



ARNL

Installation & Operations Manual

© 2009 MobileRobots Inc. All rights reserved.

This document, as well as the software described in it, is provided under license and may only be used or copied in accordance with the terms of the respective license.

Information in this document is subject to change without notice and should not be construed as a commitment by MobileRobots Inc.

The software on disk, CD-ROM and firmware which accompany the robot and are available for network download by MobileRobots customers are solely owned and copyrighted or licensed for use and distribution by MobileRobots Inc.

Developers and users are authorized by revocable license to develop and operate custom software for personal research and educational use only. Duplication, distribution, reverse-engineering or commercial application of MobileRobots software and hardware without license or the express written consent of MobileRobots Inc. is explicitly forbidden.

PeopleBot™, AmigoBot™, PowerBot™, PatrolBot®, Motivity™, SetNetGo™, MobilePlanner™, MobileSim™ and MobileEyes™ are trademarks of MobileRobots Inc. Other names and logos for companies and products mentioned or featured in this document are often registered trademarks or trademarks of their respective companies. Mention of any third-party hardware or software constitutes neither an endorsement nor a recommendation by MobileRobots Inc.

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 or fur.
- ✓ 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
- ✓ Getting the robot wet
- ✓ Continuing to run the robot after hair, yarn, string, or any other items have become wound around the robot's axles or wheels
- ✓ Opening the robot with charger attached and/or batteries inserted
- ✓ All other forms of inappropriate operation or care

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

TABLE OF CONTENTS

Chapter 1 Introduction.....	1
THE LRF ADVANTAGE.....	1
BASIC COMPONENTS.....	1
<i>Pioneer and PeopleBot only.....</i>	1
<i>All Platforms.....</i>	2
USER-SUPPLIED COMPONENTS / SYSTEM REQUIREMENTS	2
OPTIONAL	2
ADDITIONAL RESOURCES	2
<i>Newsgroups</i>	2
<i>Support.....</i>	2
Chapter 2 QUICK START	3
INSTALLING THE SOFTWARE	3
ARNL SIMULATION.....	3
<i>Step 1. Run MobileSim</i>	3
<i>Step 2. Run arnlServer.....</i>	4
<i>Step 3. Run MobileEyes.....</i>	5
<i>Gracefully Shutting Down.....</i>	6
EXERCISING ARNL WITH MOBILEEYES	6
<i>Map Display and Operating Parameters.....</i>	6
<i>Going to a Goal</i>	6
<i>Lost from Home</i>	7
Chapter 3 Hardware Installation.....	8
REQUIRED TOOLS AND PARTS:.....	8
HARDWARE INSTALLATION.....	8
<i>Step 1: Power OFF and Remove Batteries.....</i>	8
<i>Step 2: Remove Gripper and Nose.....</i>	8
<i>Step 3a: PTZ Robotic Camera Owners</i>	8
<i>Step 3b: Lift/Remove Console.....</i>	8
<i>Step 4: Attach Signal Cables</i>	10
<i>Step 4: Attach Signal Cables</i>	11
<i>Step 5: Connect Power.....</i>	11
<i>Step 6: PTZ Robotic Camera Owners</i>	11
<i>Step 7: AT Owners</i>	12
<i>Step 8: Reassembly.....</i>	13
<i>Step 9: LRF Assembly.....</i>	13
<i>Step 10: Final Assembly.....</i>	13
OPERATING AND TESTING THE LRF	14
<i>Basic Functionality</i>	14
<i>ARIA Demo</i>	14
Chapter 4 ARNL Mapping	16
THE 2D SCAN.....	16
<i>Step 1. Prepare the space.</i>	16
<i>Step 2. Install a joystick.</i>	16
<i>Step 3. Invoke Map Creation with MobileEyes.</i>	16
<i>Step 3alt. The sickLogger Alternative.....</i>	17
<i>Step 4. Joystick the robot and set goals.</i>	17
<i>Step 5. Finish the 2d scan.</i>	17
MANAGING MAPS WITH MAPPER3.....	17
<i>Points and Lines</i>	17
<i>File Management.....</i>	18
<i>Converting and Cleaning a 2d Scan into a Map.....</i>	19
<i>Editing, Homes, and Docks.....</i>	20
Appendix A.....	21
Warranty & Liabilities.....	22

Chapter 1 Introduction

Congratulations on your purchase and welcome to the rapidly growing community of researchers, developers and enthusiasts of MOBILEROBOTS intelligent robots. This *Advanced Robotics Navigation and Localization Installation and Operations Manual* provides both the general and technical details you need to install and operate a laser range finder (LRF) with Advanced Robotics Navigation and Localization (ARNL) software on your Pioneer 2/3-DX, -AT, PeopleBot™, PowerBot™ and PatrolBot™ MOBILEROBOTS platforms.



Figure 1. Pioneer 3-DX, PatrolBot and PeopleBot with laser range finders.

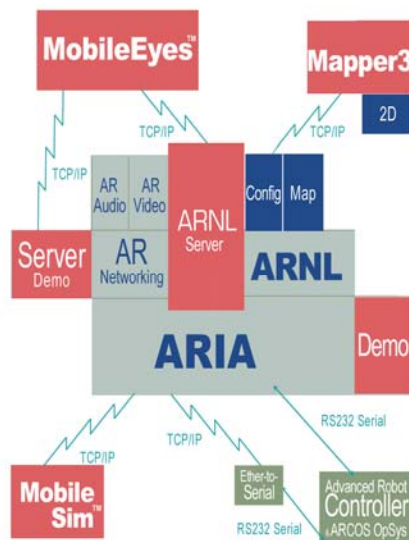


Figure 2. The complete localization and path planning solution.

For your Pioneer or PeopleBot, the laser integration hardware kit contains the mounting bracket and power and signal connectors you need to connect the SICK LMS200 LRF onto your platform. PatrolBot's and PowerBot's LRF require factory installation.

ARNL is a complete localization and autonomous navigation solution, as well as a comprehensive suite of development tools and example applications by which, with your LRF-enabled robot, you create and perform highly advanced mobile robotics tasks. ARNL also provides the tools you need to quickly and easily create, edit, and deploy the maps and floor plans required for those advanced robotics applications.

The LRF Advantage

Within a single plane, the precision of the SICK LRF is some 25 times that of other common range-finding sensors: 180 readings in 180 degrees versus one reading in eight degrees for a sonar disc, for example. Moreover, because the laser beam is highly focused and not readily distorted or absorbed by the reflecting medium, the precision of range-finding with the laser is much superior to sonar with many fewer false-positive readings.

A laser can sense objects much further away, too: out to 10-50 meters, while sonar are limited to 3-6 meters. Since rooms, and especially commercial spaces commonly are eight or more meters square, a sonar-based robot can easily get into voids wherein reflecting objects like walls are outside their detection limits. Without feature clues, navigation becomes severely imprecise; lost in space, so to speak. The precision and range of a laser, on the other hand, make it possible to achieve nearly error-proof localization and very fine gradient-based navigation.

The one disadvantage of the laser is its scanning height – a thin plane of less than one degree. Sonar cone of operation provides broader sensing for proprioception, such as acute obstacle avoidance, which is why MOBILEROBOTS platforms operate best when equipped with both sonar arrays and an LRF.

Basic Components

Pioneer and PeopleBot only

- ✓ One SICK LMS200 LRF (optional)
- ✓ One mounting bracket with power switch and USB port
- ✓ USB joystick
- ✓ Platform-specific cables (must specify robot and onboard PC model)

Introduction

All Platforms

- ✓ CD with ARNL and supporting software
- ✓ This manual also in PDF on the CD-ROM, as well as embedded documentation

User-Supplied Components / System Requirements

- ✓ 1G or faster, Windows or Linux PC (either onboard robot or piggyback laptop) with 32 MB RAM, >2 MB free hard-disk storage and one RS-232 serial interface

Optional

- ✓ Wireless Ethernet

Additional Resources

New MOBILEROBOTS customers get three additional and valuable resources:

- ✓ A private account on our support Internet website for downloading software, updates, and manuals
- ✓ Access to private newsgroups
- ✓ Direct access to the MOBILEROBOTS technical support team

Newsgroups

We maintain several email-based newsgroups through which robot owners share ideas, software, and questions about the robot. Visit the support <http://robots.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, suggestion, 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 pioneer users**>

Access to the `pioneer-users` e-mail newslit 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 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://robots.MobileRobots.com/techsupport>

Please include your robot's **serial number**—we often need to understand your robot's configuration to best answer your question. Your message goes directly to the MOBILEROBOTS technical support team. There a staff member will help you or point you to a place where you can find help. 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.

¹ Note: Leave out the `-requests` part of the email address when sending messages to the newsgroup.

Chapter 2 QUICK START

You don't need an LRF or robot to operate MOBILEROBOTS' ARNL software. ARNL lets you connect a client, such as MobileEyes™, with a MobileSim™-simulated MOBILEROBOTS server that has a simulated SICK LMS200 sensor. Accordingly, without fiddling with hardware you can exercise all of ARNL's laser-enabled robot control, localization and navigation, as well as the many other MOBILEROBOTS software features of the Advanced Robotics Interface for Applications (ARIA), ARNetworking and other MOBILEROBOTS software that come with your platform.

In fact, we recommend that you install ARNL, MobileSim and MobileEyes software and familiarize yourself with their components and interfaces before you install and work with the hardware. That knowledge will be most useful when you get ready to test and operate the robot with its LRF.

Installing the Software

If your robot came with an LRF and onboard PC, you do not need to install ARNL or any other MOBILEROBOTS software there—they're already installed on the robot. Otherwise, or if you want to run the software on another PC, you need to install at least three distributions: MobileSim, MobileEyes and ARNL. Find them on the CD that comes with your robot and LRF or download fresh versions from our support website:

<http://robots.MobileRobots.com>

ARNL comes in two packages: ARNL itself and BaseARNL. BaseARNL includes ARIA and ArNetworking, as well as the localization and navigation software in common with SonARNL and mOGS. Unlike earlier versions, ARNL and SonARNL may be co-installed. The only thing to make sure is that the version numbers sync.

For Windows, ARNL and BaseARNL come as self-extracting archives (*.exe executables) and typically get put into their respectively named directories under the MobileRobots\ main directory; e.g., C:\Program Files\MobileRobots\Arnl. Shortcuts to the programs and files also get put into Start:Programs:MobileRobots.

With Linux, you'll need to be root or otherwise have superuser privileges in order to install the distributions.² ARNL, for example:

```
rpm -ivh ARNL-1.7.0-0.i386.rpm BaseArnl-1.7.0-0.i386.rpm (RedHat)
dpkg -i libarnl_1.7.0_i386.deb arnl-base_1.7.0_i386.deb (Debian)
```

The software get put into /usr/local/Arnl.

ARNL Simulation

The simulated ARNL robot has three parts: MobileSim robot simulator, ARNL's arnlServer and the MobileEyes client. Start them up in that same order on one or more networked computers.

Step 1. Run MobileSim

Load a simulated world and the simulated robot into MobileSim from its start-up dialog, or by explicit declaration as arguments on the start-up command line. The default is a Pioneer 3-DX simulated robot. For example:

Linux

Launch MobileSim from an X-Windows terminal (X required):

```
/usr/local/bin/MobileSim -m columbia-OL.map -r p3dx &
```

² Be careful to assign useful write as well as read permissions in the respective directories. Later, when making a map, for example, the user will need to be able to write to /usr/local/Arnl/examples. Ditto for developing new versions of arnlServer and other ARNL applications.

Windows

Select MobileSim in Start:Programs:MobileRobots:MobileSim and load the map when requested on start up. Otherwise, use a command window (Start:Run:cmd.exe):

```
cd C:\Program Files\MobileRobots\MobileSim
MobileSim -m columbia-OL.map -r p3dx
```

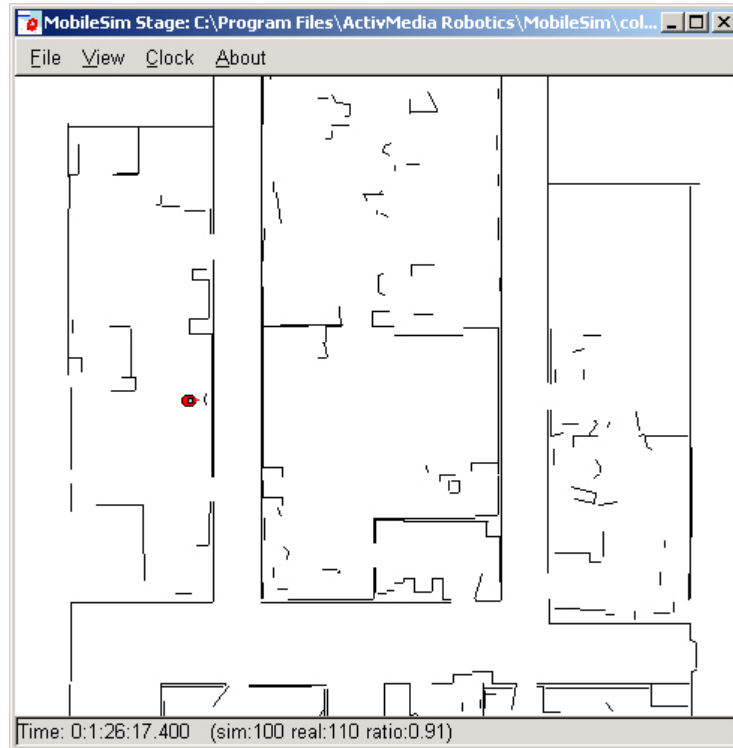


Figure 3. The MobileSim-simulated Pioneer 3-DX at its auto-recharging station in the columbia-OL map.

Step 2. Run arnlServer

All MOBILEROBOTS platforms operate in a client-server environment. ARNL comes with a demonstration server called arnlServer. It is through server applications like arnlServer that clients like MobileEyes get monitor and control access to the robot and its accessories, including the LRF, as well as to all the many features and functions of MOBILEROBOTS software, including ARIA, ARNL and ARNetworking.

Like any ARIA-based application, arnlServer takes several arguments that direct its connection with the robot and with the LRF (see Appendix A for details).³ Also, arnlServer uses parameter files that specify the operating characteristics for the connecting robot and its accessories like the LRF (Arnl/params/p3dx.p, for example), as well as for ARNL itself (Arnl/params/arnl.p). You may modify these files or use the parameter names with a single dash ("-") prefix followed by the new value (-NumSamples 1000, for example), as arguments when starting arnlServer to override the params file's defaults. Once running, clients like MobileEyes also may send commands to arnlServer that change these and other operating parameters.

Finally, arnlServer needs a map.⁴ An ARNL map defines not only the robot's operating space for localization and navigation, but also defines goals like Dock for the auto-recharging station accessory and other predetermined destinations. See subsequent chapters for Mapper3 details.

³ ARIA also lets you embed startup options in a text file /etc/Aria.args or as an environment variable ARIA.ARGS.

⁴ Use MobileEyes and Mapper3 to make maps of real places. See subsequent chapters for details.

In the `Arnl/examples` directory, we include a demonstration map that matches the simulated world map loaded into MobileSim by default. For the purposes of this *Quick Start* demonstration, let's just use arnlServer's defaults with the sample `columbia.map`.

The arnlServer application is text-based (X not required), so it may be started from any text console, but cannot run in the background.⁵

Linux

```
cd /usr/local/Arnl/examples
./arnlServer -map columbia.map
```

Windows

```
Start:Programs:MobileRobots:Arnl:ArnlServer
```

Or from a command window:

```
cd C:\Program Files\MobileRobots\Arnl\bin
arnlServer -map columbia.map
```

ARIA, through ARNL's arnlServer, prints out a lot of useful connection details on the console, including robot type and networking information. Read it, but please don't weep—it should connect well with the MobileSim-simulated Pioneer 3-DX robot and its simulated accessories like the LRF. For example:

```
ArNetServer started on tcp port 7272 and udp port 7272.
Connecting to simulator through tcp.

Syncing 0
Syncing 1
Syncing 2
Connected to robot.
Name: MobileSim
Type: Pioneer
Subtype: p3dx
Loaded robot parameters from p3dx.p
ArSick: Connected to simulated laser.
Mode stop added as default mode
Initializing log from config
Loaded map file 'columbia.map'
Directory for maps and file serving: C:\Program
Files\MobileRobots\ARNL\examples\
See the ARNL README.txt for more information
Checking default on stop mode
Activated stop mode as default
Start localization at 0 0 0.0
Home pose set to 34 119 -4.6
Localized robot (using home position) to 14 213 0.4 with score 0.994475 (min
0.3).
To exit, press escape.
```

Step 3. Run MobileEyes

This one is easy: Simply launch MobileEyes and from its startup dialog specify the IP address or hostname of the computer that is running arnlServer. The hostname `localhost` or IP address `127.0.0.1` usually works if arnlServer is running on the same PC as you launch MobileEyes.

Linux

```
cd /usr/local/MobileEyes/bin
./MobileEyes &
```

Windows

```
Start:Programs:MobileRobots:MobileEyes:MobileEyes
```

⁵ Use the Linux tool 'screen' in order to launch arnlServer and eventually disconnect telnet from the robot.

If you hadn't specified a map when starting up `arnlServer`, then you weren't following our directions or you use Windows. That's okay—MobileEyes warns you that there is no map when first connecting with the robot server and gives you the opportunity to load one then and there. However, if you do select a map this way, MobileEyes automatically inserts it into `arnl.p` for future sessions. Thereafter, use another map either by editing `arnl.p` or through MobileEyes' `Tools:Robot Configuration` dialog.

Gracefully Shutting Down

You need not do so immediately, of course, but now is a good time to know how to gracefully shut down the demonstration: In the opposite order of starting up. Click MobileEyes' close box (upper right-hand corner of the window) or choose `Quit` from its `Robot` main menu.

In the `arnlServer` console, press the `Esc` key to quit the program (includes automatic disconnection from the robot and LRF). And, finally, click MobileSim's close box (upper right-hand corner of the window) or choose `Quit` from its `File` main menu.

Exercising ARNL with MobileEyes

MobileEyes has numerous features, some of which we demonstrate briefly here. We encourage you to read the `README` that comes with the program and to investigate its online `Help` menu and try out its many other functions on your own.

First notice that, upon initial connection, `arnlServer` sends its map to the MobileEyes client, which subsequently displays it (black lines) around a red icon representing the robot and its location in that map. Use the various MobileEyes menu icons, or click in the map window and use the mouse (scroll wheels rule!) to resize and shift the map for better views.

Map Display and Operating Parameters

The robot icon's relative onscreen dimensions are related to the robot's actual dimensions. These, as well as many other characteristics such as maximum speeds for translation and rotation, are specified in the robot's `params` file (`p3dx.p`, e.g.) and from the robot controller's FLASH-based parameters, and are passed from `arnlServer` to MobileEyes. ARNL's operating parameters are in there, too, including those related to localization and navigation. In MobileEyes' `Tools` menu, you can alter many of these characteristics. For instance, use a different map by changing the map name in the `Tools:Robot Configuration` option.

Also, blue dots represent the nearly instantaneous LRF range readings and small, light-blue triangles represent sonar reflections.⁶ These should conform to what the real or simulated robot senses in its world at the time.

Going to a Goal

Using `Mapper3`, you may embed destinations with final headings (goals) into the map. Thereafter, they need only be referenced by name in order to have the robot automatically drive to that position and turn to the prescribed heading once it gets there. MobileEyes parses those goal names and displays them in its `Goals:` window as well as distinctive markers in the map.⁷

The demonstration `columbia.map` has several goals. Double-click, for example, the `Hallway 1` goal in the map, or choose it from the `Goals:` window, to have ARNL plan a path and drive your simulated

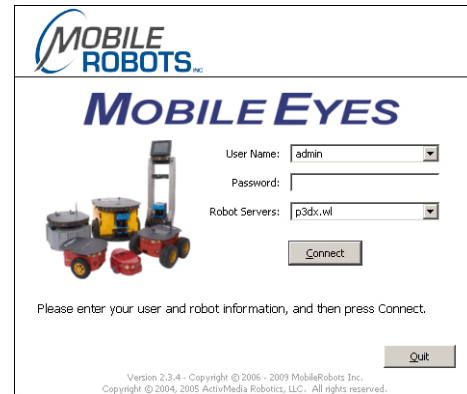


Figure 4. In MobileEyes' startup dialog, enter `arnlServer`'s hostname or IP address, not MobileSim's.

⁶ Turn the various display features on and off from their respective `Map` menu options.

⁷ Choose the `View:Goals/Routes` menu option to display the window in MobileEyes.

robot to it. MobileEyes displays ARNL's planned path to the goal in light blue and refines that displayed path if and when the robot encounters an unexpected obstacle along the way.

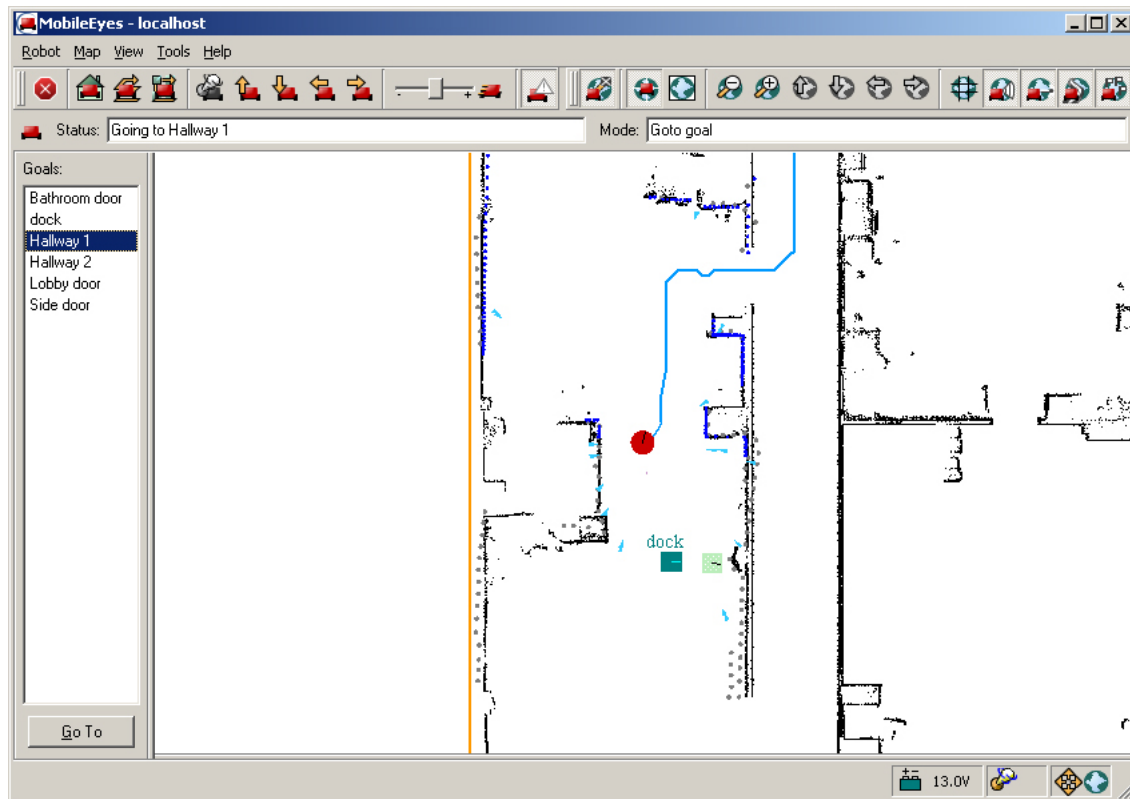


Figure 5. The simulated P3DX on its way to the Hallway 1 goal.

Lost from Home

Home is the special place in the map where the robot expects to be when it first starts up and connects with arnlServer. MobileSim places its simulated robot at the first Home in its map.⁸ If you start the real robot from some place other than Home or somehow displace it manually, the robot might become lost. With MobileSim, move the simulated robot away from its Home position, exit, then restart arnlServer to lose localization.

A good clue that ARNL cannot localize is the LRF and sonar readings don't match the map's features. MobileEyes will complain, too.

If your robot gets lost, orient the map in MobileEyes to where the robot actually is (you may have to click the `Stop` centering the robot icon on the main menubar or deselect the `Center on Robot` option in the `Map` menu). Then, select the `Tools:Robot Tools:Localize to point` option, click and hold the mouse on the spot in the map where the robot really is. Drag the mouse in the direction the robot is facing and release the mouse button.

Repeat as needed until MobileEyes stops complaining and the LRF scans snap onto the map, meaning that ARNL has "found" the robot and is successfully localizing it in the map.

⁸ Home defaults to the map's 0,0,0 position unless you add Home goals with Mapper3. Notice the plural in goals? Yes, you can have several Home goals. See the *ARNL Mapping* chapter for details.

Chapter 3 Hardware Installation

If you order the option at the time of purchase, your robot should arrive with the SICK LMS200 accessory fully cabled and tested, but, for safe shipping, the hardware does need some final assembly. Skip to *Step 9: LRF Assembly* in this chapter. Performance PeopleBot, PowerBot and PatrolBot come with the LRF attached, so you may skip this Chapter altogether.

To add a new LRF to a previously purchased robot, please read through this manual carefully before you attempt first-time installation. If for any reason you do not wish to perform the work yourself, contact us (support@mobilerobots.com) and make arrangements to have the assembly installed at the factory.

Required Tools and Parts:

- ✓ SICK LMS200 Laser Range Finder
- ✓ Pioneer 2/3-DX, -DXE, -AT, or Performance PeopleBot mobile robot
- ✓ Laser Integration Kit (includes mounting bracket, power, signal, and USB cabling)
- ✓ PTZ Robotic Camera owners get extension cables and replacement mounting bracket
- ✓ Hex wrenches that accompanied the robot

Hardware Installation

Perform these steps only when adding a new LRF to a robot. Otherwise, skip to *Step 9*.

Step 1: Power OFF and Remove Batteries

Switch **Main Power** off and remove the batteries from your robot. This eliminates the possibility of inadvertent electrical shorts which may damage your robot or you.

Step 2: Remove Gripper and Nose

If you have the Pioneer Gripper accessory, remove its four retaining screws from the robot's Nose and carefully disconnect it and its power and signal cables.

Use the hex wrenches that came with your robot to remove the two screws that fasten the robot's Nose to the underside of the sonar ring. On earlier models of the Pioneer 2, another screw on the bottom of the robot secures the nose to the body. Remove it, as well.

Once loosened, depending on the model, either lower the nose on its hinge or slide it on the pin guides away from the robot. For the latter operation, disconnect the power to the nose fan and rotate the nose to the left side of the robot, disturbing the other wires and connectors as little as possible.

Step 3a: PTZ Robotic Camera Owners

If you have the PTZ camera accessory on your robot, unplug the control, power, and video signal cables from the rear of the PTZ camera. Then, using the hex wrench that came with your robot, remove the screws that secure the camera mounting plate to the robot's top plate. Set the PTZ camera and its mount aside.

Step 3b: Lift/Remove Console

The signal wire for the LRF attaches to a serial port in the onboard computer, which is just beneath the front of the top plate and behind the nose.

Non-hinged Pioneer 2 models

With older models of Pioneer 2, you have to completely remove your robot's top plate to access the computer and route the LRF signal cables. Using the hex wrenches that came with your robot, remove the seventeen (AT) or fifteen (DX) screws that attach the struts (DX), center console (AT), and top plate. Please be careful to remove only the screws indicated in the Figure nearby for your robot. Removing additional screws will complicate reassembly.

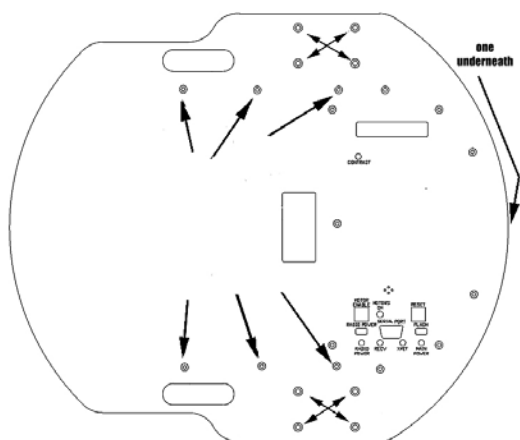


Figure 10. Pioneer 2-DX Top Plate retaining

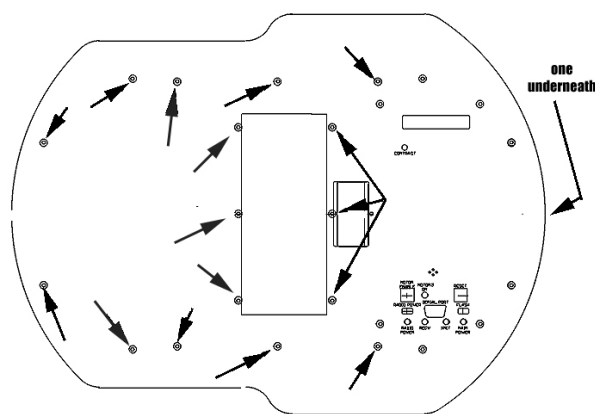


Figure 9. Pioneer 2-AT original Top Plate retaining screw locations

Performance PeopleBot

For the Performance PeopleBot, use the hex wrenches that came with the robot to remove its wheel-strut covers. Then remove the ten screws that retain the lower Plate to the robot body, including the four that secure the Plate on the sides. Lift the Plate off and set aside.

Hinged top-plate Pioneer 2/3 models

For all other Pioneer 2/3-DX and -AT models, simply remove the screws that attach the front portion of the robot's top plate and lift the hinged section, being careful not to pinch any cables at the hinge joint.

Do not remove any other screws as that would complicate reassembly.

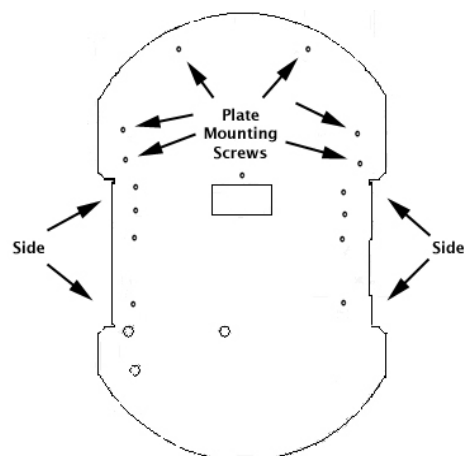


Figure 6. Performance PeopleBot's lower Plate retaining screws

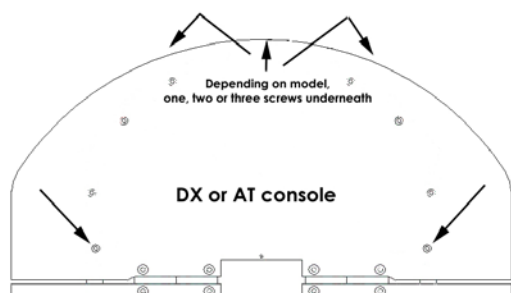


Figure 8. Pioneer 3-DX and -AT console retaining screws. Locations identical with Pioneer 2-DXE and -AT.

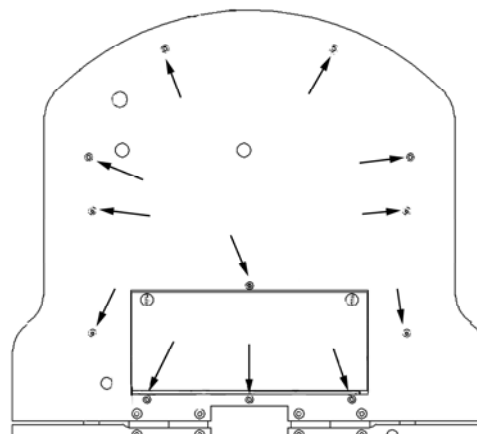


Figure 7. AT rear plate retaining screws

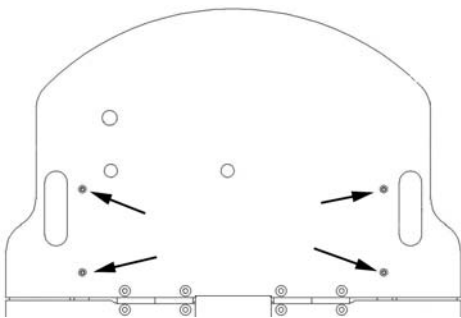


Figure 11. DX rear plate retaining screws

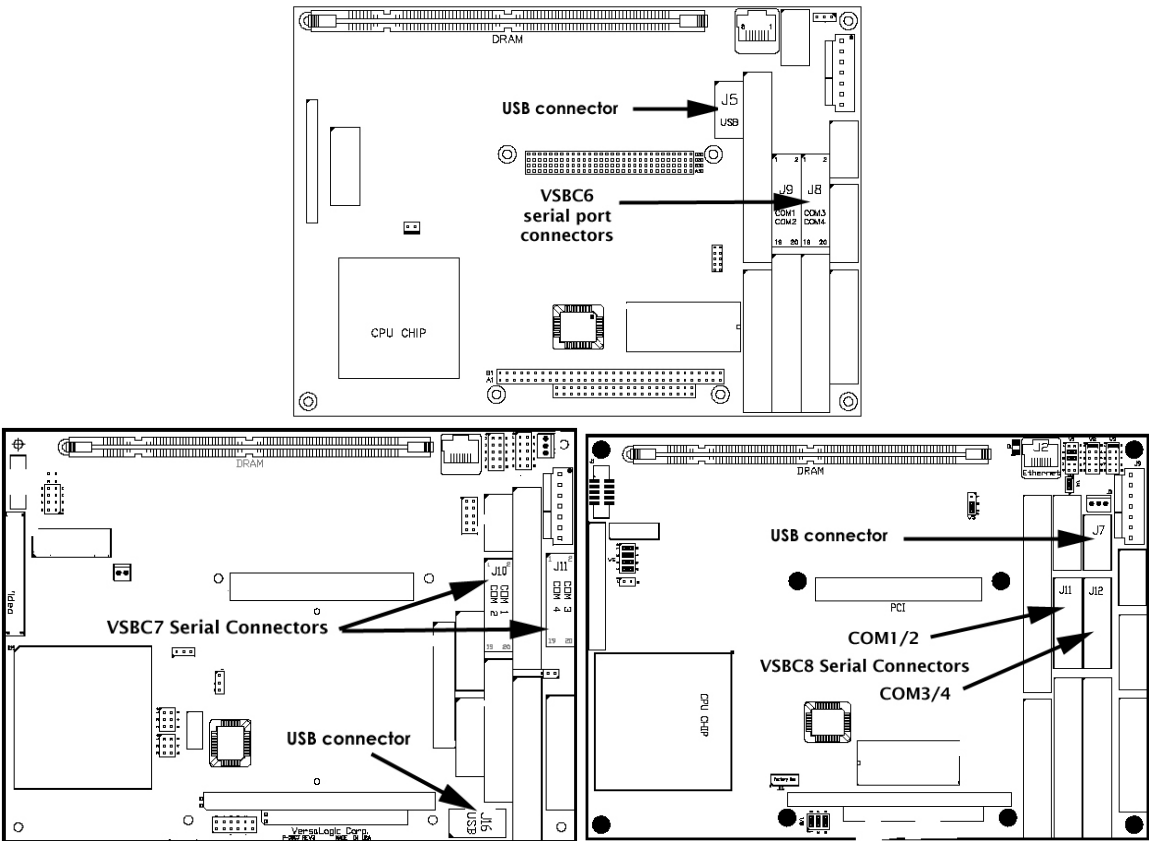


Figure 12. Versalogic VSB6, 7, and 8 serial port connectors

Step 4: Attach Signal Cables

Working through the robot's nose section, locate the serial and USB connectors on the onboard PC. Locations vary, depending on PC model. If you have the Versalogic Cobra, the USB and serial connectors already are attached. If you don't have a Versalogic 6/7/8 PC, contact support@mobilerobots.com for additional details.

The COM1/2 serial connector should already have a cable attached. You may have to unplug one or more of the other cables to access the COM3/4 connector. If you have a DX, DXE, or Performance PeopleBot, route the IDC plug end of the LRF signal cable through one of the slots on either side of the computer. With an AT, route the serial cable over the top of the computer to avoid possible entanglement with the power train. Plug the connector into the COM3/4 PC socket.

Step 5: Connect Power

For newer models of the Pioneer 2/3, re-secure the front plate by reinstalling the screws you had removed in an earlier step and then remove the screws that secure the rear plate to the robot, as indicated in the figures nearby. Remove only those screws indicated.

AT owners should now lift the rear plate away from the robot and set it aside. DX owners should simply tilt up the rear plate on its hinges to access the robot's insides.

If you have a Pioneer 3 or Pioneer 2 *Plus*, locate the LRF power terminal on the Motor-Power board. For older

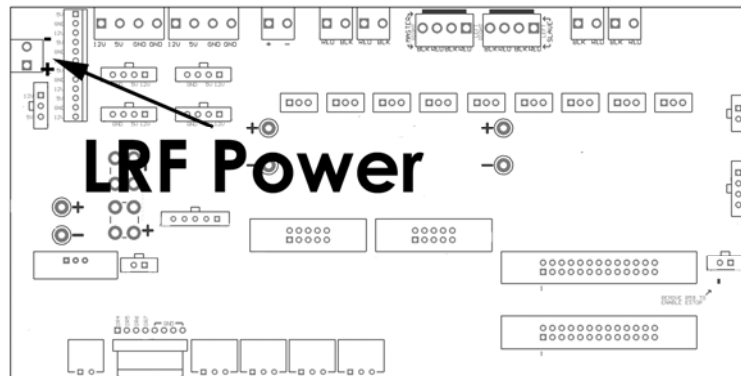


Figure 14. LRF power connector on the Pioneer 2 Plus and Pioneer 3 Motor-Power board

battery connector board at the rear of the battery box. Insert the the LRF power cable (yellow and black wires) into the BATT (+) and GND (-) screw terminals, respectively.

CAUTION
Attach the YELLOW wire to + (BATT) and the BLACK wire to (-) GND.
Polarity is critical!

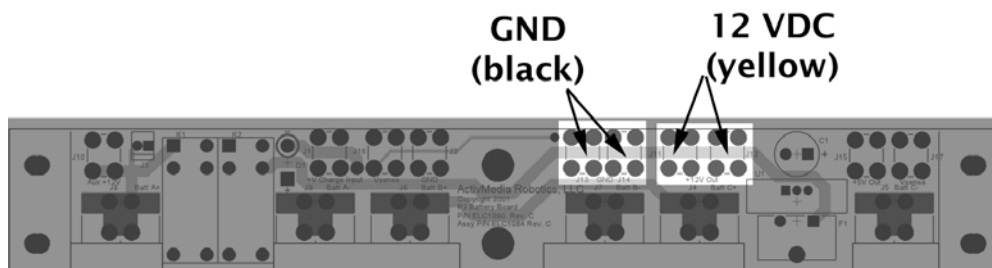


Figure 13. Battery board main power screw terminals

Step 6: PTZ Robotic Camera Owners

Those of you with a DX, DXE, or

AT robot with the PTZ Robotic Camera need to extend the accessory's power and signal cables for attachment at the top of the LRF bracket. Attach those cables now.

For old pre-Pioneer 2 *Plus* robots, unplug the 12-position auxiliary power connector from your robot's Motor-Power board. Have the PTZ Robotic Camera power extension cable ready. Using a pointed instrument, press into and release the locking tabs and slide out the metal clips attached to the two camera power wires (one yellow and one black). Note the wire color and location from which you removed each—you will need to reinstall the extension cable in the same locations.

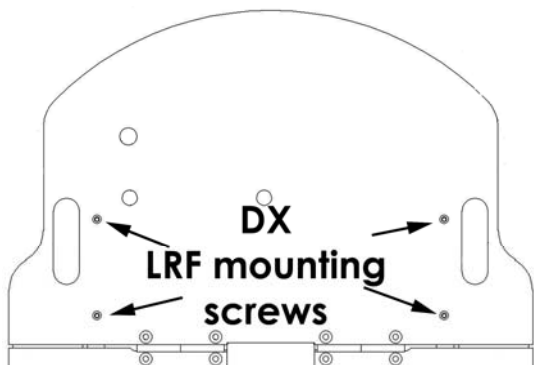


Figure 15. Pioneer 2- and 3-DX LRF bracket mounting points.

Insert the metal clips from the kit's extension power cable into the 12-position auxiliary power connector, with the same locations and polarity as for the original cable. Then insert the original PTZ power cable connectors into the two-position socket—yellow wire to yellow wire and black wire to black wire—at the other end of the extension cable. Note: Slide the latchlock-style clips into the socket such that the locking tab clicks into the locking slot.

Also, locate and unplug your camera's control cable from the AUX serial port of the microcontroller and replace it with the longer cable supplied with your Laser Integration kit.

Step 7: AT Owners

This procedure is **only** for owners of the older Pioneer 2-AT: If still attached, remove the rear of the top plate by removing the four hinge screws along the center of the robot.

Locate the LRF mounting bracket's rear mounting holes to the two existing body-mounting holes on the plate. Mark the bracket's center mounting holes on the plate, then drill and tap for M3 x 0.5 screws.

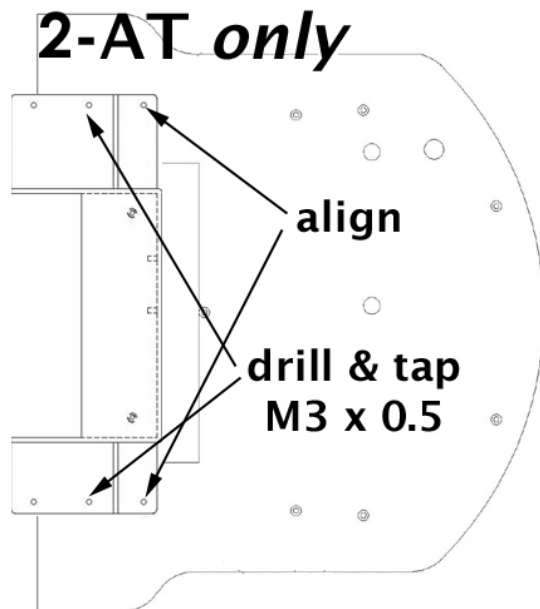


Figure 16. Pioneer 2-AT **only**: align laser mounting bracket to rear plate, then drill and tap the center mounting holes.

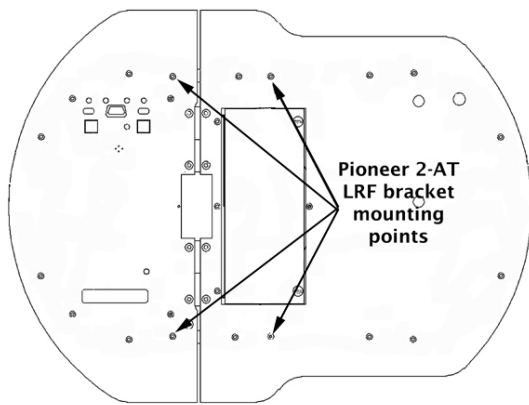


Figure 18. Pioneer 2-AT bracket mounting points same as for Pioneer 3-AT.

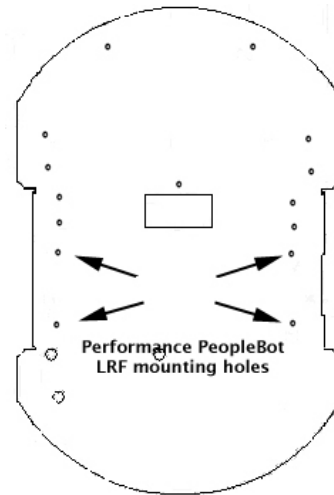


Figure 17. Performance PeopleBot LRF mounting points

Step 8: Reassembly

Carefully collect the LRF signal and power cables, and the PTZ robotic camera cables (if you have the accessory) and feed them through the access portal in the center of the robot's top plate (lower plate on PeopleBot). You may have to remove the plastic plug.

Align and reattach the top plate to the robot's body using the tools and hardware from Step 3. Leave out the four screws for mounting the laser bracket.

Also align and reattach the nose and Gripper assemblies, in the reverse order and using the same tools and screws from Step 1.

Step 9: LRF Assembly

Place the LRF with bracket on the center of the robot's top plate. If you have the PTZ Robotic Camera accessory on the AT or DX/DXE model robot, feed its extended power, control, and video cables up from the bottom between the back of the LRF and its bracket and out through the access hole near the top of the bracket.

Plug the LRF's signal, USB, and power cable connectors with their mates that come up through the access portal and tuck them under and inside between the bracket.

Using the longer 3x10mm screws that come with the integration kit, attach the LRF and bracket to the top plate and body of your robot through the four mating mounting holes.

Attach the power and signal (serial and USB) connectors for the LRF to their mating connectors on the bracket.

Step 10: Final Assembly

PTZ Robotic Camera Owners

Mount the camera and bracket to the top of the LRF bracket.

Insert the video and control connectors, but **not** the power connector, into their respective sockets on the camera.

Turn the robot Main Power on and test the voltage on your camera's power plug with a voltmeter. The center pin of the connector should be +12-14 volts (battery level) and the outer housing of the connector should be ground. If the polarity got reversed when you installed the power extender, you could severely damage the camera in a fashion that will not be covered by its warranty.

CAUTION

Test the PTZ power cable before inserting into the camera.

Performance PeopleBot Owners

Reattach the wheel strut covers.

All robots

Reattach and secure all remaining parts and accessories, including the nose and Gripper as removed in earlier steps.

Switch off all accessories, including the onboard computer, and install your robot's batteries. Phew! Wipe your brow and relax.

Operating and Testing the LRF

Power to the LRF passes through a manual rocker switch on the back of the mounting bracket. When off, the LRF will never get power. When on, the indicator lights on the LRF will blink briefly when you first start the robot, then go off. This happens because the onboard PC ultimately is in control of LRF power—the DTR line of the serial interface toggles power. It takes a moment to reset that line while the computer starts up.

Basic Functionality

Run the Motors Self-Tests (press User Control Panel Motors button twice) to make sure the basic systems are still functional after you attach the LRF. Then switch on and test each accessory as you normally would without the LRF.

See your robot's *Operations Manual* and other systems and accessories manuals for additional detail, as needed.

Switch LRF power from the rear of the mounting bracket.

LRF power also may be under control of the onboard PC. Run ARIA demo to verify operation *before* contacting MOBILEROBOTS support.

ARIA Demo

Running ARNL with MobileEyes ultimately tests your LRF-enabled robot, but there is some setup time involved. See the next chapter. Instead, to quickly test your hardware, we recommend that you use ARIA's demo program. It is a quick and easy tool to test not only your new LRF, but also all of your other accessories.

Install ARIA as needed—it comes preinstalled on all robots shipped with onboard PC, and it can be found on the CD-ROM that came with your robot.⁹ Otherwise, download a fresh copy from our support website, <http://robots.MobileRobots.com>.

ARIA normally installs in `/usr/local/Aria` (Linux) or in `C:\Program Files\MobileRobots\Aria` under Windows. The demo program is in the `examples` directory with its sources and `Makefile` (MS Visual® C++ .sln files, too). Additional resources are in the `src` directory.

Launch demo on the same PC as you have connected with the robot controller and the LRF. For this application, you are connecting with the robot's controller through a serial port, not to the MobileSim simulator. The LRF is connected to another serial port. The defaults should work, but refer to your robot's *Operations Manual* for connection details if it doesn't connect properly.

To exercise the LRF with demo, press the 'L' key. The LRF should power up and flash its indicator lamps. If not, check its on/off rocker switch on the back of the mounting bracket. Alternatively, the LRF may not be connected with the default serial port. In that case, you need to specify it when starting demo, or add that port reference to the robots params file. For example, if the laser is on COM4:

```
./demo -lp /dev/ttyS3
```

When successfully connected and operating, ARIA's demo reports:

```
You are in 'laser' mode currently.
Laser mode connects to a laser, or uses a previously established connection.
Laser mode then displays the closest and furthest reading from the laser.
ArModeLaser is connecting to the laser.
ArSick: waiting for laser to power on.
ArSick: Connected to the laser.
Laser mode has connected to the laser.
Close:      501.99mm  1.93 deg  Far:      3863.23mm -97.26 deg    79 readings
```

The LRF should remain on while you operate other demo features.

⁹ ARIA and ArNetworking come with ARNL, but only their libraries, not their sources or demonstration programs.

You might also test the USB port on the back of the LRF bracket with the USB joystick that came with ARNL. Press the “fire” button #1 to enable JoyDrive mode while operating demo in ‘T’eleoperation mode.¹⁰

When finished testing, press the `Esc` key to quit demo and power down the LRF.

¹⁰ You may have to restart demo if it does not recognize the joystick.

Chapter 4 ARNL Mapping

As you discovered in the *Quick Start* chapter, your LRF-enabled robot with ARNL's arnlServer and MobileEyes software is able to navigate indoor spaces autonomously. But unless you plan to operate your robot only at the Columbia Drive plant of MOBILEROBOTS, you're going to have to make a map of your robot's operating space. This is where 2d files and Mapper3 come in.

Users of MOBILEROBOTS' foundation SonARNL software or even the now ancient Basic Suite know the tedium and frustration of manually drawing maps. With ARNL and Mapper3, you let your robot and LRF do the dirty work.

You start the mapping process by creating a "2d" scan file with MobileEyes and arnlServer or with the sickLogger tool, and by driving your robot around with a joystick in the places where you plan to operate it with ARNL. You then load the 2d file into Mapper3 and, voil  , a map emerges, to which you may edit, add features and goals, indicate forbidden places, and much more.

The 2d Scan

Presuming that you have installed and tested your robot and its attached LRF, connect with MobileEyes and make a 2d scan. The setup is nearly identical with the one you made with MobileSim in the *Quick Start* chapter, though here, you will be connecting ARNL's arnlServer with the real robot and its LRF.

The following instructions presume that you have a PC built into your robot, or that you have a laptop or other PC in which one serial port is connected with the LRF and another with the robot's controller. There are other ways to connect which we leave for you to discover. Read your robot's *Operations Manual* and ARNL documentation for clues.

Step 1. Prepare the space.

Your LRF-enabled robot is going to use the map you are making to figure out where it is and how to get from one place to another. Ideally, the space will be rich in solid and immovable landmarks. Not likely, though.

ARNL is able to maintain localization in very dynamic environments, and you can add and erase important features that move, change, or were otherwise hidden when you first made the map. Nonetheless, it is best to start with a space that is quiet and uncluttered by moveable objects. Onlookers who gather and walk alongside while you make a map are okay, as long as they stay behind the robot out of view of the laser scan, just as you should do while driving, too.

Step 2. Install a joystick.

To make the scan, you are going to drive the robot manually through its operating space. Though you could use the arrow keys on a keyboard to do that, it's much easier to drive with a joystick.

Install the USB joystick that comes with ARNL into the USB port on the back of the LRF mounting bracket. Or insert a MOBILEROBOTS accessory joystick into its socket on the robot. The difference is that the accessory joystick interfaces and drives directly with the robot's controller, whereas the USB one interfaces through ARIA on the PC.¹¹

Step 3. Invoke Map Creation with MobileEyes.

When ready to start the scan, start up your robot and arnlServer.¹² Connect with MobileEyes, and then select its Tools:Map Creation:Start Mapping option. MobileEyes pops up a dialog that requests a name for your scan file, which it then stores, by default, in the *robot's* arnl/examples directory with the suffix ".2d". Provide a pathname to change directories.

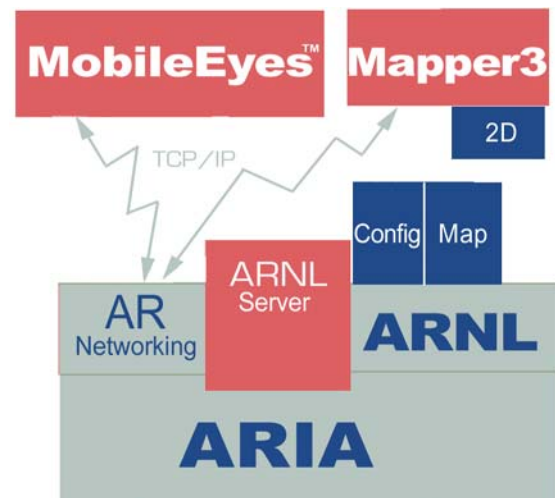


Figure 19. Software components for mapping

¹¹ Restart arnlServer if it does not respond to the USB joystick.

¹² Same procedure as with MobileSim, except don't start MobileSim. Also, some of the arnlServer and LRF startup arguments may apply to your configuration. See Appendix A.

Step 3alt. The sickLogger Alternative

There is an alternative to MobileEyes and arnlServer for making a 2d scan: sickLogger. Like arnlServer, sickLogger is an ARIA program. It is found in the Arnl/examples (Linux) or Arnl\bin (Windows) directory. And like arnlServer, it connects with the robot's controller, lets you drive with the joystick, and accumulates laser scans coordinated with odometry readings into a .2d scan file which you later use with Mapper3 to create a map.

Unlike arnlServer however, sickLogger does not interface with MobileEyes or Mapper3 over the network. So you won't be able to take immediate advantage of their advanced network-file management features. However, sickLogger takes twice as many readings from the LRF to create sometimes better maps. And the .2d files you create with sickLogger are identical otherwise, and you may include them with any others you create, edit, and manage through MobileEyes, arnlServer, and Mapper3. You decide.

Start sickLogger with the same ARIA-related arguments as you would for arnlServer or demo. Also, provide an optional 2d scan path-name as the startup argument; lscans.2d is the default.

Step 4. Joystick the robot and set goals.

Carefully joystick the robot around the space you plan to operate the robot with ARNL. Take your time: the scan file grows by distance traveled, not by time spent traveling. Peek into and around corners and nooks. You are trying to give the map as much detail as you can.

Also, as you drive into rooms, come back around and cross over the path where you entered; go beyond the door and come back again, for example. And come back to and beyond the place where you started the map. These strategies help "close loops" in the map-making process and improve its resolution. (Read the Mapping.txt file in Arnl/docs for more details.)

Of course, avoid dangerous places like stairways—you can block those later with forbidden lines applied from Mapper3. And avoid running into things or places where the tires might slip—that can ruin the map. Also, try and stay behind the robot, out of view of the LRF's 180-degree scanning range so you don't become a member of the map.

As you drive, drop goal markers into the map by pressing the second joystick button. Its best to stop and orient the robot as you want it to be when you later send it to that goal, though you may edit the goal position and pose later with Mapper3.

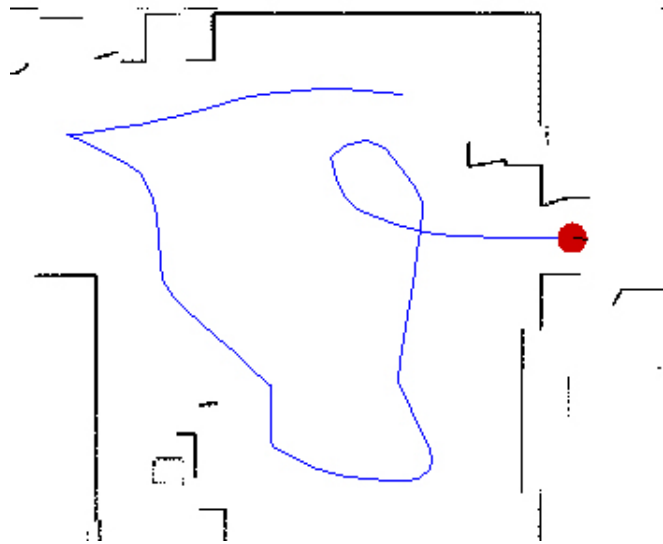


Figure 20. While mapping, drive into nooks and crannies, and cross your loops.

Step 5. Finish the 2d scan.

If you are making a 2d scan using arnlServer, go back to MobileEyes and choose the Stop Mapping option from the Tools:Map Creation menu.

Press the 'esc' key to stop sickLogger and finish the scan. Or reset the robot's controller to break the client-server connection and finish the scan.

Managing Maps with Mapper3

Use the Mapper3 program to transform your 2d scan files into maps, as well as add and edit the map's features, forbidden spaces, and goals. Mapper3, in collaboration with arnlServer, also lets you manage your robot's map files remotely over the network for deployment and use on your robots.

Points and Lines

First, a quick word about the various map types and elements. The 2d scan contains raw laser and odometry readings, as well as positions and poses for the goals that you inserted while making the scan. Mapper3 converts the 2d scan into an occupancy grid of points, cleaning and reorienting them into a best-fit map consisting of lines and points. The 2d scan-embedded goals also get put into their corrected positions in the map.



Figure 21. Maps consist of occupancy-grid points (left) and extracted lines (right).

For localization, SonARNL can only use lines like those you may have drawn yourself with Mapper3-Basic or add with Mapper3. For path-planning, SonARNL uses both points and lines. MobileSim uses points and lines, too, though earlier versions could only use lines. ARNL uses both lines and points for both localization and path-planning.

Both a map's points and lines normally get displayed by Mapper3. To view the differences, select **Lines** or **Points** alone under Mapper3's **Map:Data** menu option. Of course, if you show only points, a Mapper3Basic map consisting of only lines won't show up in Mapper3's map display, and vice versa.

File Management

Mapper3 gives you several ways to access and manage 2d and map files locally and over the network. Click the **Open** icon to engage a dialog in which you choose a 2d or map file to load into Mapper3 from local storage or from a networked filesystem. Similarly, click the **Save** button to commit your work to permanent storage.

Mapper3's **File** menu on the main menu bar has many more file-management features. **New** opens and lets you draw and edit a map. **Open** under **File** works just like the **Open** icon. **Save** and **Save As...** work similarly to store your maps.

Open on robot bypasses the hassles of networking filesystems and connects you with a selected robot's 2d and map files through arnlServer.¹³ **Save on Robot** uses the same mechanism, but in reverse to store your Mapper3 work.

The **File:Download/Upload** option also bypasses networked filesystems and connects you via arnlServer with a specific robot's files. The option provides a dialog in which you select and transfer nearly any type of file—2d, map, or otherwise—to and from the robot and your networked filesystems. It doesn't open any of the files for processing by Mapper3, though.

For example, you might want to convert the 2d scan file you just made with MobileEyes and arnlServer into a map. That 2d file, by default, got stored in the `Arnl/examples` directory on your robot. You could access and run Mapper3 on the robot over the network and convert the file that way. But, instead, you want to operate Mapper3 from your desktop PC.

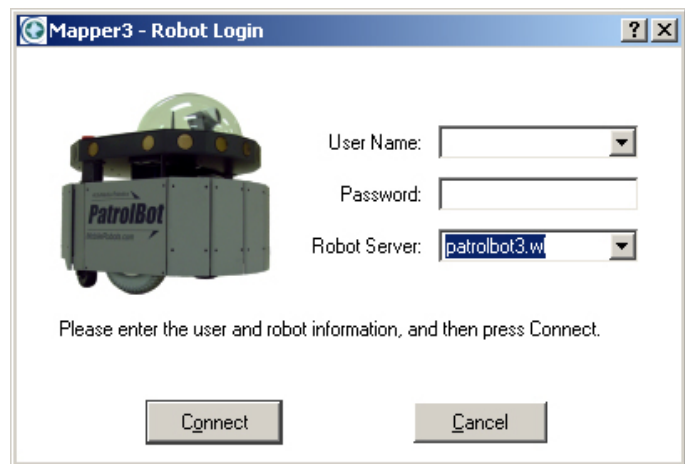


Figure 22. Access maps on your robot with Mapper3's **Open on robot** and **Save on robot** options.

¹³ Of course, your robot needs to be running arnlServer or similar ARNetworking-based application and have access to your network.

Accordingly, use Mapper3's **File:Open on robot** option to connect with your robot through arnlServer over the network, and select the 2d file from the file-selection dialog. Mapper3's scan-to-map conversion process then happens on your PC, and you may edit the map there, too. Put the resulting map file onto your robot using the **File:Save on Robot** option.

Note, too, that a map created on one robot may be used without modification on any other ARNL- or SONARNL-enabled robot.¹⁴ Use Mapper3 to connect and distribute maps to your fleet.

Converting and Cleaning a 2d Scan into a Map

Simply opening a 2d file initiates conversion of it into a map by Mapper3.¹⁵ At the beginning, a special **Scan Tools** toolbar appears and a robot icon automatically begins retracing your robot's path as you drove it around with the joystick while creating the scan. Around the robot icon, walls, hallways, doors, and other features develop on screen.

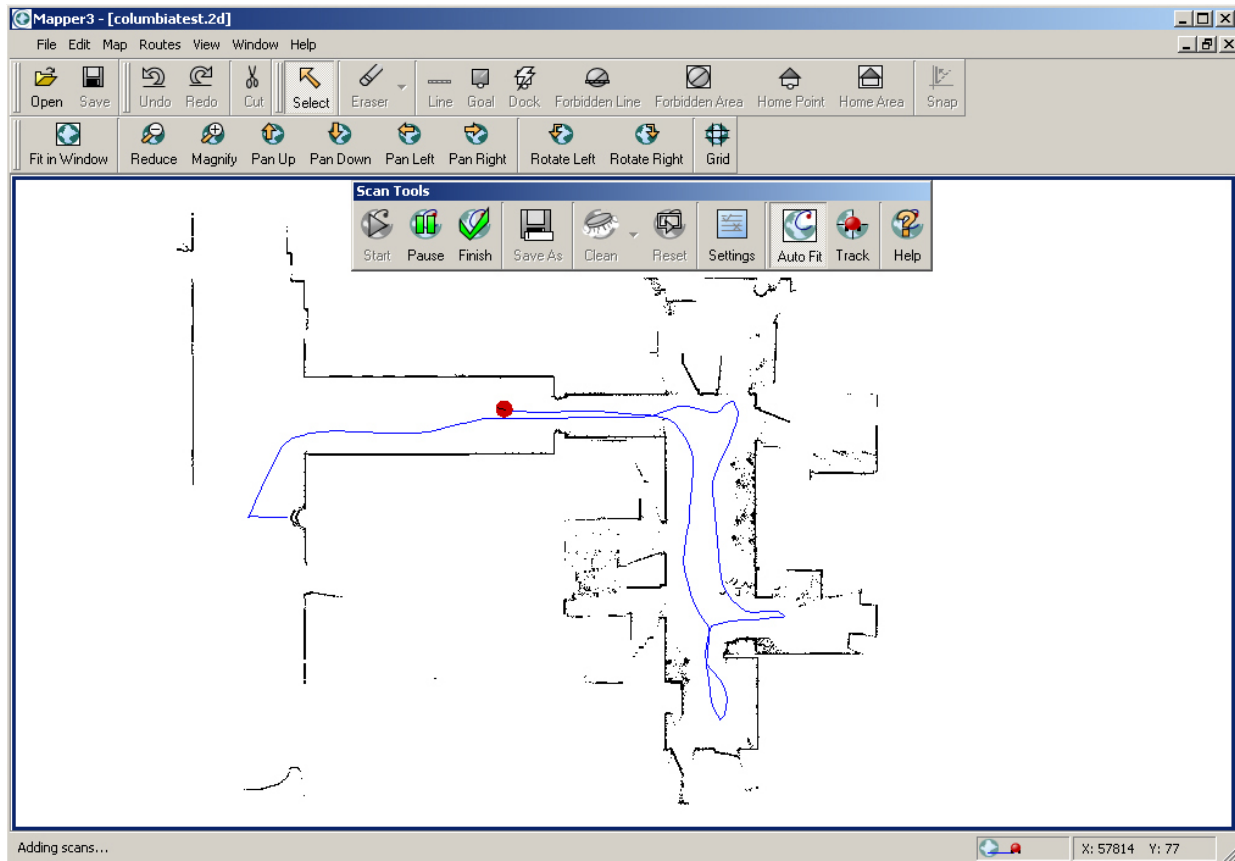


Figure 23. A 2d scan file during conversion into a map

Once all of the scan data have been added to the map window, a registration phase begins. During this time, Mapper3 compares the data and adjusts the map elements to best represent the real space. Closing loops is one mechanism of registration.

When registration is complete, the robot icon and its trail disappear from the display, and the scan icon on Mapper3's status bar stops animating. Unless disabled from the **Scan Tools Settings** options, Mapper3 then proceeds automatically to clean the map.

Cleaning is the process by which extraneous points get removed from the map, such as those created by a black cat that walked in front of the robot while you were creating the 2d scan. Use the **Scan Tools** toolbar **Clean** option to adjust the parameters which influence cleaning and click its icon to manually re-clean the map.

¹⁴ The only caveat is that the ranging sensor must "see" nearly the same elements of the map: A laser mounted one meter off the floor, for instance, might have a different view and, consequently, requires a different map of the same place than for a robot with laser mounted 4 cm off the floor.

¹⁵ You may disable this and other automated 2d conversion-related processes under **Scan Settings** of the **Scan Tools** toolbar.

From the `Scan Tools` toolbar `Settings` option, also adjust parameters which influence other map-making processes, such as maximum laser range and cleaning grid size, to best reproduce your robot's operating environment. Please consult Mapper3's `Scan Tools Help` menu for much more detail.

When you are finished converting and cleaning the map, click the `Finish` button. `Finish` initiates the formation of lines from an analysis of the current points in the map, as well as computed adjustments to the map's resolution. These can be processor and time consuming, so once again, please be patient. When truly finished, the goals that you designated during the scanning process will now appear in the display.

Editing, Homes, and Docks

Mapper3 provides a wide range of tools to massage your map into the best representation of your robot's real operating environment. Add lines and erase lines and points, for instance, to update areas that have changed in your map. Mapper3 also lets you designate and modify places that your robots may go (goals) or are forbidden to cross (forbidden lines and areas). Mapper3 even lets you stitch maps together into larger maps (`File:Insert Map...`).

Most of a map's elements and the tools to manage them are obvious for even the casual user, so we don't go into detail here.¹⁶ Please consult Mapper3's online `Help` for additional details, and read its accompanying `README.txt` file for updates.

Some elements—Origin, Home, and Dock—need further explanation. Origin is $X,Y,Theta = 0,0,0$ in 2-D space. The map radiates out around the origin. It may be the place where you started making your 2d scan if you hadn't moved your robot after you started it up. It is the original Home.

Home, as described earlier in the *Quick Start* chapter, is where ARNL expects the robot to be when it starts up and, consequently, can localize initially. You can have more than one Home in the map to match all the places where your robot may be stored. In fact, the map's Origin is probably not where your robot really is most times it starts up, so please add one or more Homes to your new maps.

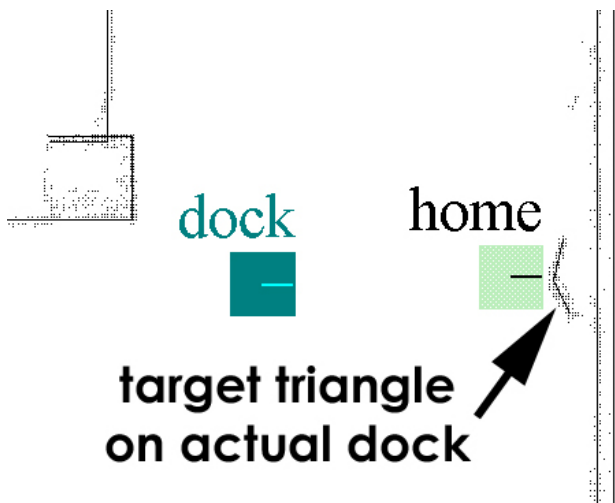


Figure 24. Place Dock goal one to three meters away from actual auto-recharging accessory.

ARNL examines each Home on start up and choose the one with the highest probability of matching its location in the map as its own start-up position for localization.

Dock is special in that it designates the place **one to three meters away from** and the respective heading **facing at** the auto-recharging station accessory that your robots may use to automatically recharge their batteries. When sent to an auto-recharge station, your robot's ARNL navigates to the Dock goal and then uses the station's special shape to guide it the remaining distance into the charging contacts. That's why the Dock goal is not actually on the station, though you should put a Home there if you plan to shut down and store the robot on the dock.

¹⁶ Zoom and shift your map with the mouse scroll wheel and right button. Or use Mapper3's toolbar buttons.

Appendix A

Table 1. arnIserver's ARIA-based robot connection start-up options

-remoteHost <Host Name or IP> (abbreviated -rh)	Connect with robot through a remote host over the network instead of a serial port; requires special serial Ethernet hardware or IPTHRU software mediation.
-robotPort <Serial Port> (abbreviated -rp)	Connect with robot through specified serial port name; COM3, for example. COM1 or /dev/ttyS0 is the default.
-robotBaud <baudrate> (abbreviated -rb)	Connect with robot using the specified baudrate; 19200 or 38400, for example. Default is 9600.
-remoteRobotTcpPort <Number> (abbreviated -rrtp)	Remote TCP host-to-robot connection port number; default is 8101.

Table 2. Some LRF-related, ARIA-based arnIserver startup arguments

-laserPort <Serial Port> (abbreviated -lp)	Connect with the LRF through specified serial port name; COM2, for example. Default in robot's param file.
-laserBaud <baudrate> (abbreviated -lb)	Connect with LRF using the specified baudrate; 19200 or 38400, for example. Default is 9600.
-remoteLaserTcpPort <Number> (abbreviated -rltp)	Remote TCP host-to-laser connection port number; default is 8102.

Warranty & Liabilities

Your 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 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 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 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 MOBILEROBOTS product take responsibility for support of any special or custom modification to MOBILEROBOTS platforms or their software.



10 Columbia Drive
Amherst, NH 03031
(603) 881-7960
(603) 881-3818 fax
<http://www.MobileRobots.com>