used for software control of the robot. Other computer I/O connections are used to connect to other devices on the robot (refer to accessory documentation for details). If more than one computer was ordered, most accessory devices including GPS and pan/tilt/zoom analog camera are connected to the first onboard computer (PC#1). Ethernet devices (LMS-111 laser rangefinder, ethernet cameras) may be used by any of the onboard computers. If multiple MobileRanger C3D stereo cameras were ordered (note that these are NOT ethernet devices), generally each camera will be connected to a different computer.

Power to the computers (and disks) are supplied by Seekur's power distribution board (PDB). Power to the first computer (PC#1) is automatically switched on at startup. Power to additional computers must be switched on by software commands to the robot. See **Power Distribution Board (PDB) for Component Power Supply** on page 32 for details.

An ethernet network (LAN) links the onboard computers, the WRAP wifi interface, and any ethernet devices (such as the LMS-111 laser rangefinder), as well as an external ethernet 8P8C (RJ45) connector.

To access the computer for maintenance, remove the side access panel of the robot.

More technical information including a detailed computer manual is available from the MobileRobots support website: http://robots.mobilerobots.com/wiki/Onboard_Computers. Note that the standard computers for Seekur are extended temperature versions.

Networking

The WRAP wifi interface serves as an 802.11 wireless access point (AP) and allows you to join the Seekur's internal LAN using a laptop computer outside the robot. The WRAP and onboard computers have been preconfigured as follows:

Seekur network configuration:

WRAP wifi network name (ESSID)	Seekur
WRAP wifi network security/WEP key	none
WRAP wifi network assignment	Automatic (dynamic DHCP)
First computer (PC#1) network IP address	10.0.125.32 (static)
Second computer (PC#2, optional) network IP address	10.0.125.33 (static)
Third computer (PC#3, optional) network IP address	10.0.125.34 (static)
Fourth computer (PC#4, optional) network IP address	10.0.125.35 (static)
Fifth computer (PC#5, optional) network IP address	10.0.125.36 (static)
Network gateway	10.0.125.1
WRAP configuration interface	https://admin:mono@10.0.125.1
WRAP configuration login	admin
WRAP configuration password	mono

The WRAP may be reconfigured instead to link to an existing wifi network if desired. It also includes a firewall, network address translation (NAT) and other features. Wifi access point settings such as ESSID name and WEP key may also be configured. For details, see http://robots.mobilerobots.com/wiki/WRAP_Wireless_Bridge/Router

The optional 900 MHz radio bridge provides for longer range communications than wifi. It simply bridges from the robot's internal LAN to your laptop or to an existing network. If bridging to an existing network, you must reconfigure the robot's onboard

computers and other ethernet devices (such as LMS-111 laser) according to your existing network policy. Otherwise, you can simply attach to your laptop or switch and configure settings similar to the robot's onboard computers (on 10.0.125.xxx subnet).

***Warning**: Power to the WRAP or 900MHz radio is supplied by a 12V power-over-ethernet connection. Do not attach any other equipment to the power-over-ethernet/PoE external port on the Seekur.*

Information about changing networking settings on the onboard computers if necessary is available at http://robots.mobilerobots.com/wiki/Onboard_Computer_Network_Configuration

Some devices on the robot (such as the LMS-111 laser rangefinder) use the network for communication to software on the onboard computer, but use a different logical network. The LMS-111 uses a 192.168.0.xxx subnet, for example. Virtual ethernet interfaces of onboard computers using these devices have been configured to allow them to access this alternate logical network.

Logging In

An onboard computer operating system may be accessed by remote connection over the network.

If an onboard computer is running Linux, a remote login connection can be made using **ssh** (Secure Shell). Files may be copied using **sftp** (Secure FTP) or **scp** (Secure Copy). (To establish an ssh connection from Windows to the onboard Linux computer, use the **Putty** application. To establish an ssh connection from Mac OSX to the onboard Linux computer, run ssh from the **Terminal**. To establish an sftp/scp connection from Windows to the onboard Linux computer, use with **WinSCP** application. To establish an sftp/scp connection from MacOSX to the onboard Linux computer, use the **Blowfish** application.)

If an onboard computer is running Windows, a remote connection can be made using Remote Desktop.

If an onboard computer is running Linux, you can log in as **guest** (normal unprivileged user) or **root** (privileged administrative account). The default passwords are **mobilerobots**. *Please change these passwords*. Passwords are changed in Linux using the **passwd** command. New users can be added using the **adduser** command or the "Users and Groups" utility in the "Administration" section of the "System" menu.

Software Overview

The robot comes with the following software. If ordered with an onboard computer, all software has been installed on the onboard computer. Software is provided on CD-ROM, and is available for download (including any updated versions) at http://robots.mobilerobots.com.

Some optional software packages (including ARNL and MOGS libraries) are restricted to customers who purchased these packages. Use the login name and password provided with the robot documentation to download these packages.

All software is available for Windows and Linux.

SeekurOS

At the lowest level, the embedded microcontroller running SeekurOS firmware handles the details of mobility, including maintaining the robot's drive speed and heading over uneven terrain, coordinating all 4 traction motors and 4 steering motors to achieve the requested velocity, as well as determining a position estimate using the encoders, gyroscope, and optional IMU if present, and managing the robot's power and emergency stop systems, batteries, and bumpers. The Seekur controller computes and reports the robot's position estimate (X, Y, and heading), and a variety of operating state information via an RS-232 serial channel. See **Programming and Command Protocol** on page 41 for details on this protocol, more information on how SeekurOS operates, and details on its behavior via configuration parameters stored by the robot.

The embedded controller is not user-programmable. Instead, software running on a computer communicates with SeekurOS.

SeekurOS updates are available for download at http://robots.mobilerobots.com

SeekurParamManager

SeekurParamManager is the utility used to configure various SeekurOS parameters stored by the robot in its nonvolatile "flash" memory. It can also be used to upgrade SeekurOS. SeekurParamManager can be found in **`/usr/local/SeekurParamManager`** on Linux or **`C:\Program Files\MobileRobots\SeekurParamManager`** on Windows and can be downloaded from http://robots.mobilerobots.com.

Onboard Computer Operating System

The robot's onboard computers (if present) have been configured with either a Linux or Windows XP Embedded operating system, as requested, along with all drivers and software needed for devices included with the computer.

ARIA

ARIA is the core development library or SDK for use with the Seekur. It is a C++ library (with wrapper libraries also available for Python and Java).

ARIA is available with all robots.

On Linux is can be found at **`/usr/local/Aria`**, and on Windows at **`C:\Program Files\MobileRobots\Aria`** and in the Start Menu. ARIA includes full API reference documentation in its doc subdirectory, as well as example programs in the examples directory, and full source code distributed as free software under the terms of the GNU General Public License.

ARIA updates and additional information are available for download at http://robots.mobilerobots.com/wiki/ARIA.

ARNL

ARNL is a development library or SDK for including accurate indoor laser localization and flexible, reliable autonomous navigation capabilities in your software.

ARNL is included with all robots ordered with the Laser Navigation or MOGS Outdoor Navigation packages.

ARNL can be found installed on Linux at **`/usr/local/Arnl`**, and on Windows at **`C:\Program Files\MobileRobots\ARNL`**. The ARNL installation includes the localization and navigation libraries, as well as compatible ARIA libraries. It includes a full API reference manual in the doc subdirectory, as well as example programs in the examples directory. Refer to ARNL's README.txt file and API reference manual for more information on getting started using it.

ARNL updates and additional information are available for download at http://robots.mobilerobots.com/wiki/ARNL.

MOGS

MOGS is a development library or SDK for incorporating GPS positioning and flexible, reliable autonomous navigation capabilities in your software.

MOGS can be found installed on Linux at **`/usr/local/Arnl`** and on Windows at **`C:\Program Files\MobileRobots\ARNL`**. The MOGS installation includes the MOGS localization libraries, as well as compatible ARIA libraries. It includes a full API reference manual in the doc subdirectory, as well as example programs in the examples directory. Refer to MOG's MOGS-README.txt file and API reference manual for more information on getting started using MOGS.

MOGS updates and additional information is available at http://robots.mobilerobots.com/wiki/MOGS

Mapper3

Mapper3 is an application used for converting and editing maps for use with ARNL, MOGS and MobileSim.

Mapper3 is available for download at http://robots.mobilerobots.com/wiki/Mapper3.

MobileSim

MobileSim is the MobileRobots simulator. If you run MobileSim first, ARIA will automatically connect to MobileSim instead of the real robot. This allows software to be tested with the simulator on any computer before using the real robot, without recompilation or any changes.

MobileSim is available for download at http://robots.mobilerobots.com/wiki/MobileSim.

MobileEyes

MobileEyes is a graphical application for remote visualization, teleoperation, and software configuration. It communicates with onboard robot software via the wireless network and the ArNetworking system (included with ARIA), and can run on any PC or laptop.

MobileEyes is available for download at http://robots.mobilerobots.com/wiki/MobileEyes.

ArVideo

ArVideo is a library for acquisition of images from cameras, and optional ArNetworking server components that provide images to MobileEyes or other ArNetworking client software.

ArVideo is available at http://robots.mobilerobots.com/wiki/ArVideo.

Software for Accessory Devices

Additional development libraries for use with some accessory devices and options are provided by MobileRobots or the original manufacturer of the device. These libraries can be downloaded from http://robots.mobilerobots.com/wiki/Software. All other accessory devices are supported in ARIA.

Software Demonstrations and Quick Start

This section will walk you through running some example programs from ARIA and MOGS.

You can run these examples either on the robot using actual robot hardware, or, you can simulate the robot on your own laptop or desktop workstation, by first simply running the MobileSim simulator and selecting the **seekur** robot type (however, not all hardware features and devices are simulated.)

ARIA Demo

ARIA includes an example called demo. This is as simple text mode program that connects to the robot and other devices and displays information read from them. You may also drive (teleoperate) the robot using the keyboard, test movement of a pan/tilt camera, etc.

Onboard Computer Running Linux:

1. If using simulation, install MobileSim and ARIA from the CD or from http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN)** on page 17 above). Log in as **guest** with password **mobilerobots**.
3. Change to the ARIA examples directory with the following command:

```
cd /usr/local/Aria/examples
```

4. Run demo with the following command:

```
./demo
```

Onboard Computer Running Windows:

1. If using simulation, install MobileSim and ARIA from the CD or from http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using Remote Desktop (See **Onboard Computers, Internal Network (LAN)** on page 17 above). Log in as **Administrator** with password **mobilerobots**.
3. Double click the ARIA Demo icon on the desktop to run ARIA Demo, or run it from the Start menu (All Programs -> MobileRobots -> ARIA -> demo).
4. Or, to run it from a command prompt:
5. Open a command prompt (Start->All Programs->Accessories->Command Prompt)
6. Change to the ARIA programs directory with the following command:

```
cd "\Program Files\MobileRobots\ARIA\bin"
```

7. Run demo with the following command:

```
.\demo.exe
```

Demo will connect to the robot, displaying information about the connection and the robot such as Name and Subtype.

Demo starts in teleoperation/drive mode. Use the arrow keys to drive the robot. Use the space bar to stop.

Ensure there is adequate clearance (at least 5 meters) on all sides of the robot before driving. The Seekur moves fast!

Seekur's motors must be enabled with the MOTORS button before it can drive.

Use other keys on the keyboard to switch to different modes. Press ? for a list of modes. For example, laser mode (L key) connects to the laser and displays information read from it. Direct command mode (D key) lets you send individual commands directly to the SeekurOS firmware on the robot controller (See **Client Commands** on page 46 for list of commands).

Press Control-C or Escape to exit.

ArNetworking Demo Server and MobileEyes

ARIA's ArNetworking library includes an example server which can be used with the MobileEyes remote graphical user interface to observe and control the robot. You can use the source code of the example server to integrate ArNetworking into your own software for use with MobileEyes or other remote client.

To use MobileEyes on your laptop or other computer, download and install MobileEyes from the CD or from http://robots.mobilerobots.com/wiki/MobileEyes.

Onboard Computer Running Linux:

1. If using simulation, install MobileSim and ARIA from the CD or http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN**) on page 17 above). Log in as **guest** with password **mobilerobots**.
3. Change to the ArNetworking examples directory with the following command:

```
cd /usr/local/Aria/ArNetworking/examples
```

4. Run serverDemo, with connection to the laser rangefinder with the following command:

```
./serverDemo -connectLaser
```

Onboard Computer Running Windows:

1. If using simulation, install MobileSim and ARIA from the CD or http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using Remote Desktop (See **Onboard Computers, Internal Network (LAN**) on page 17 above). Log in as **Administrator** with password **mobilerobots**.
3. Double click the ArNetworking serverDemo with laser icon on the desktop or run it from the Start menu (All Programs -> MobileRobots -> ARIA -> ArNetworking serverDemo).
4. Or, to run it from a command prompt:
    a. Open a command prompt (Start->All Programs->Accessories->Command Prompt)
    b. Change to the ARIA programs directory with the following command:

```
cd "\Program Files\MobileRobots\ARIA\bin"
```

    c. Run serverDemo with connection to the laser rangefinder with the following command:

```
.\serverDemo.exe -connectLaser
```

serverDemo will connect to the robot, displaying information about the connection and the robot such as Name and Subtype. When it finishes connecting to the robot and laser, it will indicate that it has opened a server port for remote connections.

5. Next, run MobileEyes on your laptop.
6. For Robot Server, enter the address of the onboard computer, 10.0.125.32 (no user name or password are necessary, leave these fields empty) and click Connect.

MobileEyes will connect to serverDemo. The robot is represented as a red oval. Readings from the laser rangefinder are represented by blue dots. Data about the robot (position, velocity, debugging information) are displayed in the Details window, to open these enable Details and Custom Details in the View menu. Battery level and other indicators are shown in the status bar at the bottom of the MobileEyes window. To drive the robot using MobileEyes, click the Drive button on the toolbar, and adjust the speed slider to a low speed. Drive by clicking the arrow buttons in the popup tool window which appears, or using the arrow keys on the keyboard. If a USB joystick or game controller was connected when MobileEyes runs, then you can also use that to drive the robot (click the joystick button in the bottom status bar of the robot).

Ensure there is adequate clearance (at least 5 meters) on all sides of the robot before driving. The Seekur moves fast!

Seekur's motors must be enabled with the MOTORS button before it can drive.

Press Control-C or Escape to exit from serverDemo.

Note, if the remote connection to the robot is closed, then serverDemo will automatically quit as well. For information about running serverDemo or other programs in the background, rather than attached to a remote connection terminal in Linux, see http://robots.mobilerobots.com/wiki/Running_A_Linux_Program_In_The_Background.

MOGS or ARNL Demo Server and MobileEyes

MOGS and ARNL each include an example server which can be used with the MobileEyes remote graphical user interface to observe and control the robot. You can use the source code of the example server to integrate MOGS or ARNL into your own software.

To use MobileEyes on your laptop or other computer, download and install MobileEyes from the CD or from http://robots.mobilerobots.com/wiki/MobileEyes.

Onboard Computer Running Linux:

1. If using simulation, install MobileSim and ARIA from the CD or http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN)** on page 17 above). Log in as **guest** with password **mobilerobots**.
3. Change to the Arnl examples directory with the following command:

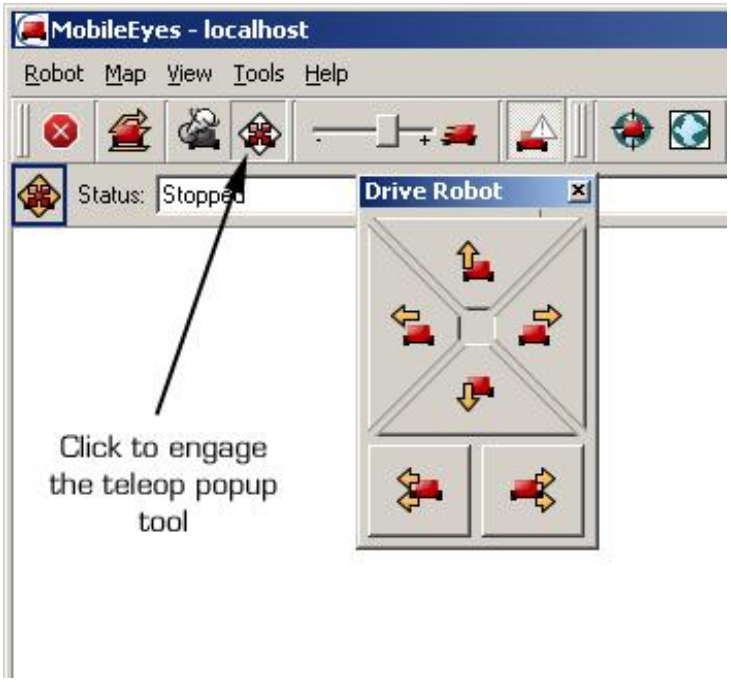
```
cd /usr/local/Arnl/examples
```

4. Run mogsServer with the following command (the ARNL example is called arnlServer).

```
./mogsServer
```

Onboard Computer Running Windows:

1. If using simulation, install MobileSim and ARIA from the CD or http://robots.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekur" robot type.
2. If using the actual robot, join the Seekur wireless network, and log in to the onboard computer (address 10.0.125.32) using Remote Desktop (See **Onboard Computers, Internal Network (LAN)** on page 17 above). Log in as **Administrator** with password **mobilerobots**.
3. Double click the mogsServer (or arnlServer) icon on the desktop or run it from the Start menu (All Programs -> MobileRobots -> ARNL -> mogsServer or arnlServer).



4. Or, to run it from a command prompt,
    a. Open a command prompt (Start->All Programs->Accessories->Command Prompt)
    b. Change to the ARNL programs directory with the following command:

```
cd "\Program Files\MobileRobots\ARNL\bin"
```

    c. Run mogsServer (or arnlServer)

```
.\mogsServer
```

The server will connect to the robot and other devices, displaying information about the connection and the robot such as Name and Subtype. When it finishes connecting to the robot, GPS, and laser, it will indicate that it has opened a server port for remote connections.

5. Next, run MobileEyes on your laptop.
6. For Robot Server, enter the address of the onboard computer, 10.0.125.32 (no user name or password are necessary, leave these fields empty) and click Connect.

MobileEyes will connect to the server. The robot is represented as a red oval. Readings from the laser rangefinder are represented by blue dots. Data about the robot (position, velocity, debugging information) are displayed in the Details windows, to open these enable Details and Custom Details in the View menu. Battery level and other indicators are shown in the status bar at the bottom of the MobileEyes window.

To navigate autonomously, the robot requires a map, and initialization of MOGS or ARNL. Refer to the ARNL and MOGS guides for more information.

Quick reference instructions on creating maps and initializing MOGS are available in the ARNL docs directory (GPSMapping.txt) and at http://robots.mobilerobots.com/wiki/Getting_Started_with_MOGS.

Quick reference instructions on creating maps for ARNL is available in the ARNL docs directory (Mapping.txt).

Before MOGS or ARNL are initialized, they are in a "Lost" state. When lost, you cannot drive the robot in Safe Drive Mode. To disable unsafe drive mode, click the Safe Drive button in the toolbar in MobileEyes. You can then use the drive controls in MobileEyes to drive the robot, however, no checking for collisions using the laser rangefinder is performed when in unsafe drive mode!

NO COLLISION DETECTION is performed using sensing when in Unsafe drive mode.

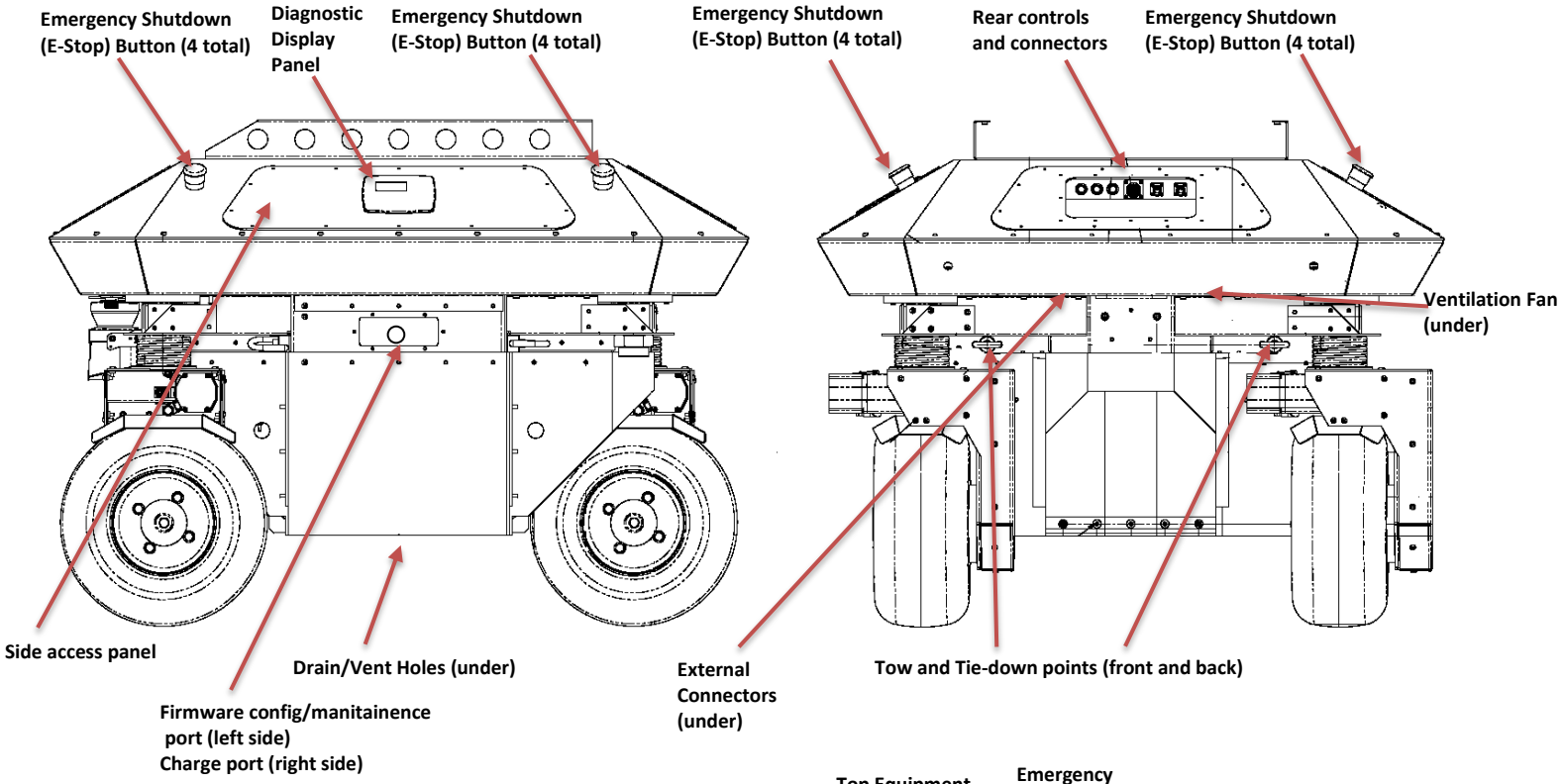
Ensure there is adequate clearance (at least 5 meters) on all sides of the robot before driving. The Seekur moves fast!

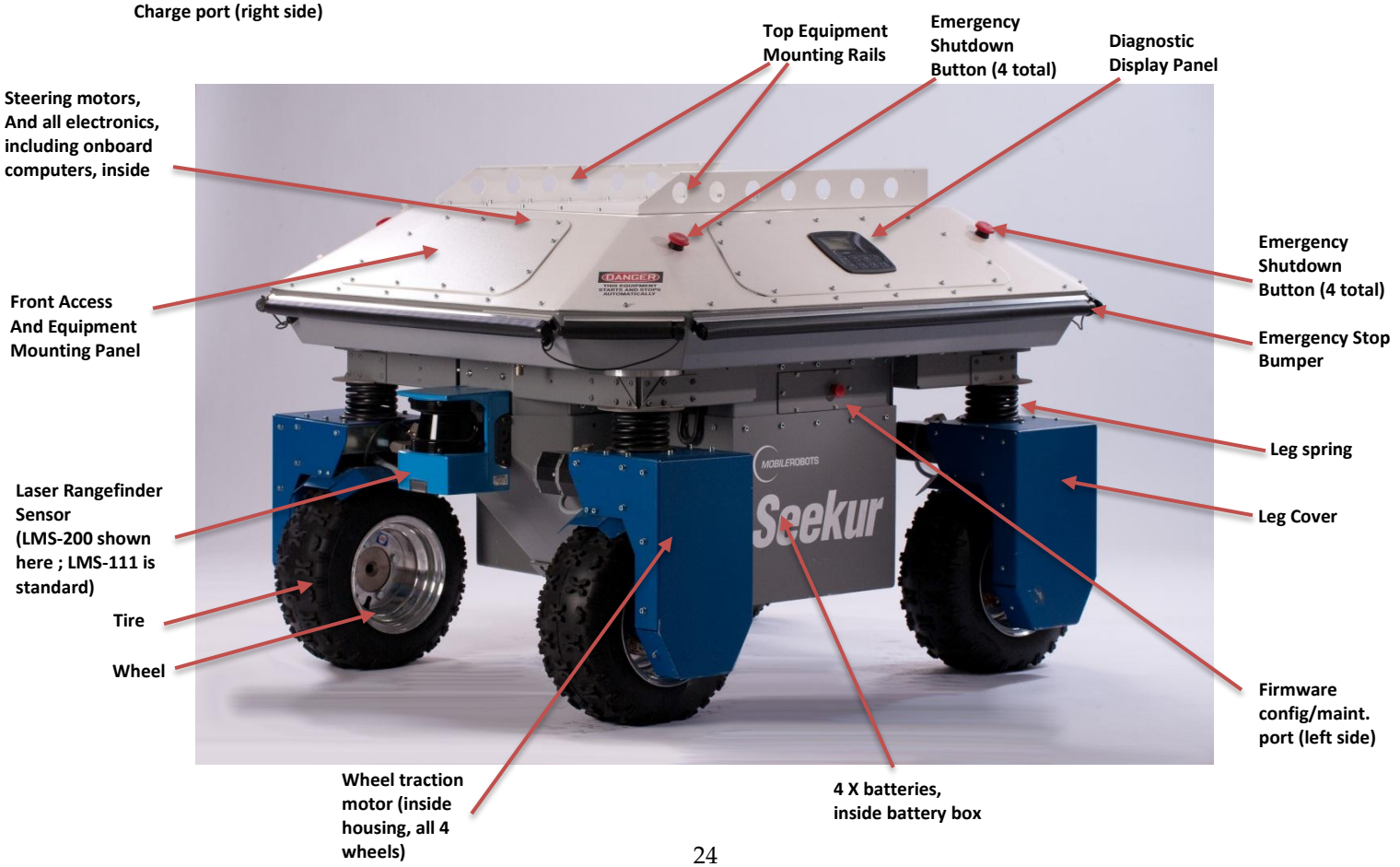
Seekur's motors must be enabled with the MOTORS button before it can drive.

Press Control-C to exit from mogsServer.

Note, if the remote connection to the robot is closed, then mogsServer will automatically quit as well. For information about running mogsServer or other programs in the background, rather than attached to a remote connection terminal in Linux, see http://robots.mobilerobots.com/wiki/Running_A_Linux_Program_In_The_Background.

Components, Specifications, Interfaces and Controls





External Ports & Connections

All external ports and connectors on Seekur are waterproof and designed to maintain the environmental rating of Seekur. Make sure all connectors are firmly connected, and connections are only changed while Seekur is in a dry, sheltered location. (Note, some external cables on Seekur, e.g. bumper wiring, camera, laser ethernet, do not use a removable connector, but instead pass through a waterproof rubber gasket.)

Name	Location	Connector	Mating Connector	www.digikey.com part #	www.mouser.com part #
Ethernet 12V PoE (passive) for WRAP wifi router, or radio	Rear of robot, under edge of body	Amphenol RJF2SA1N 8P8C (RJ45) receptacle	Amphenol RJF6MN protective housing on any CAT5/6 cable with 8P8C (RJ45) modular plug	Protective housing: APC1043-ND	Protective housing: 523-RJF6MN
Ethernet LAN (optional)	Rear of robot, under edge of body	Amphenol RJF2SA1N 8P8C (RJ45) receptacle	Any CAT5/6 cable with 8P8C (RJ45) modular plug.	Protective housing: APC1043-ND	Protective housing is 523-RJF6MN
Joystick	Rear control panel	Amphenol Amphelite AL00F9-35S	Amphenol Amphelite AL00F9-35P	n/a	654-AL00F9-35P
Aux	Rear control panel	Amphelite AL00F9-35S	Amphenol Amphelite AL00F9-35P	n/a	654-AL00F9-35P
Multipurpose (optional, included if GPS)	Rear of robot, under edge of body	66-pos Amphenol Amphelite AL00F19-35S	Amphenol Amphelite AL06F19-35P	n/a	654-AL06F19-35P
MobileRanger Stereo Camera (optional)	Front panel	Amphenol RJF2SA1N 8P8C (RJ45) receptacle	Amphenol RJF6MN housing on CAT6 cable with 8P8C (RJ45) modular plug	Protective housing: APC1043-ND	Protective housing: 523-RJF6MN
USB (optional)		Amphenol USBFTV21N type A USB receptacle	Amphenol USBFTV6N or any USB type A male plug	APC1107-ND	523-USBFTV6N
Microcontroller Maintenance	Side of robot.	13-pos Amphenol Amphelite AL00F11-35S	Amphenol Amphelite AL06F11-35P	n/a	654-AL06F11-35P
Laser Rangefinder Power	Front, under edge of body.	14-pos Tyco/TE CPC 796271-1	Tyco/TE CPC 796272-1	796273-1-ND	571-796272-1
Gasket for RVision camera cable or other cable of same size.	Front panel	Water-tight cable pass-through gasket assembly. MobileRobots part #MCH	Replacement screw-in gasket is McMaster-Carr part number 7310K15, thread size PG-21 (for cord diameter 0.51 inches – 0.71 inches)	n/a	n/a
Antennas on WRAP (2.5 Ghz, 5.5dBi omni-directional antennas for wifi)	WRAP wifi router	N-type	N-type, male, straight	n/a	n/a

***WARNING**: The external 8P8C (RJ45) ethernet connection for the WRAP or 900MHz radio is a passive power-over-ethernet (PoE) connection with 12V electrical power. Do not attach any other equipment to the power-over-ethernet 8P8C (RJ45) external port on the Seekur.*

Microcontroller maintenance serial port pin assignment:
13-pos Amphenol AL00F11-35S

Adapter to DB-9 serial is provided with the robot, spare is available from MobileRobots.

Pin	Function
2	SeekurOS maintenance/configuration connection only (RxD)
3	SeekurOS maintenance/configuration connection only (TxD)
4	Normally unused. But if no onboard computer, then RxD for robot control.
5	Normally unused. But no onboard computer, then TxD for robot control.
6	DSR
7	Signal GND
8	Signal GND
9	RI
10	Signal GND
11	Unused
12	Unused
13	Power GND

LMS111 LRF power:
14-pos CPC Tyco 796272-1

Pin	function
1-3	NC
4	Power +24V regulated
5	Power GND
6-8	NC
9	Signal GND
10-11	NC
12	Heat +24V raw
13	NC
14	Heat GND

Multipurpose connector (GPS, others):
66-pos Amphelite connector AL00F19-35S

Pin	Function
1-55	Unassigned
56	+12V GPS
57	Power GND GPS
58	Unassigned
59	RxD GPS (to DB9 pin 2)
60	TxD GPS (to DB9 pin 3)
62	Signal GND GPS (to DB9 pin 5)
63-66	Unassigned

Joystick and AUX CAN pin assignment:
Amphelite AL00F9-35S

Pin	Function
1	CAN H
2	CAN Vcc
3	Shield
4	CAN GND loopback (connection detect)
5	CAN L
6	CAN GND

Adding New Ports

Additional unused connector locations are available at the front and rear of the robot, under the edge of robot body. Connectors can be added at these locations if ordered from Adept MobileRobots at time of order or by contacting sales and support, or by removing the panel blank that covers the hole and adding your own connector.

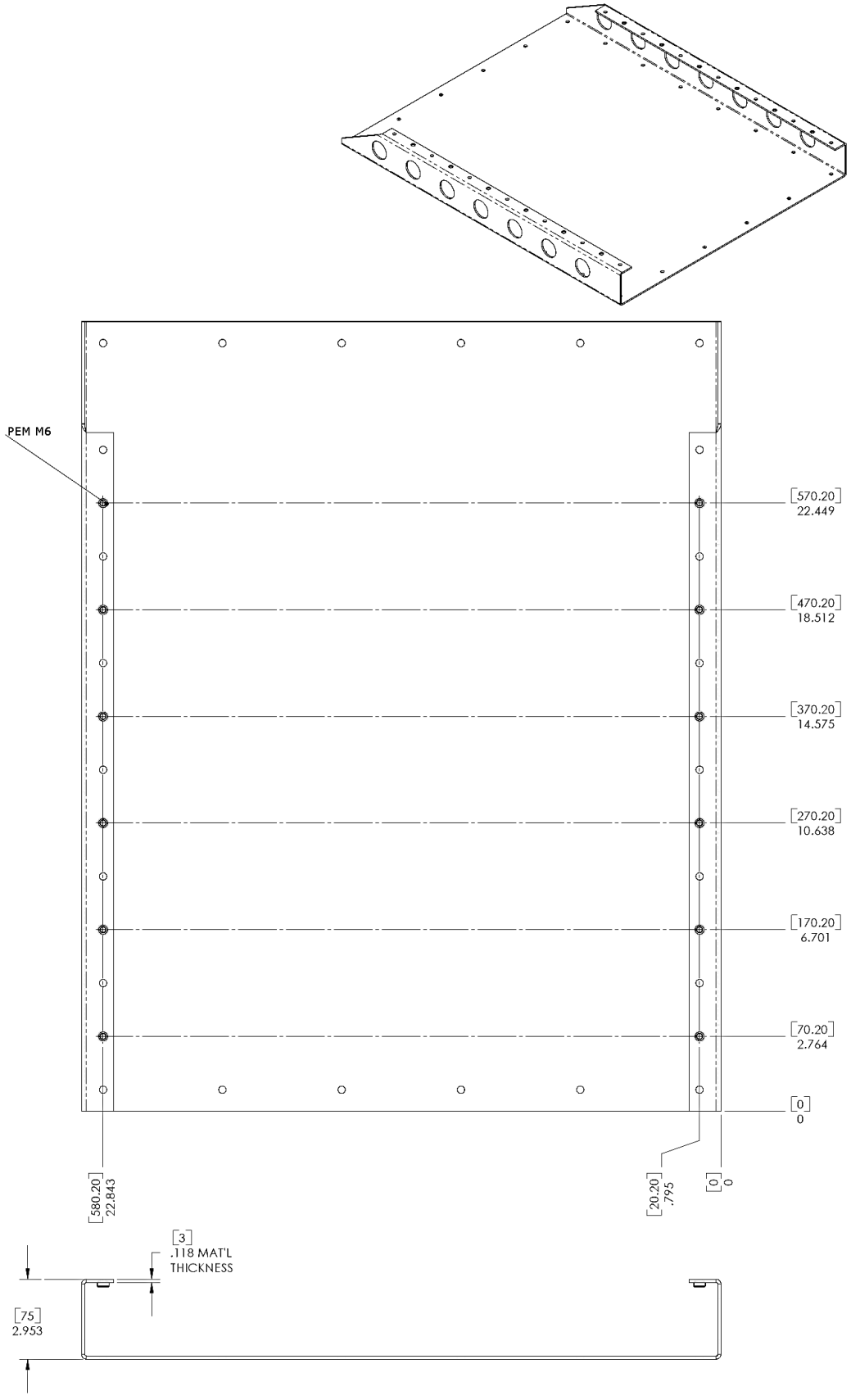
Pins may also be added to the multipurpose connector provided with all robots located at the rear of the robot under the edge of the robot body (used for GPS, for example) by pressing them though the rubber plug, and crimping wiring to the pin inside the robot, and to the connector outside the robot. (See table above for connector and pin part numbers.)

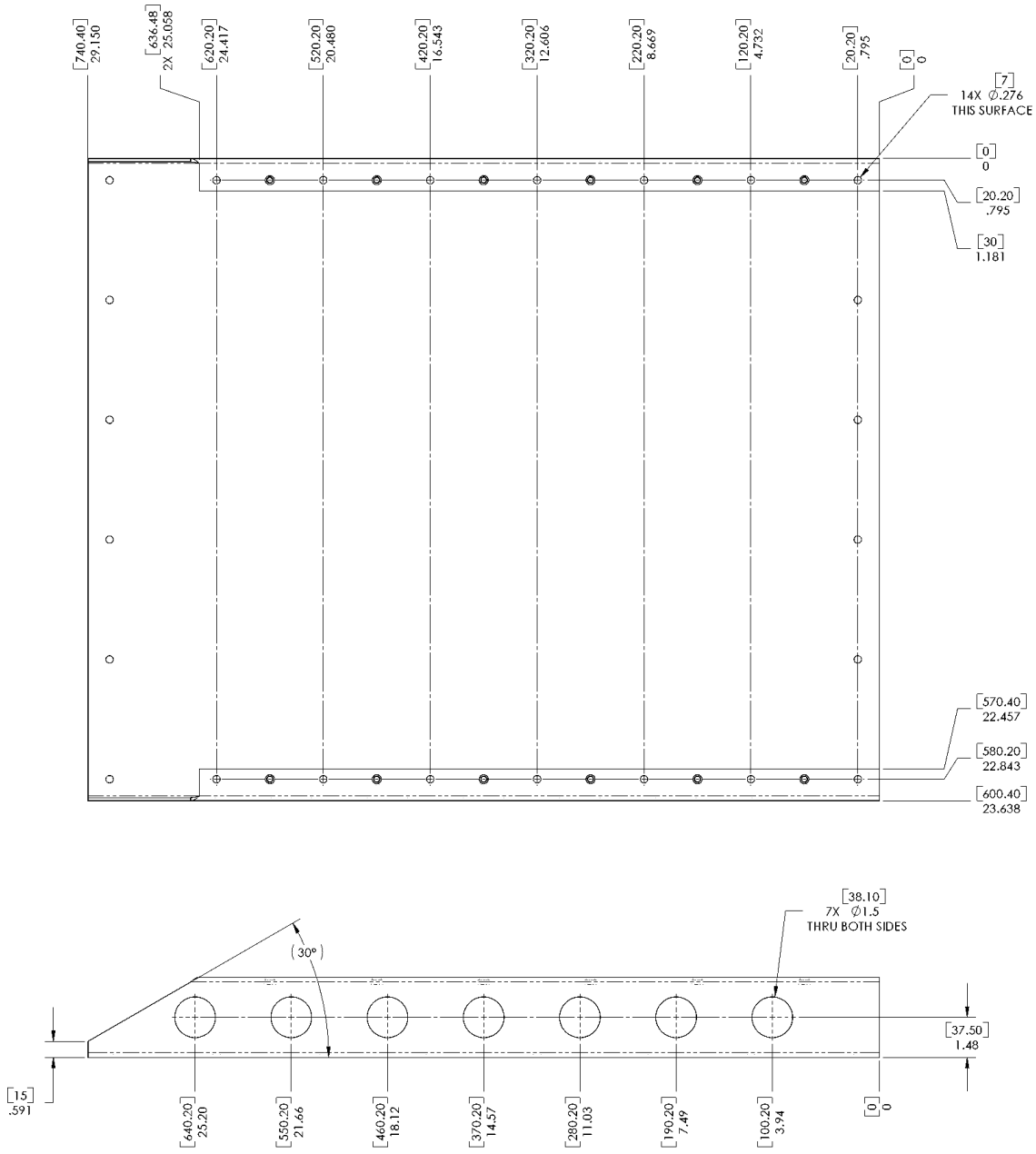
Adding Equipment

User-added equipment may be added to several parts of the robot. Information and drawings are given below. Drawings and CAD models for some parts are also available for download at http://robots.mobilerobots.com or by contacting support@mobilerobots.com.

Top Panel with Equipment Mounting Rails (MCH1875):

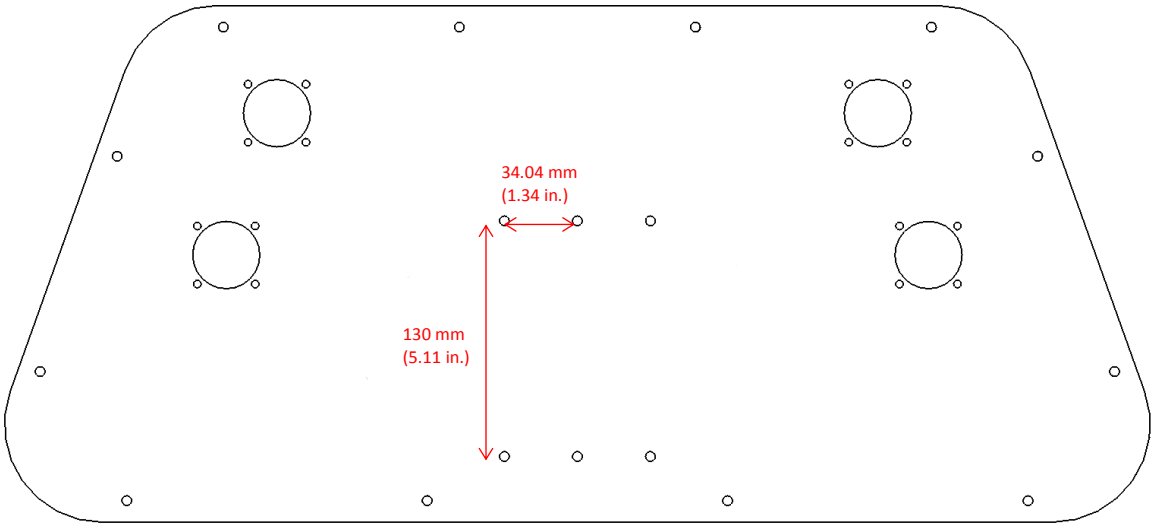
Equipment may be added to mounting rails on the top of the robot. The rails are 56 cm (22.05 in.) apart, with 7mm diameter holes drilled 10 cm (3.94 in.) apart, as well as tapped M6 screw holes spaced 10 cm (3.94 in.) apart along the rails.


PEM M6
[570.20]
22.449
[470.20]
18.512
[370.20]
14.575
[270.20]
10.638
[170.20]
6.701
[70.20]
2.764
[0]
0
[580.20]
22.843
[20.20]
.795
[0]
0
[3]
.118 MAT'L
THICKNESS
[75]
2.953

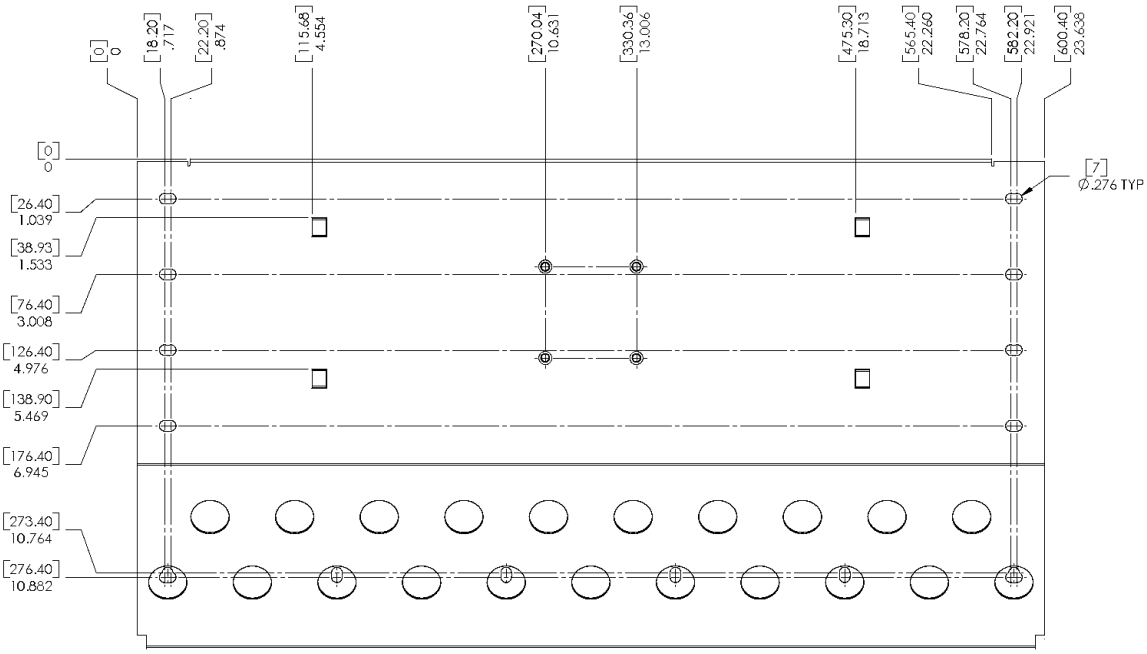


Front Panel:
Equipment may also be attached to the front access panel, or at the laser rangefinder mounting locations at the front and rear of the robot.

Optional Front Panel with Equipment Mounting Points (MCH1915):
An optional front panel MCH1915 with connector locations and mounting locations for the optional pan/tilt unit (PTU), stereo camera, or other sensors or equipment is available (specify types of connectors required, or blanks.), or is included if a PTU or stereo camera was ordered with the robot.

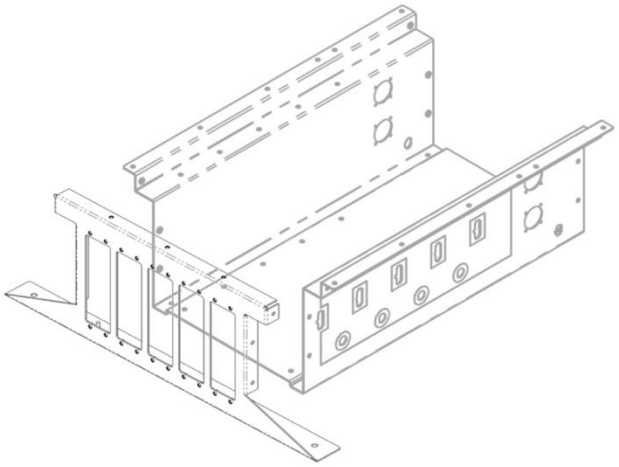


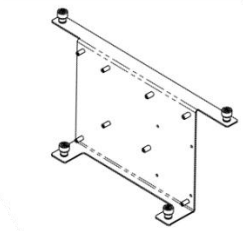
Camera Top Mounting Platform (MCH2007):
An optional mounting bracket MCH2007 for the top of the robot (on top mounting rails at front) for a camera or other sensors or equipment is available, or is included if a pan/tilt/zoom (PTZ) camera was ordered with the robot.

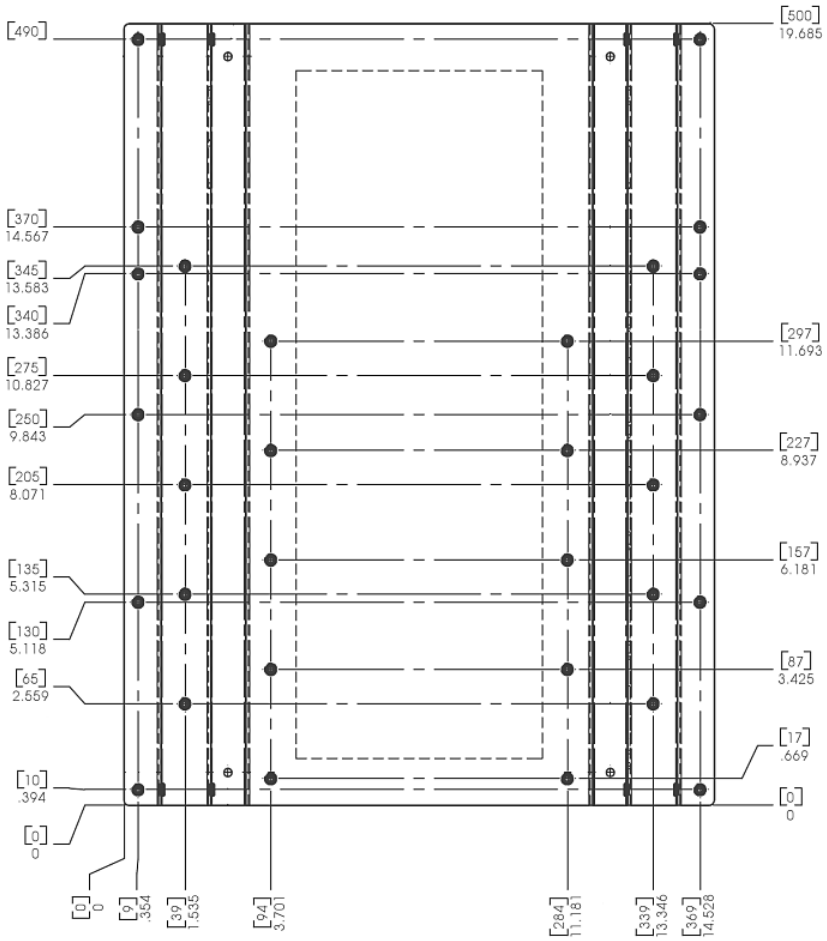


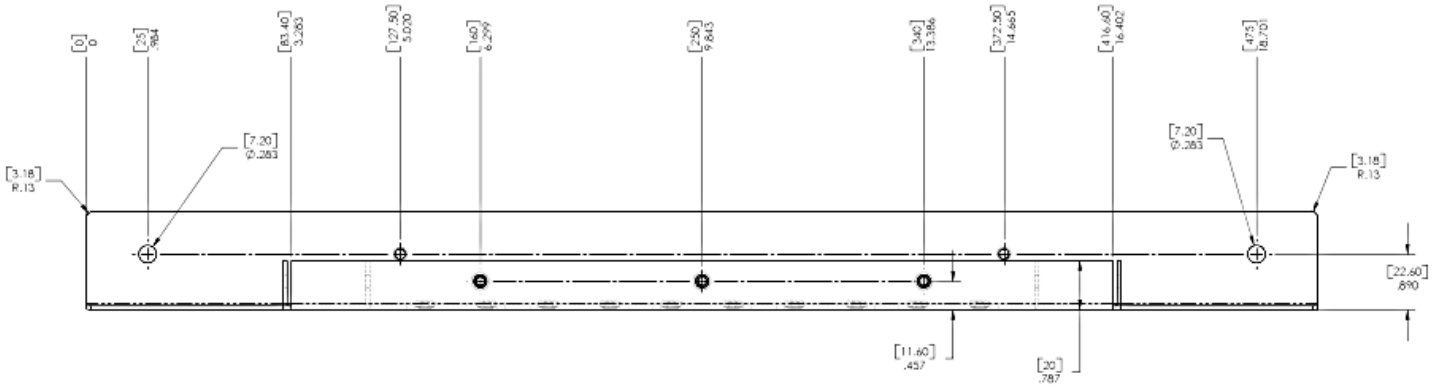
Computer Chassis (Cage) Dimensions and Mounting Locations:
Electronics and other internal equipment may be attached inside or on top of the computers/electronics chassis (cage) inside the robot. EBX onboard computers are mounted vertically on brackets inside this 27x50x15.6 cm (10.6x19.69x6.16 in.) aluminum chassis (cage) in the interior of the robot with thumbscrews and may easily be removed for maintenance. Supporting devices such as ethernet switch, power-over-ethernet injector, analog framegrabber, etc. are also attached inside this computer chassis (cage). The computers/electronics chassis is attached to the interior of the robot with four M6 screws. Use a 5mm hex wrench to remove these screws and remove the chassis.

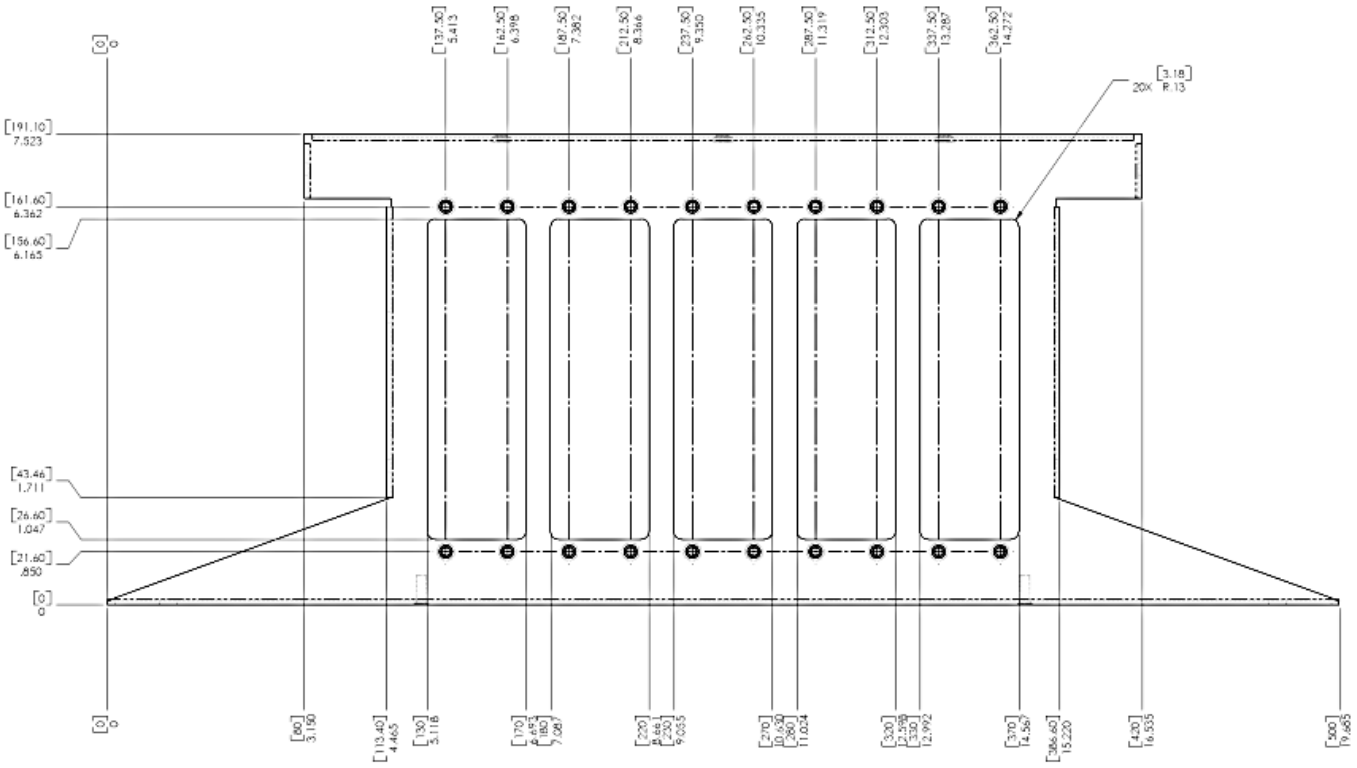
***Danger**: Use caution when working inside the robot. Make sure that all cables are secured away from belts and pulleys. Prevent all body parts, tools and loose parts from making contact with circuit boards, exposed contacts, or other electrical components. When possible, remove components from robot before servicing or modifying.*











Contact support (support@mobilerobots.com) for help and details on attaching equipment or otherwise customizing the robot.

Power Distribution Board (PDB) for Component Power Supply

Power to internal and external accessory devices and components are switched and supplied by one or more power distribution boards (PDB). Some of these power outputs are normally switched off, and must be switched on by sending the robot command number 116, which must be given with two byte arguments. The first byte argument identifies which power output to switch, and the second byte argument is the desired state (1 for on, 0 for off).

Device interface classes in ARIA which normally require power switching on Seekur and Seekur Jr. (e.g. ArTrimbleGPS, ArRVisionPTZ) will automatically send these power switching commands before attempting to connect to the device.

For example, to switch on port 6 using ARIA, use the following ArRobot method call:

```
robot.com2Bytes(116, 6, 1);
```

You can also use the **seekurPower** program from the command line, provided with the ARIA examples. For example, for port 6 (the GPS):

```
cd /usr/local/Aria/examples
./seekurPower gps on
```

Or, use the ID number:

```
./seekurPower 6 on
```

Or, using Aria's **demo** example program, enter direct command mode by pressing the **d** key, and enter the command:

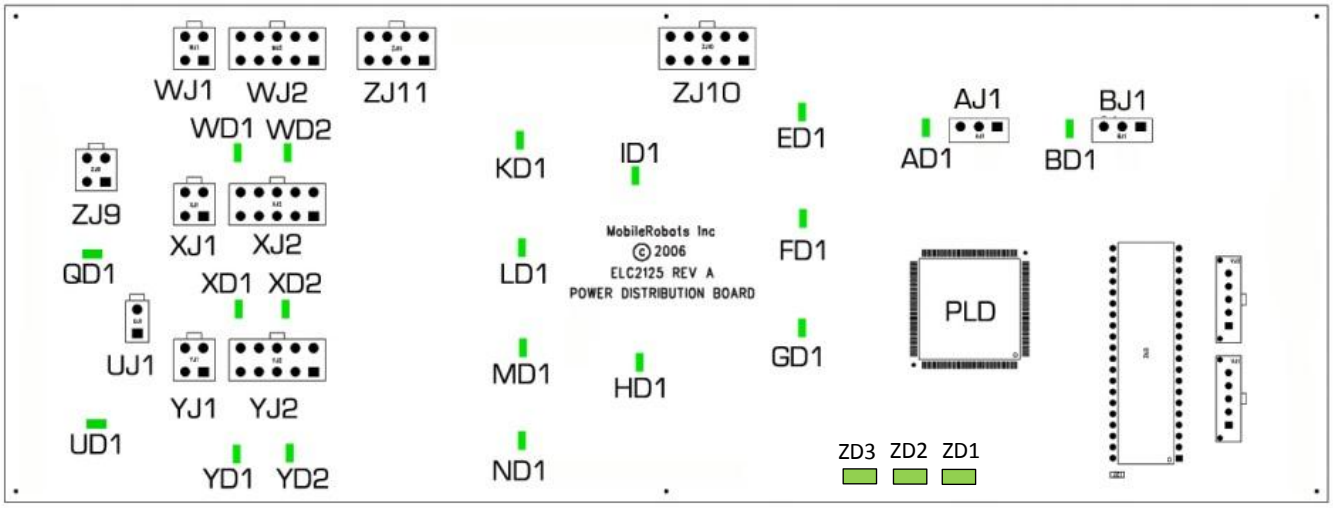
```
116 6 1
```

(Programming with ARIA and the robot command protocol is discussed in more detail in **Programming and Command Protocol** on page 41)

SeekurOS also includes some compatibility with ARCOS and uARCS by also recognizing some power switch commands used by those robot firmware variants.

The Seekur PDB switched outputs are load sensing, and will automatically shut off if the load (device) is disconnected from the PDB while the port is switched on, or it draws excessive current (beyond rated limit, given in table below). The port must be reset by turning it off before it can be switched on again. The ZD2 status LED on the PDB will flash when this has happened on any port.

Seekur Power Distribution Board (PDB) Connector Locations:



The PDB carrier board (beneath the PDB) also includes status LEDs (RAW, REG, GOOD power conversion, and 24V, 12V and 5V output indicators), visible near the left and right edges of the PDB.

PDB Connector Types:

Connector Type	Used By	Mating Connector	www.digikey.com	www.mouser.com
2-position Molex Mini-fit Jr. 39-01-2021	UJ1	Molex Mini-Fit Jr. 2-position receptacle connector 39-01-2020	WM3700-ND	538-39-01-2020
3-position Molex Mini-fit Jr. 39-01-4036	AJ1, BJ1	Molex Mini-Fit Jr. 3-position receptacle connector 39-01-4030	WM18434-ND	538-39-01-4030
4-position Molex Mini-fit Jr. (dual rows)	WJ1, XJ1, YJ1, ZJ9	Molex Mini-Fit Jr. 4-position receptacle connector (dual rows) 39-01-2040	WM3701-ND	538-39-01-2040

39-01-2041				
8-position Molex Mini-fit Jr. (dual rows) 39-01-2081	XJ11	Molex Mini-Fit Jr. 8-position receptacle connector (dual rows) 39-01-2080	WM3703-ND	538-39-01-2080
10-position Molex Mini-fit Jr. (dual rows) 39-01-2101	WJ2, X2, YJ2, ZJ10	Molex Mini-Fit Jr. 10-position receptacle connector (dual rows) 39-01-2100	WM3704-ND	538-39-01-2100

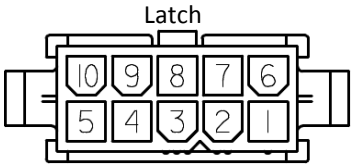
Kits are available from electronics suppliers with an assortment of Molex Mini-Fit Jr. connectors with pins (e.g. Digikey WM1047-ND). To make connectors, crimp female Molex pins to 18-24 AWG wire using Molex crimping tool and insert pins into Molex plug.

Note: Seekur contains a Secondary PDB only if more than two computers, lasers, or pan/tilt units were ordered, or if additional power output was specially requested.

In addition to the PDB outputs listed here, there are additional 24V high amperage sources available elsewhere in the robot. **Contact support** for more information.

If a port is not used (unallocated or optional device is not present), it may be used by your equipment instead.

Molex Mini-Fit Pin Numbering:



On the Seekur PDB, the first row of pins supply positive power (at different voltages, amp. limits and voltage regulation, see table below for details), and the second row are corresponding ground (return) connections. (On 3-position connectors AJ1 and BJ1, pin 1 supplies power, pin 2 is ground, pin 3 is not used)

Primary PDB (standard):

Connection	Device	Port ID	LED indicator	Power Output
AJ1	Laser Rangefinder #1 Heater On by default, but laser only uses as necessary.	11	AD1	Pin 1: +18-36V (unregulated)**, 5A Pin 2: GND Pin 3: NC
BJ1	RVision Camera -- Raw** (optional)	12	BD1	Pin 1: 18-36V (unregulated)**, 5A Pin 2: GND Pin 3: NC
UJ1	Internal (microcontroller/CAN1 power)	n/a	UD1	+5V, 2A
WJ1/WJ2	Onboard Computer #1 and disk On by default	n/a	WD1/WD2	WJ1 Pin 1: 5V WJ1 Pin 2: 5V WJ1 Pin 3: GND WJ1 Pin 4: GND WJ2 Pin 1: 5V WJ2 Pin 2: 12V, 2A WJ2 Pin 3: GND WJ2 Pin 4: GND All 5V outputs on WJ1 and WJ2 have a cumulative limit of 8A.
XJ1/XJ2	Onboard computer #2 (optional) and disk On by default	1	XD1/XD2	XJ1 Pin1: 5V XJ1 Pin 2: 5V XJ1 Pin 3: GND XJ1 Pin 4: GND

				XJ2 Pin 1: 5V XJ2 Pin 2: 12V, 2A XJ2 Pin 3: GND XJ2 Pin 4: GND All 5V outputs on XJ1 and XJ2 have a cumulative limit of 8A.
YJ1/YJ2	Unallocated	2	YD1/YD2	YJ1 Pin1: 5V YJ1 Pin 2: 5V YJ1 Pin 3: GND YJ1 Pin 4: GND YJ2 Pin 1: 5V YJ2 Pin 2: 12V, 2A YJ2 Pin 3: GND YJ2 Pin 4: GND All 5V outputs on YJ1 and YJ2 have a cumulative limit of 8A.
ZJ9 pin 1	Unallocated	7	QD1	5V, 2A
ZJ9 pin 2	Unallocated	3	QD1	5V, 2A
ZJ10 pin 1	Internal (CAN2 power)	n/a	ED1	24V, 1A
ZJ10 pin 2	Internal (CAN3 power)	n/a	FD1	24V, 1A
ZJ10 pin 3	Blower fan. Automatically switched on while charging, off after charging.	8	GD1	24V, 2A
ZJ10 pin 4	Laser Rangefinder #1	9	HD1	24V, 1A
ZJ10 pin 5	Pan/tilt unit #1 (optional)	10	ID1	24V, 1A
ZJ11 pin 1	Power-over-Ethernet 12V On by default	4	KD1	12V, 2A
ZJ11 pin 2	Ethernet LAN switch. On by default	5	LD1	12V, 2A
ZJ11 pin 3	GPS Receiver (optional)	6	MD1	12V, 2A
ZJ11 pin 4	Small arm-mounted camera (optional)	7	ND1	12V, 2A

Secondary PDB (optional):

Connection	Device	Port ID	LED indicator	Power Output
Secondary PDB, AJ1	Laser Rangefinder #2 Heater (optional) On by default, but laser only uses as necessary.	27	Secondary PDB, AD1	Pin 1: +18-36V (unregulated)**, 5A Pin 2: GND Pin 3: NC
Secondary PDB, BJ1	Unallocated	28	Secondary PDB, BD1	Pin 1: +18-36V (unregulated)**, 5A Pin 2: GND Pin 3: NC
Secondary PDB, UJ1	Unallocated Always on	n/a	Secondary PDB, UD1	5V, 2A
Secondary PDB, WJ1/WJ2	Onboard Computer #3 (optional) and disk On by default	13	Secondary PDB, WD1/WD2	WJ1 Pin 1: 5V WJ1 Pin 2: 5V WJ1 Pin 3: GND WJ1 Pin 4: GND WJ2 Pin 1: 5V WJ2 Pin 2: 12V, 2A WJ2 Pin 3: GND WJ2 Pin 4: GND All 5V outputs on WJ1 and WJ2 have a cumulative limit of 8A.

Secondary PDB, XJ1/XJ2	Onboard computer #4 (optional) and disk On by default	14	Secondary PDB, XD1/XD2	XJ1 Pin1: 5V XJ1 Pin 2: 5V XJ1 Pin 3: GND XJ1 Pin 4: GND XJ2 Pin 1: 5V XJ2 Pin 2: 12V, 2A XJ2 Pin 3: GND XJ2 Pin 4: GND All 5V outputs on XJ1 and XJ2 have a cumulative limit of 8A.
Secondary PDB, YJ1/YJ2	Onboard computer #5 (optional) and disk On by default	15	Secondary PDB, YD1/YD2	YJ1 Pin1: 5V YJ1 Pin 2: 5V YJ1 Pin 3: GND YJ1 Pin 4: GND YJ2 Pin 1: 5V YJ2 Pin 2: 12V, 2A YJ2 Pin 3: GND YJ2 Pin 4: GND All 5V outputs on YJ1 and YJ2 have a cumulative limit of 8A.
Secondary PDB ZJ10 pin 3	Laser Rangefinder #2 (optional)	24	Secondary PDB, GD1	24V, 2A
Secondary PDB, ZJ11, pin 1	Unallocated	18	Secondary PDB, KD1	12V, 2A
Secondary PDB, ZJ11, pin 2	Unallocated	19	Secondary PDB, LD1	12V, 2A
Secondary PDB, ZJ11, pin 3	Unallocated	20	Secondary PDB, MD1	12V, 2A
Secondary PDB, ZJ11, pin 4	Unallocated	21	Secondary PDB, ND1	12V, 2A

Other:

Location	Device	Port ID	LED indicator	Power Output
Fuse and Relay Board	Manipulator arm (optional) Contact support for details on using this output.	29	n/a	18-36V from battery (approx. 8A max recommended)

******NOTE*** *Raw (unregulated voltage) PDB outputs are disabled when robot is charging (since the voltage may exceed 24V)*

Fuses

Fuses are located on the three fuse/relay boards in the center of Seekur underneath the computers chassis. Fuses are ATO type (available at all electronics and auto parts stores). 30A, 20A and 3A are used. Replace with same rating fuse.

Warning*: a blown fuse may indicate a fault in wiring, charger, batteries, or other components. Contact*

MobileRobots support for help if you experience blown fuses.

Maintenance

Opening Hatches & Covers

The interior of the robot can be accessed for maintenance, troubleshooting, adding equipment, or for access to onboard computer keyboard, monitor, mouse or other ports, by removing the side, front or top access covers. The side and front access covers are attached to the robot with M4 X 12mm screws with washers, and are sealed with a sealing gasket and grounding/static dissipative gasket. Use a 3 mm hex wrench or bit to remove the screws. The top access cover is attached to the robot with M6 screws with washers. Use a 5mm hex wrench or bit to remove the screws.

Warning: *To maintain water and dust resistant integrity of Seekur, make sure the sealing gaskets are intact when replacing access covers, and replace all screws and washers. Make sure screws are properly threaded by hand before using power tools and do not over tighten to avoid stripping the threads. Use a moderate torque setting if using a power screwdriver.*

Danger*: Use caution when working inside the robot. Make sure that all cables are secured away from belts and pulleys. Prevent all body parts, tools and loose parts from making contact with circuit boards, exposed contacts, or other electrical components. Protect against static discharge.*

Recommended Maintenance

Seekur has been designed to require minimal maintenance.

Light duty recommended maintenance schedule:

Light duty is defined as occasional or periodic use in an indoor laboratory setting, and occasional use in an outdoor setting.

Once per month or after more than 6 hours of continuous use.	Fully recharge robot
Before and after outdoor use.	Inspect exterior of robot for damage and replace any missing body screws to maintain weatherproof rating. Clean any water or dirt from robot, external heat sinks, and laser rangefinder lens.
Every 4 months	If robot has not been used in past month, power on and verify proper wheel alignment and operation of robot. Recharge, verifying operation of ventilation fan.
Once per year	Recharge and verify operation of ventilation fan. Clean and verify operation of all computer CPU cooling fans. Inspect sealing gasket for damage on any access panels that are frequently removed and replace if necessary. Clean any dirt from exterior of ventilation filter with a rag or brush Check tire pressure (See Specifications, p. 59). Clean bottom of robot, ensure drain/vent holes are clear.

Normal duty recommended maintenance schedule:

Normal duty is defined as continuous or frequent use in an indoor laboratory setting, and periodic use in an outdoor setting.

Once per week or after more than 6	Fully recharge robot

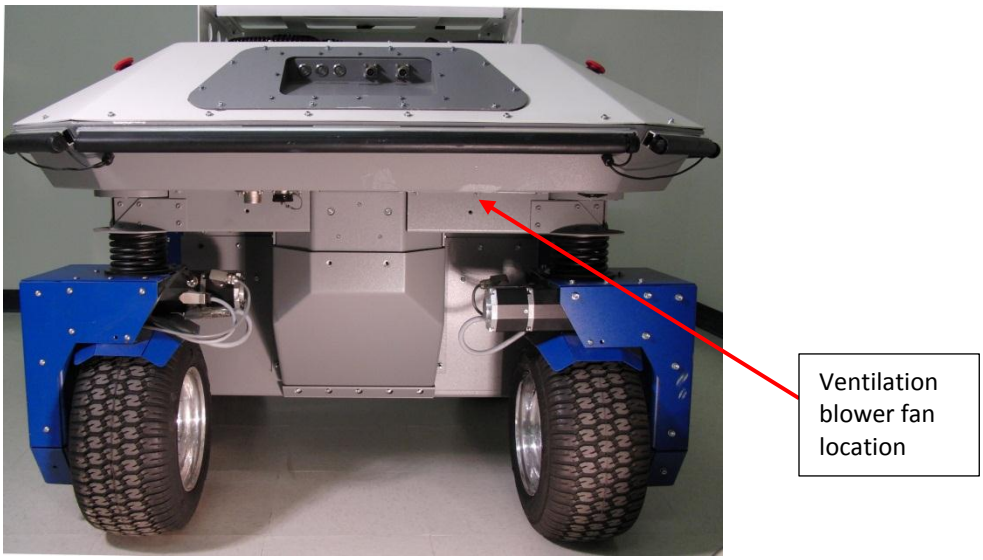
hours of continuous use.	
Once per month.	Inspect exterior of robot for damage and replace any missing body screws to maintain weatherproof rating. Clean any dirt from exterior of ventilation filter with a rag or brush. Clean any water or dirt from robot, external heat sinks, and laser rangefinder lens. Check tire pressure (See Specifications, p. 59). Recharge.
Once per year	Verify operation of ventilation fan (while recharging). Clean and verify operation of all computer CPU cooling fans . Inspect sealing gasket of any access panels that are frequently removed, for damage. Clean bottom of robot, ensure drain/vent holes are clear.

Heavy duty recommended maintenance schedule:

Heavy duty is defined as frequent use in an outdoor setting.

Once per week or after more than 6 hours of continuous use.	Fully recharge robot
Once per week	Inspect exterior of robot for damage and replace any missing body screws to maintain weatherproofing. Clean any water or dirt from robot, external heat sinks, and laser rangefinder lens. Clean any dirt from exterior of ventilation filter with a rag or brush. Check tire pressure (See Specifications, p. 59). Clean bottom of robot, ensure drain/vent holes are clear. Fully recharge robot.
Once per month	Verify operation of ventilation fan (while recharging). Clean and verify operation of all computer CPU cooling fans. Recharge.
Once per year	Inspect and replace if necessary sealing gasket of any access panels that are frequently removed. Inspect and replace worn or damaged tires. (See Specifications, p. 59.)

Location of blower fan at rear of robot:



For help with any repairs and part replacement, contact support (support@mobilerobots.com), including your robot's serial number in your message.

Battery Charging and Maintenance

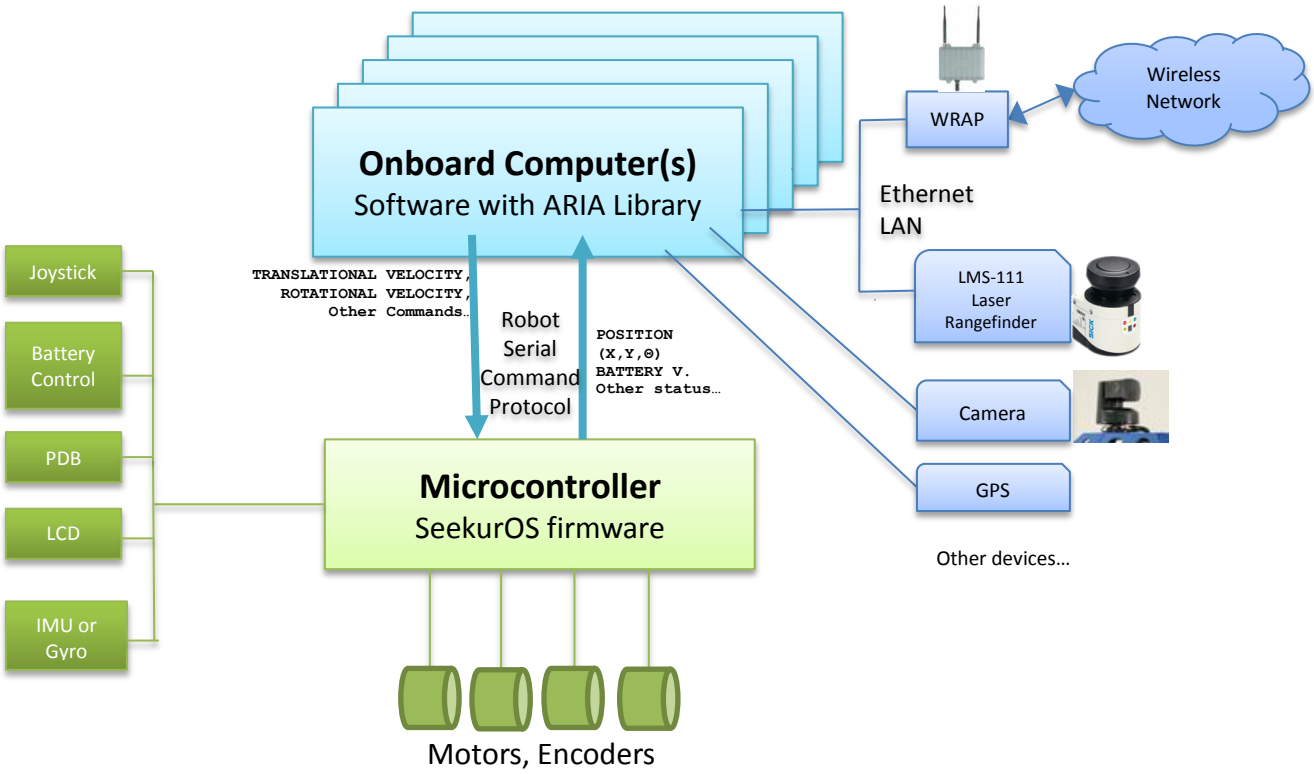
The batteries will require minimal maintenance if kept above minimum 20% charge level. If fully discharged, Seekur may not recharge or power on normally. Contact support if this is the case, or if Seekur will not retain a charge or remain powered on for more than 1-2 hours.

Wheel and Tire Replacement

To remove the wheel from the robot, remove the four bolts attaching the wheel to the hub. See Specifications, p. 59 for tire and wheel specifications.

Programming and Command Protocol

All MobileRobots platforms use a two-tier architecture. Seekur's microcontroller with embedded **SeekurOS** firmware manages all the details of the robot's mobility and internal systems including implementing velocity control of the robot platform, coordinating the motors, receiving encoder data, integrating encoder and gyro or IMU data to determine an estimate for robot position, managing power to all components, and more. SeekurOS unifies the mobile robot base into a single system with one interface channel through which software on the onboard computer can communicate and control the mobile robot base Software running on the onboard PCs communicates with SeekurOS to receive data and send commands.



ARIA

Software communicates with SeekurOS via a simple packet-based protocol (described below) via an RS-232 serial connection between the robot and an onboard computer.

To support development of software, MobileRobots provides a C++ development library called **ARIA** which implements this protocol, provides interfaces to many accessory devices, and also includes many useful tools for robotics and cross-platform programming. It is also possible to use ARIA in Python and Java via wrapper libraries. ARIA installation packages can be found on the CD included with your robot, preinstalled on the onboard computer and the latest version as well as all future updates can be downloaded at http://robots.mobilerobots.com/wiki/ARIA. ARIA can be used on Linux with the standard GNU C++ compiler and linker (g++), or on Windows with Microsoft Visual C++ (either Visual Studio 2003, 2008 or 2010). ARIA is provided as open-source software, under the terms of the GNU General Public License (GPL). Full reference documentation is included.

In addition to the ARIA library, MobileRobots provides additional development libraries, including the **ArNetworking** framework for network programming over TCP and UDP, and the **ARNL** and **MOGS** intelligent navigation libraries. Useful tools for robot development also include the **MobileSim** simulator, the **MobileEyes** user interface application, and the **Mapper3** map editing tool. This software can be found on the CD provided with the robot, and at http://robots.mobilerobots.com/wiki/Software.

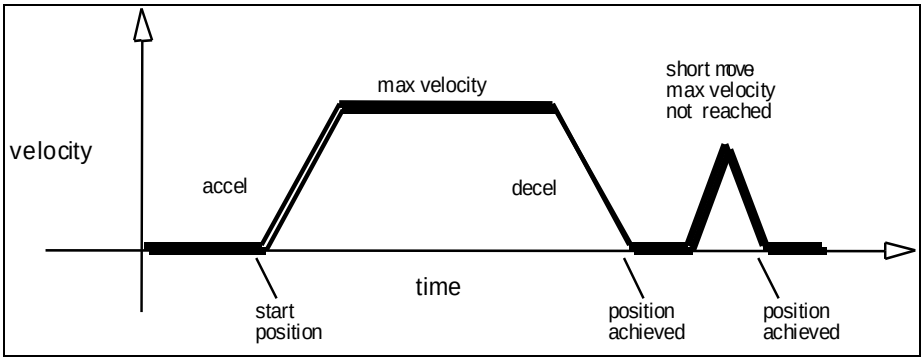
ARIA automatically handles all communication with the robot and many accessory devices, including but not limited to laser rangefinders, the robot's built in sonar and bumper sensors, pan/tilt cameras and pan/tilt units, GPS receivers, and more. To communicate with the robot, ARIA sends and receives messages with the robot's embedded firmware. The rest of this chapter describes this protocol in detail. When using ARIA or other development software, however, you do not need to implement this protocol directly (though some of the information included below regarding the behavior of SeekurOS, as well as how to configure its parameters, will be useful to the ARIA programmer.)

Robots in Motion

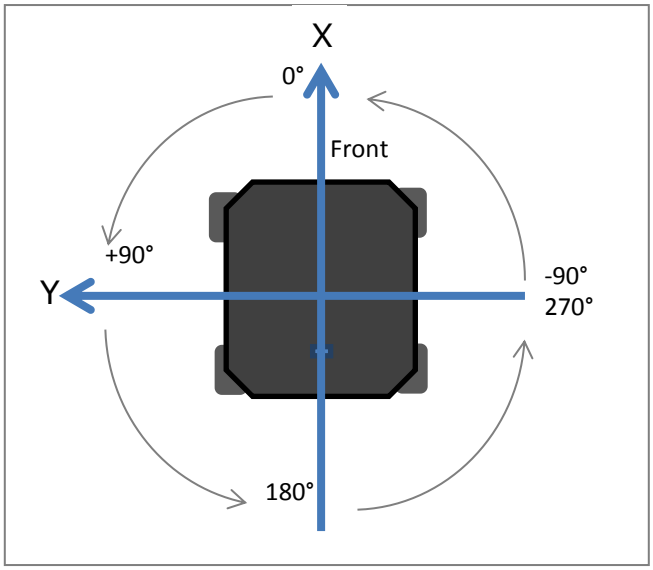
When Seekur receives a motion command, it accelerates or decelerates the robot according to acceleration or deceleration parameters previously set by SETA (command #5, for translation) and SETRA (command #23, for rotation) until the platform either achieves the requested speed (for VEL, LATVEL and RVEL commands) or nears the requested movement distance (when performing MOVE, HEAD and DHEAD commands). Rotation headings and translation setpoints are achieved by a trapezoidal velocity function, which SeekurOS recomputes each time it receives a new motion command.

SeekurOS automatically limits velocities, acceleration and deceleration to client-modifiable maximum limits (set via SETV, SETRV, SETA and SETRA) and ultimately by internal constants. Initial default values for these limits may be set in SeekurOS configuration parameters using seekurParamManager.

Limits set via SETV, SERV, SETA commands and SETRA take effect on subsequent commands, not on the current translation or rotation activity, and are reset when the client disconnects or SeekurOS is reset.



The orientation commands Head (#12), DHEAD (#13) and DCHEAD (#22) turn the robot with respect to its internal dead-reckoned angle to an absolute heading (0-359 degrees), relative to its immediate heading, or relative to its current heading setpoint (achieved or last commanded heading), respectively.



The STOP command is equivalent to requesting both translation and rotation velocities of 0; the robot will decelerate to 0. The E-STOP command #55 overrides normal deceleration and abruptly stops the robot in the shortest distance and time possible. Accordingly, the robot brakes to zero translational and rotational velocities with very high deceleration and remains stopped until it receives a subsequent translation or rotation velocity command from the client.

Position Integration

MobileRobots platforms track their position and orientation based on wheel motion sensed by encoder readings and from the integrated gyroscope or IMU accessory (if present).

On start-up, the robot position is initialized to (0mm, 0mm, 0°), pointing along the positive X-axis at 0 degrees. As the robot moves, the position is updated with reference to this initial coordinate frame,

and the latest calculated position estimate is reported in the standard SIP data packet (see below) as XPos, YPos and Theta. X and Y coordinates are provided in millimeters. Angles vary between -179 to 180 degrees. ARIA uses these position estimates to update its own stored position, which may optionally have transformations automatically applied to place the robot in any coordinate system, or to make corrections.

Be aware that registration between external and internal coordinates deteriorates rapidly with movement due to gearbox play, movement in robot suspension, wheel imbalance, wheel slip, accumulated small errors in encoder sensing, and many other real-world factors. You can rely on the dead-reckoning ability of the robot for a short range—on the order of several meters and one or two revolutions, depending on the surface. (ARNL and MOGS address these problems by using additional sensing and sophisticated positioning/localization algorithms to correct the position of the robot.)

You may reset the internal coordinates back to 0,0,0 with the SETO command #7 (see below).

Communication Packet Protocol

All MobileRobots mobile robot platforms implement the same Pioneer protocol and are compatible. All versions of SeekurOS firmware are backwards-compatible with previous versions.

The protocol is a bidirectional byte stream, in which sequences of byte **packets** represent individual **commands** (when sent from client software to SeekurOS), or Server Information Packets, or **SIPs** or simply **packets,** (when sent from SeekurOS back the client software). Packets consist of five main elements: a two-byte header, a one-byte count of the number of subsequent packet bytes, a one-byte command or packet type identifier followed by packet data, and finally a two-byte checksum.

General Packet Format:

Component	Bytes	Value	Description
header	2	0xFA, 0xFB	Packet header; same for client and server
byte count	1		Number of command/argument bytes plus Checksum's two bytes, but not including Byte Count itself or the header bytes. Maximum of 249.
packet type or command identifier	1	0 - 255	Packet type or command identifier (see below for lists).
packet data	n		Packet or command data (if any)
checksum	2		Packet integrity checksum

Packet data is divided into one or more value fields. The meaning and sequence of fields are specific to each packet or command type. Each field has a data type which determines the size of that field (in bytes). Integer data values contained in a packet may be signed and unsigned, and are 16-bit (2-byte) integers, least-significant byte first. Single-byte values are one 8-bit byte. Strings may be either length-prefixed or NULL terminated (depending on packet type).

Packet Checksum

Calculate the client-server packet Checksum by successively adding data byte pairs (most-significant byte first) to a running checksum (initially zero), disregarding sign and overflow. If there are an odd number of data bytes, the last byte is XORed to the low-order byte of the checksum.

NOTE: The checksum integer is at the end of the packet, with its bytes in the reverse order of that used for data; that is, b_0 is the high byte and b_1 is the low byte.

```
// packetBuf is a character buffer char[] containing one data packet
// read from the robot.
int i;
unsigned char n;
```

```
int c = 0;
i = 3;
n = packetBuf[2] - 2;  // data length
while (n > 1) {
    c += ((unsigned char)packetBuf[i]<<8) | (unsigned char)packetBuf[i+1];
    c = c & 0xffff;
    n -= 2;
    i += 2;
}
if (n > 0)
    c = c ^ (int)((unsigned char) packetBuf[i]);
```

ARIA automatically verifies packet checksums when receiving and parsing robot packets from SeekurOS, and automatically provides correct checksums with client commands sent to SeekurOS.

Server Information Packets

Once a client establishes a connection and sends the OPEN command, SeekurOS automatically and periodically sends a set of Server Information Packets (SIPs) over the connection back to the connected software. The SIP period is 100ms (10Hz. One type of packet, the Standard SIP (also referred to just as "SIP", or as "Motors Packet"), is always sent. In addition to the Standard SIP, other types of information packets may be requested via client commands to include accessory or additional data. These packets are sent immediately before or after the Standard SIP. The Standard SIP has priority, it will be sent first on each cycle.

ARIA contains classes which serve as interfaces to the data received by these packets. See the ARIA reference documentation and examples for details.

Standard SIP contents

"byte" indicates a single byte value (usually unsigned). "int" indicates a signed 16-bit (2-byte) integer value (LSB). "uint" indicates an unsigned 16-bit (2-byte) integer value (LSB).

Some items in the SIP are provided by other MobileRobots platforms but are not relevant or implemented in Seekur (these are shown in grey italic in the table below).

The standard SIP is automatically received and the data stored by ARIA's **`ArRobot`** class.

Field	Size	Description
Packet Header	2 bytes	Exactly 0xFA (250), 0xFB (251)
Packet Byte Count	byte	Number of packet bytes that follow, which includes 2 bytes for checksum (does not include preceding header or this byte count)
Packet Type	byte	0x32 when motors stopped or 0x33 when robot moving.
XPOS	int	Estimated robot position coordinates (from 0,0 at robot startup) in millimeters (DistConvFactor[1] = 1.0).
YPOS	int	
THPOS	int	Orientation (theta) in angular units of 2π/4096 radians (AngleConvFactor[1] = 0.001534).
L VEL	int	Average velocity of left side of robot in millimeters per second (used by ARIA to determine translational and rotational velocity components)

[1] *ARIA stores conversion factors in its robot parameter files for different types of robots in robot parameter files and automatically applies the conversion when SIP is received*

<table>
<tr><td>R VEL</td><td>int</td><td>Average velocity of right side of robot in millimeters per second (used by ARIA to determine translational and rotational velocity components)</td></tr>
<tr><td>BATTERY</td><td>byte</td><td>Battery charge in decivolts (tenths of volts) (e.g. 101 = 10.1 V) (See also BATTERYX10 and STATEOFCHARGE fields below)</td></tr>
<tr><td>STALL</td><td>uint</td><td>Motor stall indicators:<table><tr><th>Bit</th><th>Condition if set (1)</th></tr><tr><td>0</td><td>Set if any bumper hit (“left” or “right” stall in ARIA)</td></tr><tr><td>1-5</td><td>Bit set if front or side bumpers hit</td></tr><tr><td>6-7</td><td>Unused on Seekur</td></tr><tr><td>8</td><td>Set if any bumper hit (“left” or “right” stall in ARIA)</td></tr><tr><td>9-11</td><td>Bit set if rear bumpers hit</td></tr><tr><td>12-15</td><td>Unused on Seekur</td></tr></table></td></tr>
<tr><td>CONTROL</td><td>int</td><td>unused</td></tr>
<tr><td>FLAGS</td><td>uint</td><td>General status flags:<table><tr><th>Bit</th><th>Condition if set (1)</th></tr><tr><td>0</td><td>Motors enabled</td></tr><tr><td>1</td><td>Pioneer sonar array #1 enabled (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>2</td><td>Pioneer sonar array #2 enabled (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>3</td><td>Pioneer sonar array #3 enabled (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>4</td><td>Pioneer sonar array #4 enabled (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>5</td><td>Pioneer stop button pressed (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>6</td><td>Pioneer E-stall engaged (N/A on Seekur, Seekur Jr.)</td></tr><tr><td>7</td><td>(unused on Seekur)</td></tr><tr><td>8</td><td>(unused on Seekur)</td></tr><tr><td>9</td><td>(unused on Seekur)</td></tr><tr><td>10</td><td>(unused on Seekur)</td></tr><tr><td>11-15</td><td>Reserved</td></tr></table></td></tr>
</table>

COMP	*byte*	*unused*
Pioneer Sonar Readings Count	*byte*	*Unused on Seekur or Seekur Jr. (always 0).*
GRIP_STATE	*byte*	*Pioneer gripper state (unused in PatrolBot, Seekur, Seekur Jr, always 0.)*
ANPORT	*byte*	*Selected Pioneer analog I/O Port (see ADSEL command) (unused in PatrolBot, Seekur, Seekur Jr, always 0.)*
ANALOG	*byte*	*Pioneer analog input value (0-255=0-5 VDC) (unused in PatrolBot, Seekur, Seekur Jr., always 0)*
DIGIN	*byte*	*Pioneer digital input (unused in PatrolBot, Seekur, Seekur Jr., always 0)*
DIGOUT	*byte*	*Pioneer digital output (unused in PatrolBot, Seekur, Seekur Jr., always 0)*
BATTERYX10	int	Actual battery voltage in units of 0.1 V
RECHARGESTAGE	*byte*	*Unused on Seekur, always 0*
ROTVEL	int	Current rotational velocity in units of degrees X 10 per sec.
FAULTFLAGS	*uint*	*Unused on Seekur, always 0*
LATVEL	int	Current Seekur lateral (sideways) velocity (mm/s). (Seekur only)
TEMPERATURE	byte	Reading from internal temperature sensor, in deg. C, or -127 if unavailable.
STATEOFCHARGE	byte	Amount of remaining battery charge, 0-100% (unused in Pioneer, PatrolBot)
CHECKSUM	2 bytes	checksum

Client Commands

Client commands are packets sent from client software to SeekurOS. The packet ID indicates the command, and depending on the command, argument data may follow.

ARIA's **`ArRobot`** and other classes which serve as interfaces to the robot which will send commands and requests. ARIA's **`ArRobot`** class also contains functions to easily send any command with arguments.

Not all commands listed below are implemented by Seekur. These are indicated in grey italic.

A command's argument may be a two-byte (16-bit) integer, ordered as least-significant byte first (little endian). The integer absolute value is provided in the packet; whether it should be interpreted as a positive or negative number is indicated by the argument type byte (0x3B or 0x1B). The argument may also be a string or buffer of up to a maximum of 200 bytes, prefaced by a length byte and may or may not be NULL terminated. Some commands have other command-specific argument formats as well. How exactly a command's argument should be interpreted is documented with commands is specified in the list of commands below.

Command Packet Format

Component	Bytes	Value	Description
header	2	0xFA, 0xFB	Packet header; same for client and server
byte count	1		Number of command/argument bytes plus Checksum's two bytes, but not including Byte Count itself or the header bytes. Maximum of 249.
packet type or command identifier	1	0 - 255	Packet type or command identifier (see below for lists).
data type	1	0x3B (positive integer), 0x1B (negative or unsigned integer), 0x2B (string)	0x3B indicates that a 16-bit (2-byte) integer follows. 0x1B indicates that a 16-bit (2-byte) integer follows, whose value should be negated 0x2B indicates that a length-prefixed string follows. The length prefix is one byte.
data	n		Argument data
checksum	2		Packet integrity checksum

Connection Initialization and Maintenance Commands:

"byte" indicates a single byte value (usually unsigned). "int" indicates a signed 16-bit (2-byte) integer value (LSB). "uint" indicates an unsigned 16-bit (2-byte) integer value (LSB).

During Connection Initialization Handshake:

Command	#	Args	Description
SYNC0	0	none	Start connection. Send in sequence. ARCOS echoes
SYNC1	1	none	synchronization commands back to client, and
SYNC2	2	none	robot-specific auto-synchronization after SYNC2.

After Established Connection:

PULSE	0	none	Reset server watchdog timeout. Send this command periodically if not sending any other commands to maintain connection and prevent watchdog from triggering. (See below.)
OPEN	1	none	Start sending data.
CLOSE	2	none	Stop sending data and close client connection.
RESET	253	none	Force reset of the microcontroller.
HOSTBAUD	50	int	Request firmware to switch its connection baud rate. (You must also change rate on sending computer serial port.) 0=9600, 1=19200, 2=38400, 3=57600, 4=115200 baud.

Robot Motor, Velocity and Position Control:
See **Robots in Motion** discussion below on page 42. To stop the robot, send VEL, RVEL and LATVEL commands with 0 argument values.

ENABLE	4	int	1=enable; 0=disable the motors.
VEL	11	int	Set velocity for forward/backward translation (mm/sec). May be combined with RVEL (and LATVEL on Seekur) for simultaneous motion. Maximum speed is limited with SETV and stored firmware parameter.
RVEL	21	int	Set rotation velocity (degrees/sec). Positive argument values rotate counter-clockwise, negative value rotate clockwise. May be combined with VEL (and LATVEL on Seekur) for simultaneous motion. Maximum speed is limited with SETRV and stored firmware parameter. (This command replaces ROTATE; commands are equivalent.)
ROTATE	9	int	See RVEL.
LATVEL		int	(Seekur only) Set velocity for sideways translation (mm/sec). Positive argument values translate to the right, negative values translate to the left. May be combined with VEL and RVEL for simultaneous motion. Maximum speed is limited with SETV and stored firmware parameter. Only available on Seekur.
HEAD	12	int	Turn to given absolute heading. (+ = counterclockwise). Speed used is given by SETRV.
DHEAD	13	int	Turn to given offset from current heading (degrees) (+) counter- or (–) clockwise. Speed is given by SETRV.

Configure Acceleration and Deceleration Parameters
See **Robots in Motion** discussion on page 42.

SETA	5	int	If positive, set acceleration used for VEL, VEL2, MOVE commands, in mm/sec^2 If negative, set rotation deceleration used for VEL, VEL2, MOVE commands, in mm/sec^2
SETRA	23	int	If positive, set rotation acceleration used for RVEL, ROTATE, HEAD, DHEAD and DCHEAD commands, in degrees/sec^2 If negative, set rotation deceleration used for RVEL, ROTATE, HEAD, DHEAD and DCHEAD commands, in degrees/sec^2
LATACCEL	113	Int	If positive, set acceleration used for LATVEL commands in mm/sec^2. If negative, set deceleration used for LATVEL commands in mm/sec^2.

Configure Maximum Speeds
See **Robots in Motion** discussion on page 42.

SETV			Sets maximum translation velocity, which is also used by MOVE command. Degrees/sec.
SETRV	10	int	Sets maximum turn rotation velocity, which is also used by HEAD, DHEAD commands; degrees/sec.

Reset Position

SETO	7	none	Reset position estimate given in standard SIP to 0,0,0..

Joystick

JOYDRIVE	47	int	1=allow joystick drive from port while connected with a client; 0 (default) disallows.
JOYREQUEST	17	int	Request one or continuous stream (>1) or stop (0) joystick packets

Additional SIP Requests

CONFIG	18	none	Request one robot configuration SIP. See CONFIGpac description below.
ENCODER	*19*	*int*	*Request one, a continuous stream (>1), or stop (0) encoder SIPs. See ENCODERpac description below. (Pioneer, PeopleBot, PowerBot, AmigoBot and PatrolBot only;* ***not available on Seekur or Seekur Jr****.)*
IMUREQUEST	26	Int	Use argument value 1 to request one IMU packet. Use argument value 2 to request continuous IMU packets every cycle. Send argument value 0 to stop. See **Inertial Measurement Unit (IMU)** on page 53 for details.

Power Control

See **Power Distribution Board (PDB) for Component Power Supply** on page 32.

SEEKURPDB	116	2 byte	Turn a Seekur PDB power port on or off. First byte indicates which port, second byte indicates state (1=on, 0=off). See Power Supply for locations of power output connectors and typical devices attached to which ports.. Seekur and Seekur Jr. only.

Other

SKRMAINT	117	none	Engage Seekur microcontroller maintenance mode (factory use only).
BATTEST	250	int	Artificially set the battery voltage; argument in tens volts (100=10V); 0 to revert to real voltage
DIGTEMPTEST	251	Int	Artificially set the digital temperature; argument in degrees C; 0 to revert to real temperature.
ANTEMPTEST	252	Int	Artificially set the analog temperature; argument in degrees C; 0 to revert to real temperature.
SYSKILL_RECENTER	119	None	Recenter the wheels, then turns off the robot. (Seekur only)
RECENTERWHEELS	120	None	Recenter the wheels. (Seekur only)

Command Packet Errors

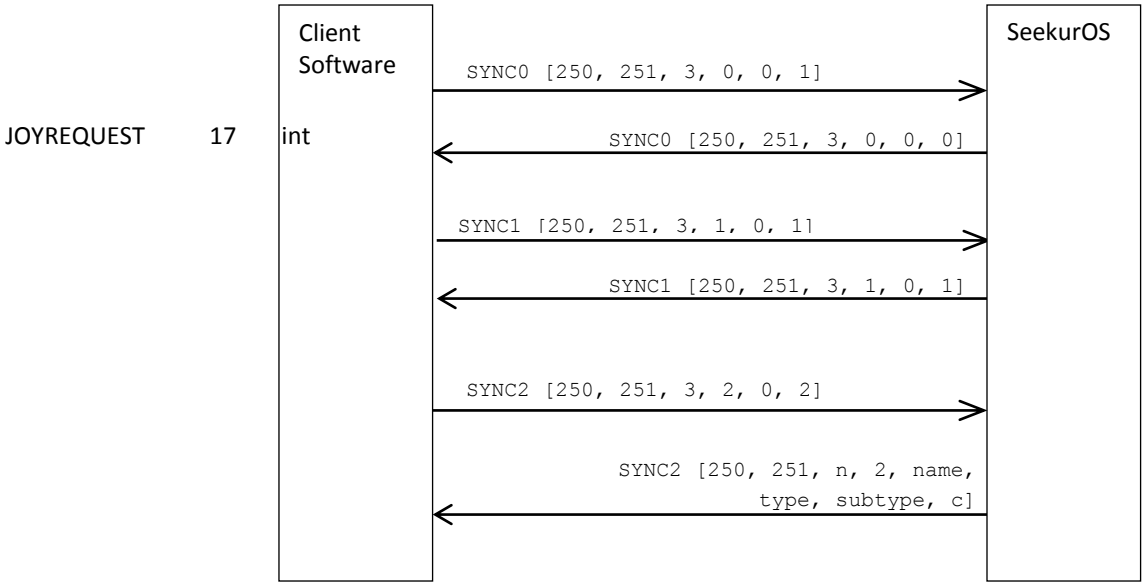
SeekurOS ignores a client command packet whose Byte Count exceeds 204 (total packet size of 207 bytes) or has an erroneous Checksum. The client should similarly ignore erroneous SIPs.

Because of the real-time nature of client-server mobile-robotics interactions, we made a conscious decision to provide an unacknowledged communication packet interface. Retransmitting server information packets typically serves no useful purpose because old data is useless in maintaining responsive robot behaviors.

Nonetheless, the client-server interface provides a simple means for dealing with ignored command packets: Most of the client commands alter state variables in the server. By examining those values in respective SIPs, client software may detect ignored commands and re-issue them until achieving the correct state. Or, it may simply continually request its desired state on each SIP cycle (ARIA does this in its "state reflection" loop. See ARIA reference manual.).

Establishing the connection

When first started or reset, SeekurOS is in a special wait state, listening for communication packets to establish a client-server connection. To establish a connection, the client application must open the serial connection at 9600 baud (no stop bits, no parity bit), then send a series of three synchronization packets containing the SYNC0, SYNC1 and SYNC2, waiting for response from SeekurOS after each packet, before sending the next. SeekurOS echoes each SYNC packet verbatim back to the client. The client should listen for the returned packets and only issue the next synchronization packet after it has received the appropriate echo response.



The response from SeekurOS to the last synchronization packet, SYNC2, also includes robot identifying information as three NULL-terminated character strings: robot name, robot class or type, and robot subclass or subtype. You may set the name to any string using **`SeekurParamManager`** (see next section). For the Seekur, the class is always "Pioneer" and the subclass is always "seekur". Clients use these identifying strings to configure their own operating parameters. ARIA, for instance, uses the Subclass to identify the robot subtype and load the matching parameter file found in the Aria/params directory (for subclass "seekur" ARIA loads "seekur.p"), followed by a custom robot parameter file corresponding to the name, if one exists.

Opening the session—OPEN

Once you've established a connection with SeekurOS, the client should send the Open command #1 (250, 251, 3, 1, 0, 1, which causes SeekurOS to perform a few housekeeping functions, start its various internal processes, such as the motor controllers, and begin transmitting information to the client.

When first connected, the motion commands are disabled. To enable the motors after starting a connection, send the ENABLE command #4 with an integer argument of one.

Keeping the Beat—PULSE

A SeekurOS safety watchdog expects that, once connected, it receives at least one command packet from the client program every WatchDog seconds, as defined in the robot's FLASH configuration. (default is 2 seconds). Otherwise, it assumes the client-server connection is broken and stops the robot.

Some clients—ARIA-based ones, for instance—use the good practice of sending a PULSE client command #0 (250, 251, 3, 0, 0, 0) just after OPEN. And if your client application will be otherwise distracted for some time, have it periodically issue the PULSE command to let your robot server know that your client is indeed alive and well. It has no other effect.

If the robot shuts down due to lack of communication with the client, it will revive upon receipt of a client command and automatically accelerate to the last-specified speed and heading.

Closing the connection—CLOSE

To close the client-server connection, which automatically disables the motors and other server functions, simply issue the CLOSE command #2. Most of the controller's operating parameters return to their configured default values upon disconnection with the client.

ARIA automatically uses OPEN and CLOSE at start and end of session, and will send a PULSE command on every SIP cycle if there are no other pending commands to send.

CONFIG Packet and CONFIG command

Send the CONFIG command #18 without an argument to have SeekurOS send back a CONFIG packet type 32 (0x20) containing the robot's current operational parameters. Use the CONFIG packet to examine many of your robot's default settings and their working values, where appropriate, as changed by other client commands, such as SetV and RotKV.

The information in this packet is automatically used by ARIA internally, and is also made available through interface classes, such as **`ArRobot`** and **`ArRobotConfigPacketReader`**.

CONFIG packet contents:

Label	Data	Description
Packet Header	int	Common packet header = 0xFAFB
Packet Byte count	byte	Number of following data bytes
Packet type	byte	CONFIG packet type identifier = 0x20
Robot type	str	"Pioneer"
Subtype	str	"seekur"
Sernum	str	Serial number for the robot.
4mots	*byte*	*unused*
Rotveltop	int	Maximum rotation velocity; deg/sec
Transveltop	int	Maximum translation speed; mm/sec
Rotacctop	int	Maximum rotation (de)acceleration; deg/sec^2
Transacctop	int	Maximum translation (de)acceleration; mm/sec^2
PWMmax	int	Maximum motor PWM (limit is 500).
Name	str	Unique name given to your robot (can be changed using seekurParamManager).
SIPcycle	byte	Server information packet cycle time; ms.
Hostbaud	byte	Baud rate for client-server HOST serial: 0=9.6k, 1=19.2k, 2=38.4k, 3=56.8k, 4=115.2k.
Aux1baud	*byte*	*Baud rate for AUX1 serial port; see HostBaud.* *Unused on Seekur, Seekur Jr.*

Gripper	*int*	*0 if no Pioneer Gripper; else 1.* *Always 0 on PatrolBot, Seekur, and Seekur Jr.*
Front Sonar	*int*	*1 if robot has front sonar array enabled, else 0.* *Always 0 on 3rd gen. PatrolBot, Seekur, and Seekur Jr.*
Rear Sonar	*byte*	*1 if robot has rear sonar enabled, else 0.* *Always 0 on Seekur and Seekur Jr.*
Lowbattery	int	In 1/10 volts; alarm activated when battery charge falls below this threshold value.
Revcount	*int*	*Pioneer wheel encoder calibration value.* *Unused on Seekur, Seekur Jr.*
Watchdog	int	Time (ms) before robot automatically stops if it has not received a command from the client. Restarts on restoration of connection.
P2mpacs	*byte*	*Unused.*
Stallval	*int*	*Maximum PWM before Pioneer stall detection. If > PWMMAX, never.* *Unused on Seekur, Seekur Jr.*
Stallcount	*int*	*Time (ms) after a stall for Pioneer stall recovery. Motors lax during this time.* *Unused on Seekur, Seekur Jr.*
Joyvel	int	Joystick translation velocity setting, mm/sec
Joyrvel	int	Joystick rotation velocity setting in deg/sec
Rotvelmax	int	Current max rotation speed; deg/sec.
Transvelmax	int	Current max translation speed; mm/sec.
Rotacc	int	Current rotation acceleration; deg/ sec^2
Rotdecel	int	Current rotation deceleration; deg/ sec^2
Rotkp	int	Current Proportional PID for rotation
Rotkv	int	Current Derivative PID for rotation
Rotki	int	Current Integral PID for rotation
Transacc	int	Current translation acceleration; mm/ sec^2
Transdecel	int	Current translation deceleration; mm/ sec^2
Transkp	int	Current Proportional PID for translation.
Transkv	int	Current Derivative PID for translation.
Transki	int	Current Integral PID for translation.
Frontbumps	byte	Number of front bumper segments.
Rearbumps	byte	Number of rear bumper segments.
Charger	*byte*	*Autocharger/dock type. 0=none; 1 = Pioneer/PeopleBot; 3 = PatrolBot; 4 = PowerBot (Always 0 on Seekur, Seekur Jr.)*

SonarCycle	*byte*	*Sonar duty cycle time in milliseconds.* *Always 0 on Seekur, Seekur Jr.*
Autobaud	byte	1 if the client can change baud rates; 2 if auto-baud implemented.
HasGyro	byte	Indicates type of gyro or IMU in this robot. 1, 2 or 3 = single axis gyro. 4 = optional IMU. 0 = no gyro or IMU. Default is 1.
Driftfactor	*int*	*Pioneer DriftFactor parameter.* *Unused on Seekur, Seekur Jr.*
Aux2baud	*byte*	*Baud rate for Pioneer AUX2.* *Unused on Seekur, Seekur Jr.*
Aux3baud	*byte*	*Baud rate for Pioneer AUX3.* *Unused on Seekur, Seekur Jr.*
Ticksmm	*int*	*Pioneer TicksMM.* *Unused on Seekur, Seekur Jr.*
Shutdownvolts	int	DC volts X10 at or below which the onboard PC will shut down
Firmware Version	str	Null-terminated string containing firmware version identifier.
GyroCW	int	Gyro calibration factor clockwise
GyroCCW	int	Gyro calibration factor counterclockwise
KinematicsDelay	byte	Time delay (ms) between acquisition and reporting of rotation
LatVelTop	Int	Absolute maximum lateral velocity allowed (mm/sec) (Seekur Only.)
LatAccTop	Int	Absolute maximum lateral acceleration allowed (mm/sec^2) (Seekur Only.)
LatVelMax	Int	Currently set lateral velocity maximum (mm/sec) (Seekur Only)
LatAccel	Int	Currently set lateral acceleration (mm/sec^2) (Seekur Only)
LatDec	Int	Currently set lateral deceleration (mm/sec^2) (Seekur Only)
PChargeThresh	*Int*	*Unused on Seekur*
PowerCommands	Int	Power command availability for PatrolBot compatibility
BatteryType	byte	Battery type. 1 = Lead, 2=NiMH.
LowSOC	Int	Threshold state of charge value at which to warn (default is 10%)
ShutdownSOC	Int	Threshold state of charge value at which to shut down the robot (default is 5%)
Packet checksum	2bytes	Packet checksum

Inertial Measurement Unit (IMU)

An optional inertial measurement unit (IMU) may be ordered with the robot which provides additional tilt information. The IMU provides SeekurOS with short term averages, which SeekurOS stores until the next packet cycle, when it sends them in the IMU packet, if IMU data has been requested by software. SeekurOS also uses the Z axis to computer robot heading estimate (Theta) as provided in the SIP. To request IMU packets, send the IMUREQUEST command #26. Send with argument 1 to return one response, send with an integer argument 2 to start sending a response packet every cycle, or send with argument 0 to stop.

The IMU packet is generalized for several types of IMU and gyroscopic/accelerometer, so the packet includes field which indicates whether different types of data are available and for which axes.

IMU packet contents:

<table>
<tr><th>Field</th><th>Size</th><th>Value</th><th>Description</th></tr>
<tr><td>Packet header</td><td>2</td><td>0xFA, 0xFB</td><td>Packet header</td></tr>
<tr><td>Byte count</td><td>1</td><td></td><td>Number of following bytes</td></tr>
<tr><td>Packet type</td><td>1</td><td>0x9A</td><td>IMU packet type identifier</td></tr>
<tr><td>TimeSinceSIP</td><td>1</td><td></td><td>Time in ms. Between last SIP sent and sending this packet</td></tr>
<tr><td>NumReadings</td><td>1</td><td></td><td>Number of readings stored since last packet sent</td></tr>
<tr><td>NumAxes</td><td>1</td><td></td><td>Number of axes of speed data available on this IMU (normally 3)</td></tr>
<tr><td>Data</td><td></td><td></td><td>For each recent reading (NumReadings):
<table>
<tr><th>Field</th><th>Size</th><th>Description</th></tr>
<tr><td>Range</td><td>1</td><td>Configured range mode (mode 2, 160/150 deg/s, by default).
1=80/75 deg/s
2=160/150 deg/s
3=320/300 deg/s</td></tr>
</table>
If NumAxes is 3:
<table>
<tr><td>AvgX</td><td>2</td><td>Recent average on X axis</td></tr>
<tr><td>AvgY</td><td>2</td><td>Recent average on Y axis</td></tr>
<tr><td>AvgZ</td><td>2</td><td>Recent average on Z axis</td></tr>
</table>
If NumAxes is 1:
<table>
<tr><td>AvgZ</td><td>2</td><td>Recent average on Z axis</td></tr>
</table></td></tr>
<tr><td>NumAccelAxes</td><td>1</td><td></td><td>Number of axes of acceleration data available on this IMU.</td></tr>
<tr><td>AccelData</td><td></td><td></td><td>Omitted if NumAccelAxes is 0. Otherwise, for each reading (NumReadings):
If NumAccelAxes is 3:
<table>
<tr><td>AvgAccelX</td><td>2</td><td>Recent average on X axis</td></tr>
<tr><td>AvgAccelY</td><td>2</td><td>Recent average on Y axis</td></tr>
<tr><td>AvgAccelZ</td><td>2</td><td>Recent average on Z axis</td></tr>
</table>
If NumAccelAxes is 1:
<table>
<tr><td>AvgAccelZ</td><td>2</td><td>Recent average on Z axis</td></tr>
</table></td></tr>
<tr><td>NumTempAxes</td><td>1</td><td></td><td>Number of temperature sensors (usually one per motion axis)</td></tr>
<tr><td>TempData</td><td></td><td></td><td>Omitted if NumTempAxes is 0. Otherwise, for each reading (NumReadings):</td></tr>
</table>

			I If NumTempAxes is 3: AvgTempX 2 Recent average on X axis AvgTempY 2 Recent average on Y axis AvgTempZ 2 Recent average on Z axis If NumTempAxes is 1: AvgTempZ 2 Recent average on Z axis

Joystick Packet

Use the JoyRequest command #17 with an argument value of 0, 1 or 2 to request data from the robot joystick. An argument value of 1 requests a single JOYSTICK packet (type = 248) to be sent by the next client-server communications cycle. An argument value of 2 requests that packets are sent continuously, after each standard SIP, at approximately one per cycle depending on serial port speed and other pending packets. Use argument value 0 to stop continuous JOYSTICK packets.

ARIA provides an interface to the robot joystick in its **`ArRobotJoyHandler`** class.

JOYSTICK packet contents:

Field	Size	Value	Description
Packet header	2	0xFA, 0xFB	Common header
Byte count	1	11	Number of following bytes
Packet type	1	0xF8	Joystick packet type identifier
Connected	1	0 or 1	1 if joystick is connected, 0 if joystick is not connected or disconnect button is engaged.
Button	1	0 or 1	1 while wheel re-align button is pressed, 0 otherwise. Seekur will always re-align the wheels if this button is pressed, but it may be monitored here for software to perform any additional actions.
Rotation	2	varies 0-1023	0 is fully to the left, 1023 is fully to the right. (ARIA converts this to [-1.0,1.0])
Forward/Back	2	varies 0-1023	0 is fully back, 1023 is fully forward. (ARIA converts this to [-1.0,1.0])
Speed	2	varies 0-1023	Speed knob setting
Packet checksum	2	varies	Computed checksum

SeekurParamManager

SeekurParamManager may be used to change configuration parameters stored by SeekurOS. SeekurParamManager connects to over a serial port connection, either from the onboard computer, or from an external computer connected via the firmware maintenance serial port.

seekurParamManager is a text-based console

In Linux, run seekurParamManager using the following commands.

```
$ cd /usr/local/SeekurParamManager
$ ./seekurParamManager
```

In Windows, use the SeekurParamManager icon in the start menu or in **`C:\Program Files\MobileRobots`**.

To view the list of parameters with current values, type ‘v’ or ‘view’ followed by a return (Enter).

To change the value of a parameter, enter its list number, then enter the new value.

To save changes to Seekur, type ‘save’ followed by the return (Enter) key.

Troubleshooting

For additional help, contact support at http://robots.mobilerobots.com/ or by email support@mobilerobots.com. Include your robot's serial number, and a detailed description of your problem including visible symptoms, error messages, status light state, steps you are taking and their results.

Suggestions for troubleshooting some of the most common problems follow.

ON button does not power robot on. All indicator lights remain off and power remains off.

- Verify that all four E-Stop buttons have been released and popped up. If any E-Stop button is depressed, all power to the robot will be disconnected.
- Recharge robot (see **Charging** on page 14)
- Contact support for help testing internal power circuits or determining battery failure.

Robot suddenly shuts down.

- If an E-Stop button was pressed, release the E-Stop button and power on the robot.
- If the robot powers on, press F2 on the diagnostic display and check the battery voltage. The robot automatically shuts down at 20V. If the voltage is low, recharge the robot immediately. (see **Charging** on page 14)
- If the robot will not power on or charge, contact support.

An accessory device does not start, has no power.

- If the device power supply is switchable, use a power control command to turn it on. (see **Power Distribution Board (PDB) for Component Power Supply** on page 32). If power was switched on, but the device still does not have power, reset the switchable output by turning it off, then on again.
- Check the status LED ZD2 on the Power Distribution Board (PDB). If flashing, the device may have drawn above the current limit for that power connection.
- Verify power and signal wiring and connections to robot and onboard computer.

Robot is on, but will not drive.

- Enable motors with MOTORS button if not enabled. Wait for robot and wheel initialization. (see **Enable Motors**, p. 13) The MOTORS button will glow solid blue if motors are enabled.
- If MOTORS button is blinking rapidly, there was an error initializing the robot or wheels. Power cycle the robot and re-enable the motors again.
- Check diagnostic display. If "Bumper Triggered!" is displayed, then a bumper collision was detected. (See **Emergency shut-off (E-Stop) buttons and bumpers** on page 14). To re-enable the robot, first disable any motion control being performed by running software (e.g. enter Stop Mode in ARNL/MOGS). Then press the enter/return key on the diagnostic display to re-enable the robot.
- If software is running and connected to the robot, then If software is running, that software must also enable motors with the ENABLE command.
- If software is running and connected to the robot, then it may be limiting or preventing movement of the robot due to sensor readings or other software logic. It may be requesting joystick control (See **Joystick Software Control Mode** on page 15), overriding direct joystick control. The joystick's indicator light will blink if under software control.

Software will not connect to the robot.

- Only one program may connect to the robot at one time. Close or disconnect other programs that may be running on the onboard computer and connected to the robot. Close SeekurParamManager if running on the onboard computer, or if connected via SeekurOS maintenance port.
- Check the serial cable connection between the onboard computer on which you wish to run the software and the robot microcontroller, inside the robot (Serial cable is usually labeled "Microcontroller). This cable is normally connected to COM1 on onboard PC #1.
- Refer to log messages printed by the software.

For more frequently asked questions and troubleshooting steps, see
http://robots.mobilerobots.com/wiki/Category:Troubleshooting and http://robots.mobilerobots.com/wiki/Category:FAQ

Parts and Accessories

To inquire about replacement parts, contact Adept MobileRobots support (support@mobilerobots.com) to verify the correct part and order. Include the robot's serial number in your message.

Below is a partial list of replacement parts. Contact support for other parts or with any questions.

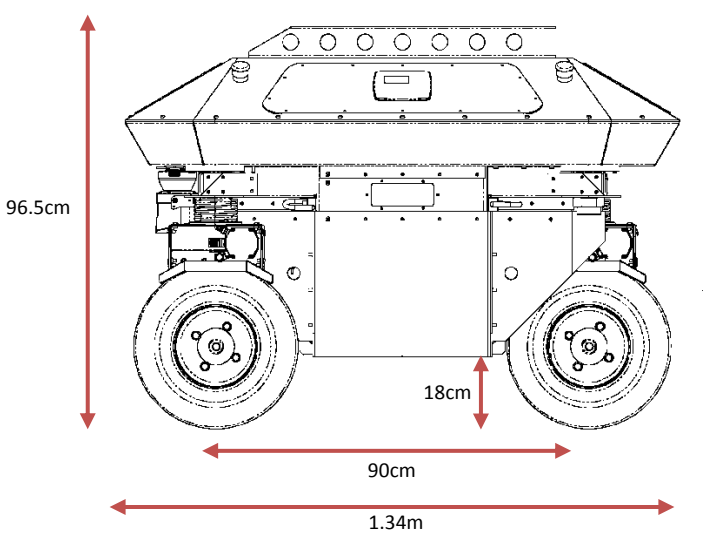
Replacement Tire, Seekur (one is included with robot)	
Replacement Wheel, Seekur	
Replacement bumper strip, long edge, Seekur	
Replacement bumper strip, corner, Seekur	
Access panel, blank, front, Seekur	AMR Part# MCH1877
Access panel, blank, side, Seekur	AMR Part# MCH1881
Front panel with cutouts for connectors and mounting points for cameras or other equipment. Specify connectors to add when ordering.	AMR Part# MCH1915
Replacement conductive gasket for access covers, specify which cover.	AMR Part# MCH1833/MCH1834
Replacement water sealing neoprene strip	
Top panel/cover with equipment mounting rails, Seekur	AMR Part# MCH1875
Top mounting bracket for camera or other equipment, Seekur	AMR Part# MCH2007
Top mounting bracket for GPS or other equipment, Seekur	
Blank/cover for connector cutout	AMR Part# MCH1964
External USB port, weatherproof	
External ethernet port (8P8C/RJ45 modular jack), weatherproof	
External power port	
Cable pass-through gasket (for cables 0.51 inches – 0.71 inches diameter)	
Replacement E-Stop button	
Computer cage, EBX, Seekur	
SeekurOS maintenance serial cable	
Ball joint ¼"X20 camera mount	
Seekur Parts Kit (one included with robot). Includes set of M6x16mm and M4x12mm screws, wire ties, hex keys and fuses.	
Charger power supply	
Antenna for WRAP wireless router (2.4 Ghz 5.5dBi with male N-type connection)	AMR Part# ELC2396

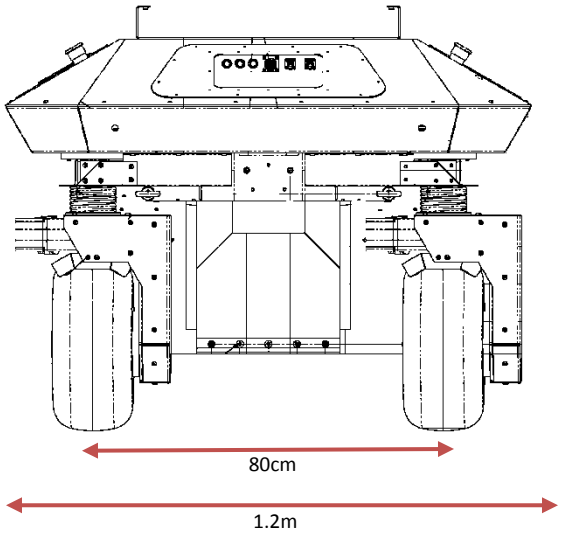
Many additional accessory devices (cameras, sensors, manipulators, etc.) are available for the robot, and accessories are upgraded or added periodically

Below is a partial list of available accessories. For a current list of Seekur accessories available, or for pricing, see http://www.mobilerobots.com and contact sales@mobilerobots.com. Include your robot's serial number to verify compatibility.

- GPS outdoor navigation package, includes GPS receiver and MOGS navigation software (ACT0360)
- Laser navigation package, includes laser rangefinder and ARNL navigation software (ACT0340)
- Rugged pan/tilt/zoom camera and image acquisition interface for onboard computer (ACT0320)
- MobileRanger3D stereo camera and depth sensing system with processor
- High performance pan/tilt unit(s) (ACT0325)
- Additional onboard computers (up to 5), ethernet switch
- Additional power-distribution board
- Inertial Measurement Unit (IMU, automatically used by Seekur to correct rotation) (ACT0310)
- Manipulator arms
- Wireless ethernet interface/access point (WRAP) (ACT0330)
- 900Mhz radio bridge for ethernet (ACT0410)
- Wireless joystick control (ACT0335)
- GPS receiver (hardware only) (ACT0350)

Specifications





Physical:

Width: 1.2 meters (47.2 in.)
Length: 1.34 meters (52.7 in.)
Height: 96.5 cm (36.8 in.)
Wheelbase: 90 cm (35.4 in.)
Wheel track width: 80 cm (31.5 in.)
Ground Clearance: 18 cm (7 in.)
Body construction: Powder-coated aluminum
Robot Weight: 300 kg (661 lbs.)
Turn Radius: 0 cm
Swing Radius: 83 cm (32.7 in.)
Suspension: Independent spring coil shock absorbers

Environment and Terrain:

IP Rating: IP-54
Operating Temperature: -20° to 50° C (-4° to 122° F)
Ground Clearance: 18 cm (7 in.)
Max. Traversable Step: 17 cm (6.7 in.)
Max. Traversable Slope: 35% (unladen)
Max. Traversable water depth: 17 cm (6.7 in.)
Indoor surfaces: tile, concrete, flooring
Outdoor surfaces: Pavement, grass, light snow, light rocky terrain, packed dirt, gravel, sand.
All-weather (rain, snow, dust) capable.

Movement:

Max. Forward/Backward/Left/Right Speed: 1.8 m/s (6.5 kph, 5.9ft/s, 4 mph)
Max. Rotation Speed: 150 deg/sec.
Control: omnidirectional (holonomic), with simultaneous translation and rotation, velocity control with configurable acceleration and deceleration.

Payload:

Rated Operating Payload: 70 kg (154.3 lbs.)

Tires and Wheels:

Tire pressure: 28 psi (193 kPa)
Tire size: 16 X 6.50 – 8
Wheel diameter: 8 inches (20.32 cm)
Tire diameter: 16 in. (400mm)

Screws and Bolts:

Front/side Access Panels: M4 X 12mm with 3mm hex head
Top Panel: M5 with 4mm hex head

Power:

Run Time: 3-8 hours approx. (with some accessory devices)
Complete Charge Time: 8 hours
Battery Capacity: 100 Ah
Battery Chemistry: NiCad
Software-Switchable Power Supplies (some are used by accessory devices, some available for user custom use):

- 2 x 5 VDC, 2 A regulated (4 with second PDB)
- 4 x 12 VDC, 2 A regulated (8 with second PDB)
- 2 x 24 VDC, 1 A regulated (4 with second PDB)
- 2 x 24 VDC, 5 A (4 with second PDB)
- 1 X 24 VDC, 8 A max raw battery

See **Power Distribution Board (PDB) for Component Power Supply** on page 32

Battery Charging Power Supply:

Input: 100-120 or 200-240 VAC (47-63 Hz), 2600 VA
Output (robot-controlled): 27-36V DC max 35A

Index

R

S

T

U

V

W