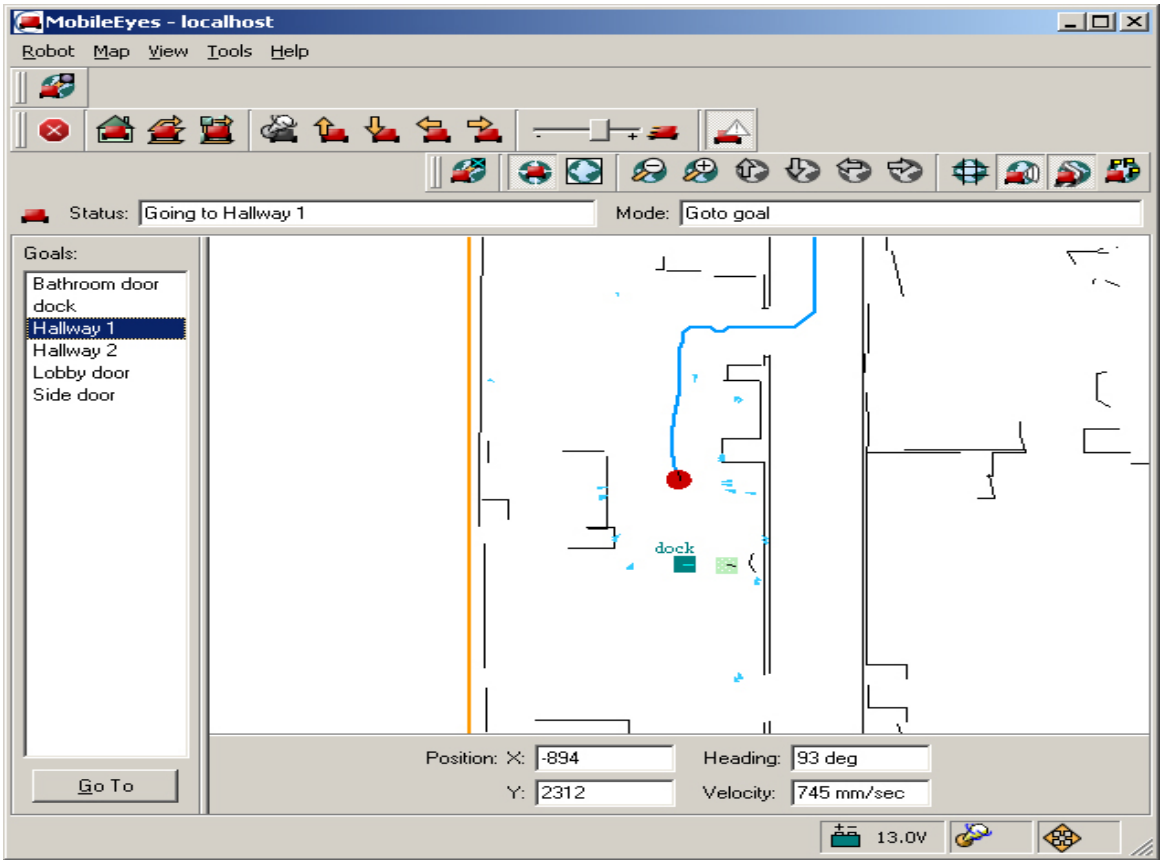
SonARNL

with

MobileEyes™



Installation & Operations Manual

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**

User-Supplied Components / System Requirements 1
Optional 1
Additional Resources 1
Newsgroups *1*
Support *2*

CHAPTER 2 **QUICK START** **3**

Installing the Software 3
SonARNL Simulation 4
Step 1. Run MobileSim *4*
Step 2. Run SonARNL guiServerSonar *4*
Step 3. Run MobileEyes *5*
Gracefully Shutting Down *6*
Exercising SonARNL with MobileEyes 6
Map Display and Operating Parameters *6*
Going to a Goal *6*
Lost from Home *7*
Working with the Real Robot 8
Tethered *8*
Wireless Ethernet *8*

CHAPTER 3 **MAPPER3BASIC** **9**

Points and Lines *9*
File Management *9*
Creating and Editing Maps *10*
Homes and Docks *10*

APPENDIX A **11**

WARRANTY & LIABILITIES **12**

Chapter 1 Introduction

Congratulations on your purchase and welcome to the rapidly growing community of researchers, developers and enthusiasts of MOBILEROBOTS intelligent robots. With your new robot, you get a an advanced robot control development environment and a suite of advanced robotics control and navigation software. This *SonARNL with MobileEyes™ Installation and Operations Manual* provides both the general and technical details you need to install and operate MOBILEROBOTS' Sonar-based Advanced Robotics Navigation and Localization (SonARNL) software on your sonar-enabled robot with the MobileEyes GUI client and Mapper3Basic map-building applications. In fact, SonARNL works with any and all sonar-enabled MOBILEROBOTS platform—new or old, including Pioneer 2- and 3-DX and -ATs, Performance PeopleBot™, PowerBot™, PatrolBot™ and AmigoBot™.

We also encourage you to investigate the higher performance navigation, mapping, and control features of MOBILEROBOTS' Advanced Robotics Navigation and Localization (ARNL) software that uses a laser range-finder accessory and onboard PC. See `http://www.MobileRobots.com` or contact `sales@MobileRobots.com` for details.

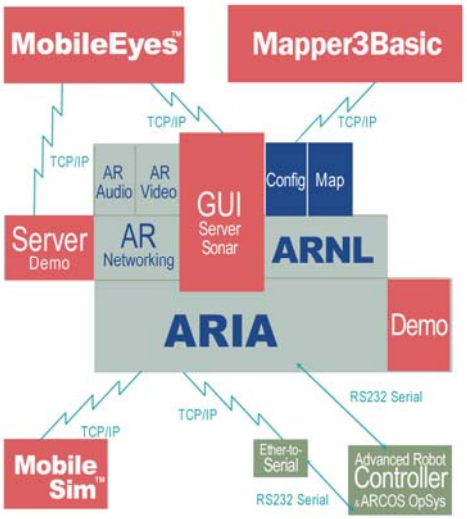


Figure 1. The complete localization and path planning solution.

User-Supplied Components / System Requirements

- ✓ 586-or faster, Win32 or Linux PC with 32 MB RAM
- ✓ >2 MB free hard-disk storage and one RS-232 serial interface

Optional

- ✓ Radio 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]

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

> 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 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 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`

or by email:

`support@MobileRobots.com`

Please include your robot's ***serial number*** (look for it beside the `Main Power` switch)—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 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.

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

Chapter 2 QUICK START

You don't need a robot to operate MOBILEROBOTS' SonARNL software. SonARNL lets you connect a client, such as MobileEyes™, with a MobileSim™-simulated robot server. Accordingly, you may exercise all of SonARNL's sonar-based robot-control, -localization, and -navigation, as well as the many other MOBILEROBOTS software features of ARIA, ARNetworking and so on, without fiddling with hardware.

In fact, we recommend that you install SonARNL, MobileSim, MobileEyes and Mapper3Basic and familiarize yourself with their components and interfaces before you install and work with the robot. 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 onboard PC, you do not need to install SonARNL 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 in order to exercise SonARNL, you need to install at least three distributions: MobileSim, MobileEyes, and SonARNL. Install Mapper3Basic when you want to make and edit maps. Find them all on the CD-ROM that comes with your robot or download fresh versions from our support website:

```
http://robots.MobileRobots.com
```

A cautionary note before you begin installation: If you have ARNL installed (it comes with the laser range-finder accessory), stop right now and rethink why you want to exercise SonARNL: ARNL with Mapper3 is much more advanced software and hardware for localization and navigation. If you plan to forge ahead, uninstall ARNL first before installing SonARNL.

With Windows, either use the Control Panel's `Add/Remove Programs` utility, or

```
Start:Programs:MobileRobots:ARNL:Uninstall ARNL.
```

ATTENTION

You must UNINSTALL ARNL before installing SonARNL
(though we seriously question why you'd want to abandon the more advanced ARNL hardware and software).

With Linux, use the appropriate OS command:

```
rpm –ev ARNL (RedHat®)
apt-get remove libarnl (Debian)
```

Now install the new software. With Windows, the distributions 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 \MobileEyes`. Shortcuts to the programs and files also get put into `Start:Programs: MobileRobots`.

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

```
rpm –ivh SONARNL-1.1-1.i386.rpm (RedHat)
dpgk -i libsonarnl_1.1-1_i386.deb (Debian)
```

The software get put into their respectively named directories under `/usr/local`; e.g., `/usr/local/Arnl`.[3]

[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 guiServer and other ARNL applications.

[3] Yes, that's correct: Arnl, not SonArnl.

SonARNL Simulation

The simulated SonARNL robot has three parts: MobileSim robot simulator, SonARNL's guiServerSonar, and the MobileEyes client. Start them up in 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:

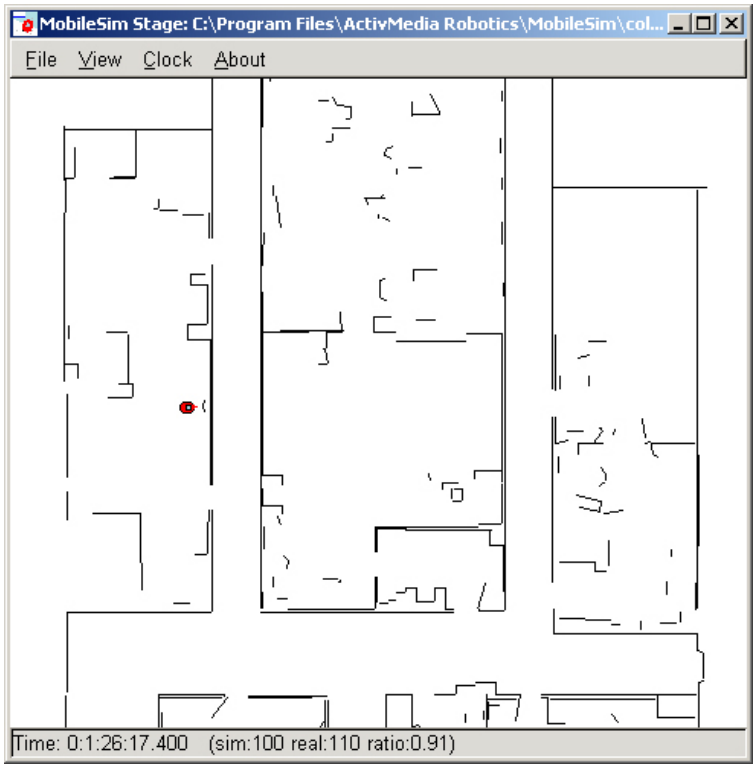


Figure 2. The MobileSim-simulated Pioneer 3-DX at its dock in the columbia-OL map.

Linux

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

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

Windows

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

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

Step 2. Run SonARNL guiServerSonar

All MobileRobots platforms operate in a client-server environment. ARNL comes with a demonstration server called guiServerSonar. It is through server applications like guiServerSonar that clients like MobileEyes get control access to the robot and its accessories, such as the pan-tilt robotic camera, as well as to all the many features and functions of MobileRobots software, including ARIA, SonARNL, ARNetworking and so on.

Like any ARIA-based application, guiServerSonar takes several arguments that direct its connection with the robot (see Appendix A for details).[4] Also, guiServerSonar uses parameter files that specify the operating

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

characteristics for the connecting robot and its accessories (e.g., in `Arnl/params/p3dx.p`), as well as for SonARNL itself (`Arnl/params/sonarnl.p`). You may modify these files or use the parameter names with a single dash ("-") prefix followed by the new value (e.g., `-NumSamples 1000`), as arguments when starting guiServerSonar to override the params files defaults. Once running, clients like MobileEyes also may send commands to guiServerSonar that change these and other operating parameters.

Finally, guiServerSonar needs a map.[5] A SonARNL map defines not only the robot's operating space for localization and navigation, but also defines goals like Dock for auto-charging and other predetermined destinations. See subsequent chapters for Mapper3Basic details.

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

The guiServerSonar application is text-based (X not required), so it may be started from any text console, but cannot run in the background.[6]

Linux

```
cd /usr/local/Arnl/examples
./guiServerSonar –map columbia.map
```

Windows

```
Start:Programs:MobileRobots:SONARNL:GuiServer (with sonar localization)
```

Or, from a command window:

```
cd C:\Program Files\MobileRobots\Arnl\bin
guiServerSonar –map columbia.map
```

SonARNL's guiServerSonar, through ARIA, 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 associated, also simulated, 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
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 guiServerSonar. The hostname `localhost` or IP address 127.0.0.1 usually works if guiServerSonar is running on the same PC as you launch MobileEyes.

[5] Use MobileEyes and Mapper3 to make maps of real places. See subsequent chapters for details.
[6] Use the Linux tool 'screen' in order to launch guiServerSonar and eventually disconnect telnet from the robot.

Linux

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

Windows

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

If you hadn't specified a map when starting up guiServerSonar, 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 `sonarnl.p` for future sessions. Thereafter, use another map either by editing `sonarnl.p` or through MobileEyes' `Tools:Robot Configuration` dialog.

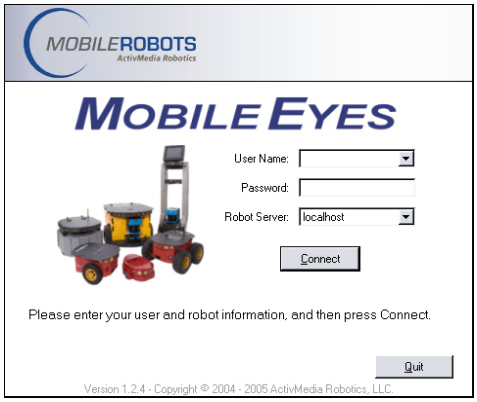


Figure 3. In MobileEyes' startup dialog, enter guiServerSonar's hostname or IP address, not MobileSim's.

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 the `Robot` main menu.

In the guiServerSonar console, press the `Esc` key to quit the program (includes automatic disconnection from the robot and laser).

And, finally, click MobileSim's close box (upper right-hand corner of the window) or choose `Quit` from its `File` main menu.

Exercising SonARNL 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 as well as to investigate its online `Help` menu and try out its many other functions on your own.

First notice that, upon initial connection, guiServerSonar 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 guiServerSonar to MobileEyes. SonARNL'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, the small, light-blue triangles around the robot icon represent sonar reflections.[7] These should conform to what the real or simulated robot senses in its world at the time.

Going to a Goal

Using Mapper3Basic, 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

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

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.[8]

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 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.

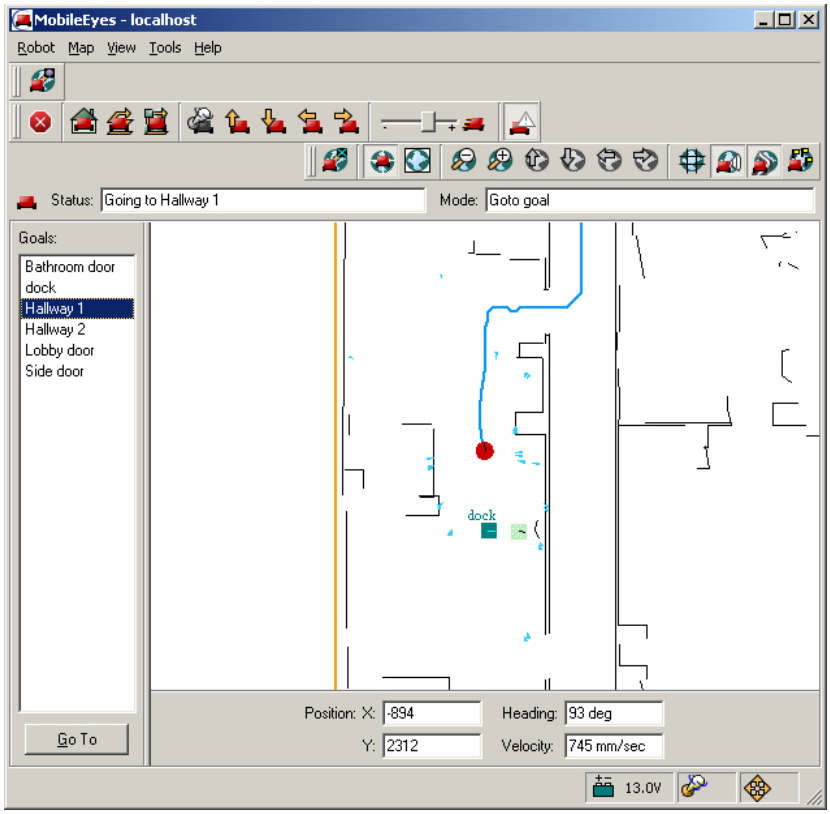


Figure 4. The simulated P3DX on its way to the Hallway 1 goal.

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 guiServerSonar. MobileSim places its simulated robot at the first Home in its map. 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, and then restart guiServerSonar to lose localization.

A good clue that SonARNL 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] Choose the `View:Goals/Routes` menu option to display the window in MobileEyes.

Working with the Real Robot

SonARNL works with any sonar-enabled MOBILEROBOTS platform—new or old, including Pioneer 2- and 3- DX and ATs, Performance PeopleBot, PowerBot and AmigoBot. The trick is where to run the various components and connections. And you'll need to make a map of the robot's real operating space. For that, see the next Chapter.

Tethered

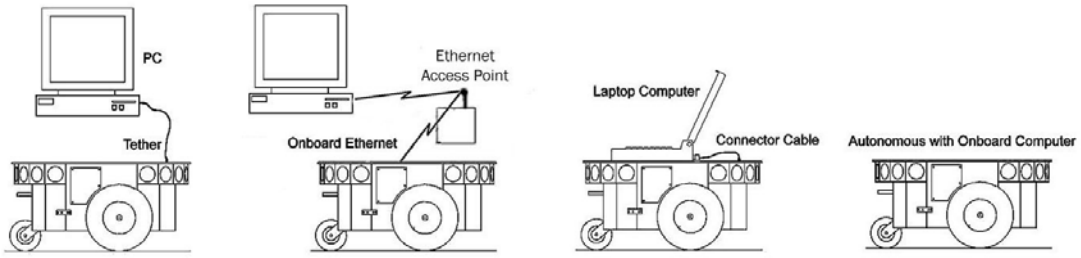


Figure 5. guiServerSonar connection options

The simplest way to connect SonARNL with your robot is via a serial tether from the `COM1` (Linux `/dev/ttyS0`) serial port on your host PC, piggyback laptop PC, or the onboard PC to the HOST serial port on the robot. Then, simply start up the guiServerSonar and MobileEyes software as described above.

In the absence of MobileSim, guiServerSonar defaults to making a connection with the robot through that default serial port. To attach through an alternative serial port, use the '-rp' guiServerSonar startup command line argument; `guiServerSonar -rp COM3`, for example.

Wireless Ethernet

The tethered method works okay for a piggyback laptop and onboard PC; not so well for a desktop PC. And for the piggyback or onboard PC options, you'll have to chase down the robot every time you want to give it a new goal through MobileEyes.

Radio modems may replace the serial wire so that the desktop PC option running guiServerSonar and MobileEyes is viable, and the option still is in use by some legacy robot owners. However, radio modems are fraught with operational limitations and, accordingly, are no longer available from MOBILEROBOTS. Instead, for about the same price, we recommend wireless Ethernet.

There is the Ethernet-to-serial accessory available for all MOBILEROBOTS platforms that don't have an onboard PC, especially AmigoBot. In this case, run guiServerSonar on a networked PC that has access to the wireless network, or a PC equipped with a comparable wireless Ethernet card and link with the device peer-to-peer. Run guiServerSonar on the PC and direct it to connect with the Ethernet-to-serial device's IP:

```
guiServerSonar -rh <Ethernet-to-serial IP or hostname here>
```

If guiServerSonar's host PC is on your LAN, then you may run MobileEyes from any other networked PC. Simply give MobileEyes guiServerSonar's host PC's IP address or hostname, not the Ethernet-to-serial device's network address.

Alternatively, and of course the best option is to have wireless Ethernet onboard the robot. Run guiServerSonar on that PC and connect MobileEyes from anywhere on the LAN. Like with all other options, always give MobileEyes guiServerSonar's host PC IP address or hostname.

Chapter 3 Mapper3Basic

Use the Mapper3Basic program to create maps, as well as add and edit forbidden spaces and goals. Mapper3Basic, in collaboration with guiServerSonar and comparable servers, 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. If you have an ARNL-enabled MOBILEROBOTS platform, use it to make maps for all your robots. SonARNL and ARNL maps are virtually interchangeable, with a few limitations (see below).

Also, Mapper3 and Mapper3Basic are the same application with identical features except the former also converts "2d" scan files created with an ARNL-enabled robot and containing raw laser and odometry readings into a map. Mapper3Basic cannot.

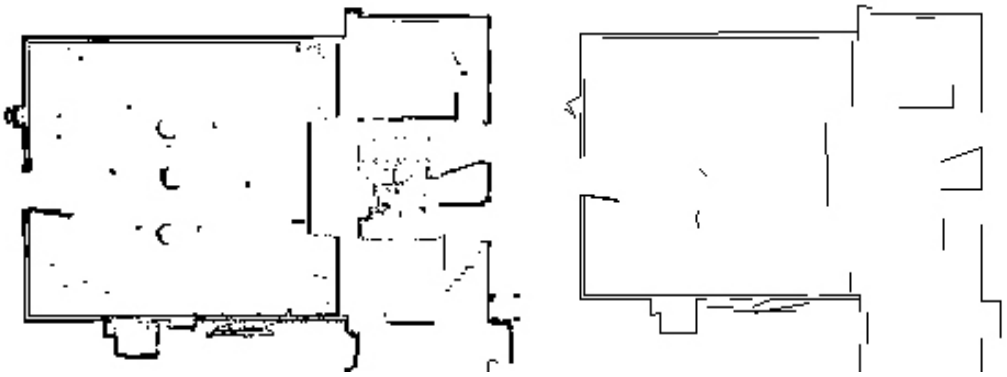


Figure 6. Maps consist of occupancy-grid points (left) and extracted lines (right).

Maps created through Mapper3 and ARNL consist of occupancy grid-points and derived lines. Maps made with Mapper3Basic consist only of lines; no points.

For localization, SonARNL can only use lines like those you may have drawn yourself with Mapper3Basic or add with Mapper3. For path-planning, SonARNL uses both points and lines. MobileSim uses only the lines in a map as its view of the world. ARNL uses both lines and points for both localization and path-planning.

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

File Management

Mapper3Basic gives you several ways to access and map files locally and over the network. Click the `Open` icon to engage a dialog in which you choose a map file to load into Mapper3Basic from local storage or from a networked filesystem. Similarly, click the `Save` button to commit your work to permanent storage.

Mapper3Basic'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 guiServerSonar.[9] `Save on Robot` uses the same mechanism, but in reverse to store your Mapper3Basic work.

[9] Of course, your robot's PC needs to be running guiServerSonar or similar ARNetworking-based application and have access to your network.

The `File:Download/Upload` option also bypasses networked filesystems and connects you via guiServerSonar with a specific robot's files. The option provides a dialog in which you select and transfer nearly any type of file—map or otherwise—to and from the robot and your networked filesystems. It doesn't open any of the files for processing by Mapper3Basic, though.

Accordingly, use Mapper3Basic to connect and distribute maps to your MOBILEROBOTS fleet.

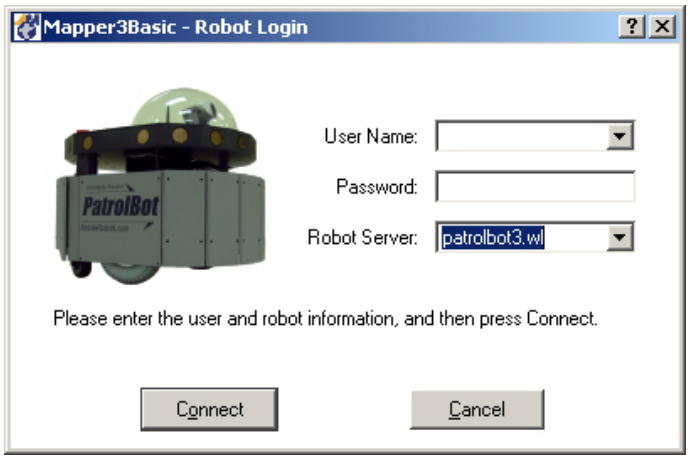


Figure 7. Access maps on your robot with Mapper3Basic's `Open on robot` and `Save on robot` options.

Creating and Editing Maps

Mapper3Basic provides a wide range of tools to create and edit your maps into the best representations of your robot's real operating environments. The important thing to note, however, is that the map is only as good as your measurements, and your robot's ability to navigate the map is directly related to the fidelity of the map. Measure twice and once more again.

Add and erase lines to create and update areas that have changed in your map. Mapper3Basic also lets you designate and modify places that your robots may go (goals) or are forbidden to cross (forbidden lines and areas). Mapper3Basic 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.[10] Please consult Mapper3Basic's online `Help` for additional details, and read its accompanying README.txt file for updates.

Homes and Docks

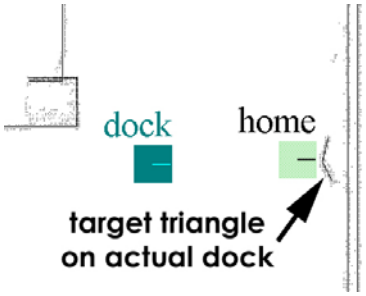


Figure 8. Place Dock goal one to three meters away from actual auto-recharging accessory.

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. And it is the original Home.

Home, as described earlier in the *Quick Start* chapter, is where SonARNL 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.

SonARNL 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 dock accessory that your robots may use to automatically recharge their batteries. When sent to a Dock, your robot's SonARNL navigates to the Dock goal and then uses the dock's special shape to guide it the remaining distance into the charging contacts. That's why the Dock goal isn't actually on the robot's power station, though you should put a Home on the station if that's where you plan to normally shut down the robot.

[10] Zoom and shift your map with the mouse scroll wheel and right button. Or use Mapper3Basic's toolbar buttons.

Appendix A

Table 1. guiServer'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.

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