OMRON | adept

Omron Adept Mobile Robots
For Research, Education and Development

Seekur Jr.

SKR0100, SKR0200

Manual

Revision 5, January 31, 2017

Omron Adept MobileRobots • 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

Important Safety Instructions

- ✓ Read the installation and operations instructions before using the equipment.
- ✓ Avoid using power extension cords.
- ✓ To prevent fire or shock hazard, do not expose the equipment to rain or moisture.
- ✓ Refrain from opening the unit or any of its accessories.
- ✓ Keep wheels away from long hair.
- ✓ Never access the interior of the robot with charger attached or batteries inserted.

Inappropriate Operation

Inappropriate operation voids your warranty! Inappropriate operation includes, but is not limited to:

- ✓ Dropping the robot, running it off a ledge, or otherwise operating it in an irresponsible manner
- ✓ Overloading the robot above its payload capacity
- ✓ Continuing to run the robot after hair, yarn, string, or any other items have become wound around the robot's axles or wheels
- ✓ Working inside the robot when power is ON
- ✓ All other forms of inappropriate operation or care
- ✓ Submerging the robot in water beyond rated depth
- ✓ Operating the robot in hazardous environments

Contents

Introduction.

This ***Seekur Jr Operations Manual*** provides both the general and technical details you need to operate your new intelligent robot base and to begin developing your own robotics software.

Every Seekur Jr platform comes complete with a sturdy aluminum body, balanced drive system (two-wheel differential with casters or four-wheel skid-steer), reversible DC motors, motor-control and drive electronics, high-resolution motion encoders, and battery power, all managed by an onboard microcontroller capable of moving the robot platform at commanded vehicle velocity and estimating the robot's position in the world as it moves, and communicating with your custom high-level software running on an onboard or offboard computer.

Our experienced manufacturing staff put your mobile robot and accessories through a "burn in" period and carefully tested them before shipping the products to you. In addition to the companion resources listed above, we warrant your Seekur Jr platform and our manufactured accessories against mechanical, electronic, and labor defects for one year. Third-party accessories are warranted by their manufacturers, typically for 90 days.

Even though we've made every effort to make your robot package complete, please check the components carefully after you unpack them from the shipping crate.

Basic Components (all shipments)

- ✓ One fully assembled Seekur Jr mobile robot with battery or batteries
- ✓ Microcontroller with SeekurOS mobility firmware (internal)
- ✓ Internal heading correction gyroscope (internal)
- ✓ Joystick
- ✓ Power supply and battery charger
- ✓ ARIA, MobileEyes, MobileSim, Mapper3-Basic software
- ✓ CD-ROM containing licensed copies of software and documentation
- ✓ Set of print manuals
- ✓ Hex wrenches and assorted replacement screws
- ✓ Replacement fuse(s)
- ✓ Registration and Account Sheet

User-Supplied Components / System Requirements

- ✓ Computer with Windows or Linux operating system

Optional Components and Attachments (partial list)

- ✓ Additional batteries
- ✓ One or two onboard computers with requested operating system (Linux or Windows) and software preinstalled
- ✓ Wifi access point/bridge (WRAP)
- ✓ Laser range finder with Advanced Robotics Navigation and Localization (ARNL) software
- ✓ Global Positioning System
- ✓ MOGS outdoor navigation and localization software
- ✓ Stereo Vision Systems
- ✓ Pan-Tilt-Zoom Cameras
- ✓ High precision pan-tilt unit
- ✓ Manipulator arm

- ✓ Other accessory or device not listed here. See www.mobilerobots.com or contact sales@mobilerobots.com for more information on accessories and options.

Unpacking and Assembly

Seekur Jr. is shipped fully assembled, with the exception of the removable batteries, and some external accessory devices.

First, lift Seekur Jr. from its shipping box. Seekur Jr. weights approximately 70 kg. (661 lbs.) so several people should carefully lift the robot out of the box. To move the robot, the manual joystick may be used. Insert one or all batteries (see below for instructions), ensure all four E-Stop buttons are released, turn on the robot, attach the joystick, then enable motors. Make sure the joystick is set at a low speed, then carefully drive the robot using the joystick. See "Specifications and Controls" and "Accessories" below for more details on operating the robot and the joystick.

Attach any optional external accessories to the equipment mounting decks of the robot. Cabling for these accessories has been included and labelled.

- The WRAP wifi access point bracket may be attached anywhere on the rear equipment deck of the robot. Connect it to the WRAP/PoE (Power-over-ethernet) port on the back of the robot using the included CAT-6 cable.

Refer to supplementary documentation included for the optional accessories for more information on attaching, configuring and using them.

If you have any questions or concerns about unpacking and assembling your robot, please contact technical support at support@mobilerobots.com. Include your robot's serial number (located on a label under the removable center battery hatch.)

Additional Resources

New customers get three additional and valuable resources:

- ✓ Login and password for our support Internet website http://support.mobilerobots.com for downloading software, updates, and manuals
- ✓ Access to user discussion groups
- ✓ Direct access to the technical support team (support@mobilerobots.com)

Support Website

We maintain a 24-hour, seven-day per week World Wide Web server where customers may obtain software and support materials:

http://support.mobilerobots.com

Some areas of the website are restricted to licensed customers. To gain access, enter the username and password written on the *Registration & Account Sheet* that accompanied your robot.

User Discussion Groups

We maintain several email-based discussion groups through which robot owners share ideas, software, and questions about the robot. Visit the support http://support.mobilerobots.com website for more details. To sign up for pioneer-users, for example, send an e-mail message to the –requests automated newsgroup server:

To: pioneer-users-requests@MobileRobots.com

From: <your return e-mail address goes here>

Subject: <choose one command:>

help (returns instructions)

lists (returns list of newsgroups)

subscribe

unsubscribe

Our e-mail list server will respond automatically. After you subscribe, e-mail your comments, suggestions, and questions intended for the worldwide community of Pioneer users:[1]

To: pioneer-users@MobileRobots.com

From: <your return e-mail address goes here>

Subject: <something of interest to Seekur users>

Access to the pioneer-users e-mail newslist is limited to subscribers, so your address is safe from spam. However, the list currently is unmoderated, so please confine your comments and inquiries to issues concerning the operation and programming of ADEPT MOBILEROBOTS platforms.

Support

Have a problem? Can't find the answer in this or any of the accompanying manuals? Or do you know a way that we might improve our robots? Share your thoughts and questions with us from the online form at the support website:

http://support.mobilerobots.com/techsupport

or by email:

support@MobileRobots.com

Please include your robot's ***serial number*** (see label near battery hatch)—we often need to understand your robot's configuration to best answer your question.

Tell us your robot's SERIAL NUMBER.

Your message goes directly to the technical support team. There a staff member will help you or point you to a place where you can find help.

[1] Note: Leave out the –requests part of the email address when sending messages to the newsgroup.

Because this is a support option, not a general-interest newsgroup like pioneer-users, we reserve the option to reply only to questions about problems with your robot or software.

Use authorized parts *ONLY*;
warranty void otherwise.

Pioneer SDK

All Adept MobileRobots platforms operate as the server in a client-server environment: Their microcontrollers handle the low-level details of mobile robotics, including maintaining the platform's drive speed and heading, acquiring sensor readings and managing some attached accessories. To complete the client-server architecture, robot platforms require a PC connection: software running on a computer connected with the robot's microcontroller via the HOST serial link and which provides the high-level, intelligent robot controls, including obstacle avoidance, path planning, features recognition, localization, navigation, and so on.

An important benefit of the client-server architecture is that different robot servers can be run using the same high-level client. Several clients also may share responsibility for controlling a single mobile server, which permits experimentation in distributed communication, planning, and control.

The Pioneer SDK is a collection of libraries and applications that come with every mobile robot and with selected accessories. The standard Pioneer SDK bundled with Seekur Jr at no extra charge includes the open-source ARIA and ArNetworking, the MobileEyes GUI application and MobileSim.

ARIA provides an interface and framework for controlling and receiving data from the robot, as well as most accessories (some devices also have separate interface libraries). ARIA also has utilities useful for writing robot control software as well as tools for writing cross-platform (Windows and Linux) code and support for network sockets and threads. **ArNetworking** provides a simple, extensible

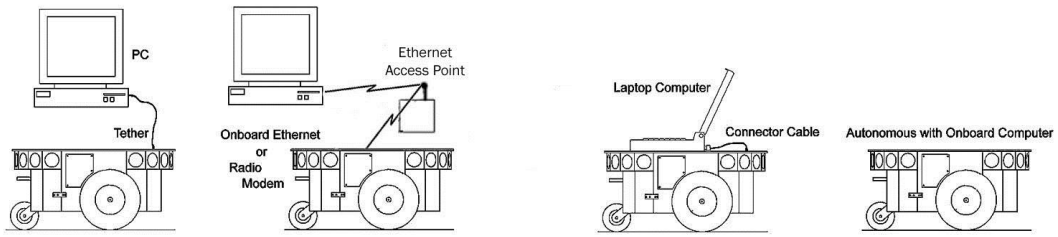


Figure 1. Adept MobileRobots platforms require a PC to run client software for intelligent robotics command and control operations.

framework for client-server network programming. ARIA is a C++ library that can be used on Linux or Windows with minimal additional dependencies. Interfaces are also available to use ARIA from Python, Java and Matlab.

MobileSim is an open-source high level abstract simulation of all Pioneer platforms and some accessories.

MobileEyes, enabled through ARIA and ArNetworking, is a GUI application for configuration, operation and monitoring of your robot platform over the network.

If a navigation package was purchased, then the ARNL or MOGS library will be available. **ARNL** enables a robust, laser-based autonomous localization and navigation indoors or other flat spaces with many stationary features. ARNL is the best-in-class software foundation for Adept MobileRobots' Motivity commercial- and industrial-ready mobile localization and navigation systems. **MOGS** enables autonomous navigation in open outdoor areas by fusing GPS position data with robot odometry .

Several other robotics applications development environments also have emerged to support Pioneer mobile robots, including Ayllu, Pyro, Player, and ROS. See documentation specific to those development environments for more details.

Specifications & Controls

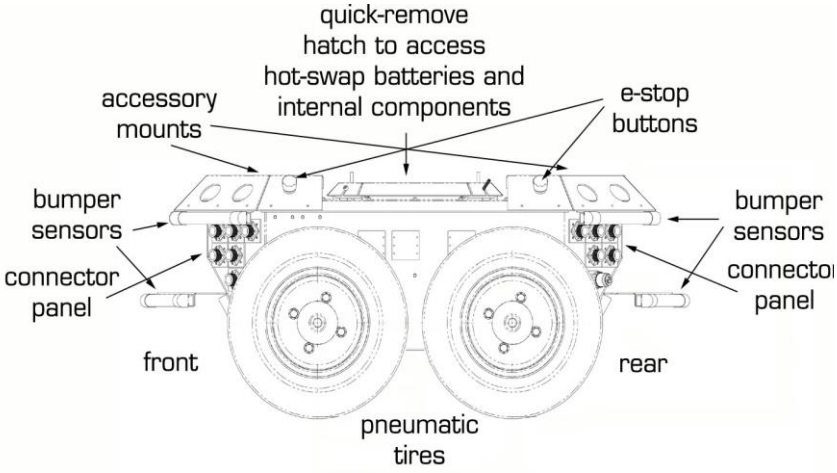
Seekur Jr is a sturdy platform that can handle everything from open fields to parking garages. This skid-steer, differential-drive platform operates on moderate terrain with good stability.

Physical Characteristics and Components

Seekur Jr weighs 70 kilograms (154 pounds) with one battery installed. Two powerful reversable DC motors drive the left and right wheels, providing differential drive, skid-steer mobility for the robot.

Seekur Jr's built-in controller implements velocity control of the robot, automatically controlling motor speeds and reading motor encoder signals to achieve desired vehicle translation and rotational velocities using desired acceleration and deceleration profiles.

Seekur Jr. is ruggedized for all-season outdoor operation on moderatelty rough terrain, and it is water resistant. Space is available inside the robot and on its accessory mounting decks for customizing the robot with your own sensors and other payloads.



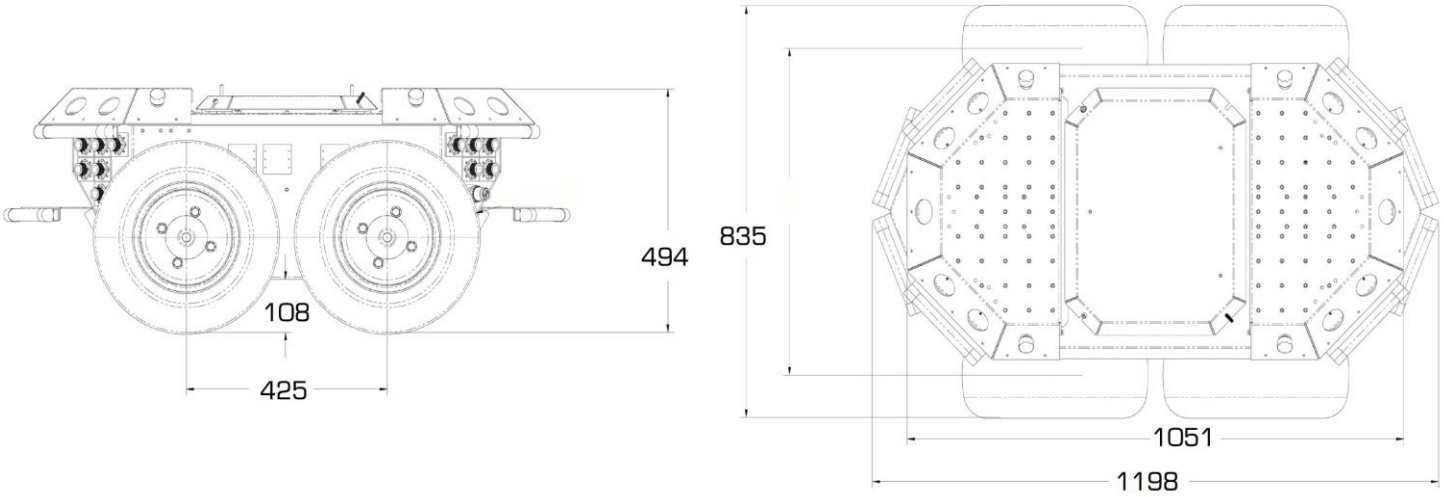


Figure 5. Physical dimensions

Operating Environment and Terrain

Seekur Jr and all standard accessories are designed to operate indoors and outdoors, on most surfaces. It is resistant to light rain and snow, and can operate in temperatures ranging from -5° to 35° C (23° to 95° F). It is capable of climbing a maximum 70% grade (but restricting operation to less steep hills is recommended with higher payload weight added or when operating autonomously, especially if robot may rotate or drive across (sideways) a slope.)

Seekur Jr uses differential-drive controls to turn, requiring the tires to slip or skid as the robot turns, especially when rotating in place with no forward movement. This can be damaging on high-friction surfaces, such as carpet, so we recommend operation either on unconsolidated or slip-easy surfaces like grass, sand or smooth tile or concrete to significantly reduce motor and motion stresses.

Seekur Jr's tires have been selected for efficient operation on smooth surfaces such as asphalt, concrete, smooth tile, packed dirt and gravel. If a more aggressive tread is desired, the tires may be replaced. See **Appendix C on page 048** for specifications.

Warning: Do not submerge Seekur Jr in water. Do not operate in water more than 12 cm (5 in.) deep. Do not operate Seekur Jr in blowing rain or spray Seekur Jr with water under pressure. Do not operate Seekur Jr in the presence of chemicals or other hazardous substances.

Before operating Seekur Jr. outdoors or in any situation in which it will encounter water:

1. Ensure that the computer access plates have undamaged seals and are attached with all screws.
2. Ensure that all four screw-down bolts on the battery access hatch are tightened.
3. Ensure that no connectors or buttons are loose or missing.
4. Cover the charger port with its cap.

After operating Seekur Jr. outdoors or in wet conditions:

1. Clean any dirt, mud or debris from the robot, especially behind the wheels, on any connectors or buttons, near the battery access hatch, and on the SICK laser rangefinder or any other external accessories (use a soft nonabrasive lens cleaning cloth when cleaning the laser rangefinder lens).
2. Inspect the interior of the robot to verify that no water, dust or dirt has entered the robot.

Park/store, recharge, and service Seekur Jr. in a dry, covered area only.

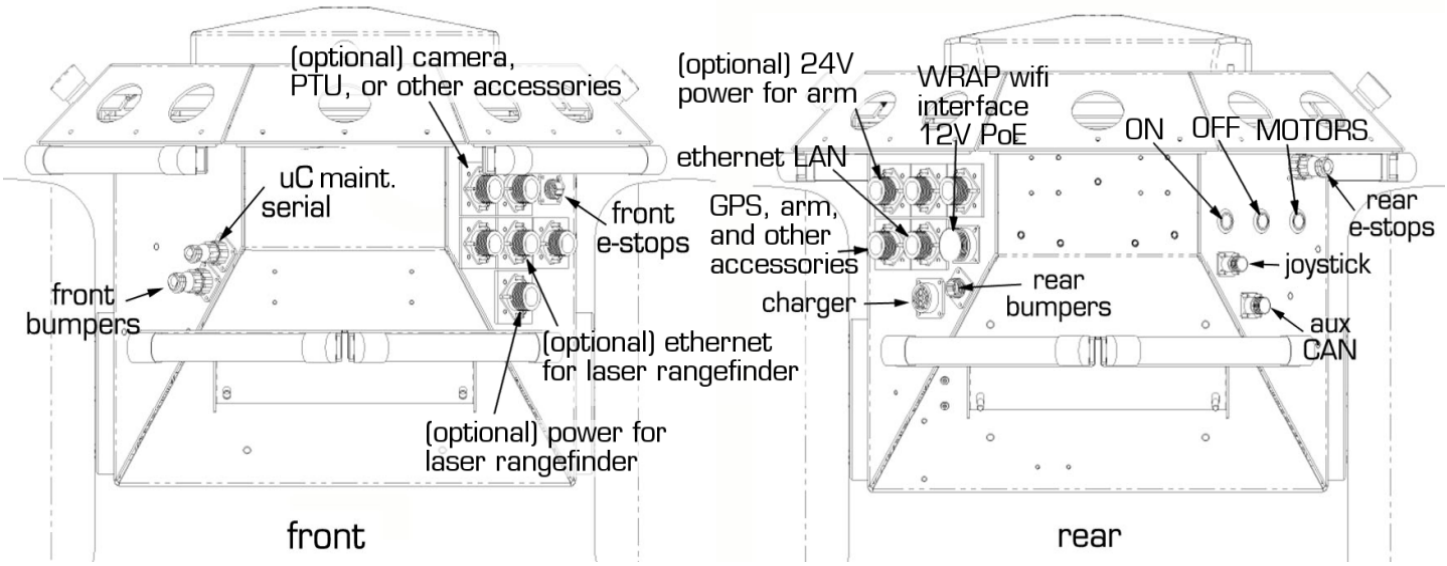


Figure 6. Seekur Jr typical connectors; exact connectors present on an individual robot may vary depending on accessories.

Connectors and Interfaces

Seekur Jr supports eight standard connectors and up to 12 additional ones depending on accessories. Blank cover plates are added when a connector is not present. This plate can be removed and a weatherproof connector added in its place. Pinouts, part numbers, and mating connector specifications are provided in *Appendix A*.

> ***Note*: The E-stop buttons and bumpers must be connected in order to operate Seekur Jr.**

The microcontroller (uC) maintainence serial port can be used for updating and configuring SeekurOS (see **SeekurParamManager on page 36** for details) instead of using an onboard computer. It can also be used for robot control if your robot does not have any onboard computer.

The Joystick connector is used to connect the joystick for manual driving. See *Joystick and Joystick Modes* on page 16.

The Aux CAN ports are reserved for future Seekur Jr components and accessories.

The large 66-pin connector is used for data I/O to the GPS and other accessories. Unused pins in this connector can be used for your own device I/O, and low power. See *Appendix A* for more information about this connector.

The PoE port for the WRAP interface includes 12V power. Do not attach any other ethernet device to this connector.

The unpowered ethernet connectors connect to the robot's internal ethernet LAN. On Seekur Jr. robots built after March 2013, a modular ethernet connector is used on the front of the robot for the SICK LMS-111 laser rangefinder. On some robots built prior to March 2013, the laser rangefinder ethernet cable passes through a gasket into the interior of the robot without using a connector. The ethernet connector on the rear of the robot may be used to attach a laptop, or to connect the robot's internal LAN to an external network.

See *Onboard Computers, Internal Network (LAN) and Wireless (wifi) Network* on page 16 for more about networking in the robot.

The optional MobileRanger stereo camera uses a modular connector (also known as "RJ11"), but it is not an ethernet device. Do not connect this port to an ethernet device or network. Do not connect the MobileRanger stereo camera to an ethernet connector.

Pinouts, part numbers, and mating connector specifications are provided in *Appendix A*.

Batteries and Charging

Seekur Jr operates with one, two or three nickel-metal hydride (NiMH) 24 VDC hot-swappable battery packs, located just beneath the top battery access hatch.

Installing and Removing the Batteries

One of the first things that you will do when Seekur Jr arrives into your workspace is install its batteries. It is a five-step process to insert or remove one or more battery packs: Remove the top hatch, open the battery pack restraints, slide one and up to three battery packs into position, flip down the restraints to secure the pack(s), and attach a power cable to each battery. See adjacent illustrations for details.

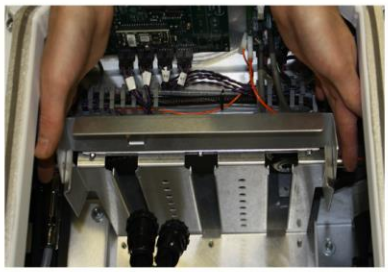
Carefully align the cable and connector keys to avoid damage.

To remove a battery pack, open the top hatch, remove the power cable(s), relieve the restraints, lift out the battery pack(s), and then re-secure the restraints.

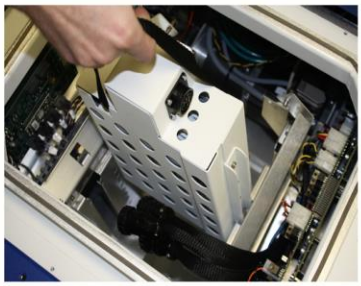
Finally, re-secure the top hatch with its four latch screws.

Only open the hatch and change batteries in a dry area to prevent water from entering the robot.

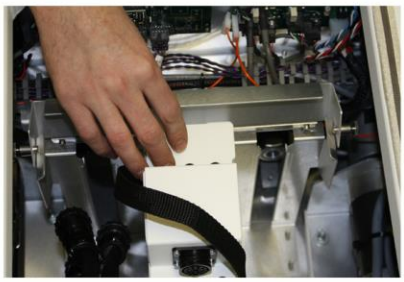
STEP 1: Loosen the four latch screws at the corners and lift off Seekur Jr's top hatch.



STEP 2: Inside, pull the spring-loaded pin latches on each side of the battery mount and open the two restraints on their hinges.



STEP 3: Slide one to three battery packs into position along the guides with their power connector facing the power wires.



STEP 4: Flip the two battery restraints down and latch them to secure the packs.

STEP 5: Attach a power cord to each pack (no order – any cord to any battery) and twist the connector to secure.

Charging and Runtimes

Seekur Jr's onboard systems will continue to operate while the power supply and battery charger is attached. Battery operating times vary depending on mobile activity mostly: Seekur Jr will run for up to three hours of continuous motion starting with three fully charged batteries. One battery provides nearly one hour of continuous run time. The onboard systems will run over four times as long on batteries if the robot is not driven.

Driving is disabled while the charger is attached. The blue MOTORS button will flash while charging to indicate that the motors are disabled.

For nearly continuous battery operation, use hot-swapping with additional battery packs: Either temporarily connect the Power/Recharging Module to provide systems power while removing exhausted battery packs, or keep at least one battery in place while replacing the others.

Batteries can be recharged while installed in the robot with the power supply/recharger attached to the charge port on the rear of Seekur Jr. A battery can also be recharged by attaching the power supply/recharger directly to the battery

It takes three or more hours to recharge one or more of Seekur Jr's NiMH battery pack(s).

Note: Some devices (including RVision pan/tilt/zoom camera and Terabot arm) may be attached to raw battery power supply in the robot, and will be automatically disconnected while the charger is attached to the robot to avoid damage to these devices, since voltage may exceed the specified 30VDC. These devices will be switched back on when the charger is disconnected.

Recharge Seekur Jr. while in a dry, sheltered area only.

Charge Indicators

SeekurOS keeps tabs on the state of charge for each battery and notifies a connected client like the ARIA demo of the charge state, including when the batteries get below 10% (LowSOC parameter may be changed using SeekurParamManager). If the batteries go below 5% state of charge (ShutdownSOC parameter) SeekurOS shuts down all Seekur Jr systems.

When starting up Seekur Jr with its batteries below ShutdownSOC, systems power is removed after about ten seconds. Currently, there is no other visual or audio warning of a low-battery condition.

Power and Motors

Three buttons on the right rear body of Seekur Jr control main power and power to the motors. Four Emergency Stop (e-stop) buttons at the top corners of Seekur Jr also control power to the robot (see E-Stop below).

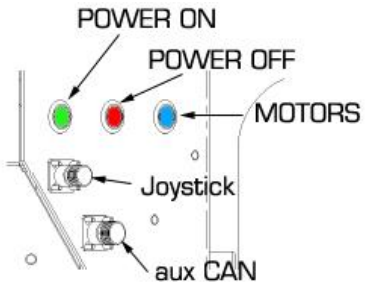


Figure 2. Seekur Jr power and motor buttons (right rear panel)

Main Power

Press the left-most POWER ON button to supply power to the onboard systems. The button will light green when the robot has successfully powered all internal systems. Press the center POWER OFF button (red when lit) to immediately remove power from all the onboard systems.

If the robot does not respond, make sure all the e-stop buttons are connected and released by pressing each down then rotating clockwise to release, and that the robot has batteries or is connected to an external power supply, and press POWER ON again. If power shuts down ten seconds or so after startup, battery charge probably is below ShutdownSOC of 5% and need recharging.

E-Stops

Press any one of the four red e-stop buttons mounted on top near the four corners of Seekur Jr to immediately remove main power from the robot. Cables to the buttons run beneath the accessory mount panels front and rear and connect to the interior power through the front and rear body panels. Seekur Jr will not operate without proper e-stop connections.

To relieve an e-stop, twist the cap clockwise until it pops up. You'll then have to restart the robot to continue operation.

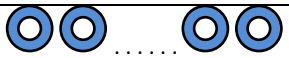
MOTORS

The third user-operated button next to the main-power buttons is the MOTORS button (blue when lit). It must be pressed at least once after starting power to the robot in order to engage hardware-mediated power to the motors. The button does not engage the motor drivers, however. That sequence is a bit more complicated and depends on the readiness of the robot microcontroller (µC), motor-control initialization and client or joystick connections.

As shown in the truth table nearby, motors status depends on whether and when the MOTORS button gets pressed and whether or not SeekurOS has completed its startup and has initialized its motor-control systems. Startup takes about ten seconds. Motor initialization takes about 45 seconds to complete and happens only after SeekerOS power startup and after the MOTORS button gets pressed. Accordingly, most users will press the MOTORS button immediately after pressing the main POWER ON button.

Table 1. MOTOR power sequence and status indication

MOTORS button	Robot ready?	MOTORS LED indicator	Result
Button not pressed	Robot internal initialization, not ready yet.	Off	**Motors are disabled**
Enable requested by button press	Robot not ready (Internal initialization is still in progress, or robot is recharging)	. . . . . Single blink	**Motors are disabled**
Enable requested by button press	Robot ready		**Motors are enabled**

		Solid On	
Disable requested by button press	Robot ready	 double blink	**Motors are disabled**

The MOTORS button will double blink when motors are automatically disabled during charging.

Once the motors have power and the control systems are fully initialized, you may press the MOTORS button again to manually disable power to the motors and thereby immobilize Seekur Jr without removing main power. Press the MOTORS button again to immediately restore power to the motors—the motor-control systems don't need to re-initialize.

To actually drive the robot from software, the motors need to also be enabled by software by sending the ENABLE command (e.g. using `ArRobot::enableMotors()` in ARIA).

Bumpers

Front and rear bumpers provide emergency collision sensing for when other sensing has failed to detect an obstacle. When a bumper receives contact, the robot will stop immediately and motors will be disabled. Use the MOTORS button to re-enable motors after recovering the robot from collision and ensuring that it will be safe to continue operating the robot. The robot will indicate STALL state to client software as well as which bumper was hit (bumper segments are numbered clockwise from one to four starting with the front-left bumper).

Your robot may not move if you unplug one or both bumpers.

Communications Timeout and Stall Monitors

SeekurOS contains a communications "watchdog" monitor that will halt the robot's motion if communications between a PC client and the robot server are disrupted for specified time interval (two seconds, by default). The robot will automatically resume activity, including motion, as soon as communications are restored.

SeekurOS also contains a stall monitor. If SeekurOS detects that the wheels fail to turn despite active motion commands, motion is halted for a configurable amount of time (StallWait parameter). SeekurOS also notifies the client which motor is stalled. When the time elapses, motion continues as requested. It is recommented that you controlling software detect stall events and change its behavior or motion plan accordingly.

Adding Equipment

Equipment may be added to the front and rear exterior equipment mounting decks supplied with Seekur Jr. These decks include threaded holes for M6 machine screws, as well as ¼-inch untapped holes for cameras and other devices. If your robot does not have the optional GPS receiver mounted, then equipment can be mounted on the center battery hatch as well.

The maximum rated payload for Seekur Jr. is 40 kg. (88 lbs.) Exeeding this payload weight will cause reduced performance and possibly damage to Seekur Jr. When mounting payload, be aware that it may raise the center of mass of the robot, reducing the maximum slope on which it can operate without turning over.

To remove the equipment mounting decks, use a 3mm hexagonal wrench to remove all screws which attach the decks to the robot. Before lifting the decks, disconnect the wiring for the red e-stop switches, to avoid breaking the wiring.

Equipment may be added to the interior of the robot in place of the optional second onboard computer, if desired. A blank mounting bracket is available from MobileRobots.

Accessories

For convenience, we include a description of the more commonly integrated accessories in this document. Please also refer to the detailed documents that come with the accessory.

Joystick and Joystick Modes

Figure 3. Seekur Jr joystick

Seekur Jr has a joystick interface for manual operation of the robot.

Connect the joystick to the joystick port on the back of the robot. Adjust the speed knob counter-clockwise (to the left) to set a slow initial speed. Hold the green GO button, and move the joystick to drive the robot. Release the joystick to stop motion. Release the GO button or disconnect the joystick to disengage joystick driving.

The joystick may be connected or removed from the robot at any time.

If no software client is running and connected to the robot, the robot's own controller will drive the robot from joystick input. No sensing is performed in this mode. The green LED will be lit when the robot is ready to drive in this direct joystick driving mode, and you have pressed the green GO button to enable the joystick. The robots motors must be enabled by pressing the MOTORS button to drive with the joystick.

> ***Warning**: When the GO button is released or the joystick is disconnected from the robot, the robot controller leaves joystick driving mode and will no longer hold the motors in position on a hill. The robot may roll down the hill.*

However, if a software client is running and is connected to the robot, and it has requested joystick control via the JOYDRIVE command, then instead of driving the robot directly, joystick input is sent to the software instead. The software may drive the robot using its own speed limits, obstacle sensing, or other logic, or use the joystick input for another purpose. The yellow GOAL button may also be used by software as an input (when scanning for a new map for ARNL, this button places a goal in the map). Robot joystick input may be accessed in ARIA using the **`ArRobotJoyHandler`** class.

The green LED on the joystick will flash when in this mode.

Client Connections and Accessories

All Adept MobileRobots platforms are servers in a client-server architecture. You supply the client computer to run your intelligent mobile-robot applications. The client can be either a piggy-back laptop or embedded SBC, or an off-board PC connected through radio modems or wireless serial Ethernet.

In all cases, the client computer connects with the microcontroller through a dedicated HOST serial port in order for the robot and your software to work. The connection may be direct internally or, in the absence of an onboard SBC, via the uC serial port on the front right panel using the provided adapter cable.

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

WRAP wifi interface

One or more industrial single-board computers (SBCs) may be mounted internally to the robot. Either Linux or Windows has been preinstalled on each computer, along with all 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.

To access the onboard computer(s), first remove the equipment mounting decks. Use a 3mm

hex head wrench to remove all screws attaching the deck to the robot, and carefully lift the deck enough to reach underneath and disconnect the e-stop wiring without damaging it. Then, use the 3mm wrench to remove all screws attaching the computer cover to the robot. When replacing the computer cover, check that the waterproof and conductive gaskets are in place and undamaged. Replace *all* screws to maintain water tightness.

Power to the computers and disks are supplied by Seekur Jr'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 43 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.

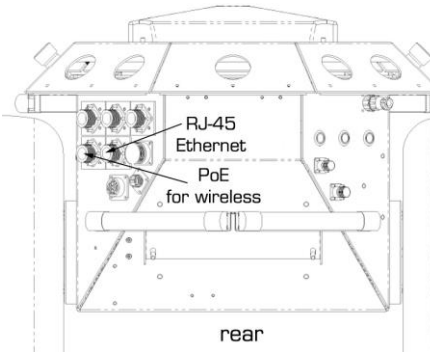


Figure 4. Ethernet and PoE/WRAP connectors

More technical information including a detailed computer manual is available from the MobileRobots support website: http://support.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)	SeekurJr
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

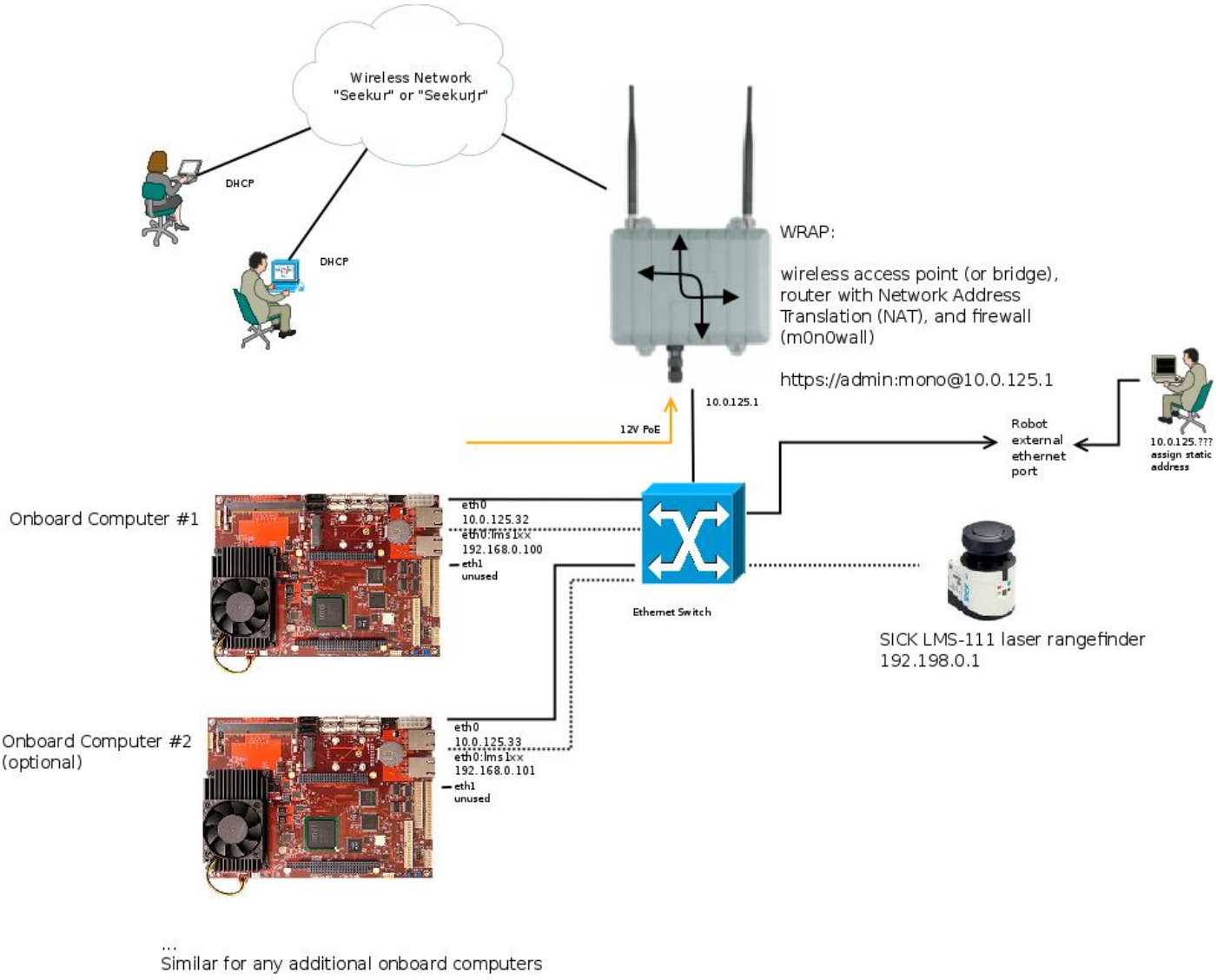
The WRAP may be reconfigured instead to link to an existing wifi network if desired. It also includes a firewall, network address translation (NAT) and other features. Wifi access point settings such as ESSID name and WEP key may also be configured. For details, see http://support.mobilerobots.com/wiki/WRAP_Wireless_Bridge/Router

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://support.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 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 **`seekurjr`** 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://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN)** on page 16 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://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr 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 16 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:
   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 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 28 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://support.mobilerobots.com/wiki/MobileEyes.

Onboard Computer Running Linux:

1. If using simulation, install MobileSim and ARIA from the CD or http://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN)** on page 16 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://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr 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 16 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"
```

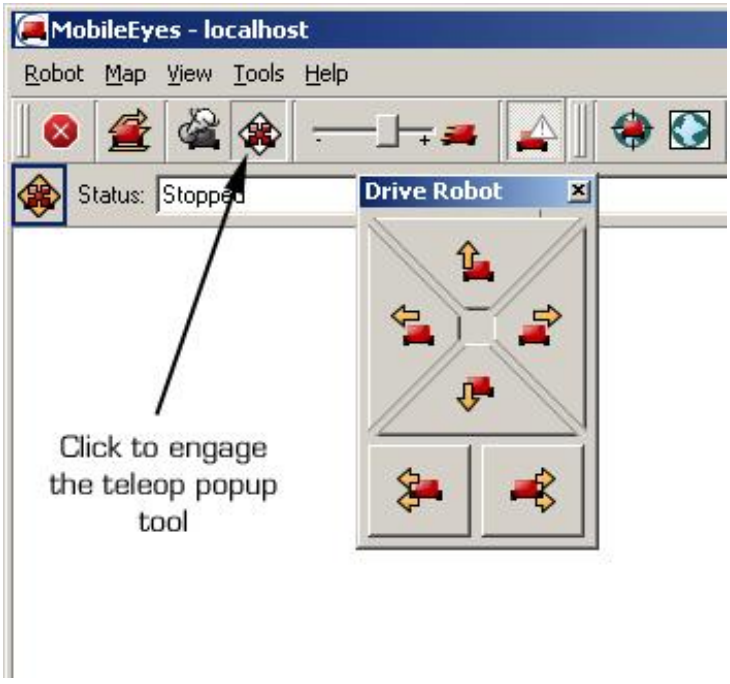
   c. Run serverDemo with connection to the laser rangefinder with the following command:

```
.\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. Seekur Jr. moves fast!

Seekur Jr.'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://support.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://support.mobilerobots.com/wiki/MobileEyes.

Onboard Computer Running Linux:

1. If using simulation, install MobileSim and ARIA from the CD or http://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr wireless network, and log in to the onboard computer (address 10.0.125.32) using ssh (See **Onboard Computers, Internal Network (LAN)** on page 16 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://support.mobilerobots.com/wiki/Software. Run MobileSim and select the "seekurjr" robot type.
2. If using the actual robot, join the SeekurJr 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 16 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://support.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!

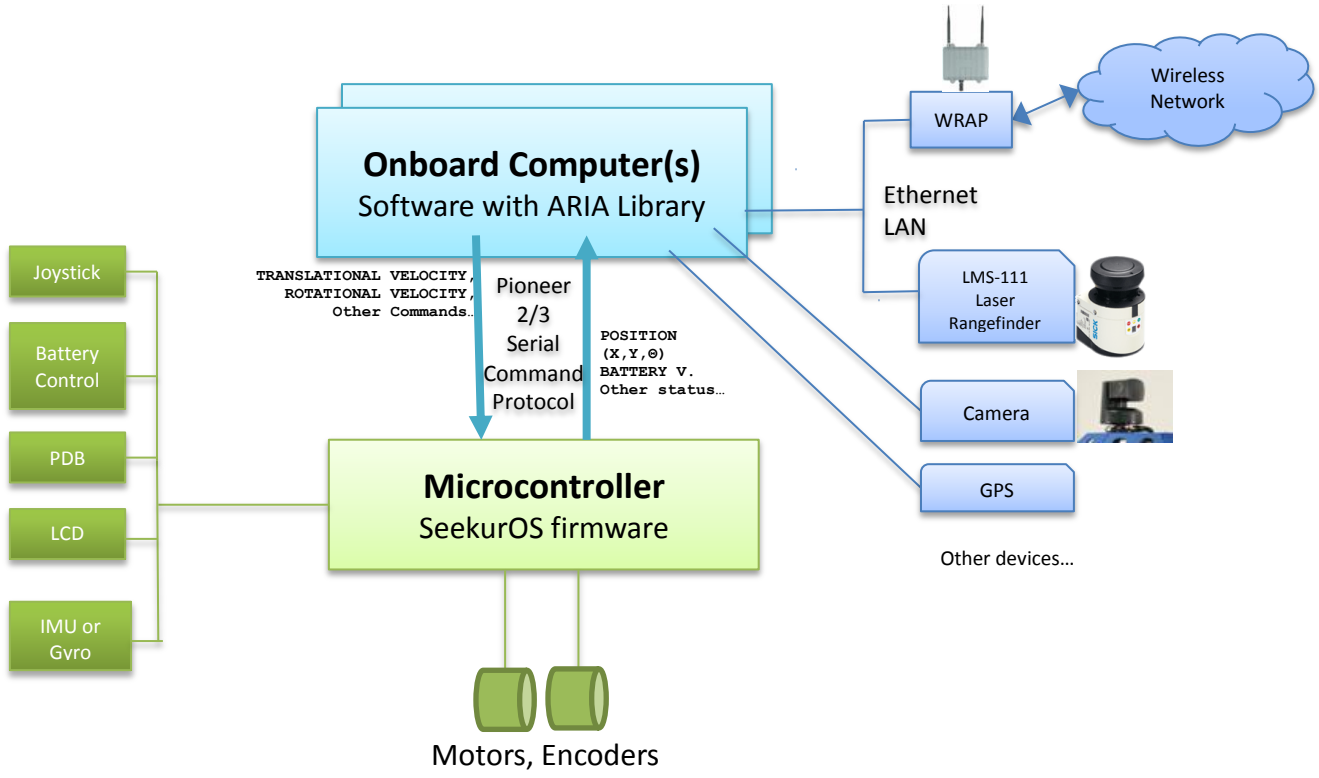
Seekur's motors must be enabled with the MOTORS button before it can drive.

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://support.mobilerobots.com/wiki/Running_A_Linux_Program_In_The_Background.

Programming and Command Protocol

All Pioneer 2/3-compatible platforms use a two-tier architecture. Seekur Jr'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, we provide 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://support.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, we provide 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://support.mobilerobots.com/wiki/Software.

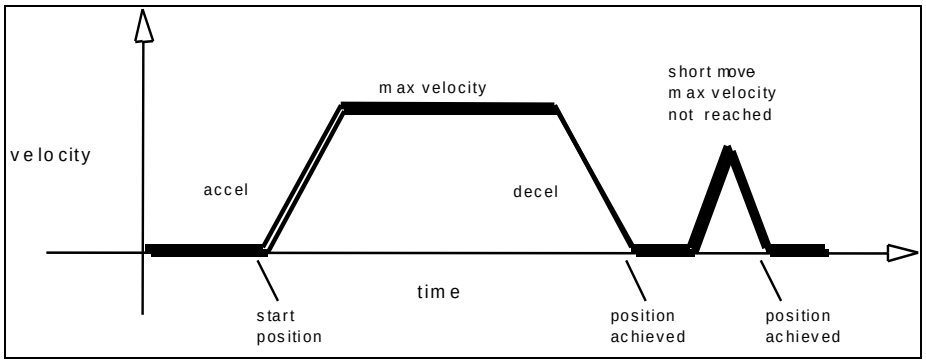
ARIA automatically handles all communication with the robot and many accessory devices, including but not limited to laser rangefinders, the robot's built in sonar and bumper sensors, pan/tilt cameras and pan/tilt units, GPS receivers, and more. To communicate with the robot, ARIA sends and receives 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 Jr 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 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.



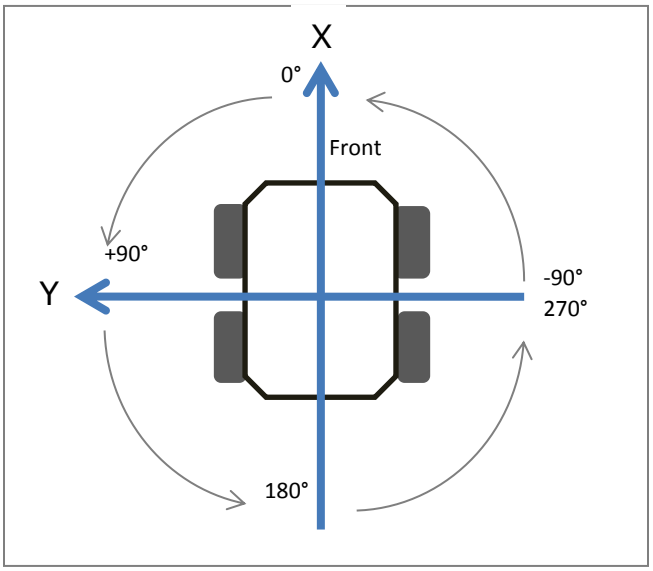
The orientation commands HEAD (#12), DHEAD (#13) and DCHEAD (#22) turn the robot to an absolute heading (0-359 degrees), relative to its current heading, or relative to its current heading setpoint (achieved or last commanded heading), respectively.

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

The robot controller firmware tracks platform 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 over time 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 Pioneer 2/3-compatbile MobileRobots platforms implement the same Pioneer 2/3 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 bytes called **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 or unsigned, and are 16-bit (2-byte) integers, least-significant byte first. Single-byte values are one 8-bit byte. Text strings may be either length-prefixed or NULL terminated (depending on packet type).

Packet Checksum

Calculate the 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 c = 0;
int i = 3;
unsigned char 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 periodically sends packets over the connection back to the connected client software. The default sending interval is 100ms (10Hz), which can be modified by changing the SIPCycle firmware parameter. 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 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 robot platforms but are not relevant or implemented in Seekur Jr. (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.

<table>
<tr><th>Field</th><th>Size</th><th>Description</th></tr>
<tr><td>Packet Header</td><td>2 bytes</td><td>Exactly 0xFA (250), 0xFB (251)</td></tr>
<tr><td>Packet Byte Count</td><td>byte</td><td>Number of packet bytes that follow, which includes 2 bytes for checksum (does not include preceding header or this byte count)</td></tr>
<tr><td>Packet Type</td><td>byte</td><td>0x32 when motors stopped or 0x33 when robot moving.</td></tr>
<tr><td>XPOS</td><td>int</td><td rowspan="2">Estimated robot position coordinates (from 0,0 at robot startup) in millimeters (DistConvFactor[2] = 1.0).</td></tr>
<tr><td>YPOS</td><td>int</td></tr>
<tr><td>THPOS</td><td>int</td><td>Orientation (theta) in angular units of 2π/4096 radians (AngleConvFactor[2] = 0.001534).</td></tr>
<tr><td>L VEL</td><td>int</td><td>Average velocity of left side of robot in millimeters per second (see also VEL and ROTVEL below)</td></tr>
<tr><td>R VEL</td><td>int</td><td>Average velocity of right side of robot in millimeters per second (see also VEL and ROTVEL below)</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>
</table></td></tr>
</table>

[2] 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

		2	*Pioneer sonar array #2 enabled (N/A on Seekur, Seekur Jr.)*
		3	*Pioneer sonar array #3 enabled (N/A on Seekur, Seekur Jr.)*
		4	*Pioneer sonar array #4 enabled (N/A on Seekur, Seekur Jr.)*
		5	*Pioneer stop button pressed (N/A on Seekur, Seekur Jr.)*
		6	*Pioneer E-stall engaged (N/A on Seekur, Seekur Jr.)*
		7	*(unused on Seekur)*
		8	*(unused on Seekur)*
		9	*(unused on Seekur)*
		10	*(unused on Seekur)*
		11-15	*Reserved*
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 Jr. 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.

Robot Motor, Velocity and Position Control:

See **Robots in Motion** discussion below on page 24. To stop the robot, send STOP or VEL and RVEL 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.
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.
STOP	29	None	Stop robot motion.

Configure Acceleration and Deceleration Parameters

See **Robots in Motion** discussion on page 24.

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

Configure Maximum Speeds

See **Robots in Motion** discussion on page 24.

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 34 for details.

Power Control

See **Power Distribution Board (PDB) for Component Power Supply** on page 43.

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.

Command Packet Errors

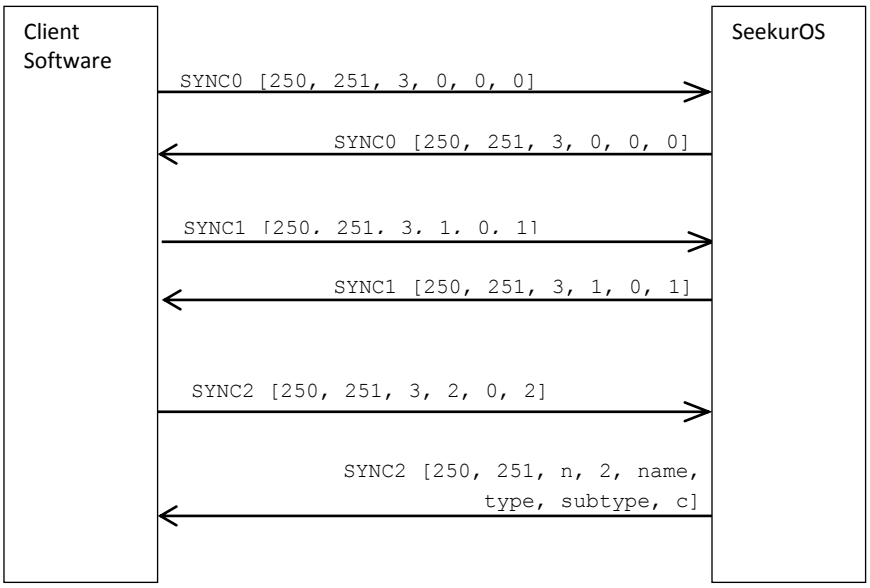
SeekurOS ignores a client command packet whose Byte Count exceeds 204 (total packet size of 207 bytes) or has an erroneous Checksum. The client should similarly ignore erroneous SIPs.

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 Jr, the class is always "Pioneer" and the subclass is always "seekurjr". ARIA uses the subclass to identify the robot type, set default parameters for that robot type, then load the matching parameter file found in the Aria/params directory (for subclass "seekurjr" ARIA loads "seekurjr.p"), followed by a custom robot parameter file corresponding to the robot 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 safety "watchdog" timer expects that, once connected, it receives at least one command packet from the client program every 2 seconds (may be changed via WatchDog firmware parameter), as defined in the robot's FLASH configuration. 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.

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	"seekurjr"
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=9600, 1=19200, 2=38400, 3=56800, 4=115200. Unused on Seekur, Seekur Jr. (always 9600 baud)*
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 after connecting using HOSTBAUD command; 2 if auto-baud implemented; 0 if baud rate cannot be changed.
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	Battery voltage (given here in decivolts) at or below which the robot will enter shutdown
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. NiMH batteries provide state of charge (SOC) level, lead batteries provide voltage measurement only.
LowSOC	Int	Threshold state of charge value at which to signal warning to software (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 ADIS16362 6-axis inertial measurement unit (IMU) may be ordered with the robot instead of the standard single-axis "gyro" accelerometer which provides additional tilt information. SeekurOS automatically uses the horizontal Z rotation axis (yaw) information of either the 6-axis IMU or single axis gyro along with wheel encoder information to automatically calculate the orientation (theta) component of its robot position estimate. The 6-axis IMU also measures acceleration along the X, Y and Z axes, and rotation rate around X (roll) and Y (pitch) axes in addition to rotation rate around the Z axis (yaw), this additional raw data is available for your software to use in your own custom applications and features. Since internal temperature of the sensor can affect its response behavior, temperature data is also provided as a diagnostic tool.

The 6-axis 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. To access the data, use ArSeekurIMU class in ARIA (added in ARIA 2.9.2), or you can request and parse IMU data packets yourself.

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. In your handler code/function, read the following fields, if present, from the packet data.

IMU packet contents:

<table>
<tr><th>Field</th><th>Size (bytes)</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>NumRotAxes</td><td>1</td><td></td><td>Number of axes of speed data available on this IMU (normally 3)</td></tr>
<tr><td>Rotation Rate 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 (roll speed)</td></tr>
<tr><td>AvgY</td><td>2</td><td>Recent average on Y axis (pitch speed)</td></tr>
</table></td></tr>
</table>

			AvgZ | 2 | Recent average on Z axis (yaw speed) If NumAxes is 1: AvgZ | 2 | Recent average on Z axis (yaw speed)
NumAccelAxes	1		Number of axes of acceleration data available on this IMU.
AccelData			Omitted if NumAccelAxes is 0. Otherwise, for each reading (NumReadings): If NumAccelAxes is 3: AvgAccelX | 2 | Recent average acceleration on X axis AvgAccelY | 2 | Recent average acceleration on Y axis AvgAccelZ | 2 | Recent average acceleration on Z axis If NumAccelAxes is 1: AvgAccelZ | 2 | Recent average on Z axis
NumTempAxes	1		Number of temperature sensors (usually one per motion axis)
TempData			Omitted if NumTempAxes is 0. Otherwise, for each reading (NumReadings): 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 (uC) maintenance/configuration serial port on the front of the robot using an adapter to a DB9 serial connector.

seekurParamManager is a text-based console

In Linux, run seekurParamManager using the following commands.

```
/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, type ‘save’ followed by the return (Enter) key.

Calibration & Maintenance

Your SEEKUR JR platform is built to last a lifetime and requires little maintenance.

Tire Inflation

Maintain even tire inflation for proper navigation of your Seekur Jr. We ship with each pneumatic tire inflated to its maximum 28 psi.

Tighten Loose Bolts and Screws

Check for loose bolts and screws periodically, especially wheel bolts and the bolts holding the bearing housing (knuckle). The forces involved with skidding a heavy robot can loosen nuts and screws.

Batteries

It's best to keep Seekur Jr's batteries on its Power/Recharging Accessory when not in use. Fully recharge the batteries for more than 24 hours periodically, particularly before putting the batteries or robot in storage for an extended time.

A status LED indicator is located at the top of each battery module, inside the metal enclosure, as well as LED charge level indicator. When the battery module is fully charged and balanced, the status LED will be steady green when attached to the robot or charger. It will flash amber and/or green when charging. It will be red on error. There is a status button near the LED that will activate the status LEDs when the battery is disconnected from the robot or charger, and a fuse. You can disassemble the battery enclosure if necessary to access this button to help diagnose battery problems.

Before and After Operation

Before using Seekur Jr. outdoors, ensure that both top plate covers are in place using all screws, and that the battery hatch cover is firmly attached. Place connector port covers in place. If the plate covers are frequently removed and replaced, check that their gaskets are intact and replace if damaged.

Secure or remove any and all loose wires or other items from inside the robot before operation.

After using Seekur Jr. outdoors, clean any dirt or debris from the robot body and laser rangefinder lens using a soft damp cloth. Check the interior to ensure no water entered the robot.

Fuses

Fuses are located on fuse/relay boards in the front and rear inside robot near the bottom of the chassis, and inside the robot at the on/off buttons. Fuses are ATO type (available at all electronics and auto parts stores). 30A, 20A and 3A are used. Replace with same rating fuse. Each battery module contains a safety fuse as well. **Contact support for more information on fuses.**

***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.

Factory Repairs

If, after reading this manual, you're having *hardware* problems with your Seekur Jr and you're satisfied that it needs repair, contact us:

http://support.mobilerobots.com/techsupport

Describe the problem in as much detail as possible. Also include your **robot's serial number** (IMPORTANT!) as well as name, email and mail addresses, along with phone and fax numbers.

Tell us your robot's SERIAL NUMBER

Tell us when and how we can best contact you (we will assume email is the best manner, unless otherwise notified).

**Use ADEPT MOBILEROBOTS authorized parts *ONLY*;
warranty void otherwise.**

We will try to resolve the problem through communication. If the robot must be returned to the factory for repair, obtain a shipping and repair authorization code and shipping details from us first.

We are not responsible for shipping damage or loss.

Appendix A

Ports & Connectors

This Appendix contains pin-out and electrical specifications for the external I/O and communication ports and connectors for Seekur Jr.

External Microcontroller Maintenance

13-position Amphenol AL00F11-35S

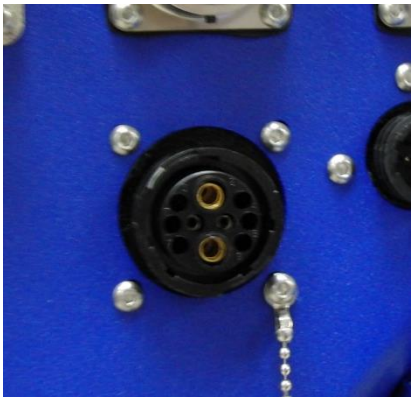
Located on left-front robot exterior.

Pin	Function	Notes
1	NC	No connection
2	RxD from µC	Maintenance connection serial RS-232 RxD
3	TxD to µC	Maintenance connection serial RS-232 TxD
4	TxD to µC	HOST connection RS-232 TxD, only on robots without onboard computer
5	RxD from µC	HOST connection RS-232 RxD, only on robots without onboard computer
6	DSR	Maintenance connection serial RS-232 DSR
7	Signal GND	HOST signal ground. Only on robots without onboard computer
8	Signal GND	HOST signal ground. Only on robots without onboard computer
9	RI	Maintenance connection serial RS-232 RI
10	Signal GND	Maintenance connection serial RS-232 signal ground
11	NC	
12	+12V @ 1A max.	Only on robots without onboard computer
13	Power GND	Only on robots without onboard computer

An adapter cable to DB9 for the firmware maintenance serial connection is provided with the robot. Contact MobileRobots to order this cable if required.

SeekurOS firmware configuration and maintenance may also be done using onboard computer.

External Charging/Power;



10-position CPC

Located on left-rear robot exterior.

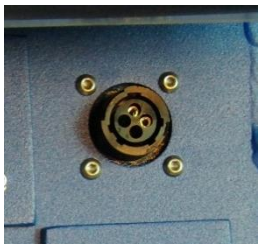
Pin	Function	Notes
1	NC	No connection
2	+ charge power	
3-4	NC	
5	charger detect	
6	charger detect	

7-8	NC	
9	- charge power	
10	NC	

E-stops

4-position CPC AMP 206430-1

Located on upper edge of each robot quarter face.



Pin	Function	Notes
1	e-stop	Short pins 1 and 2 to allow robot to power ON
2	e-stop	
3-4	NC	No connection

Bumpers

8-position CPC

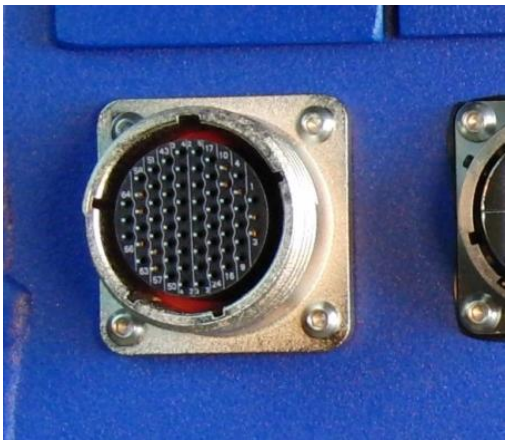
For rear bumpers, located on left-rear quarter.

For front bumpers, located on right-front quarter.

Pin	Function	Notes
1	bumper 0 or 5	Active OPEN
2	GND	
3	bumper 1 or 6	
4	GND	
5	bumper 2 or 7	
6	GND	
7	bumper 3 or 8	
8	GND	

General Purpose External Device Connector

Used for GPS and other external devices; Amphenol 66-pos Amphelite AL00F19-35S



Pin	Function	Notes
1-8	NC	No connection
9	+12V compass	
10	Power GND compass	
11	NC	
12	RxD compass	db9 pin 2
13	TxD compass	db9 pin 3
14	NC	
15	Signal GND compass	db9 pin 5
16-55	NC	

56	+12V GPS	
57	Power GND GPS	
58	NC	
59	RxD GPS	db9 pin 2
60	TxD GPS	db9 pin 3
61	NC	
62	Signal GND GPS	db9 pin 5
63-66	NC	

Ethernet with 12V PoE

If Seekur Jr is ordered with the WRAP wifi access point/bridge (see *Onboard Computers, Internal Network (LAN) and Wireless (wifi) Network*, page 16), then a modular socket provides both ethernet connection to the onboard computer(s) as well as 12V power (power-over-ethernet or PoE).

***Warning**: do not attach other ethernet devices that may be unable to handle 12V power-over-ethernet to this connection. Add a second ethernet port, or disconnect power inside the robot.*

The external connector/passthrough is Amphenol part RJF21B (MobileRobots part ELC1997), also available with a screw-cover as well.

To order additional ethernet connectors from MobileRobots, contact support@mobilerobots.com.

Other External Connectors

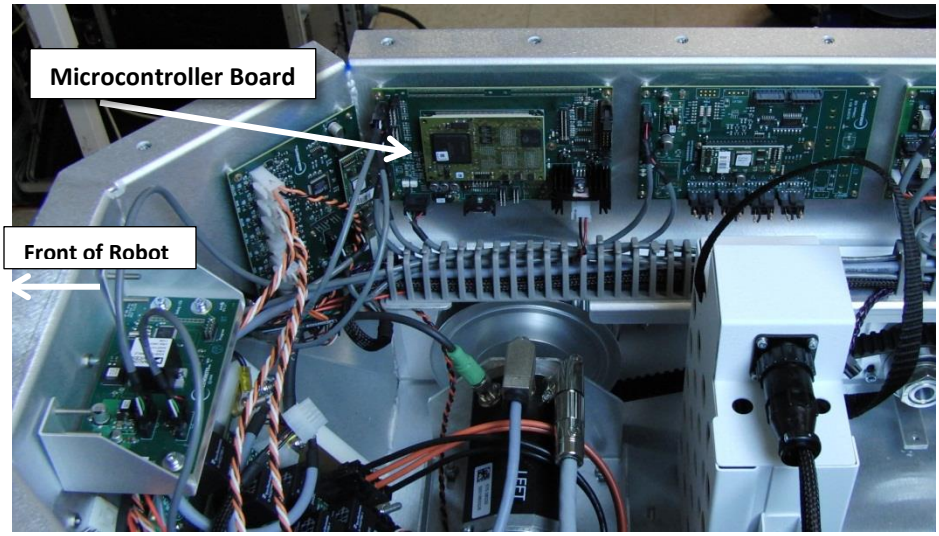
Other external connectors that may be added to Seekur Jr. include USB type A (e.g. Amphenol USBFTV21N), USB A/B adapter, Firewire (Amphenol FWFTV21N or FWFTV6N), DSUB9, and various connectors for power. Contact support@mobilerobots.com if you would like to order any additional connectors or custom cable assemblies from MobileRobots.

Internal Robot Microcontroller Serial Communications

Molex 8-pin Microfit

Pin	Function	Notes
1	TxD	DB9 pin 3
2	RxD	DB9 pin 2
3	DSR	DB9 pin 9
4	RI	DB9 pin 6
5	GND	DB9 pin 5
6	NC	
7	NC	
8	NC	

The robot microcontroller is located in the interior of the right-front quarter panel of the robot.

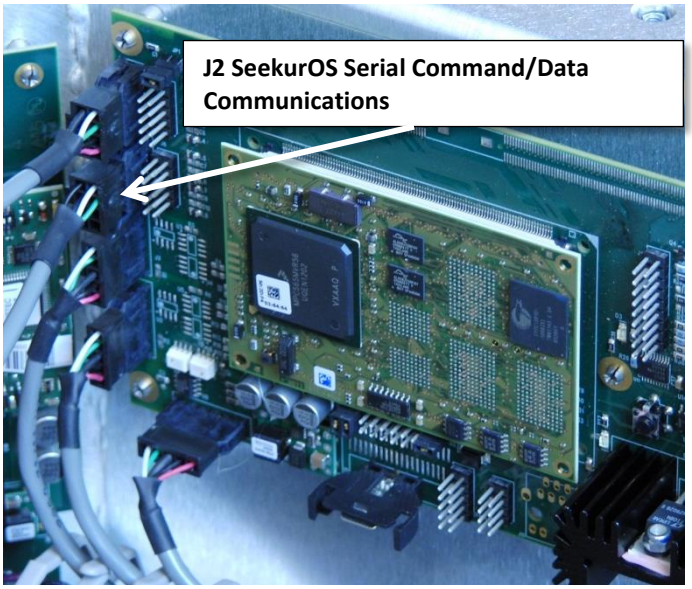


Location of Seekur Jr. microcontroller board inside robot

Communications between the SeekurOS firmware on this microcontroller is performed via RS-232 serial communications through the J2 connector on this board, an 8-pin Molex Microfit connector.

This is normally connected to the primary serial port COM1 on the primary onboard computer via DB9 adapter cable (MobileRobots part #CAB2021)

If the robot does not have an onboard computer, however, then it is instead connected to the microcontroller maintenance connector on the exterior of the robot (See ***External Microcontroller Maintenance*** above for location and pinout.)



Appendix B

Power Distribution Board (PDB) for Component Power Supply

Power to internal and external accessory devices and components are switched and supplied by the power distribution boards (PDB). The PDB is located on the left side of the robot, accessible through the battery hatch. 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:

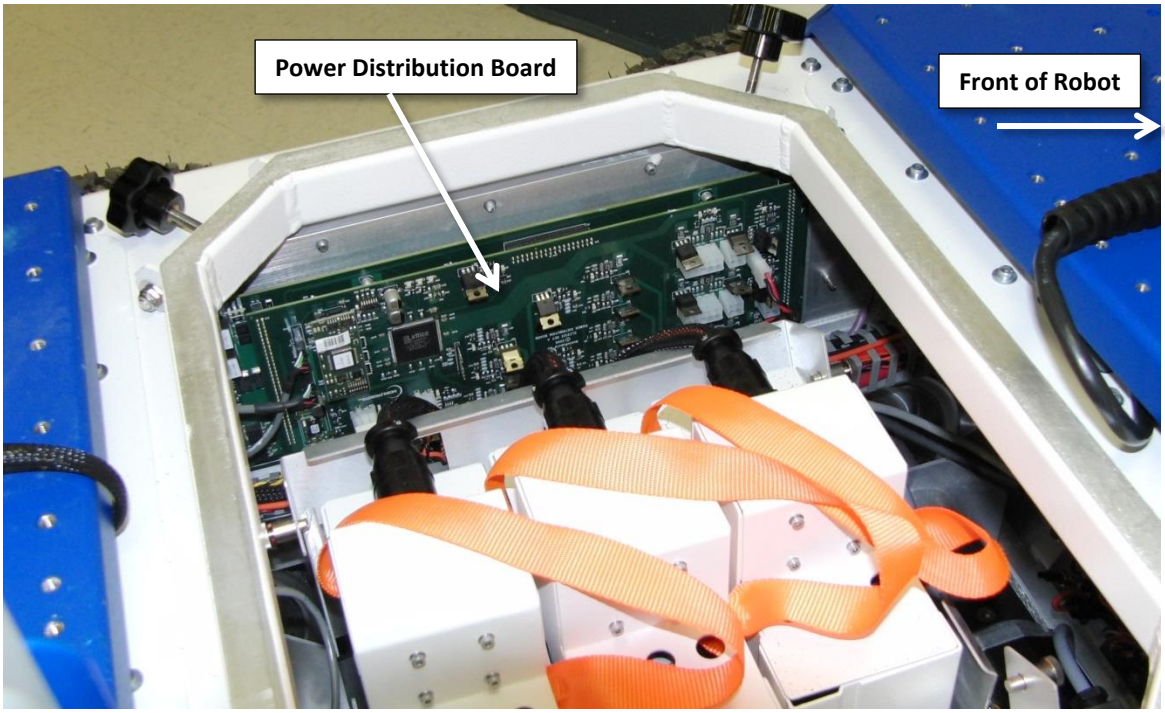
```
116 6 1
```

(Programming with ARIA and the robot command protocol is discussed in more detail in **Programming and Command Protocol** on page 23)

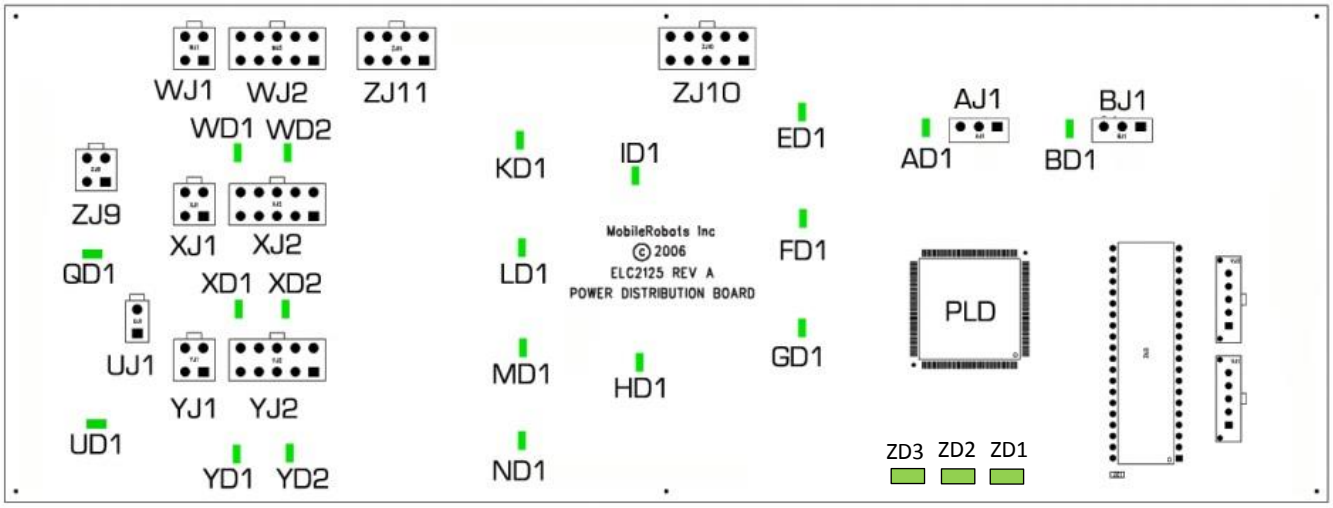
SeekurOS also includes some compatibility with ARCOS and uARCS by recognizing some power switch commands used by those robot firmware variants.

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 Jr. Power Distribution Board (PDB) Location:



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:

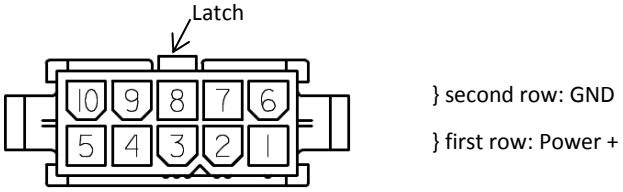
Connector Type	Used By	Mating Connector	www.digikey.com	www.mouser.com
2-position Molex Mini-fit Jr. 39-01-2021	UJ1	Molex Mini-Fit Jr. 2-position receptacle connector 39-01-2020	WM3700-ND	538-39-01-2020
3-position Molex Mini-fit Jr. 39-01-4036	AJ1, BJ1	Molex Mini-Fit Jr. 3-position receptacle connector 39-01-4030	WM18434-ND	538-39-01-4030
4-position Molex Mini-fit Jr. (dual rows) 39-01-2041	WJ1, XJ1, YJ1, ZJ9	Molex Mini-Fit Jr. 4-position receptacle connector (dual rows) 39-01-2040	WM3701-ND	538-39-01-2040
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.

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 total combined 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 total combined 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 total combined 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	Seekur ventilation fan (unused on Seekur Jr.). Note: may be automatically switched on/off by Seekur Jr. during charging and other conditions.	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

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 fuse/relay boards in the front and rear inside robot near the bottom of the chassis, and inside the robot at the on/off buttons. Fuses are ATO type (available at all electronics and auto parts stores). 30A, 20A and 3A are used. Replace with same rating fuse. Each battery module contains a safety fuse as well. **Contact support for more information on fuses.**

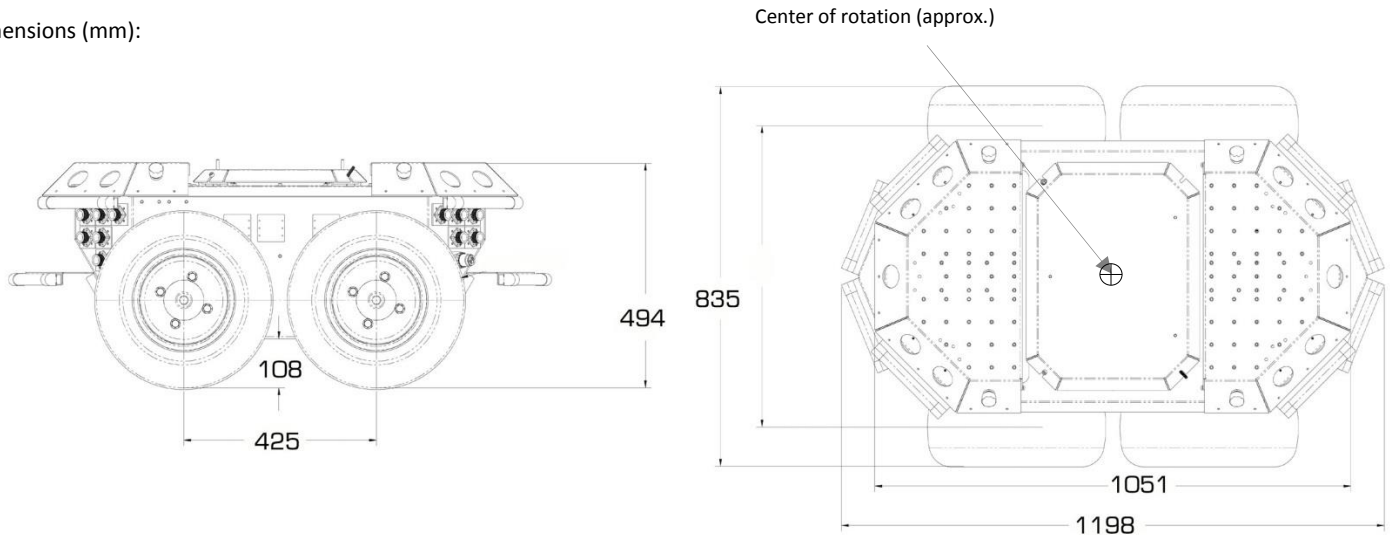
***Warning**: a blown fuse may indicate a fault in wiring, charger, batteries, or other components.*

Contact MobileRobots support for help if you experience blown fuses.

Appendix C

Specifications

Dimensions (mm):



Physical:

Width: 84 cm (32.6 in.)
Length: 1.2 meters (47.2 in.)
Height: 50 cm (19.7 in.)
Wheelbase: 42.5 cm (16.7 in.)
Ground Clearance: 10 cm (4 in.)
Body construction: Aluminum unibody
Robot Weight: 70 kg (661 lbs.) with one battery installed
Turn Radius: 0 cm
Swing Radius: 60 cm (23.6 in.)

Environment and Terrain:

IP Rating: IP-54
Operating Temperature: -5° to 35° C (23° to 95° F)
Ground Clearance: 10 cm (7 in.)
Max. Traversable Step: approx. 20 cm (8 in.)
Max. Traversable Slope: 70% (unladen)
Max. Traversable water depth: 12 cm (5 in.)
Indoor surfaces: tile, concrete, flooring
Outdoor surfaces: Pavement, grass, light snow, light rocky terrain, packed dirt, gravel, sand.
Light rain, snow, dust capable.

Movement:

Max. Forward/Backward/Left/Right Speed: 1.1 m/s (3.9 km/h, 3.6ft/s, 2.5 mph)
Max. Rotation Speed: 100 deg/sec.
Movement Type: differential, skid-steer rotation
Control: velocity command with simultaneous translation and rotation, with configurable acceleration and deceleration.

Payload:

Rated Operating Payload: 40 kg (88 lbs.)

Tires and Wheels:

Tire Type: Pneumatic rubber
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:

Equipment mounting on decks: M6 threaded; ¼ inch holes
Equipment decks attachment: M4 X 12mm with 3mm hex head
Computer Access Panels: M4 X 12mm with 3mm hex head
Battery Hatch: hand knobs

Power:

Run Time: 3 hours approx.
Complete Charge Time: 3 hours approx..
Battery Capacity: 10Ah each module, 30 Ah with three modules
Battery Chemistry: NiMH sealed battery modules
Software-Switchable Power Supplies (some are used by accessory devices, some available for user custom use):

- 2 x 5 VDC, 2 A regulated
- 4 x 12 VDC, 2 A regulated
- 2 x 24 VDC, 1 A regulated
- 2 x 24 VDC, 5 A
- 1 X 24 VDC, 8 A max raw battery

See **Power Distribution Board (PDB) for Component Power Supply** on page 43

Charger Power Supply:

Input: 100-240V AC (10A max)
Output: 27-36V DC 22.2A max

Warranty & Liabilities

Your Omron Adept MobileRobots platform is fully warranted against defective parts or assembly for one year after it is shipped to you from the factory. Accessories are warranted for 90 days. Use only Omron Adept MobileRobots authorized parts or warranty void. This warranty also explicitly *does not include* damage from shipping or from abuse or inappropriate operation, such as if the robot is allowed to tumble or fall off a ledge, or if it is overloaded with heavy objects.

The developers, marketers and manufacturers of Omron Adept MobileRobots products shall bear no liabilities for operation and use of the robot or any accompanying software except that covered by the warranty and period. The developers, marketers or manufacturers shall not be held responsible for any injury to persons or property involving Omron Adept MobileRobots products in any way. They shall bear no responsibilities or liabilities for any operation or application of the robot, or for support of any of those activities. And under no circumstances will the developers, marketers or manufacturers of Omron Adept MobileRobots products take responsibility for support of any special or custom modification to Omron Adept MobileRobots platforms or their software.

Index