

# ER1™

## User Guide

Version 1.0.2



130 W. Union Street  
Pasadena, CA 91103

*Customer Service:*  
(626) 229-3198 or  
(866) ROBO4ME (toll free)  
customerservice@evolution.com

[www.evolution.com](http://www.evolution.com)

© 2002 All Rights Reserved. Evolution Robotics, Inc.  
Evolution Robotics, the Evolution Robotics logo and ER1  
are trademarks of Evolution Robotics, Inc. All other  
trademarks are the property of their respective owners.

.....

© 2002 Evolution Robotics, Inc. All rights reserved. Evolution Robotics and the Evolution Robotics logo are trademarks of Evolution Robotics, Inc.

Microsoft<sup>®</sup> DirectX is a trademark of Microsoft Corporation.

Microsoft<sup>®</sup> Speech SDK 5.1 is a trademark of Microsoft Corporation.

Microsoft<sup>®</sup> Windows is a trademark of Microsoft Corporation.

DirectX<sup>™</sup> is a trademark of Microsoft Corporation.

MyWanIp<sup>™</sup> is a trademark of Tzolkin Corporation.

XBeam<sup>™</sup> is a trademark of XBeams, Inc.

Other product and brand names may be trademarks or registered trademarks of their respective owners.

Part number MC6023.

Last revised 9/12/02.

.....



## **IMPORTANT SAFETY INFORMATION**

Before using the robot, basic safety precautions should always be followed including the following:

### **READ ALL INSTRUCTIONS BEFORE USING KEEP AWAY FROM WATER**

#### **WARNING: IMPORTANT SAFEGUARDS**

- To prevent fire or shock hazard, do not expose the robot to rain or moisture.
- To avoid electrical shock, do not operate in or near water.
- To avoid breakage or damage to the robot or your laptop computer, use caution in operating the robot. Do not operate the robot outdoors, near stairs or on top of furniture. Use the robot on a level surface.
- To avoid damage to property or injury to persons, always operate robot under close supervision. **THE ROBOT SHOULD NEVER BE LEFT TO OPERATE UNATTENDED OR UNSUPERVISED.**
- To reduce risk of damage to your laptop computer, use caution in ensuring that your laptop computer is securely fastened to the robot. **BY USING THE ROBOT WITH YOUR LAPTOP COMPUTER, YOU ARE ASSUMING ALL RISKS FOR ANY DAMAGE TO YOUR LAPTOP COMPUTER.**
- Do not under any circumstances use the robot to move, touch or handle dangerous or hazardous materials. Doing this could result in injury or death to the user or damage to property.
- The robot is not a toy and is not recommended for children.
- Keep hair, loose clothing and all body parts away from openings and moving parts.
- Beware of loose objects that may be caught in the motors or wheels.
- Use the switch on the back of the robot to turn the robot off when you wish to stop the robot.
- Do not open any of the electrical components housed in sheet metal units, as this might result in fire or shock hazard.
- Our product parts may contain sharp edges and should be handled with caution to avoid injury.
- Store packing materials safely out of reach of children to prevent the risk of suffocation from plastic bags.
- Do not operate the robot near pets.
- Use only as described in the instruction manual.



Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. 47 C.F.R. section 15.21.

---

*Important Note:* This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.



DO NOT OPERATE ROBOT  
WHILE CONNECTED TO CHARGER



## **DANGER: Rechargeable Battery Pack**

If the rechargeable battery pack is incorrectly replaced, there is a danger of an explosion. The battery pack contains a small amount of harmful substances. To avoid possible injury:

- Replace only with a battery of the type recommended by Evolution Robotics.
- Keep the battery pack away from fire, water and rain.
- Do not attempt to disassemble the battery pack.
- Ensure that the battery pack is securely fastened to the robot.
- Do not short-circuit the battery pack.
- Keep the battery pack away from children.

When disposing of the battery pack, comply with local ordinances or regulations.

Nothing herein shall constitute a warranty, expressed or implied, including any warranty of merchantability or fitness. In no event shall Evolution Robotics be responsible for any special, incidental or consequential damages.

## ■ Table of Contents

---

### **Chapter 1: Introduction**

Robot Control Center Software.....	1-2
Possibilities .....	1-3
Customer Support .....	1-4
Registration.....	1-4
ER1 Community .....	1-4

### **Chapter 2: Getting Started**

ER1 and the RCC software .....	2-2
Robot Specifications .....	2-3
Laptop Requirements.....	2-3
Optional Second Camera .....	2-3
Optional Remote Control.....	2-4
Optional Email .....	2-4
Trying Out the Software .....	2-5
Making the Robot Count.....	2-5
Making the Robot Move.....	2-7

### **Chapter 3: Behavior Programming Overview**

Behavior Tabs .....	3-2
Behavior Configurations.....	3-2
Behavior Files .....	3-2
If Conditions .....	3-3

---

Sight Condition .....	3-3
Sound Condition .....	3-3
Time Condition .....	3-4
Message Receive Condition .....	3-4
Sequence Condition .....	3-4
Then Actions .....	3-4
Move .....	3-5
Play a Sound .....	3-5
Run a Program .....	3-5
Message Send .....	3-5
Record .....	3-5
Trying Out Actions .....	3-5
Running and Stopping Behaviors .....	3-5
Behavior Execution .....	3-5
Behavior Error and Logging .....	3-7

## **Chapter 4: Audio, Video and Navigation Controls**

Video Display .....	4-1
Recognized Objects .....	4-2
Obstacle Camera (optional) .....	4-2
Mute Sound Checkbox .....	4-2
Navigate .....	4-2
Battery .....	4-3
Stop Recognizing Objects .....	4-3

## **Chapter 5: Function Buttons**

New Button .....	5-1
Open Button .....	5-2
Save Button .....	5-2
Save Only Active Fields .....	5-2
Settings Button .....	5-3
Help Button .....	5-4

---

---

## **Chapter 6: If Conditions**

Sight Condition .....	6-2
Color Selection Tool .....	6-2
Object .....	6-2
Motion .....	6-6
Sound Condition .....	6-6
Time Condition .....	6-6
Message Receive Condition .....	6-8
Sequence Condition .....	6-9

## **Chapter 7: Then Action**

Reset and Try It Buttons .....	7-1
Move .....	7-2
Play a Sound .....	7-3
Run a Program .....	7-3
Message Send .....	7-4
Record .....	7-4

## **Chapter 8: Settings**

Sight Tab .....	8-2
Camera Tab .....	8-3
Device .....	8-3
Resolution .....	8-4
Remote Transmission .....	8-4
Sound Tab .....	8-5
Remote Control Tab .....	8-6
Email Tab .....	8-9
Motion Tab .....	8-11
Obstacle Avoidance Tab .....	8-12
Positioning the Obstacle Avoidance Camera .....	8-12
Avoidance Camera Calibration .....	8-13
Avoidance Camera View .....	8-13
Obstacle Avoidance At Work .....	8-14

---

Log Tab.....	8-14
About Tab.....	8-15

## **Chapter 9: Behavior Settings**

Sight Tab.....	9-1
Image Recognizer.....	9-1
Color Recognizer.....	9-2
Motion Detection.....	9-2
Sound Tab.....	9-3
Photo Tab.....	9-3
Video Tab.....	9-4
Motion Tab.....	9-5
Rotation Angular Velocity.....	9-6
Motion Linear Velocity.....	9-6
Motor Power.....	9-7
Detect Collisions.....	9-7
Obstacle Avoidance.....	9-8
Run Tab.....	9-9
Apply This Page to All Behaviors Now.....	9-9

## **Chapter 10: Application Programmer's Interface**

Introduction.....	10-1
Accessing the API.....	10-2
Telnet Windows XP Setup.....	10-3
Older Windows Telnet Setup.....	10-3
Using the Telnet Connection.....	10-4
Commands.....	10-5
Examples.....	10-5
Command Quick Reference.....	10-5
Move Command.....	10-6
Move Distance.....	10-6
Move Rotate Toward Object.....	10-7
Move Rotate Toward Color.....	10-7
Move Drive Toward Object.....	10-8

---

---

Move Drive Toward Color .....	10-8
Objects Command.....	10-8
Objects .....	10-8
Play Command.....	10-9
Play File.....	10-9
Play Phrase.....	10-9
Position Command.....	10-9
Position .....	10-9
Stop Command.....	10-10
Sense Command.....	10-10
Clear Command .....	10-10
Events Command .....	10-11
Set Command.....	10-12
Set Voice.....	10-12
Set Linear Velocity .....	10-12
Set Angular Velocity .....	10-12
Set Power Stopped.....	10-13
Set Power Moving .....	10-13
Set Collision Detection .....	10-13
Set Obstacle Avoidance.....	10-13
Set Confidence Threshold.....	10-14
Set Color Tolerance .....	10-14
Set Color Percentage.....	10-14
Digital Input/Output Commands.....	10-15
Electrical Characteristics .....	10-15
Input Digital .....	10-16
Output Digital .....	10-16
Input Analog .....	10-18
Building ER1 Scripts .....	10-19
Download Java SDK.....	10-20
Compiling .....	10-20
Running the Script .....	10-20
TestClient.java .....	10-21
Java API Example Programs.....	10-22

---

ER1Client.java ..... 10-23

**Chapter 11: Tricks of the Trade**

Robot Search Pattern ..... 11-1  
Motor Noise ..... 11-2  
Wireless Network Disconnect ..... 11-2  
Frozen Robot ..... 11-2  
Camera Shuts Off When Launching RCC/Unplugging Power Supply ..... 11-2  
Textbox Input Size ..... 11-3

**Appendix A: Assembly Instructions**

Getting to Know the Parts ..... A-1  
Things You Need ..... A-6  
Charging the ER1 ..... A-6  
Assembling Your “Assembled” ER1 ..... A-7  
Assembling Your Unassembled ER1 ..... A-11  
Installing the Software ..... A-27  
Connecting the Laptop ..... A-28  
Testing the Robot ..... A-29  
Uninstalling ..... A-30

**Appendix B: Telepresence Over the Internet**

**Appendix C: 50 Ways to Love Your Robot**

Easy ..... C-1  
    Take a video of the kids and send it to Grandma ..... C-1  
    Take a picture of the family and email it to Uncle Fred ..... C-2  
    Send an email if ER1 hears a loud sound while you are away ..... C-2  
    Remind you to take the muffins out of the oven ..... C-3  
    Remind you to call home if you are running late at the office ..... C-3  
    Remind you to pick up the dry cleaning ..... C-3  
    Remind you to feed the fish every day ..... C-3  
    Remind you to call your Mom every Wednesday night ..... C-4  
    Give you five-minute, advance notice before your next class ..... C-4

---

---

Play your favorite song at the same time every day .....	C-4
Play your favorite music with a voice command.....	C-4
Have ER1 play your favorite 20 songs .....	C-5
Tell your dog to “Be quiet!” if it barks too loud.....	C-5
Teach your parrot some new words .....	C-6
Stand outside your door and tell anyone who knocks you are in a meeting....	C-6
Play “You’re Free!” at quitting time every Friday .....	C-6
Play “Shush!” when it hears a loud sound.....	C-7
Give homework reminders every day before the end of class .....	C-7
Tell people when it’s time to cut the cake at your child’s birthday party.....	C-7
Email video of the party every hour to a sick friend who couldn’t make it ....	C-7
Medium .....	C-8
Teach the ER1 to dance and entertain your friends .....	C-8
Read books out loud to your child (while under your supervision).....	C-9
Teach your child the names of things (while under your supervision).....	C-9
Play a lullaby (under your supervision) if the baby cries loudly .....	C-10
Help you recognize familiar objects if you have lost your glasses.....	C-10
Tell the kids a bedtime story in your voice while you entertain nearby .....	C-11
Hard.....	C-11
Play a disco song every time somebody says the word “party” .....	C-11
Take a photograph of anyone who says “Hello, robot” at a party .....	C-12
Remind your guest holding a beer not to drink and drive .....	C-12

---

■ **Table of Contents**

---

## ■ Chapter 1

---

# Introduction

Evolution Robotics™ is proud to introduce you to the ER1 Personal Robot System and pleased to welcome you to our family of owners. The ER1 is the first genuinely useful and affordable personal robot that is also extendable and programmable by you. To give you the full programming and processing power of the personal computer, as well as wireless capabilities, we designed the ER1 around a laptop as its brain. Included in this package is everything you need – all the beams, connections, motion control electronics, software, camera – to get up and running in just a couple of hours. Just supply the laptop and you're ready to go!

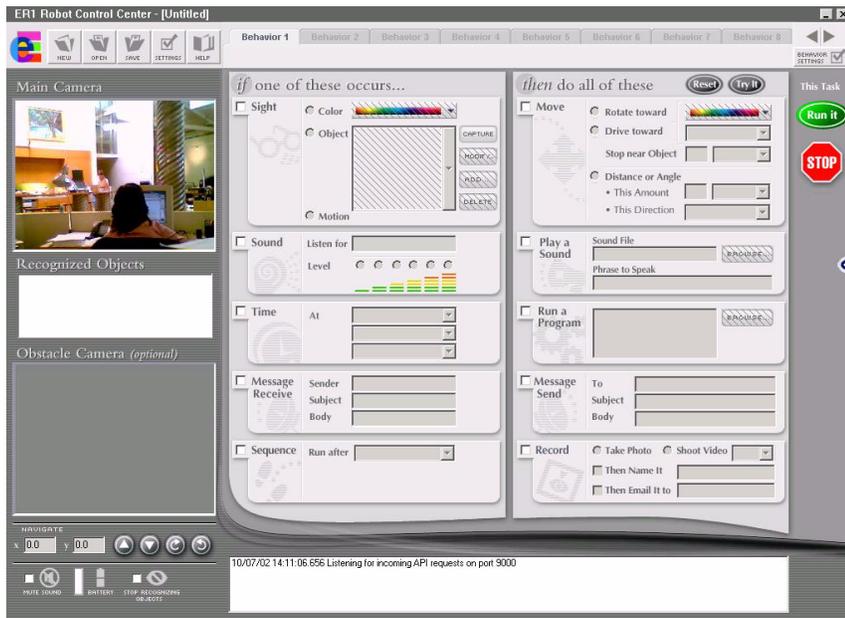


The heart of ER1 is a powerful vision algorithm that can recognize thousands of objects– plus you can teach ER1 to recognize objects in your home. The vision system can also recognize many obstacles, and can instruct your robot to move around them. The Robot Control Center (RCC) software, allows you to control this vision system, as well as all of the other RCC functions, including recognizing colors, sounds, words, sending and receiving email, acting on schedule, moving around autonomously or by remote control, playing sounds and music, or taking pictures and video. You can use the RCC software to combine these individual behaviors into any number of complex behaviors.

Your first steps are to assemble your robot, add your laptop, install the software, and start the ER1 RCC software. If you have the ER1 kit, the assembly will take about an hour. The instructions are in Appendix A, [Assembly Instructions](#). If you have the assembled robot, the [Assembly Instructions](#) in Appendix A will explain how you connect the camera carrier and mount your laptop on the robot. It will take about 5 minutes. You will be plugging two items into your laptop, both provided. One is the camera that will be used for your robot's vision. The other is our proprietary Robot Control Module (RCM) that will coordinate your robot's navigation. Both are USB connections to your computer. A separate 12-Volt 50 Watt power module is provided for the RCM.

## Robot Control Center Software

The ER1 RCC software is used to control and receive feedback from your robot. On the left side of the screen is an area to show you what your robot is seeing, a list of any recognized objects, any additional video input from an obstacle avoidance camera and robot movement controls. The rest of the RCC software consists of a set of **If** conditions and **Then** actions that can be used to create a wide variety of robot behaviors or to perform tasks. You can have up to 96 behaviors that can run in any sequence you specify, or individually. You can save what you create and share it with others at our ER1 community at [www.evolution.com](http://www.evolution.com). We have provided a number of tasks and activities for you to try in “[50 Ways to Love Your Robot](#)”, but the true range of the ER1 is unlimited. A quick tutorial for this software is available in the “[Getting Started](#)” section. Later sections will describe this software in detail.



To create a new robot task, here is the strategy:

- Decide what you want to trigger the task: a recognized object or color, a sound, a certain time, or email.
- Break the task into subtasks that the robot can achieve, one at a time, such as moving, speaking, sending an email or recording video.

- 
- Link the individual tasks one after another in any order you choose.
  - Save the sequence of behaviors you have created.
  - Click on the **Run it** button. As soon as the trigger condition in the **If** column has been met, the behaviors will be executed.

With this basic strategy, you can program your ER1 to do amazing things. With accessories such as the Gripper Arm (soon to be released) you can have the robot carry things from one place to another. With the XBeams Expansion Pack you can change the size and shape of the robot. With an extra camera, you can both view obstacles on the floor and recognize objects while looking forward. We would love to hear what new ideas you come up with for using ER1.

## Possibilities

ER1's Robot Control Software gives you access to a wide range of robot functionality. Here are just a few examples:

- **Object Recognition** allows you to train your robot to recognize specific objects (furniture, books, vases, etc.) in its environment. Later, you can train the robot to perform an action based on object identification. For example, say "Television remote" when it recognizes your television's remote control.
- **Color Detection** allows the robot to detect colors you specify. For instance, you can train the robot to recognize the color of your couch and to move toward it.
- **Voice Recognition** uses the Microsoft® Speech SDK 5.1 voice recognition engine to enable the robot to understand certain spoken commands. (Note that your robot does not come with a microphone. You can use the microphone built into in your laptop or an external one.)
- **Timer** allows the robot to perform user-specified behaviors at a preprogrammed time.
- **Navigation Commands** allow you to direct your robot's movements, including moving a specified distance at a specified velocity, moving towards an object and stopping a predetermined distance from that object, rotating towards a color, and more.
- **Text-to-Speech** uses built-in Microsoft® Windows text-to-speech capabilities to allow the robot to speak the words from written text.
- **Play a Sound** allows you to tell the robot to play a specific sound in response to a spoken command or visual cue, or at a specified time.
- **Run a Program** allows you to launch and execute any Windows program directly from the Robot Control Center software window.
- **Send and Receive Email** allows your robot to perform pre-programmed actions upon receipt of an expected email. Your robot can also send email alerting you that a preprogrammed condition has occurred.
- **Take Photo/Video** allows the robot to take a picture or record video.

- **Remote Control** allows you to use a customer-supplied wireless network to teleoperate the robot from your PC.
- **Obstacle Avoidance** provides the robot with a rudimentary visual obstacle avoidance capability through a second camera (second camera not included in base package).
- **Behavior-Based Programming** allows you to specify certain events as triggers for one or more robot actions. For example, you can tell the robot to move forward 5 feet when it hears you say “Come here robot”. You can string these condition/action behaviors together to perform complex tasks.
- **Standard Menus** provide standard Windows features for setting preferences, saving and retrieving programs, etc.

## Customer Support

Evolution Robotics Customer support is available by email at [customerservice@evolution.com](mailto:customerservice@evolution.com) or by filling out the form at [www.evolution.com/support/](http://www.evolution.com/support/). Customer Service representatives are available by calling toll free at 866-ROBO4ME or, for international customers 626-229-3198, Monday through Friday, 8:30 A.M. to 5:30 P.M. Pacific Time.

## Registration

Remember to register your robot online at [www.evolution.com](http://www.evolution.com). By registering you get the following benefits:

- World class customer support
- RobotMail email account
- Join the ER1 development community
- Software updates and bug fixes
- Information on new accessories and software

The serial number sticker for your robot is on the corrugated box under the top flap. Make sure to put your serial number sticker someplace safe, e.g. the inside cover of your manual. If you lose this sticker, you can also find the robot serial number on the Robot Control Module.

## ER1 Community

The ER1 community is a place to share development ideas, lessons learned, shortcuts and applications. After you have completed registration, visit the ER1 community at [www.evolution.com](http://www.evolution.com).

## ■ Chapter 2

---

# Getting Started

**Note:** The assembly instructions are in the “[Assembly Instructions](#)” Appendix. Before turning to them, please read the following instructions.

For both the assembled and unassembled robot, here is an overview of the steps you need to follow to get up and running:

1. Make sure that your laptop is fully charged.
2. Remove the robot (or the unassembled robot's components) from the shipping box.
3. Whether you bought the assembled robot or the unassembled robot kit, the first thing you'll want to do is charge the robot's battery. The battery is shipped with a minimal charge, and needs a full charge before you start using it.
  - Plug the robot's battery into the charger supplied with the robot, and plug the charger into an electrical outlet. Note that the robot may be turned on while it is charging, but the robot's motors should not be engaged until the charger is disconnected.
  - The robot only operates from the battery's power and cannot draw power directly from the wall outlet. In the Robot Control Center (RCC) software

Settings, you can specify how much motor power the robot should use when it is performing normal operations, when it is ON but idle, etc.

- The robot's battery takes 4-5 hours to charge.
  - A fully charged robot will run for approximately 3 hours (driving) before needing to be recharged. See our website at [www.evolution.com](http://www.evolution.com) to purchase an extra battery.
  - Remember that your laptop also requires recharging, and that you must separately plug the laptop's charger into a wall outlet to charge it. We recommend that, if you're using the robot to perform stationary actions, you leave your laptop plugged in. Note that the robot and the camera(s) will add to the load on your laptop's battery.
  - **UNPLUG THE LAPTOP'S CHARGER BEFORE INSTRUCTING THE ROBOT TO MOVE.** The robot should detect whether or not the laptop is plugged into a wall outlet and should not move until the laptop is unplugged. However, take care to ensure that the laptop is unplugged, because if the robot fails to detect that the laptop is plugged in, it will move in response to your commands until it reaches the limit of the laptop's power cord, then will continue to try to move, potentially damaging the laptop, the laptop's charger, the robot, and the electrical outlet. This could start an electrical fire.
4. Whether you purchased the unassembled or the assembled robot kit, turn now to Appendix A, "[Assembly Instructions](#)". Follow the robot assembly instructions to build your robot, attach your laptop, install and test the software. Then, return here to Step 5 to continue.
  5. When the software installation is complete, double-click on your desktop's new RCC icon to start the application. (See the "[Installing the Software](#)" section of the "[Assembly Instructions](#)" for help, if needed.)
  6. If you'll be operating your robot remotely from another computer, use the CD to load the ER1 software onto that machine as well. This use is permitted by your RCC software license.
  7. Your robot needs to be trained to recognize your speech patterns, and to adjust to your microphone, if you plan to use sound related features. See the Speech Condition section later in this document for instructions on how to do this. Until you do this, the robot's speech recognition capabilities will be poor.
  8. For remote operation, you need to configure the preferences as described in the [Settings](#) section and in the [Telepresence Over the Internet](#) section. You will also need wireless network capabilities.
  9. When you are done, you can close the RCC software by clicking the X in the upper right corner or by holding down the Alt and F4 keys.

## **ER1 and the RCC software**

Using the RCC software, ER1 is an integrated robotics system comprising an XBeams ultrastructure, motors, and a laptop for a brain. Making sure that you pick the correct laptop for your robot is very important. Here are some specifications and requirements.

---

## Robot Specifications

Your robot's specifications are:

- Size: 24" x 16" x 15" (H x W x D)
- 20 lbs without the laptop

## Laptop Requirements

*Important Note:* The laptop is not included with your purchase.

Your robot requires a Windows-based laptop with:

- Pentium® III class, Intel® Celeron®, or AMD processor - 500 MHz or faster
- 128 MB RAM
- 250 MB Hard Disk Space
- Two USB ports (directly on laptop or via hub). One port connects the camera to the laptop, the other connects the robot to the laptop. (Note that you will need three USB ports or a USB hub if you opt to attach a second camera or a microphone. Check our website at [www.evolution.com](http://www.evolution.com) for suggested USB hub brands and models.)
- Laptop monitor must be able to support a 1024 x 768 screen setting.
- One of the following installed Windows Operating Systems: Microsoft® Windows 98, Windows 2000, Windows XP, or Windows Me.

*Important Note:* The ER1 Robot Control Center software is a 1024 by 768 application. It will run on all displays with 1024 by 768 or higher resolution. Because the application has a large number of fixed fields, it will not scale/resize, so on screens larger than 1024 by 768 the RCC will have a border around it, but will continue to work perfectly. The RCC will not run on 800 by 600, or 640 by 480 displays unless you have scrolling software that allows you to view a larger window on your monitor.

## Optional Second Camera

The camera shipped with your robot can be used by the robot for object recognition and obstacle avoidance. However, the software allows you the option of using an additional, separate camera.

- The camera for obstacle avoidance is best mounted pointing toward the floor.
- Requirements for the second camera are:
  - USB camera mountable on the robot (not too big or heavy)
  - USB cable to connect the second camera to the robot
  - A third USB port on the laptop (usually this requires a hub)

## **Optional Remote Control**

If you are planning to teleoperate your robot, you need the following:

- A separate PC that meets or exceeds the laptop requirements listed previously
- A wireless network card for your laptop
- A wireless network card for your PC

*Important Note:* This remote control configuration should operate within a distance of approximately 30 to 60 feet between the robot and the controlling PC, depending on interference caused by other signals, thick walls, etc.

## **Optional Email**

If you plan to program your robot to send and receive email, you need the following:

- Access from your robot laptop to a Internet connection.
- An incoming Evolution Robotics RobotMail account for your robot. We will provide a RobotMail account that will allow you to send your robot email to initiate preprogrammed commands.
- An outgoing email account for the robot provided by the ISP of your choice.
  - Many Internet providers allow you to set up multiple email accounts. Also, there are a number of providers of free email accounts. Contact your Internet provider and/or a free email service for more information on setting up a 2nd email account.

*Important Note:* The messages transmitted to the robot are not direct instructions, but are messages which, based on the sender, subject, and/or body will result in activation of a pre-programmed task.

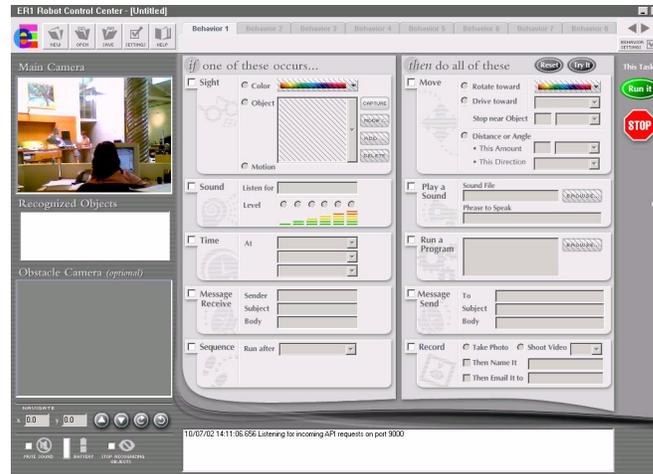
## Trying Out the Software

Ok, now for the fun stuff. The following are examples to help you learn to use your robot. These examples will walk you step by step through the process of creating a behavior for your robot. You can also use these behaviors as a starting point for creating your own behaviors. For more tutorials, see [50 Ways to Love Your Robot](#).

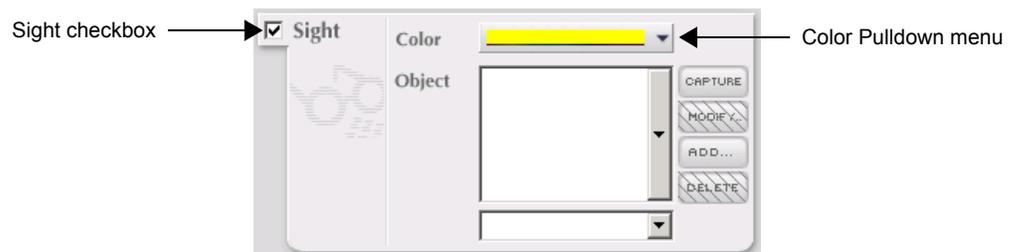
### Making the Robot Count

The following is an example of a simple exercise to test your robot.

1. Start the RCC software on the robot's laptop.

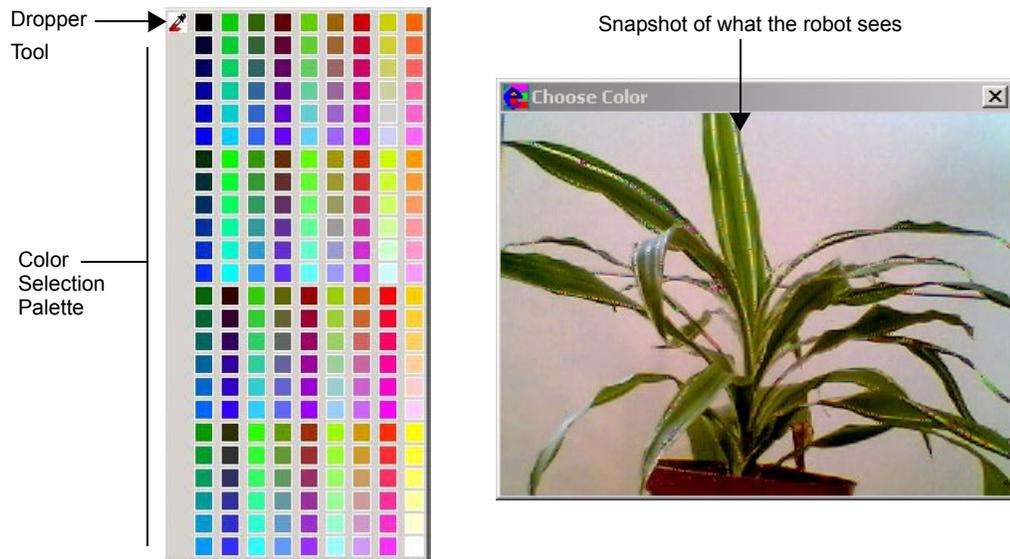


2. Select the **Sight** checkbox from the **If** column.



3. Click on the color pulldown menu.

4. Select the dropper tool in the upper left corner. A snapshot of what the robot can presently see is displayed.



5. In the snapshot, select a color. That color will be displayed in the color pulldown menu. (Note that the **Color Selection Palette** can also be used to select a color.)
6. Select the **Play a Sound** checkbox in the **Then** column.



- In the **Phrase to Speak** textbox, type the number 1.

7. At the top of the screen, select the **Behavior 2** tab.



- At the bottom of the **If** column, select the **Sequence** checkbox.
- In the **Run after** pulldown menu, select **Behavior 1**.



8. In the **Then** column, select the **Play a Sound** checkbox.
  - Type the number 2 in the **Phrase to Speak** textbox.

9. At the top of the screen, select the **Behavior 3** tab.
  - In the **If** column of conditions, select the **Sequence** checkbox.
  - In the **Run after** pulldown menu, select **Behavior 2**.
  - In the **Then** column, select the **Play a sound** checkbox.
  - In the **Phrase to Speak** textbox, type the number 3.

Congratulations! You have just programmed a behavior that will count to three when it sees a certain color. Let's run it.

10. Click on the **Run it** button.



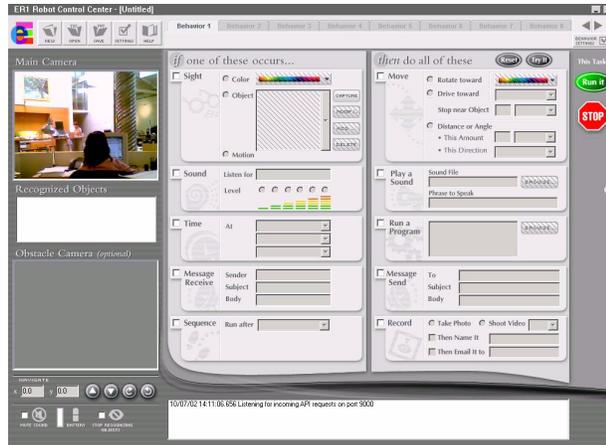
You notice that all of the behavior options are grayed out. These cannot be altered while a behavior is running. The behavior will begin as soon as the robot sees the designated color. The robot will count to three and then stop.

11. If you would like to save this behavior, click on the **Save** button in the upper left corner of the RCC screen. A **Save** dialog box will be displayed. Name the behavior and click on the **Save** button.

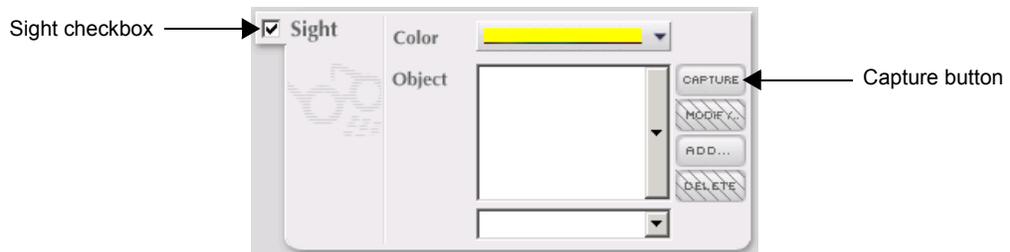
### Making the Robot Move

The following example will walk you through creating a behavior where the robot will move. Make sure that the robot has several feet of clear space all around it before running this behavior.

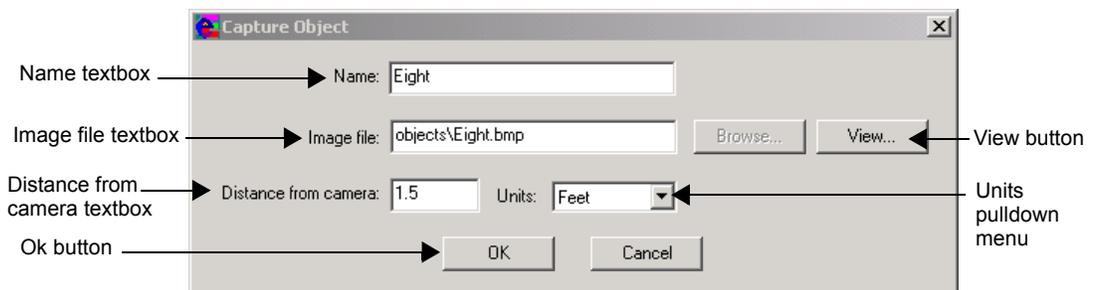
1. Get a piece of paper and a marker. Draw a large number 8 on the piece of paper. Don't be shy about applying the ink. The more clearly the number is drawn, the more easily the robot will recognize it.
2. Open the RCC software if it isn't open already. If you have previously created a behavior, click on the **New** button to save the previous behavior and to start a new one.



3. Click on the Sight checkbox.



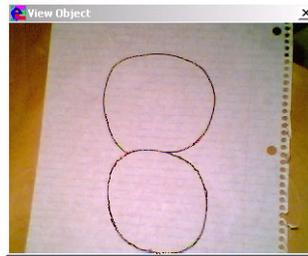
4. Position the piece of paper in front of the robot's camera. Make sure that the paper takes up most of the field of view. The paper should be well lit, but sunlight or light of that magnitude is not recommended.
5. When the paper is properly positioned, click on the **Capture** button. The image will be saved.
6. The **Capture Object** dialog box will appear.



Fill in:

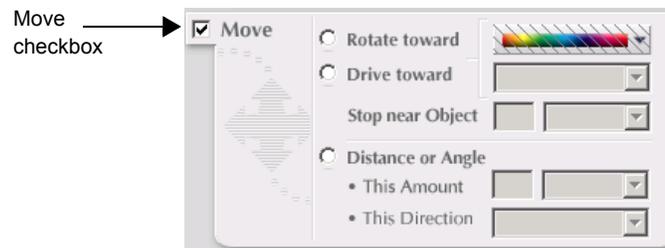
- **Name** - Type "Eight".
- **Image File** - This will automatically change to Eight .bmp.

- **View button** - Click on this and give it a spin. A preview of the image you captured will be shown. It should look something like this:

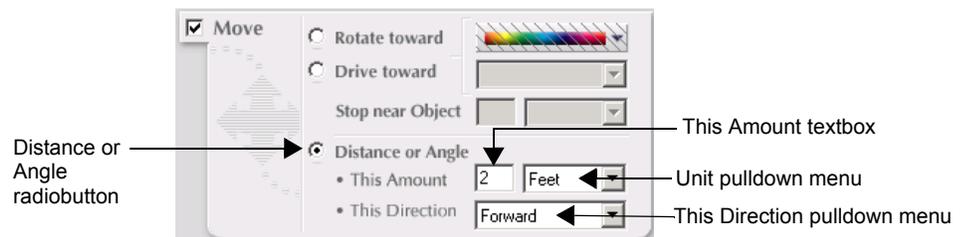


- **Distance from Camera** - Input the distance of the paper from the camera.
- **Units pulldown menu** - Select a unit for the distance from the camera.
- Click on the **Ok** button.

7. Click on the **Move** checkbox.



8. Select the **Distance or Angle** radio button. In the **This Amount** textbox, type 2. In the **Unit** pulldown menu, select Feet. In the **This Direction** pulldown menu, select Forward.



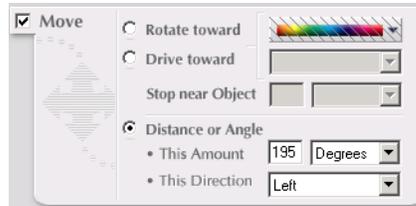
9. Select the **Behavior 2** tab at the top of the screen.



10. Click on the **Sequence** checkbox and select Behavior 1 from the **Run after** pulldown menu.



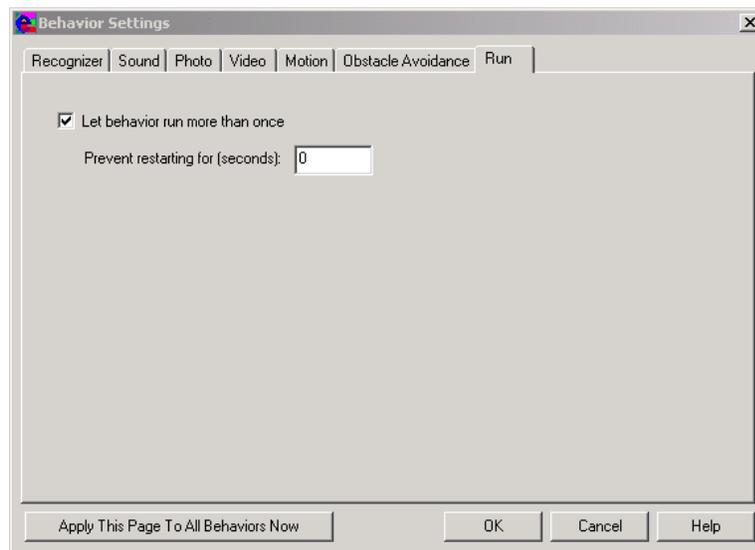
11. Click on the **Move** checkbox again.
12. Select the **Distance or Angle** radio button. In the **This Amount** textbox, type 195 and select Degrees from the **Units** pulldown menu. Select Left from the **This direction** pulldown menu.



13. Select the **Behavior 3** tab at the top of the screen.
14. Select the **Sequence** checkbox and select Behavior 2 from the pulldown menu.
15. Select the **Move** checkbox.
16. Select the **Distance or Angle** radio button. As before, type 2 in the checkbox. Select Feet from the **Units** pulldown menu. Select Forward from the **This Direction** pulldown menu.
17. Select the **Behavior 4** tab.
18. Select the **Sequence** checkbox and select Behavior 3 from the **Run after** pulldown menu.
19. Select the **Move** checkbox.
20. Select the **Distance or Angle** radiobutton. In the **This Amount** textbox, type 195. Select Degrees from the **Units** pulldown menu. Select Right from the **This Direction** pulldown menu.
21. Select the **Behavior 1** tab.
22. Select the **Sequence** checkbox and select Behavior 4. This will create a continuous loop in the behaviors.
23. Click on the **Behavior Settings** button. The **Behavior Settings** dialog will be displayed.



24. Select the **Run** tab. Select the **Let behavior run more than once** checkbox. Click on the **Apply This Page to All Behaviors Now** button. Click on the **Ok** button.



25. Ok, let's try it. Click on the **Run it** button and show the robot the paper with the number 8 on it. The robot will do a figure 8.



26. To stop the robot, click on the **Stop** button.





## ■ Chapter 3

---

# Behavior Programming Overview

Now that you've built your robot, your laptop is installed, your battery is charged, and you've moved your robot around a little bit, it's time to learn how to build behaviors that can do more sophisticated things. As we mentioned in the Introduction, there is a basic strategy to creating a behavior:

- Decide what you want to trigger the task: a recognized object or color, a sound, a certain time, or email.
- Break the task into subtasks that the robot can achieve, one at a time, such as moving, speaking, sending an email or recording video.
- Link the individual tasks one after another in any order you choose.
- Save the sequence of behaviors you have created.
- Click on the **Run it** button. As soon as the trigger condition in the **If** column has been met, the behaviors will be executed.

With that in mind, let's explore robot behaviors in more detail.

An ER1 behavior is a rule that describes what the robot should do when a certain condition is met. (For the developers among you, in programming terms, an if-then statement is used to encode the condition-action rule of a behavior.)

An example of a behavior is the following condition-action pair:

```
If hear a sound
Then shoot video and email it to the specified address
```

One or more conditions can be specified in the **If** column, and one or more actions can be specified for the **Then** column. Whenever any one of the **If** conditions is satisfied, the behavior is said to be triggered, and all its actions specified in the **Then** column are executed.

After you click **Run**, all of the **If** conditions of all of the behaviors (1-96) you programmed are continually monitored by the software until one of the conditions is met. This triggers the **Then** portion of that behavior and executes all the actions. Only one behavior can execute at a time. Once a behavior is done executing, then all behaviors are once again monitored and are eligible for triggering.

Each individual behavior is simple, but when put together, multiple behaviors can perform complex and interesting tasks in response to a variety of conditions. Behaviors can be chained together in a sequence, so that each behavior triggers the next behavior, forming a sequential program of up to 96 behaviors.

The set of all behaviors and their configuration settings is called a task, and the task can be saved to a `.rbt` file for later use.

## Behavior Tabs

Each behavior has a tab (located at the top of the screen) associated with it. Clicking on the tab selects the corresponding behavior for editing. To use the tabs:

1. Click on the left or right arrows at the top right corner of the screen to move to the subsequent or preceding page of eight tabs.
2. Double-click on a tab to change its name.

## Behavior Configurations

Each behavior also has a set of individual configuration settings accessible by clicking on the **Behavior Settings** button at the top right of the screen. See “[Behavior Settings](#)” for a complete description of these settings.

## Behavior Files

Behaviors and all their associated configuration settings can be saved to a file with the `.rbt` extension for later retrieval and reuse. To save a task to a file:

1. Click on the **Save** button at the top left of the screen.
  2. A **Save File** dialog will pop up and a name can be specified for the behavior file.
  3. The software will automatically add the `.rbt` extension.
  4. The default saved location is: `C:\Program Files\ER1\behaviors`.
-

---

## If Conditions

One or more conditions can be specified in the **If** part of a behavior. The available conditions are:

- Sight
- Sound
- Time
- Message receive
- Sequence

To select a condition and edit it for your needs, click on the checkbox to the left of it. Until you click the checkbox, you will not be able to edit any of the options within the condition. (Remember that regardless of how many conditions you specify, only one condition must be met for the behavior to be triggered and the actions executed.)

### Sight Condition

Using the **Sight** condition, you can program the robot to react to a certain color or object. To select an object, select it in the pulldown menu.

To have the behavior trigger on a color:

1. Click on the color pulldown menu to select a color.
2. The color pulldown menu also has an dropper icon. Clicking on this icon lets you select a color from the camera image.

*Important Note:* Either an object or a color can be selected as the sight condition, but not both.

### Sound Condition

Using the **Sound** condition, you can program the robot to react to an ambient volume level or to use speech recognition to respond to a spoken word or phrase.

The six radio buttons correspond to six discrete volume levels. The volume levels increase from left to right, but the actual sound level represented by each of the six depends on the microphone volume settings on your robot's laptop. You will need to experiment with your laptop's particular hardware settings to determine which level works best as your behavior trigger.

To have the robot react to a spoken word or phrase, type it into the **Listen For** text box. The ER1 RCC software uses Microsoft Speech SDK 5.1 to recognize the words typed into this text box.

Microsoft's speech recognition works better after it has been trained on the user's voice. To train the speech recognition software, open the **Microsoft Speech Applet** in the **Control Panel** and click on the **Train** button in the **Speech Recognition** tab. The speech software will prompt you from there.

## Time Condition

The **Time** condition allows you to program your robot to perform actions at a certain time or at regular time intervals. The time or time interval is specified using three pulldown menus:

- The first specifies the hour.
- The second specifies A.M. or P.M.
- The third specifies an optional interval. If an interval is specified, it automatically allows the behavior to run more than once.

## Message Receive Condition

This condition allows you to program your robot to react to an email containing expected content. The robot looks for text patterns in the sender address, subject, or body of the email. You can specify one or more of these three fields for the robot to examine. When your robot receives an email with matching text in all the specified fields, it will execute the actions that you have programmed.

## Sequence Condition

You can use the sequence condition to link behaviors together. A behavior with this condition will be triggered when the specified earlier behavior finishes. Behaviors linked together by the sequence condition are called a behavior sequence. Once a behavior set has initiated, no other non-sequence condition anywhere in the 96 behaviors will trigger until the sequence is complete. (In other words, a behavior sequence is not interruptible by any non-sequence condition.)

## Then Actions

There are five kinds of actions available to the robot:

- Move
- Play a sound
- Run a program
- Send an email
- Record a photo or video

Click on the checkbox to the left of the action to select it for the behavior and enable its internal controls.

More than one action can be specified for the behavior, and all actions will execute simultaneously once the behavior is triggered. However, some actions take longer than others, so the actions will most likely finish at different times. The behavior is not considered completed until all specified actions are completed:

- **Move** actions are completed when the robot reach the specified target.
  - The **Play Sound** action is considered completed when the sound file is finished playing.
-

- 
- The **Run Program** action is considered completed when the program starts to run.
  - The **Send Message** action is considered completed when the email is queued for sending.
  - The **Record** action is considered completed when the photo is taken or the video recording is completed.

### Move

This action moves the robot toward a color target or an object target. You can also specify a fixed distance/angle for the robot to travel or rotate.

### Play a Sound

This action plays a sound file, or speaks a phrase using the Text-to-Speech capability. The Browse button can be used to open a dialog box where you can select a sound file. Alternatively, you can type a phrase to be spoken. You can also choose from among the prerecorded sound files that ship with the RCC.

### Run a Program

This action prompts the robot to open a Windows application such as Excel. The Browse button opens a dialog box allowing you to specify the executable you want to run.

### Message Send

This action allows the robot to send an email message to one or more recipients. Fill in the sender, subject, and body fields as needed.

### Record

This action allows the robot to shoot a still photo or record video.

### Trying Out Actions

There is a **Try It** button right on the right side of the screen. Clicking on this button will execute whichever action or actions are specified in the current behavior. Use this button to try out actions to make sure they work as intended before running the actions in a program.

### Running and Stopping Behaviors

The **This Task** section of the RCC software window contains the **Run it** and **Stop** buttons used to execute and halt the behaviors you created in the **If/Then** sections. Click on the **Run it** button to begin running the behavior or behavior sequence you created. The **Stop** button stops any behaviors that are being executed. This button only becomes active while a behavior is being run.

### Behavior Execution

As mentioned earlier, a sequence of behaviors linked together by the sequence condition is uninterruptible. All sequences begin with a behavior that has a non-sequence condition

(sight, sound, time, or message receive conditions). In other words, a sequence has to begin by some external trigger condition.

Consider the following example:

- Behavior 1: **If** see teddy bear **Then** play a sound file: `teddy.wav`.
- Behavior 2: **If** hear the word email **Then** send an email to [user@ISP.com](mailto:user@ISP.com).
- Behavior 3: **If** after Behavior 1 **Then** move forward 10 feet.
- Behavior 4: **If** after Behavior 3 **Then** move backward 10 feet.

If the teddy bear is seen, the sequence of behaviors 1, 3, and 4 will commence. Even if the word email is spoken and recognized when one of these behaviors is running, it will be ignored and Behavior 2 will not run, because the sequence cannot be interrupted. The sequence has to begin with Behavior 1, which has a non-sequence condition (sight).

If a behavior has both a sequence condition and a non-sequence condition, then it can be triggered either way. For example, if Behavior 4, above, is rewritten as follows:

Behavior 4: **If** after behavior 3 or hear sound level 2, move backward 10 feet.

Then the robot may move backward 10 feet on a sound or move backward 10 feet as part of the sequence starting with Behavior 1. However, it will only do one or the other, not both, depending on which condition is satisfied first. If the robot sees the teddy bear before hearing the sound, Behaviors 1, 3, and 4 will execute in order. Even if the sound was heard during or after this execution, it will be ignored. If the sound was heard first, Behavior 4 will execute. Then, if the teddy bear was seen, Behaviors 1 and 3 will execute but not 4. (Assuming you're using the default behavior settings. This is because behaviors are set to run only once by default.)

All behaviors default to run once, meaning that once the behavior has executed, it will not run again until you once again click on the **Run it** button.

A behavior can be set to run more than once, immediately or at specified intervals, in the **Run** tab of the **Behavior Settings** menu. (Click on the **Behavior Settings** button at the top right of the application.) When a behavior is set to run more than once, it will run whenever its **If** condition is satisfied.

If Behaviors 1, 3, and 4, in the previous example above, are set to run more than once, then the sequence will be run whenever the teddy bear is recognized. For example, in the above example, the robot will end up roughly in the same position after each run of the sequence, because it just moves forward and back 10 feet. If the teddy bear is stationary, chances are it will be seen after each run, and the sequence will repeat indefinitely. Sometimes this kind of repetition is undesirable, because some trigger conditions occur more or less continually. Other times you may want this kind of a repetition.

Consider this example:

Behavior 1: **If** see teddy bear **Then** send email to [user@ISP.com](mailto:user@ISP.com).

If this behavior is allowed to run more than once, it will spam [user@ISP.com](mailto:user@ISP.com) with several emails per second, because it will see the teddy bear roughly at that rate and sending the email takes even less time. Teddy's mailbox will fill up with hundreds of emails after just a few minutes. This is why in the **Run** tab of the behavior settings dialog, there is also a **Prevent starting for \_\_ seconds** setting. Once a behavior has executed, it will not run

---

again until the specified time has elapsed. You can specify the minimum run interval in seconds.

Specifying that behaviors can run more than once also allows you to create looping sequences. Again, consider the behavior sequence:

Behavior 1: **If** see teddy bear or after behavior 4 **Then** play a sound file `teddy.wav`.

Behavior 2: **If** hear the word email **Then** send an email to [user@ISP.com](mailto:user@ISP.com).

Behavior 3: **If** after Behavior 1 **Then** move forward 10 feet.

Behavior 4: **If** after Behavior 3 **Then** move backward 10 feet.

(All behaviors set to run more than once).

The only difference between this sequence and the sequence in the first example is that Behavior 1 also has the sequence condition after Behavior 4. This condition causes this sequence to run in a continuous loop even if the teddy bear is only seen once. In this example, once this sequence gets started, Behavior 2 will never be executed because the sequence will run in a continuous loop.

Note that if multiple behaviors are triggered at the same time by the same condition, only the right-most behavior in the behavior tab sequence will be triggered. For example, if behaviors 13 and 17 have a sound level condition set to volume level 1 and 3, respectively, and the sound volume 4 is detected, satisfying both, only behavior 17 (the right-most behavior) will be triggered. If behavior 17 is set to run only once, then the next time the sound condition is triggered for both, behavior 13 will be triggered, because behavior 17 has already run and cannot be triggered again.

## **Behavior Error and Logging**

The software enables logging of events during behavior execution by default and displays the last few lines of the log at the bottom of the application. Logging can be turned on or off by clicking on the **Settings** button at the top left of the application and then selecting the **Log** tab. The generated log file is named `er1.log`, and resides in the same directory as the ER1 RCC software. During the course of executing the task, the robot might encounter an error condition that will cause it to abort the current behavior or the entire task.

Examples of this include encountering obstacles during motion, low laptop battery, etc. If the behavior has to abort due to some error condition, it will log the error if the logging option is selected. If something unexpected occurs during behavior execution, you can look at the log window or check the `er1.log` file to see the cause.



## ■ Chapter 4

---

# Audio, Video and Navigation Controls

The left side of the ER1 Robot Control Center (RCC) software window is used for direct control of the robot, and does not require that any **If / Then** conditions be set.

### Video Display

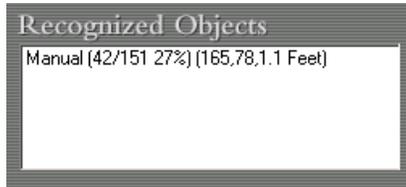
The Video window displays the live feed from the robot's camera.



This video feed changes as the robot moves. The video input is only operational when a behavior is not running (unless the behavior is using the video). You can use one camera for both object recognition and obstacle avoidance, or you can use a separate camera for each. The **Camera** tab in the **Settings** menu lets you change the settings for the cameras and video display.

## Recognized Objects

When the software recognizes an object, it will speak the object's name or a user-specified phrase, and display the name and other information in the object area. See the [Sight Tab](#) section for details on Object Recognition.



Click on the **Settings** button and then the **Sight** tab to turn on/off recognition-related features.

Depending on your **Sight** settings, the object window will display only the best match, or a list of all possible matches. The object area is only active when none of your behaviors are running.

The values displayed in the object window refer to the following:

- Manual is the name of the object that was recognized.
- 42 is the number of features matched between the current video feed and the trained object.
- 151 is the total number of features that were identified in the current video feed.
- 27% is the total percentage of features that were recognized.
- 165 is the horizontal position of the center of the object, in pixels, measured from the top of the image.
- 78 is the vertical position of the center of the object, in pixels, measured from the top of the image.
- 1.1 is the number of feet between the robot and the object recognized in the video feed, based on the distance specified when the object was captured.

## Obstacle Camera (optional)

The optional second camera is used by the software for simple obstacle avoidance. This window will only be active if you have a second camera installed on your ER1. In order to use your second camera for obstacle avoidance, you must calibrate it. Otherwise, you will find that your obstacle avoidance simply is not avoiding obstacles. To calibrate your obstacle avoidance camera, whether it is the original ER1 camera or an optional second camera, follow the procedure described in the [Obstacle Avoidance Tab](#) section of “[Settings](#)”.

## Mute Sound Checkbox



The **Mute Sound** checkbox mutes sounds coming from the robot's laptop.

## Navigate



The navigation commands are used during teleoperation to direct and describe the robot's motions. The x,y coordinates are read-only values

representing the approximate location, in centimeters, of the robot in relation to its starting point. To zero these values, you must close and then re-open the RCC application or turn the robot's battery off and then back on.

To use the buttons, simply click on them with the mouse. These buttons will only work as long as you hold down the mouse key. The robot will stop and await new input when you release the button. These buttons are inactive when a behavior is being executed.

You can also move the robot by using the arrow keys on your keyboard. Single keys, a combination of keys or a combination of the arrow keys and the RCC buttons are allowed. This way, you can move the robot forward while turning to the right and so forth. Note that buttons and keys that cancel each other out, for example left arrow and right arrow keys, should not be used together.

*Important Note:* UNPLUG THE LAPTOP'S CHARGER BEFORE INSTRUCTING THE ROBOT TO MOVE. The robot should detect whether or not the laptop is plugged into a wall outlet and should not move until the laptop is unplugged. However, take care to ensure that the laptop is unplugged, because if the robot fails to detect that the laptop is plugged in, it will move in response to your commands until it reaches the limit of the laptop's power cord, then will continue to try to move, potentially damaging the laptop, the laptop's charger, the robot, and the electrical outlet. This could start an electrical fire.

## Battery



The Battery display indicates the level of charge is left in the power module, and when you need to recharge. If the power module is fully charged, it should last for about 3 hours of driving time. This icon will be disabled if the robot is disconnected from the RCC software.

Note that the robot will be disabled when the battery falls below 10% charge. See our website at [www.evolution.com](http://www.evolution.com) for information about acquiring additional power modules.

*Important Note:* The rate at which the power module loses its charge is highly dependant on how much driving the robot is doing, the weight of the laptop and any accessories, and the surface on which the robot is driving.

## Stop Recognizing Objects



This option, if selected, specifies that RCC stop recognizing objects. Object recognition takes a lot of processing power, so turning this feature off will speed up other ER1 activities.

Note that when the **Stop Recognizing Objects** box is checked, capturing objects is still enabled. This means that you can teach the robot to recognize many different objects without having the software stop after each capture to retrain the image library. When you're done with capturing images, uncheck the **Stop Recognizing Objects** box and the software will process all the images at once. This is a great time saver. Similarly, you can also use the checkbox to delete many images quickly. Remember to uncheck this box when you want your robot to recognize objects again!



## ■ Chapter 5

---

# Function Buttons

The function buttons located at the top left-hand side of the Robot Control Center (RCC) software window allow you to create a new task, change overall behavior settings, save or retrieve behaviors you've created, and get help.



## New Button

The **New** button allows you to load a blank behavior set. To use this button, click on it. If you have unsaved changes to an existing behavior set, the software will first ask you if you want to save your changes. A dialog box will appear asking if you would like to save the changes you have made to the current set of behaviors. Click on the **Yes** button (the **No** and **Cancel** buttons can be used to go back to the main RCC screen). A **Save As** dialog box appears. Select a location for your behavior set and type in a name. Make sure that the name is unique and descriptive so it is easy to find as you add more behavior sets. The file is saved as a `.rbt` file.

## Open Button

The **Open** button allows you to retrieve and load a previously saved .rbt file. To do so:

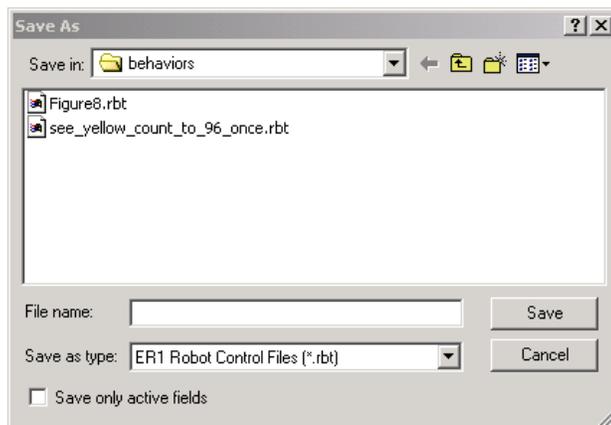
1. Click on the **Open** button.
2. If you've made changes since your last .rbt save, a dialog box will appear asking you if you want to save your current settings before retrieving. Select **Yes** or **No**. If you select **Yes**, a dialog box will appear asking you to name your file. Do so as needed, then click **Ok**.
3. A standard Windows-like file selection dialog box appears.
4. Navigate to the .rbt file you would like to use.
5. Select the file.
6. Click **Ok**. The robot behavior set that you have selected will be displayed on the screen.

## Save Button

The **Save** button saves the current settings to disk using a standard Windows dialog box. Click on the **Save** button and a **Save As** dialog is displayed. Type in a name for your behavior set. Make sure that the name is unique and descriptive so that the file will be easy to find as you add more behavior sets. Click on the **Save** Button. You may click on the **Cancel** button at any time to cancel and return to the main RCC screen. The default directory is named `behaviors` and is in the ER1 program directory. If you have already named your behavior set, just type `Ctrl S` to save your behavior set. The behavior set will be saved with no feedback.

## Save Only Active Fields

In the Save dialog box there is a new checkbox entitled, **Save only active fields**. This is disabled by default. If you click this checkbox, only the active settings will be saved. For example, if you filled in some **Move** settings, then deselected the **Move** checkbox, then used **Saved only active fields**, your **Move** settings would not be saved. Non-saved settings are not recoverable.



---

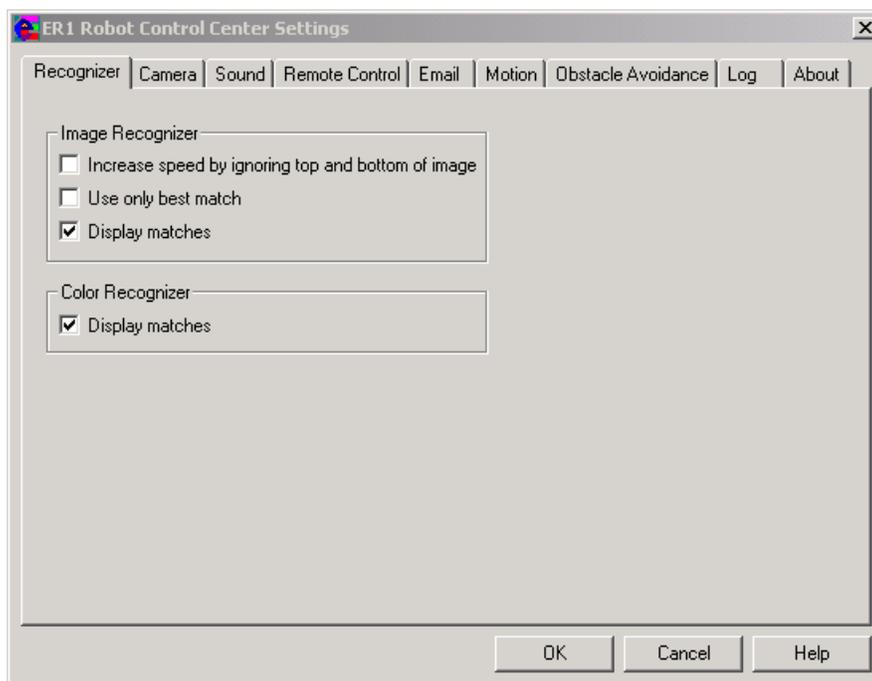
Note that if you use CTRL S to save, it will use the **Save only active fields** value (active/inactive) that was used the last time that behavior was saved.

The **Save only active fields** preference is saved as part of the behavior set.

## Settings Button

The **Settings** button allows you to customize default settings for the entire 96-behavior set. (See the “[Behavior Settings](#)” section for information on how to change the settings for one particular behavior.) Advanced users may wish to change settings that will improve speed of performance, size and resolution of image, and interval for repeating the sound file for a given image. For most users, no adjustment of the Settings is necessary.

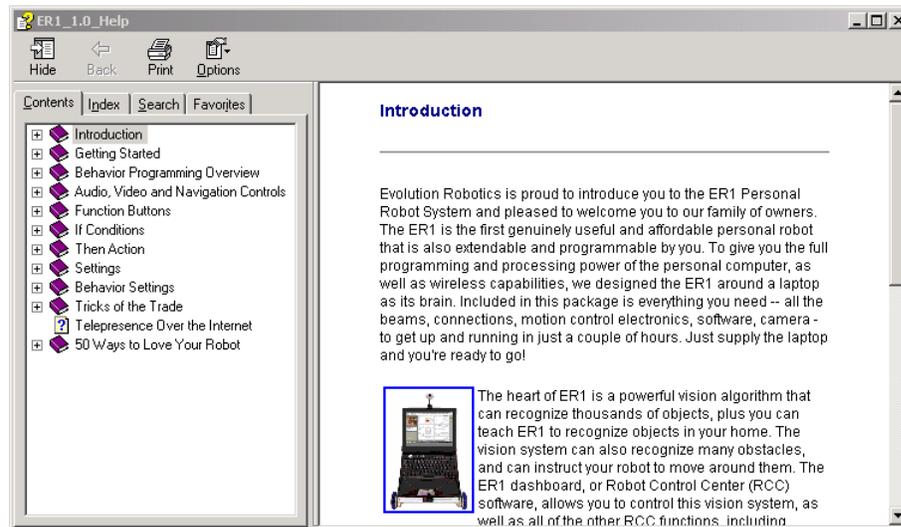
The RCC **Settings** are described in “[Settings](#)”.



## Help Button

Click on the **Help** button to get help on any of the various functions of the RCC software. The help screen can also be accessed by using the F1 button on your key board.

The help topics are organized in the same manner as this Guide. You can find help topics by clicking on a section, using the index, typing in a search topics or going to your list of favorites.



## ■ Chapter 6

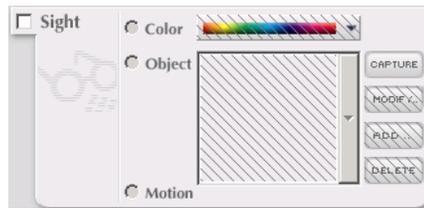
---

# If Conditions

There are five **If** condition buttons/dialog boxes, each of which corresponds to an independent condition on which to trigger one or more actions. The five conditions, described below, are:

- Sight
- Sound
- Time
- Message Receive
- Sequence

## Sight Condition



The **Sight** section allows you to set visual conditions. Click on the **Sight** checkbox to make it active.

## Color Selection Tool



This tool allows you to specify a specific color as the trigger condition.

You can choose from the 256 colors on the pull-down color palette, or you can use the **dropper** tool to capture a color displayed in the **Video** area. Either way, when the robot sees an object of that color, the corresponding action is triggered.

When you click on the **dropper** tool, a pop-up menu displays a frozen image of the current video feed, and your cursor changes to a dropper. Move the dropper to the pop-up window, and click on the color you want to specify. That color appears in the color bar. See “[Trying Out the Software](#)” for a tutorial that uses this function.

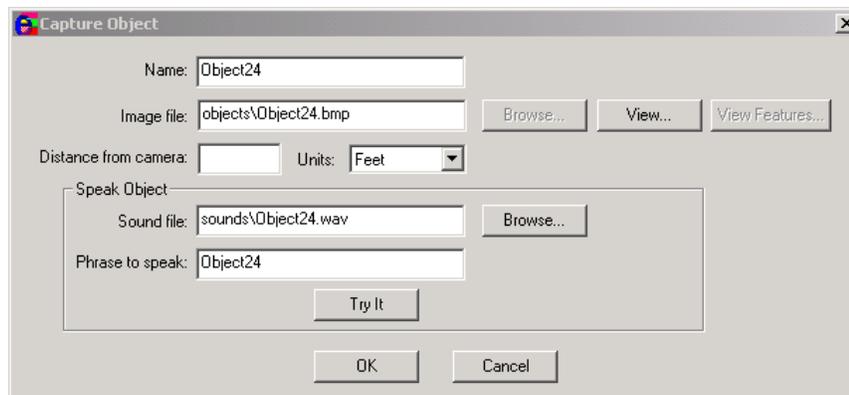
## Object

The **Object** pulldown menu allows you to specify visual recognition of an object as the trigger condition. You can select from the scrollable list of image library thumbnails displayed in the viewer with the filenames under each, or you can capture a new image. (Note that the most recently trained item is at the top of the pulldown menu list.) When the robot recognizes the specified object, the corresponding actions are triggered.

To the right of the viewer are four buttons:

## Capture Button

The **Capture** button allows you to capture an image from the robot’s video feed. If you are using a wireless connection to control your robot, note that the picture file will be stored on the robot’s laptop.



---

### *How to Capture an Image*

1. Place an object in front of the robot's camera. This will become one of the robot's learned objects.
2. Watch the RCC's video display while adjusting the object so that the image fills most of the display.
  - To maximize the robot's recognition of a learned object, the object should be placed on a neutral, not patterned, background, with good lighting, and no other objects in view.
  - For even more foolproof recognition, you can save multiple images of the same object from slightly different perspectives and variable lighting conditions.
3. When the video displays the image the way you want it, use the mouse to click the **Capture** button. A dialog box appears:
  - Click on **View** if you want to see a thumbnail of your captured image.
  - In the **Name** field, type in your name for the object image. This is the name the RCC software uses to refer to this object image. It is NOT the Windows filename for the image.
  - In the **Image file** text field, a default name is automatically filled in. This image filename is the official Windows filename for the .bmp file containing the image. You can leave the name as is, or edit the filename. If you want to edit the filename, you must do so at this time. After you quit the RCC software application, the **Image** filename can no longer be edited. All files are saved into RCC's directory. The default is: c:\Program Files\ER1\Objects.
4. In the **Distance** and **Units** fields, type in the number and select (from the pull-down menu) the correct units (inches, centimeters, etc.) of the distance between the object and the camera at the time of capture. This enables estimation of actual distance when you ask the robot to stop a specified distance from a recognized object. If you have a zoom lens on your camera, and you change its setting after you've set an image distance, the ability to judge distance correctly will be compromised.
5. In the **Sound file** textbox, you can specify a sound file to be played when the object is recognized, or you can specify a phrase in the **Phrase to speak** textbox. If you want to specify a sound file, remember to specify the path and filename. The RCC software accepts .wav, .mp3 and .wma sound files. If you aren't sure of the file's location, use the **Browse** button to find the appropriate file.
6. Click on the **Ok** button. The RCC software adds the image and associated sounds to its database. A pop-up window displays: "Training, please wait" followed by "Saving image file". Unless you have "Disable Object Recognition" checked.

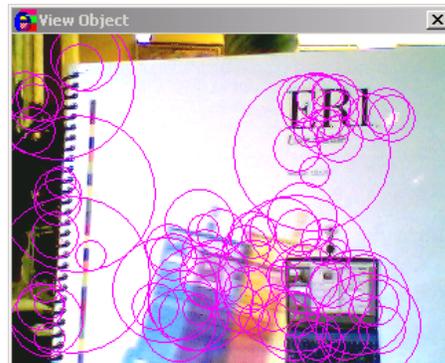
### **Modify**

The **Modify** button lets you change information associated with a stored image. Note that this function does NOT let you edit the image itself. To change information for a specific graphic, display that image in the **Viewer** by scrolling through the graphics files using the scrollbar to the right of the display. Click on the image to select it. Click on the **Modify**

---

button to display the **Modify** dialog box for that image. This dialog box allows you to change/select:

- **Name** - The ER1 image name, which is NOT the Microsoft Windows filename.
- **Image** - This textbox specifies the Windows filename. (To find a filename on your system, use the **Browse** button. You can then replace an existing, named image with your new image. Note that you cannot change the name of a Windows filename in this manner.)
- **View button** - Click this to see an enlarged version of the image you selected. Click on the x in the upper right corner to close this window.
- **View features** - The **View features** button allows you to see the features that RCC uses to recognize an object. All of the features are surrounded by purple circles. When you are done viewing the image's features, click the x in the upper right corner to close the viewing window.



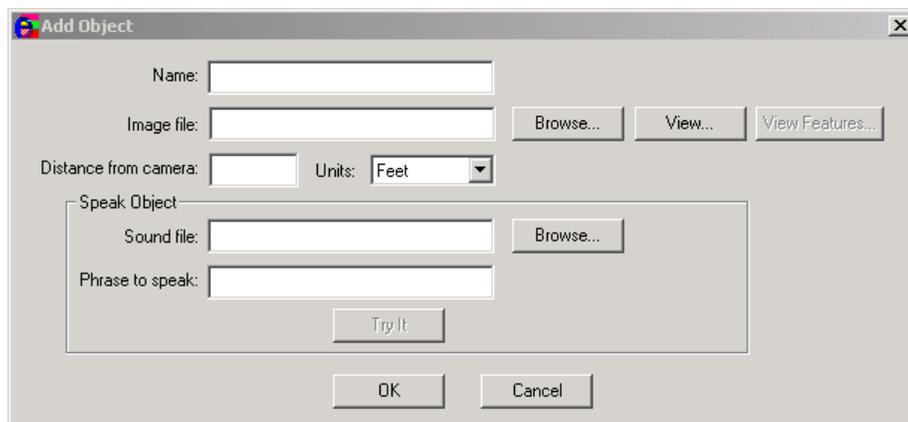
- **Distance** - This textbox is used to specify the distance between the object and the camera at the time of image capture. (You can change the default distance through the **Settings** menu.) This distance is later used when you program the robot to move to the object and stop a specified distance away.
- **Units** - To specify a unit type, use the pulldown menu. Choices are: Feet, Inches, Meters and Centimeters.
- **Speak Object** - This section is used to associate a sound file or a phrase to speak with the image file.
  - **Sound File** - To specify a sound file to be played when the image is recognized, type in the path and filename of a sound file. You can also use the **Browse** button to specify the file. RCC can use .wav, .txt, .mp3 and .wma file formats.
  - **Phrase to Speak** - To have RCC speak a phrase when the image is recognized, type the desired verbiage in this textbox.

When all of your adjustments have been made, click the **Ok** button. The changes you have selected will be implemented. You may cancel your changes by clicking on the **Cancel** button.

---

## Add

The **Add** button option allows you to add an existing image to ER1's library of recognized objects. Click on the **Add** button and the **Add Object Dialog** box will appear.



It has the following parameters:

- **Name** - This will be the RCC, not the Windows, filename for the object.
- **Image file** - This is the Windows path and filename for the image to be added. If you are unsure of the path, use the **Browse** button to find the file. Once the file has been specified, you may view it by clicking on the **View** button. Click on the x in the upper right hand corner to close the image. Use the **View Features** button to see the features that RCC is using to identify an object. The features will be outlined in purple circles. Note that RCC must have already been trained on the image in order to have identified features.
- **Distance from the camera** - This textbox is used to select the number of units the object is from the camera. Be as precise as possible, as the robot will use this information to determine its distance from a target.
- **Units** - To specify a unit type, use the pulldown menu. Choices are: Feet, Inches, Meters and Centimeters.
- **Speak Object** - This section is used to associate a sound file or a phrase to speak with the image file.
  - **Sound File** - To specify a sound file to be played when the image is recognized, type in the path and filename of a sound file. You can also use the **Browse** button to specify the file. RCC can use .wav, .txt, .mp3 and .wma file formats.
  - **Phrase to Speak** - To have RCC speak a phrase when the image is recognized, type the desired verbiage in this textbox.

## Delete

Allows you to remove an image from ER1's library of recognized images. Before it is deleted, you will be asked to confirm that you wish to delete it. To delete the image, click on the **Yes** button. To cancel the delete function, click on the **No** button.

## Motion

This radio button, if selected, sets the motion detection feature. Motion detection can be customized in the [Sight Tab](#) of the **Behaviors Settings** menu.

## Sound Condition

Click on the **Sound** checkbox to make this condition active. You will need a microphone (not included with your ER1) to use this functionality.



The robot can be programmed to react based on one of two sound conditions:

- **Listen for** - Use this textbox to type the word or phrase that the robot should listen for.

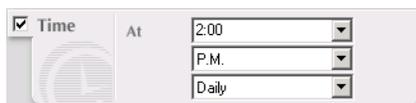
*Important Note:* In order for speech recognition to work accurately, you must take a few minutes to use the Microsoft® Windows Control Panel > Sounds > Speech Recognition submenu to train the software to recognize your microphone, your voice and your speech patterns. Until you do this, the speech recognition quality will be poor. See your Microsoft documentation for instructions on how to do this.

Speech recognition works without training, but improves in accuracy significantly with training on you voice. Speech recognition is also considerably enhanced with a noise cancelling headset microphone. This will either require a long cable or a wireless microphone system, both optional accessories available from many sources. Both of these help significantly. Short of that, you must be quite close to the microphone for speech recognition to work, because of all the extra noise and reflections that come from the room where the robot is located. Consult one of our Customer Support specialists before purchasing a microphone.

- **Level** - Select one of the six sound levels by clicking on one of the six radio buttons. When the robot hears a sound that crosses the threshold sound level selected, the behavior's **Then** condition(s) will be triggered. You might want to test the level you have chosen to determine the appropriate level.

## Time Condition

Click on the **Time** checkbox to make this condition active.



Use the three drop-down menus to have ER1 perform an action at a specific time, or at a regular interval (e.g., once, or hourly).

- The first drop down menu lets you choose time increments of one hour (1:00, 2:00, etc.). If these times are not suitable, type in an exact time (1:12, 2:17, etc.). Valid

.....

times begin with a number from 1 to 12, have a colon, then a number from 0 through 59.

- The second drop-down menu lets you specify A.M. and/or P.M. (Note: Standard Military Time is not supported.)
- The third drop-down menu lets you choose when and how often you would like the associated action(s) repeated: Once, hourly, twice a day, daily or weekly.

A couple of tips:

- If the time specified in the upper combo box is not valid, or you do not select A.M. or P.M., then the behavior will not trigger unless you specify a repeat interval. If you do specify a repeat interval, the behavior will trigger every time the repeat interval is reached after you press the **Run it** button. For example, if you leave the upper combo box blank, then specify **Every 10 minutes** in the bottom one, the behavior will trigger 10 minutes after you press the **Run it** button, then every 10 minutes after that.
- If the time specified in the upper combo box is valid and you choose A.M. or P.M., then the timed behaviors are initially triggered as follows: If the time at which you press **Run It** is already later than the time at which the behavior would run on that day, the behavior is scheduled for the next instance of the repeat interval after the missed time. If you do not choose a repeat interval or select **Once**, then the behavior occurs 24 hours after the missed time. If the time has not yet occurred that day, the behavior is first triggered when the time is reached. Here are some examples which will clarify these rules for you:
  - You schedule a behavior to run at 10:00 A.M. with a repeat interval of **Hourly**. You press the **Run it** button at 10:05 A.M. Since the initial time for the behavior to run has already passed for the day, the behavior will next run at 11:00 A.M.
  - You schedule a behavior to run at 10:00 A.M. with a repeat interval of "Once". You press the **Run it** button at 10:05 A.M. Since the initial time for the behavior to run has already passed, the behavior will next run at 10:00 A.M. the following day.
  - You schedule a behavior to run at 10:00 A.M. with a repeat interval of **Hourly**. You press the **Run it** button at 7:55 A.M. Since the initial time for the behavior to run has not passed for the day, the behavior will next run at 10:00 A.M. and then at 11:00 A.M., 12:00 P.M., etc.

When you press the **Stop** button, all scheduled behaviors are abandoned. The next time you click on **Run it**, they are rescheduled based on the above rules, starting from the time you pressed the **Run it** button.

Information about the scheduling of timed behaviors is written to the log file in the ER1 folder of your program files, so it is easy to see when your behaviors will run next. Also, if another behavior is in progress when a time triggered behavior is scheduled to run, the time triggered behavior is postponed until after the current behavior (and any behaviors sequenced to run after it) are done.

.....

## Message Receive Condition



Click on the **Message Receive** checkbox to make this condition active.

If your robot's laptop has a wireless network card and access to a wireless network, you can program ER1 to act on email from your Evolution Robotics RobotMail account based on a determination of whether the sender, subject, or body contain substrings (i.e., text) specified by you to trigger the behavior.

Fill in one or more the following:

- **Sender** - One or multiple email addresses from which to expect email. Aliases are not supported. A full email address must be provided.
- **Subject** - Exact text to look for in the Subject line of the email.
- **Body** - Exact text to look for in the body of the email.

To set how often you'd like the robot to check email - for example, once every minute, or once every five minutes, click on **Settings** button and the **Email** tab.

To open a RobotMail account:

1. When you opened the robot's box, you should have seen a registration card instructing you to go to the [www.evolution.com](http://www.evolution.com) web site. If you have not already done so, do it now. The first time you log in, you will need the default username, *evolution* and the default password, *er1*.
2. On the web site, you will be directed to the ER1 community area.
3. You will be prompted to choose a username and password for your *evolution.com* login. If the name you specify is not already taken, it will be assigned to you.
4. On the web page, there will be a description of the purpose and limitations of robot mail near or on the sign up page. Read it now.
5. You will be prompted to select a [XXXXXX@robotmail.com](mailto:XXXXXX@robotmail.com) account name for your robot. (This account name must be different than the username/password used for the *evolution.com* login. You can enter up to a 16 character name. If the name is not already taken, it will be assigned to you. If you purchased more than one robot, you will be allowed to choose multiple robot account names; one for each robot.)
6. Once the robot has been configured properly as described in the "**Settings**", the robot will monitor your RobotMail remotely for email.
7. Send the email to your new RobotMail account to trigger an action as described earlier.
8. When the robot notices that it has received an email in its RobotMail account, it will review the address, content, and subject of the email in an attempt to identify trigger words.
9. If trigger words are identified, the corresponding action(s) are performed.

---

## Sequence Condition



Click on the **Sequence** checkbox to make this condition active.

This condition lets you specify one behavior's completion as a trigger for another behavior's execution, which allows you to create a sequence of behaviors to achieve a task. The **Run After** menu lists all available behaviors. Select a behavior from the pulldown menu. For an example of how to link behaviors together to complete a task, see the [Making the Robot Move](#) section.



## ■ Chapter 7

---

# Then Action

The following section describes the actions that the robot can perform. These actions must be preceded by a trigger event that is specified in the **If** column. Remember, you can specify multiple actions to occur as a result of one trigger event. See the [Behavior Programming Overview](#) for a description of order of precedence and behavior actions.

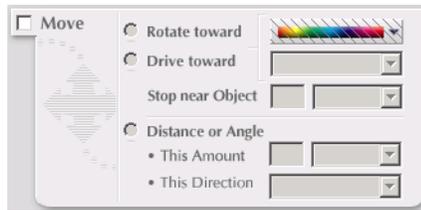
### **Reset and Try It Buttons**

The **Reset** button at the top of the action list blanks all the behavior fields to essentially create a new behavior set. Remember, there is no **Undo** button.

The **Try It** button at the top of the action list allows you to immediately try out an action you've programmed, so that you can verify that it works correctly before running the behavior.

The actions to take as a result of a triggered **If** condition are described in the following sections.

## Move



Click on the **Move** checkbox to make this action active. Options are:

- **Rotate toward** - This parameter, if set, tells the robot to rotate until it sees the object or color you specified using the adjacent color bar or the text pull-down menu adjacent to the **Drive toward** option. The color selection tool operates in the same

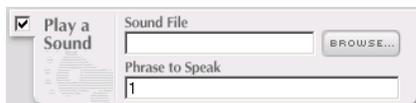
manner as the **If** color selection tool. This action is considered complete when one of the following two conditions are met:

- The robot rotates and successfully finds the specified object or color, then stops when facing the object or color.
  - The robot completes a 360 degrees rotation and fails to spot the specified object or color.
- **Drive toward** - When set, tells the robot to drive forward towards the object or color you specified using the color selection tool or the text pull-down menu. Tell the robot when to stop by specifying a distance and units in the next set of pull-down menus. Note that the robot will not move until a value greater than 5cm or 2" is specified. This action is considered complete when one of the following two conditions are met:
    - The robot drives toward the specified color or object and successfully stops at the distance specified by the user.
    - The robot fails to see and drive toward the specified color or object after 30 seconds.
  - **Stop near Object** - This parameter has two fields that allow you to specify how close the robot should come to a specified object/color before stopping. Type in a numerical value, and use the pull-down menu to specify the units. This value should not be set to less than 5cm, so as to avoid collisions.
  - **Distance or Angle** - This lets you specify the direction in which the robot should move, and/or a set distance it should travel. This includes rotational directions.
    - **This Distance** - As for **Stop near Object**, specify a numerical value and use the pull-down menu to specify the units (feet, inches, meters, centimeters or degrees). Note that this pull-down menu lists **Degrees**, allowing you to specify the number of degrees in which you would like the robot to turn. The number of degrees can be set to any positive number. Note that for every 360 degrees specified, the robot will complete one full rotation.
    - **This Direction** - This pull-down menu lets you specify a direction of travel. The options are: forward, backward, right, left.

*Important Note:* UNPLUG THE LAPTOP'S CHARGER BEFORE INSTRUCTING THE ROBOT TO MOVE. The robot should detect whether or not the laptop is plugged into a wall outlet and should not move until the laptop is unplugged. However, take care to ensure that the laptop is unplugged, because if the robot fails to detect that the laptop is

plugged in, it will move in response to your commands until it reaches the limit of the laptop's power cord, then will continue to try to move, potentially damaging the laptop, the laptop's charger, the robot, and the electrical outlet. This could start an electrical fire.

## Play a Sound



Click on the **Play a Sound** checkbox to make this action active.

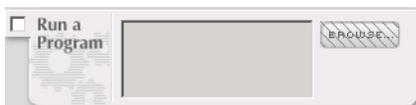
This action uses the software's built-in Text-to-Speech module. (You can change which voice is used by clicking on the **Settings** button and selecting a voice type from the **Sound** tab.) Options are:

- **Sound File** - This textbox is used to specify the path and name of a sound file. If you are unsure of the path, use the Browse button to find your file. Remember that you can create and use your own files to use in a behavior. RCC can use .wav, .txt, .mp3 and .wma file formats.
- **Phrase to Speak** - Type in what you would like the robot to say. The system will first check to see if the file name entered exists on the hard drive. If so, the system plays it. If not, the phrase is passed onto the Text-to-Speech module.

## Run a Program

Click on the **Run a Program** checkbox to make this action active. Use the **Browse** button to find the executable file, \*.exe, that you would like to execute. The selected path and file name will be displayed in the textbox.

You can also pass parameters to the launched program, but the program name (or .lnk file name) must be surrounded by double quotes (") for this to work if it has spaces in the name. The **Browse** dialog box takes care of this automatically for you.



This action allows you to specify a program to run when the **If** condition is triggered.

A couple of hints about using this feature:

- If you need to specify a path that has spaces in it, for example "C:\My Documents\etc." you must put quotes around the full path.
- If a document has a valid file association, for example Microsoft Word is used to open .doc files, you can just specify that file in the textbox. The program associated with it will open automatically.
- Web pages can also be opened without specifying a program. For example you could type [www.evolution.com](http://www.evolution.com) in the textbox. Your default internet browser will be used to open the webpage.
- If the file you wish to open is in the same directory as er1.exe, you do not need to specify the path to the file.

## Message Send

Click on the **Message Send** checkbox to make this action active.



Use this option to create an email to be sent to one or more specified addresses upon triggering of an **If** condition. The message send occurs immediately if the **If** condition is true.

To use this option, your robot's laptop must have access to the Internet. You also must have an email account set up with your ISP. Evolution Robotics RobotMail will not send outgoing messages.

The fields are:

- **To** - Enter one or multiple email addresses to which the message should be sent. Aliases are not supported; a full email address must be provided. Addresses should be separated with semicolons. Be careful in specifying behaviors that have the robot send emails. It would be easy to accidentally send multiple spurious emails if your trigger conditions are met multiple times. Click on the **Behavior Settings** button and then the **Run** tab to set the way this behavior can be run.
- **Subject** - Enter the subject of the email message.
- **Body** - Enter the body of the email message.

## Record



Click on the **Record** checkbox to make this action active.

Use it to take and save a photo or video upon triggering of the **If** condition. You can then have the photo or video emailed to one or more addresses.

To use email, your robot's laptop must have an Internet connection. You also must have set up an email account with your ISP.

- **Take Photo** - This parameter tells the robot to take a snapshot of the video image once an **If** condition is satisfied.
- **Shoot Video** - Tells the robot to record the video image for a specified amount of time (5 to 60 seconds) once the **If** condition is satisfied.
- **Then Name It** - Type in a filename for the recorded photo or video. The software will store the file on the hard drive under that name. If no directory path is specified, the image or video file will be stored in the same directory as the RCC executable.

**WARNING:** If a behavior is run more than once, a video or photo file of the same name will be created multiple times. Each time, the new file will overwrite the old file. This will occur with no warning messages. If you wish to save all of the

---

video or photo files created, then before running the behavior set, click on the tab for the appropriate behavior, then click on the **Behaviors Settings** button and select the **Video** tab. Make sure that the **Add a sequence number to each video** checkbox is selected. This will append a number to the filename that you have specified each time the behavior is run, creating a unique filename for each new file and preventing them from being overwritten by any subsequent runs of the behavior set. For example, if you specify the filename "family", the resulting files will be named: family1.bmp, family2.bmp, family3.bmp, ...etc.

You can also specify a starting number, for example, "50", in which case the resulting files will be named: family50.bmp, family51.bmp, ...etc.

- **Then Email It To** - Enter one or multiple email addresses to which the message should be sent. Aliases are not supported; a full email address must be provided.



## ■ Chapter 8

---

# Settings

The **Settings** menus let you change a wide range of settings that apply to all 96 behaviors. (If you want to customize settings for a particular behavior (or behaviors), use the **Behavior Settings** button.)

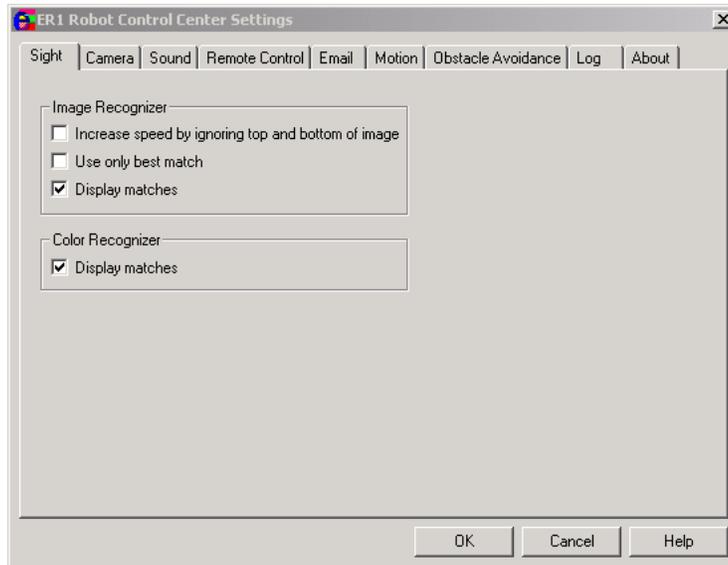
To use the **Settings** menus, click the **Settings** button. A dialog box appears. Along the top of the box are a number of tabs: [Sight Tab](#), [Camera Tab](#), [Sound Tab](#), [Remote Control Tab](#), [Email Tab](#), [Motion Tab](#), [Obstacle Avoidance Tab](#), [Log Tab](#) and [About Tab](#).

Note that if you are controlling your robot by wireless connection, the **Settings** that you will see from the remote computer are those set on the robot's laptop.

## Sight Tab

This tab is used to fine-tune aspects of the software's object recognition capabilities. These are used to recognize objects that the user has specified.

Most users will not need or want to adjust these values. Experienced users may want to experiment with them.



Here are the Image Recognizer options:

- **Increase speed by ignoring top and bottom of image** - Checking this box reduces the number of pixels required for processing recognition data by chopping off the top and bottom of the image. This speeds up the recognition process. The trade-off is that increased speed recognition only works reliably with objects that are level with the camera. It can fail to recognize objects that are a little above or below the camera. After you change this value, the software must recalibrate the saved images before continuing. This may take several minutes.
- **Use only best match** - This checkbox toggles the recognition capabilities between the **best match** and **normal match** object recognition strategies. If this option is selected, the **Recognized Object** text box will only list the image with the most matched features. If this option is not selected, the **Object** text box will list all possible matches, ranking them in descending order from top to bottom according to the confidence measure.
- **Display matches** - This checkbox, if selected, specifies the display of a blue rectangle around each recognized object. Note that selecting this option will slow the obstacle avoidance feature.

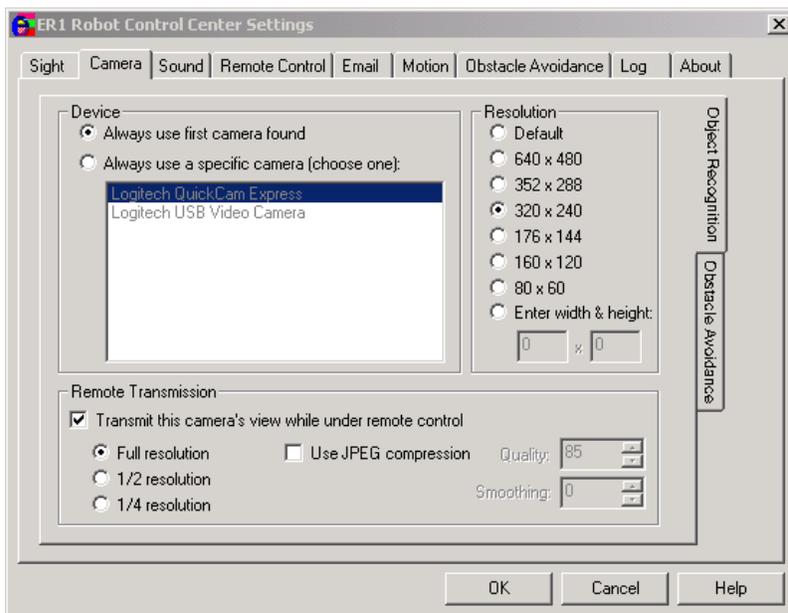
There is one Color Recognizer option:

- **Display matches** - This option lets you specify whether or not to show red boxes around colors recognized by the robot in its environment. To improve recognition speed, the robot does not look for behavior trigger colors once the behavior starts -- only colors needed for the specific behavior that was triggered are recognized.

Once the behavior is finished, the RCC goes back to looking for the trigger colors (if any are specified).

## Camera Tab

This tab allows you to set the camera parameters for both object recognition and obstacle avoidance.



The camera shipped with your robot can be used by the robot for both obstacle avoidance and object recognition. However, the software allows you the option of using a separate camera for each of these functions.

- The object recognition camera is best mounted at the robot's "eye level".
- The obstacle avoidance camera is best mounted angled toward the floor in front of the robot. Contact Evolution Robotics about getting additional hardware to mount your camera.

Click on one of the two right-hand tabs to select a camera functionality. Only one camera tab can be active at one time. Both camera tabs will be selectable even if the robot only has one camera.

## Device

This section is used to specify the camera that the robot will be using for object recognition and/or obstacle avoidance. For each camera, parameters are:

- **Always use first camera found-** If this option is selected, the default camera will be used. This is the first camera listed.
- **Always use a specific camera-** This option allows you to specify that a particular model of camera is attached to the robot. It displays the device drivers it finds by querying Windows using DirectX 8.1 (this is included in the installation). The

software default is configured for the camera that ships with your robot. You should only need to adjust this if you replace the default camera with one of your own.

## **Resolution**

This box allows you to set the resolution of the camera. If you chose the **Default** option in the **Device** box, the current resolution will be selected.

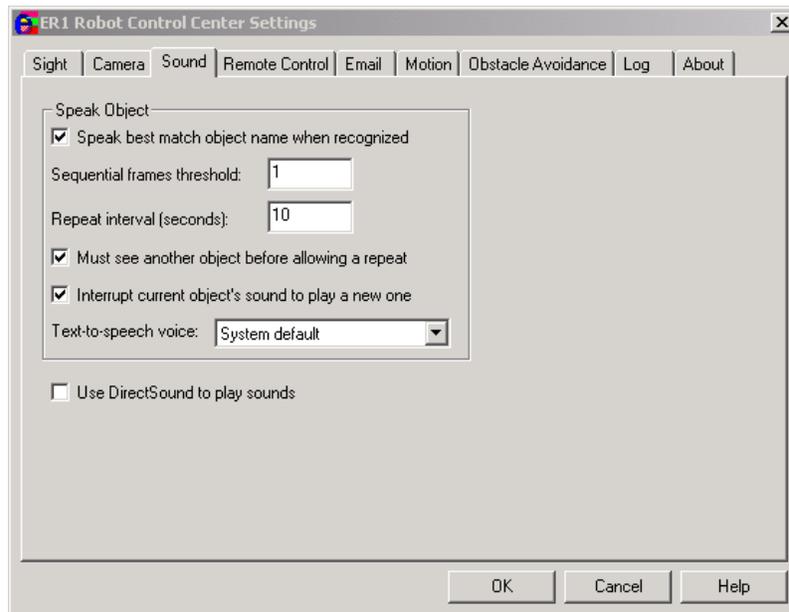
## **Remote Transmission**

Another useful feature is a checkbox which lets you individually turn on/off each camera's view under remote control. The checkbox is labeled **Transmit this camera's view under remote control**. This setting is on the **Camera** tab, rather than the **Remote Control** tab, for several reasons, including that it is a "per camera" setting and that it is mirrored across the network under remote control (whereas the **Remote Control** tab is not).

- **Full resolution** - This parameter specifies to transmit the camera feed at full resolution. Note that even if this value is set, the image may still be scaled. This occurs because the RCC's display window is a 240 x 180 image. If your camera's resolution is higher than this, the extra pixels above this value are not used, so they are not transmitted. Remember, the higher the number of pixels, the slower the transmission and processing, but the higher the quality of the picture.
- **1/2 resolution** - This parameter specifies to transmit half of the columns and half of the rows of captured pixels.
- **1/4 resolution** - This parameter specifies to transmit a quarter of the columns and a quarter of the rows of the captured pixels.
- **Use JPEG compression** - This parameter specifies whether to use JPEG compression. Compressing the images allows them to be sent more quickly, but it has a negative effect on the quality of the camera images. Also note that JPEG compression uses more CPU time to compress the images, so if you have a fast connection, you may get better frame rates from the robot by not using compression.
  - **Quality** - This parameter allows you to specify the percent of the optimal possible quality at which to save the video file. Specifying a higher value will yield higher resolution and selecting a smaller value will yield lower resolution. Valid values are 0-100. The default value is 85.
  - **Smoothing** - One of the side effects of JPEG compression is jagged edges. This parameter allows you to correct for that. Valid values are 0 to 100. The default is 0.

## Sound Tab

The **Sound** tab allows you to customize the robot's vocalization and audible output.

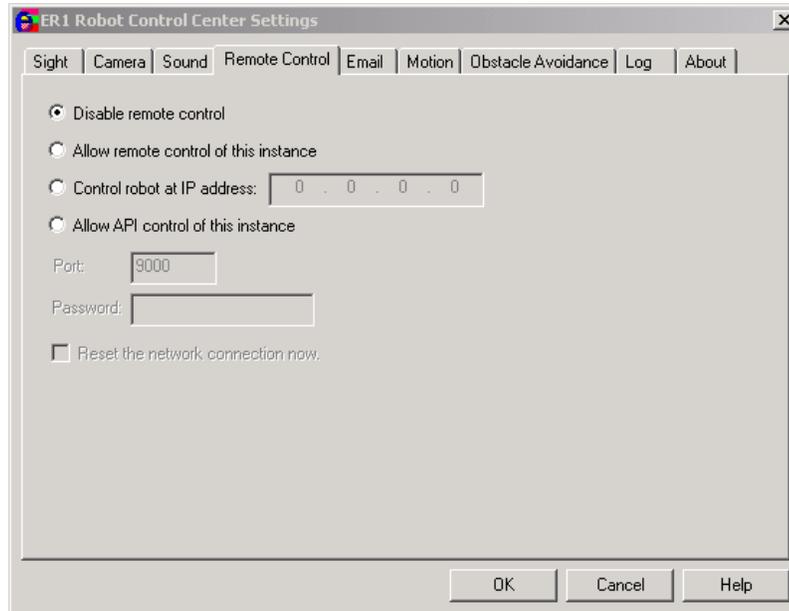


Options are:

- **Sequential frames threshold** - The minimum number of consecutive video frames in which an object is recognized before triggering the sound playback.
- **Repeat interval** - This option specifies the number of seconds in between the robot's sequential acknowledgements of a recognized object. The default value is 10 seconds.
- **Must see another object before allowing a repeat** - If this checkbox is selected, the robot will not vocally identify an object more than once per encounter. This option is set to off by default.
- **Use DirectSound to play sounds** - If this option is selected, Microsoft® DirectSound® is used to play the robot sounds. This option is set to off by default. If DirectSound is not selected, it uses the system's PlaySound() API, which has some disadvantages. One is that sounds start up more slowly, especially the first sound that is played in a session. Another is that you cannot maintain interruptability (e.g. the **Stop** button) and easily tell when a sound is done, so more processing is required to determine when a sound is finished.
- **Interrupt current sound to play a new one** - This option allows the robot to interrupt its audio output to play a new sound. For example, say your robot is playing a CD and it sees the dog. It can stop to say "Dog" and then continue playing the CD. If this box is not checked, sounds are played sequentially. This checkbox is selected by default.
- **Text to speech voice** - This pull-down menu is used to select the voice that the robot will use to read text. The system default voice is "default voice".

## Remote Control Tab

This tab is used to enable, disable and customize your remote control parameters. Before setting these options, you will need to configure your remote computer and your robot's laptop computer so that they can communicate with each other.



Here's how:

1. Install a wireless network card in your remote computer according to the manufacturer's directions. Do the same for your robot's laptop computer.
2. Install PCMCIA drivers on your remote computer according to the manufacturer's directions. Do the same for your robot's laptop computer.
3. Install the drivers that came with your wireless card on your remote computer, according to the manufacturer's directions. Do the same for your robot's laptop computer.
4. Write down the laptop's name and wireless IP address.
5. Configure the remote computer's Windows operating system to connect directly to the robot's laptop computer using a wireless connection. How you do this depends on what version of the Windows operating system your computer is running. For example, for Windows XP:
  - Double-click on the **wireless** icon in the system tray (bottom right of your Windows screen).
  - In the Properties menu, select **Wireless Net Tap**.
  - Add a new network.
  - The SSID name for the new network should be: **ADHOC**.
  - Go to **Advanced Settings**.

- 
- Select **Computer to Computer**.
  - Go to **TCP/IP** settings.
  - Call up the **TCP/IP Properties Menu** and enter the following information:
    - IP address of the robot's laptop (enter a value)
    - Subnet Mask: 255.255.255.0
  - Save the new network settings.

If you are unsure how to set up a wireless network on your version of Windows, check the Windows Help files. If you are still having trouble, contact Evolution Robotics Customer Support.

6. Next, configure the robot laptop's Windows operating system to connect directly to the remote computer using a wireless connection. How you do this depends on what version of the Windows operating system your laptop is running. For example, for Windows 2000:

- Double-click on the **wireless** icon in the system tray.
- Select **802.11 Ad Hoc network**.
- Enter the same SSID name you entered on the remote computer. In the above example, this is **ADHOC**.
- Disable encryption.
- Open the **Control Panel** and select **Network and Wireless Connections**, then **Wireless Net Tap**.
- Call up the **TCP/IP Properties Menu** and enter the following information:
  - **IP address of the robot's laptop**: Enter a value. This must be different than the IP address of the remote computer.
  - **Subnet Mask**: 255.255.255.0

7. Save the new network settings.

8. Test the wireless connection between the two computers by using **ping** on either machine. To do this:

- Open an MS-DOS window
- At the prompt, type: `ping xxx.xxx.xx.x` where instead of the x's, you should enter the IP address of the OTHER computer. Here's an example of a failed ping:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\>ping 192.168.12.1
```

```
Pinging 192.168.12.1 with 32 bytes of data:
```

```
Request timed out.
Request timed out.
```

```
Request timed out.
Request timed out.

Ping statistics for 192.168.12.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Here's an example of a successful ping:

```
C:\>ping 10.1.3.57

Pinging 10.1.3.57 with 32 bytes of data:

Reply from 10.1.3.57: bytes=32 time=31ms TTL=64
Reply from 10.1.3.57: bytes=32 time<10ms TTL=64
Reply from 10.1.3.57: bytes=32 time<10ms TTL=64
Reply from 10.1.3.57: bytes=32 time<10ms TTL=64

Ping statistics for 10.1.3.57:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 31ms, Average = 7ms

C:\>
```

9. Click on the **Settings** button and then the **Remote Control** tab. Check the box for **Allow Remote Control**.
10. Install the RCC software on your remote computer, and start the application.
11. Click on the **Settings** button again. In the **Remote Control** tab, check the box for **Control robot at this IP address**. Enter the robot's laptop IP address in the space provided.
12. Leave the default as Port 9000.

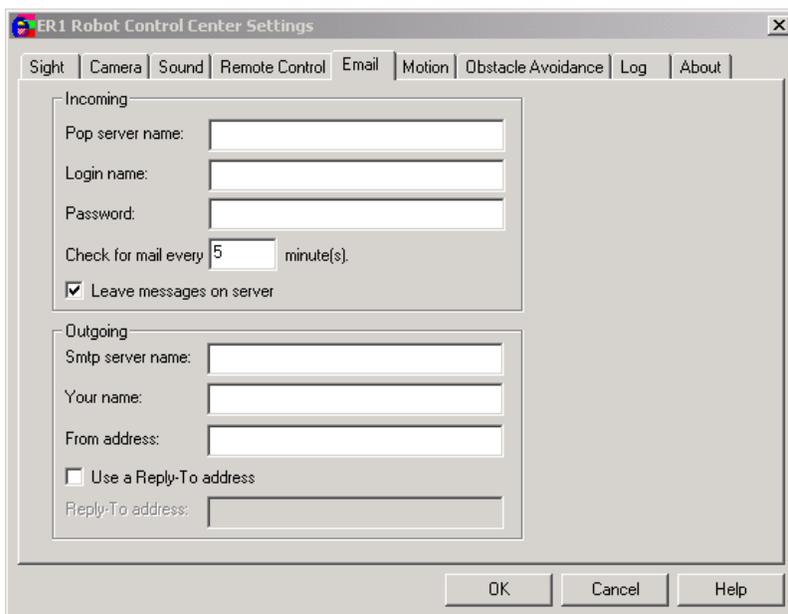
Options are:

- **Disable Remote Control** - If this checkbox is selected, remote control of your robot is disallowed.
- **Allow remote control of this instance** - If you want to remotely control your robot, click this checkbox on the robot's laptop. You will also need to go to the remote computer and click **Control robot at this IP address**. (If you close the ER1 application, then restart it, you will have to re-select this option.)
- **Control robot at this IP address** - If this option is selected, the IP address, port and password for the robot's laptop must be supplied.
- **Allow API Control of this instance** - If this option is selected, the ER1 can be controlled using the API instead of through the RCC interface. If this option is selected, a port must be specified. You may also specify a password if desired. See the [Application Programmer's Interface](#) section for details. Note that if this option is specified, you cannot control the robot using the RCC software until you discontinue your API session.
- **Reset the network connection now** - If your computer and your robot are using wireless communication, the present limitations of the technology mean

that connections can occasionally be dropped. The reset checkbox can be used to reconnect the base unit to the remote instance, assuming, of course, that a network connection is available.

## Email Tab

This tab is used to set up ER1 email parameters for both incoming and outgoing email (you must have Internet access on your laptop for this to work).



If you plan to program your robot to send and receive email, you need the following:

- A POP account for your robot.
- An email account for the robot.
  - Evolution Robotics will provide a RobotMail email account that will allow your robot to receive simple commands. You will need to provide a separate email account for sending email from the robot. We strongly recommend that you set up a separate email account for your robot. This will keep your personal mailbox free of robot command emails, and will keep the robot from downloading all your email to look for any that may be intended for the robot. See the [Message Receive Condition](#) section for information on setting up your RobotMail account.

*Important Note:* The messages transmitted to the robot are not direct instructions, but are messages which, based on the sender, subject, and/or body will result in activation of a scripted action.

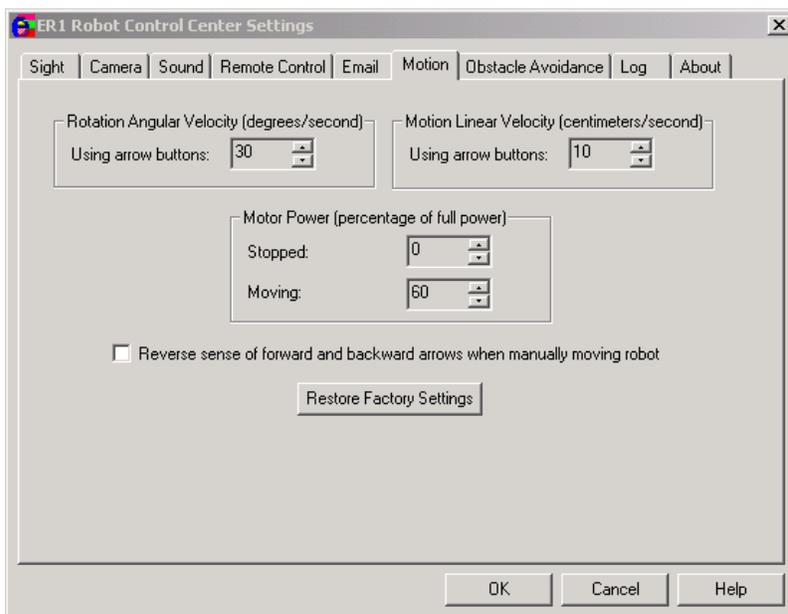
These parameters must be correctly set before the **If>Incoming email** or the **Then>Send email** features will work. These parameters do not have default values.

Options are:

- **Incoming** - Specify the following:
  - **Pop server name** - This is the name of your incoming mail server. You will get this information when you sign up for your RobotMail account.
  - **Login name** - Specifies the login name you use to access your RobotMail account.
  - **Password** - The login password you use to access your RobotMail account.
  - **Check for mail every \_\_ minutes** - Specifies how often should the ER1 software should check with the server to see if it has an email. Five minutes is the default.
  - **Leave messages on server** - If this checkbox is selected, all incoming email messages will be left on the server. This will allow you to download these messages to your robot to another email address if you so choose.
- **Outgoing** - Specify the following:
  - **Smtp server name** - This specifies your outgoing mail server. You can get this information from your Internet Service Provider.
  - **Your name** - This is the name you want to appear in your emails.
  - **From address** - Specify your email address (complete address).
  - **Use a Reply-To address** - If this checkbox is selected, you can specify a reply to address that varies from the address from which the email was sent.
  - **Reply-To address** - If **Use a Reply-To address** is enabled, specify the reply-to address here (complete address).

## Motion Tab

This tab is used to configure options related to the robot's motion. These options apply only when the direction buttons are used.



The **Rotation Angular Velocity** options allow you to specify, in degrees/second, how fast or slow the robot should rotate when performing various modes of action. The maximum value is 90 degrees/second. The minimum value is 5 degrees/second.

- **Using arrow buttons** - The maximum speed of rotation when clicking one of the two directional buttons.

The **Motion Linear Velocity** area allows you to specify, in centimeters/second, the maximum linear velocity of the robot. The maximum value is 50 cm/sec, because the robot cannot be operated safely above this speed. The minimum value is 5 cm/second.

- **Using arrow buttons** - The maximum linear velocity when clicking forward and reverse directional buttons.

The **Motor Power** options let you specify the percentage of full power that the robot should use under the following conditions:

- **Stopped** - When the robot is not moving, there is no reason for the robot's motor to be powered. It causes the battery to drain faster, and makes noise. The default value for this option is 0%.
- **Moving** - The robot does not need 100% power to perform most activities. It would only need full power if it had to travel up a steep incline or if it were traveling over a high friction surface, such as a shag rug. Leaving this value at the 60% default value should provide adequate power for all other instances. Remember, the less power you're using, the longer the robot's battery charge will last.

The **Reverse sense of forward and backward arrows when manually moving the robot** checkbox will change the upward pointing arrow (under the Obstacle Avoidance camera

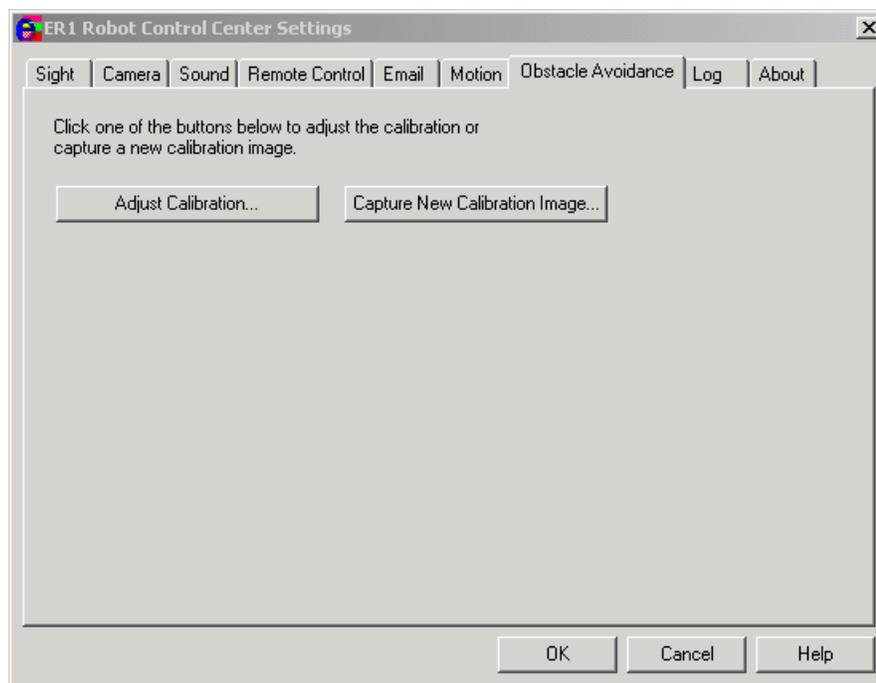
area on the RCC screen) from moving the robot forward to moving the robot backward and vice versa for the downward pointing arrow.

The **Restore Factory Settings** button resets all motion values back to the original defaults.

## Obstacle Avoidance Tab



ER1's obstacle avoidance feature uses a downward-pointing camera to detect obstacles in front of the robot. The detection area is a trapezoid in front of the robot whose size and shape depends on the height and angle of the avoidance camera. The avoidance algorithm works by detecting the lines between regions with different colors in this trapezoid. Thus, the avoidance algorithm works best on floors of uniform color without harsh edges (so objects on the floor are easy to see).



## Positioning the Obstacle Avoidance Camera

The avoidance camera (the optional second camera) should be positioned at least two feet off the ground. The camera should also be centered between the left and right wheels so that it sees the same distance to either side of the robot. When using both cameras, it is best that both cameras are centered, with the avoidance camera placed above the object recognition camera such that the latter does not obscure the former. To properly angle the avoidance camera, tilt it downward so that the bottom front edge of the robot is visible, then tilt slightly up so that the bottom front edge is no longer in the camera's view. This brings the detection area as close as possible to the robot, so that the camera can detect obstacles right in front of the robot. The size of the detection area will depend on how high the camera is and its angle. The higher the camera, the larger the detection area and the sooner the robot will see obstacles. A larger detection area is better for general

---

navigation, but it might also interfere when approaching a target. For example, the robot is approaching a painting on a wall. If it sees too far ahead, it might see the base of the wall too soon and turn away before it gets close enough to the painting. Therefore, the ideal height of the camera is dictated by the particular environment that the robot operates in and the goals it has to accomplish. You should experiment with different camera heights to see which would work best in your particular environment.

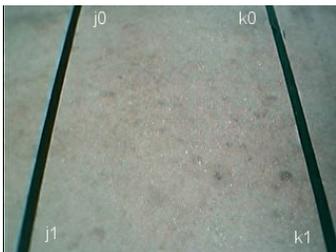
**Avoidance Camera Calibration**



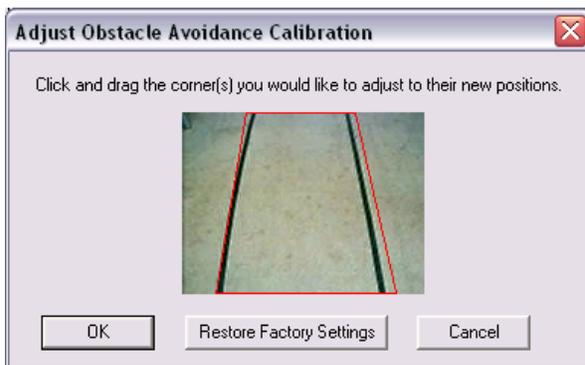
Once the camera is placed on the robot, it needs to be calibrated for best effect. To calibrate the camera, place two long straight edges or put two parallel lines of tape on the ground alongside the robot.

These edges or tape lines should be right outside of the two wheels. The avoidance camera should show a trapezoid formed from the straight edges or tape lines.

The edges or tape lines should be such that they extend to or past the top of the avoidance camera video feed as displayed on the RCC.



Click on **Capture New Calibration Image** in the **Settings>Obstacle Avoidance** tab. (Note that this must be done on the robot's laptop, not a remote computer.) The obstacle avoidance camera's current video display will be captured and displayed on a submenu, with the trapezoidal avoidance area outline overlaid on the image:



**Avoidance Camera View**



The avoidance camera view displays what the avoidance camera sees pointing downward toward the floor, along with red marks indicating where obstacles are detected. If there's just a straight red line along the top of the image, no obstacle is detected.

When obstacles are seen, the red marks move down from the top to indicate where the avoidance algorithm sees the obstacles in the image.

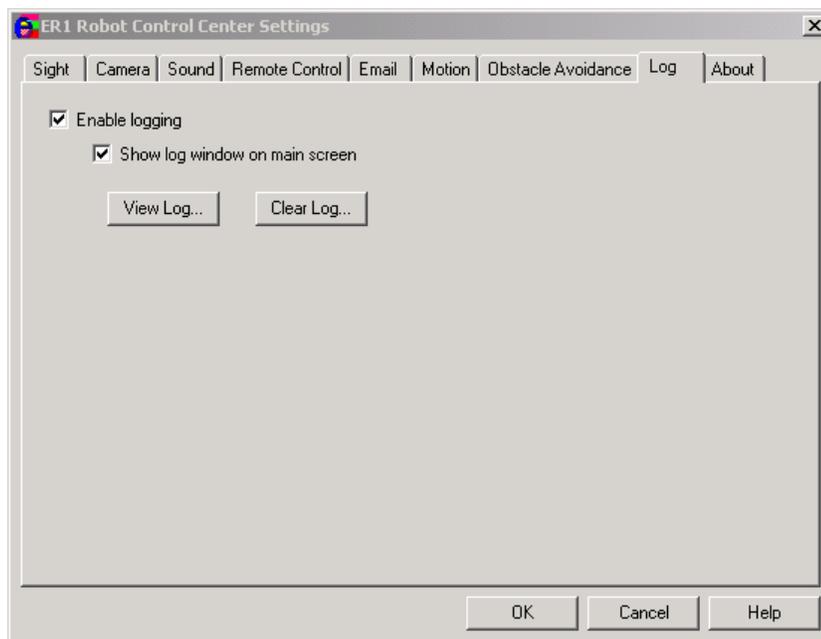
## Obstacle Avoidance At Work

Obstacle avoidance is always used in conjunction with moving toward a target. The target is either some distance, some object that the robot has been trained to recognize, or some color. If in the course of going to a target, the robot sees an obstacle in the way, it will attempt to go around the obstacle.

Due to the limited field of view from the downward pointing camera, the robot can only reliably go around small obstacles, like the legs and feet of a standing person. Large obstacles may take the robot so far away from its original path that it won't be able to reacquire the target. This is particularly true if the target is an object or a color that can only be seen or recognized from certain angles. Also, complicated or constricted environments may confuse the robot if it sees obstacles everywhere.

## Log Tab

The **Log** tab is used to adjust the RCC logging function. The log file is a good place to look if you have experienced an error or unusual behavior. The error messages should point you toward the cause and help you avoid having the same problem in the future.



The log file is named `er1.log` and is located in the RCC program directory. The following are the options available to alter the way errors are logged and viewed:

- **Enable logging** - If this checkbox is selected, errors will be logged in the log file. When logging is enabled, every action taken by a running behavior or set of behaviors is noted with a time and date stamp. The error messages are designed to describe the problem causing the error. This should be the first place that you look when you encounter a problem. If you are using a wireless connection, the incoming remote connection to the machine controlling the robot is monitored as well. This includes logging of

listening for connections, connection attempts (with the incoming IP address), password and protocol matches and mismatches, logouts, and lost connections. Please note that these will not be mirrored on the remote controlling system, since some of them occur outside the scope of the connection and some would be redundant because the remote system displays the information in various dialog boxes during the connection process.

- **Show log window on main screen** - If this checkbox is selected, the log messages will be shown at the bottom of the main screen

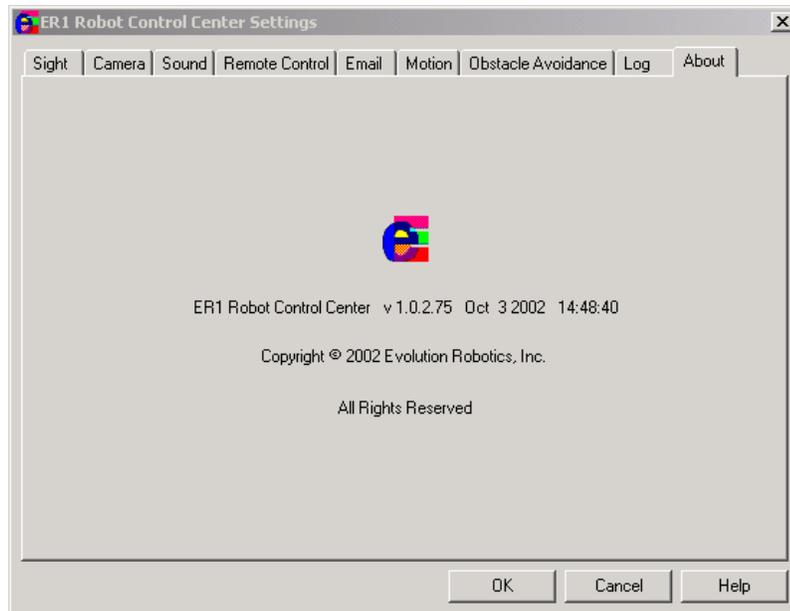


*Important Note:* Note that only the last 100 messages will be shown in this window. To view the entire contents of the log file, click on the **View Log** button.

- **View Log** - To view the log file directly, click on this button. The log file will be opened in Microsoft® Notepad. If you have a wireless connection to your robot, this file contains log messages from your robot's laptop (remote) and from the computer that you are using to control the robot (local). These can be easily differentiated. In the log file, the IP address of the remote computer, in parenthesis, will precede any error messages it produces. Any error messages that do not have an IP address were produced by the local computer. To see these messages, logging must be enabled in both computers.
- **Clear Log** - To delete all of the entries in the log file, click on this button.

## About Tab

This tab displays the RCC software's version number, date and copyright information.





## ■ Chapter 9

---

# Behavior Settings

The **Behavior Settings** button lets you change settings for each specific behavior.

To set the Behavior Settings for an individual behavior, make sure that the behavior for which the settings will apply is the active behavior and then click the **Behavior Settings** button. A dialog box appears. Along the top of the box are a number of tabs, described in the sections that follow.

*Important Note:* A good place to get ideas and help in creating behaviors is the ER1 community. If you have registered your ER1, you can join the ER1 community at [www.evolution.com](http://www.evolution.com).

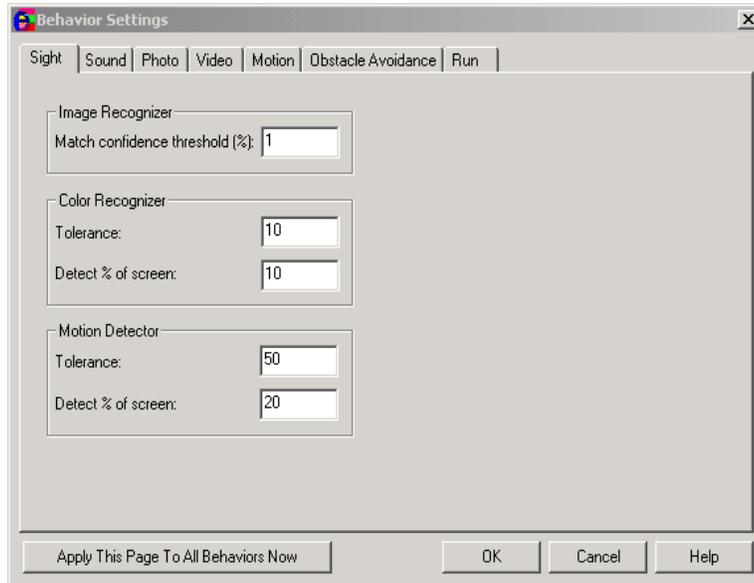
### **Sight Tab**

This tab is used to fine-tune aspects of the software's object recognition capabilities, which is used to recognize objects on which the software has already been trained.

### **Image Recognizer**

This parameter is used to determine the specificity of object recognition.

- **Match confidence threshold (%)** – Allows you to set a minimum image recognition confidence threshold. The software assigns a confidence percentage to each object it recognizes, which is displayed in the **Image Text** area after the object's name. The higher the percentage, the greater confidence the software has that it has correctly recognized an object.



## Color Recognizer

These parameters are used set the sensitivity of ER1's color recognition feature.

- **Color Tolerance** - This parameter allows you to specify a minimum color recognition tolerance value. The lower the number, the greater confidence the software has that it has identified the correct color.
- **Detect % of screen** - Setting this parameter allows you to set the percentage of the camera's video display that must contain the specified color before a trigger condition occurs. The higher the percentage, the larger the area a color must cover in order to trigger an action. Valid values range from 0-100%. The default value is 20%.

## Motion Detection

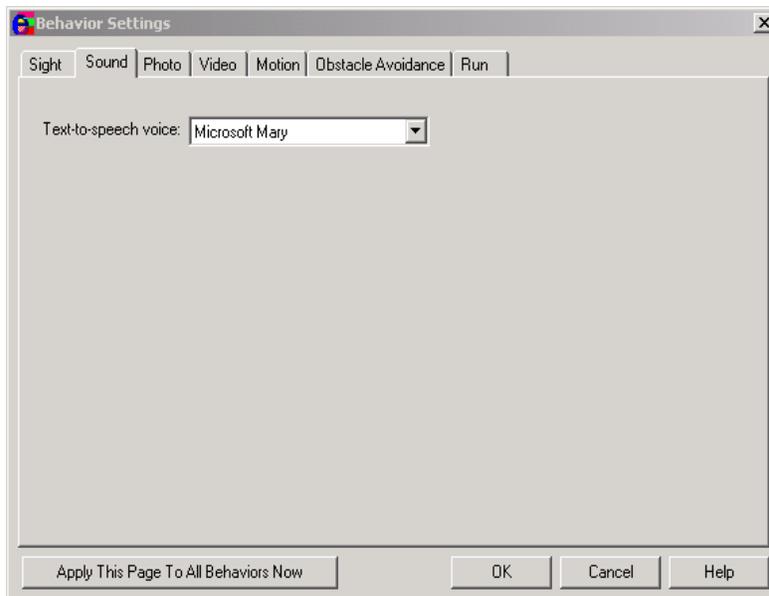
The motion detection settings allow you to specify the way that RCC detects motion. There are two options:

- **Tolerance** - The Tolerance parameter indicates the amount of change there must be between two consecutive images before the motion detection triggers. The accepted values are from 0 to 256, with a default of 50. Higher values mean that more changes must occur before the motion detection can occur.
  - **Detect % of Screen** - This parameter adjusts the percentage of pixels on the screen that have to change between two subsequent frames for motion detection to be triggered.
-

---

## Sound Tab

This tab allows you to set which Microsoft Text-to-Speech preprogrammed voice the robot should use when speaking to you. Use the pulldown menu to select an option.

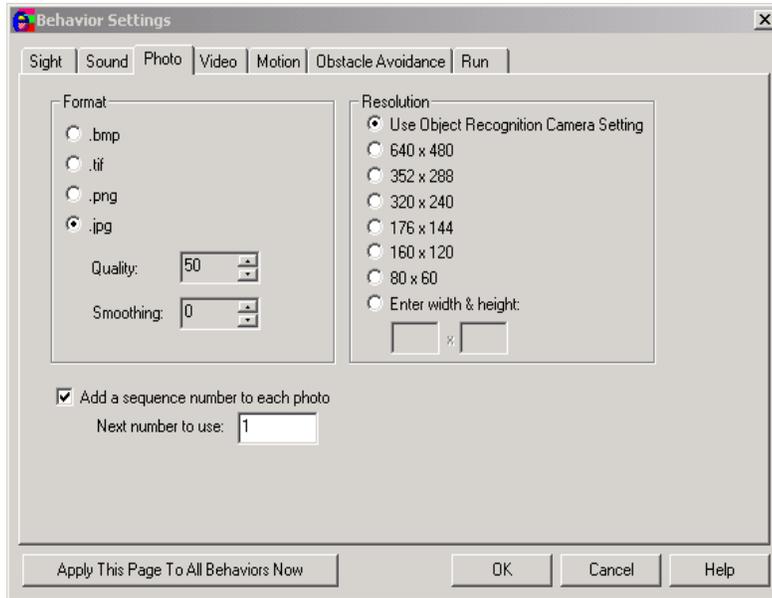


## Photo Tab

This tab is used to set up photo parameters.

Options are:

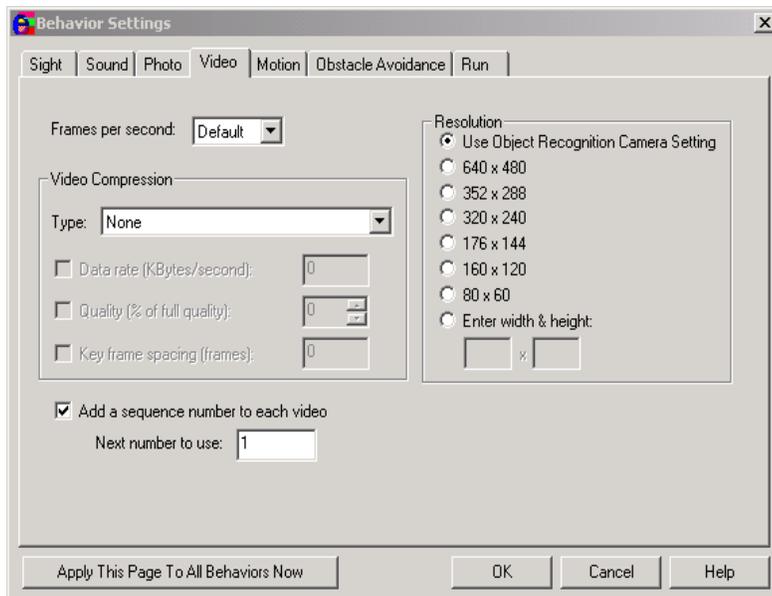
- **Format** – This parameter allows you to choose which format to use when saving a picture from the robot's webcam. The default value is `.jpg`. Choose one of the following formats:
  - `.bmp`
  - `.tif`
  - `.png`
  - `.jpg`
- **Quality** - JPEG files use lossy compression, meaning that they do not exactly recreate the original image. 100 means the least loss (best quality) and thus the biggest file size, 0 means the most loss (lowest quality) and the smallest file size.
- **Smoothing** - Compressing an image as a JPEG can cause sharp edges to appear rough or jagged. Smoothing lets you specify how much to smooth rough edges created by the compression process. 0 means no smoothing, 100 means maximum smoothing.



- **Resolution** - This box allows you to set the resolution of the photo. The higher values mean higher resolution and larger file size, whereas the lower values specify a lower resolution resulting in a smaller file size. You may also choose to accept the default value set by the object recognition camera.
- **Sequence Numbers** – This area lets you choose whether or not to add a sequence number to each photo, and if so, what is the next number to use. The sequence number is appended to the name of the file, before the extension.

## Video Tab

This tab is used to configure the video preferences.



---

The options are:

- **Frames per second** - This pulldown menu is used to specify the frames per second that the robot should use when capturing video. The available values are 1, 2, 5, 10, 15 or 30 frames per second.
- **Video Compression** - These parameters are used to specify the type of compression to use as well as the parameters associated with the compression type selected. Be sure to consider your needs and the type of camera you are using when selecting values.
  - **Type** - This pulldown menu is used to select the type of video compression to use. The default value is none.
  - **Data rate** - If the compression algorithm that you selected supports this parameter, select the checkbox. The range of values depends on the compression method chosen, the frame rate, the quality and the key frame rate. Pick a reasonable value to correspond to your other settings, should you choose to set this value manually.
  - **Quality** - If the compression algorithm that you selected supports this parameter, you can specify the percent of the optimal possible quality at which to save the video file. Specifying a higher value will yield higher resolution, but also larger file size. Selecting a smaller value will yield lower resolution, but a smaller file size. The file size should be considered when emailing video files. The default value is 85%.
  - **Key frame spacing** - To select this parameter, select the checkbox and specify a value. If you want to be able to scan quickly through your video, you will want keyframes, which do not depend on previous frames for determining what is in them. The first frame is always a key frame, since it can not depend on any other frames. If you had no other key frames, and you wanted to start watching the video from the middle, the system would have to decompress all the intermediate frames to determine what was in the middle frame. If you have keyframes, say every 10 frames, it would have to decode at most 10 frames to determine what was in that section of video.
- **Add a sequence number to each video** - Use this parameter to append a number on the end of a file name, before the file type extension. This parameter helps to create unique filenames, thus reducing the probability of overwriting files by accident.
- **Resolution** - Use this parameter to select a resolution or create a new one. Be sure that you specify a resolution that is produced by your camera.

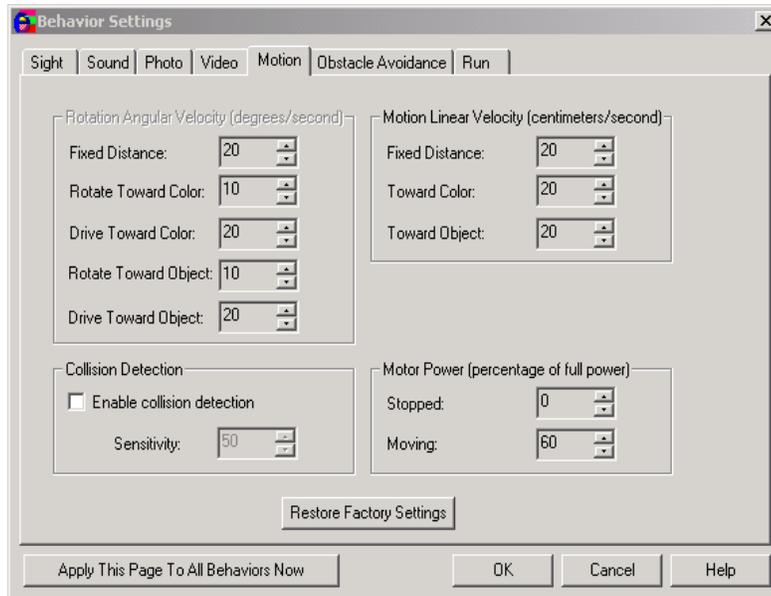
## Motion Tab

This tab is used to configure options related to the robot's motion. Parameters can be specified for a specific behavior, otherwise the default values will be used.

## Rotation Angular Velocity

The Rotation Angular Velocity options allow you to specify, in degrees/second, how fast or slow the robot should rotate when performing various modes of action. The maximum value is 90 degrees/second. The minimum value is 5 degrees/second.

- **Fixed Distance** – This parameter specifies the maximum speed of rotation when the robot rotates an angular distance specified in degrees. The default value is 20.
- **Rotate Toward Color** - This sets the maximum speed of rotation when the robot rotates towards a color. The default value is 10.



- **Drive Toward Color** – Use this parameter to set the maximum speed of rotation when the robot drives toward a color. (It will need to turn to line up with the color if it is not directly ahead. If the color happens to be moving, the robot will need to adjust its course to follow it.) The default value is 20.
- **Rotate Toward Object** - This sets the maximum speed of rotation when the robot rotates towards an object. The default value is 10.
- **Drive Toward Object** - Use this parameter to set the maximum speed of rotation when the robot drives toward an object. It will need to turn to line up with the object if it is not directly ahead. If the object happens to be moving, the robot will need to adjust its course to follow it. The default value is 20.

## Motion Linear Velocity

The **Motion Linear Velocity** area allows you to specify, in centimeters/second, the maximum linear velocity for the different modes of motion. The maximum value of all these settings is 50 cm/sec, because the robot cannot be operated safely above this speed. The minimum value is 5 cm/second.

- **Fixed Distance** - This sets the maximum linear velocity to use when traveling a fixed distance. The default value is 20.

- 
- **Toward Color** - This parameter specifies the maximum linear velocity to use when traveling towards a color. The default value is 20.
  - **Toward Object** - If set, this parameter allows you to set the maximum linear velocity to use when traveling towards an object. The default value is 20.

## Motor Power

The **Motor Power** options let you specify the percentage of full power that the robot should use under the following conditions:

- **Stopped** – When the robot is not moving, there is no reason for the robot’s motor to be powered. It causes the battery to drain faster, and makes noise. The default value for this option is 0%. However, if the robot could stop on a sloped surface, then it will need power to stay in place. In this case, the stop power should be set to greater than zero.
- **Moving** –The robot does not need 100% power to perform most activities. It would only need full power if it had to travel up a steep incline or if the robot is traveling over a high friction surface, such as a shag rug. Leaving this value at the 60% default should provide adequate power for all other instances. Remember, the less power you’re using, the longer the robot’s battery charge will last. However, too little power and the robot will not move reliably.

## Detect Collisions

The **Enable collision detection** checkbox lets you toggle collision detection capabilities on and off. When collision detection is turned off, the ER1 will move without checking to see if it has bumped into any objects. When collision detection is turned on, the ER1 will use the camera's video feed to determine if the robot has collided with some object. In that case, the ER1 responds by stopping and backing up a short distance before continuing on its intended path. This option is set to off by default.

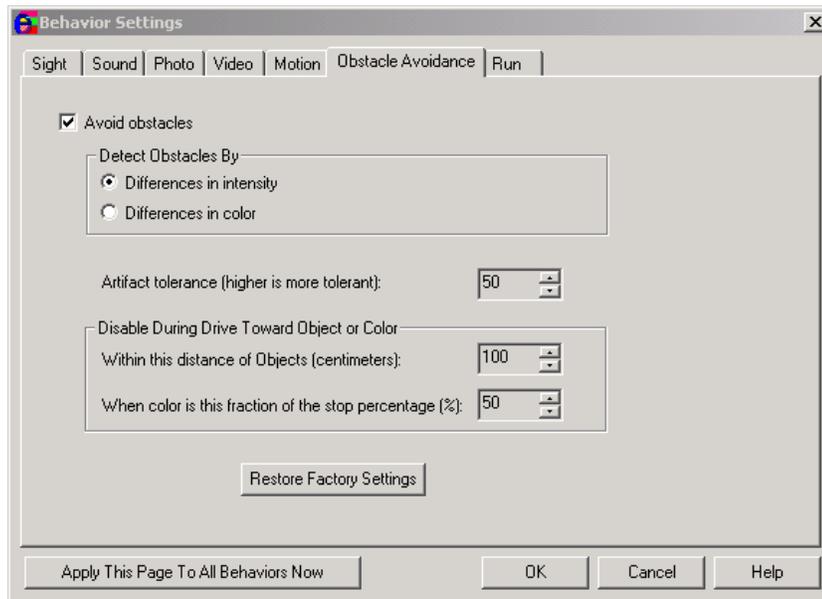
The **Sensitivity** field below the collision detection checkbox determines how tolerant the ER1 will be to potential collision situations. The higher the sensitivity number, the more readily the ER1 will detect collisions. (Conversely, the lower the number, the fewer collisions it will detect.) You may need to tailor the sensitivity value to work best in the environment that the ER1 is currently in. Keep in mind, however, that too high a **Sensitivity** will result in false collisions being detected. On the other hand, too low a **Sensitivity** will cause the ER1 not to detect actual collisions. (The default value is 50; start with this and adjust accordingly for your environment.)

An important collision detection behavior occurs when all that the ER1 can see is a featureless object, such as a blank wall. In this case, the ER1 stops, because it is most likely very close to the object, and therefore should stop moving towards it.

This behavior can be incorrectly diagnosed as a problem with the robot's motors, but if you reorient the robot so that the camera's view is of a non-featureless environment, it should continue its movement. If the robot still doesn't move, temporarily disable collision detection, and restart the robot's behavior. The robot should move correctly. If it does not, you should experiment with adjusting the robot's collision detection sensitivity so that it will work correctly in your environment. (In rare instances of extremely featureless environments, collision detection cannot be adjusted to work properly.)

## Obstacle Avoidance

These parameters are used to activate and adjust the ER1's obstacle avoidance system.



- **Avoid Obstacles** - To activate the obstacle avoidance system, select this option. This option is set to off by default.
  - **Detect Obstacles by** - These parameters allow you to select a method for detecting obstacles.
    - **Differences in intensity** - This method detects obstacles by detecting differences in light intensities.
    - **Difference in color** - This method of obstacle avoidance uses differences in color to detect obstacles.
  - **Artifact Tolerance** - The higher this value is set, the less likely that differences in colors and contrasts will be perceived as obstacles.
  - **Disable During Drive Toward Object or Color** - These parameters specify when to disengage the ER1's motors when driving toward an object or color. These parameters will prevent the robot from colliding with the intended target.
    - **Within this distance of objects** - This value specifies the number of centimeters from the target that the robot should stop. This value should never be set to less than 4cm, as there is a good chance that the robot will collide with the intended target.
    - **When color is this fraction of the stop percentage** - This parameter specifies that the robot should stop when the selected color takes up a certain percentage of the screen.

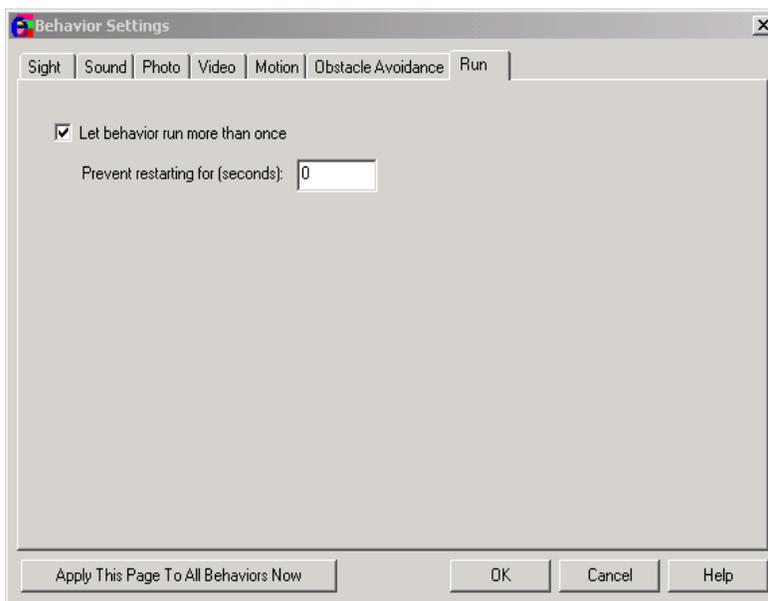
When you have finished adjusting the parameters, click the **Ok** button. You may cancel at any time by clicking on the **Cancel** button.

---

The **Restore Factory Settings** button resets all motion values back to the original defaults.

## Run Tab

The **Run** Tab allows you to specify how many times a behavior should run, and/or how long it should wait before running once again.



- **Let behavior run more than once** - Select this checkbox to allow a behavior to run again and again. This value is set to off by default.
  - **Prevent from restarting for (seconds)** - This parameter specifies the time interval in between rerunning the same behavior. Remember that the behavior must be triggered by something specified in the **If** column before it can occur.

## Apply This Page to All Behaviors Now

There is a useful feature in the Behavior Settings: the **Apply This Page To All Behaviors Now** button on the bottom on the **Behavior Settings** dialog box. If you click on the button, the current property page's settings are immediately applied to all 96 behaviors. Note that clicking the **Cancel** button does not undo the change.





# Application Programmer's Interface

## Introduction

Evolution Robotics' ER1 Robot Control Center software provides GUI functionality for controlling your robot. We also provide the APIs described in this chapter so that you can control your robot directly, instead of using the GUI. With these APIs, and some programming knowledge, you can:

- Control the robot in response to stimuli which the robot has not been programmed to know about. An example would be writing a program which moves the robot under the control of a helmet or data glove.
- Access the digital inputs and outputs and analog inputs of the robot, for use with custom sensors, or for controlling custom devices.
- Generate a sequence of motions algorithmically. For example, you could program the robot to travel around the perimeter of a circle, or to “dance”, without entering each component of the motion as a separate behavior.

- Use the ER1's vision system to control external devices. For example, you could write a program which integrates the robot with your home automation system and have it turn on your home theater system and dim the lights when you show it a particular DVD box.

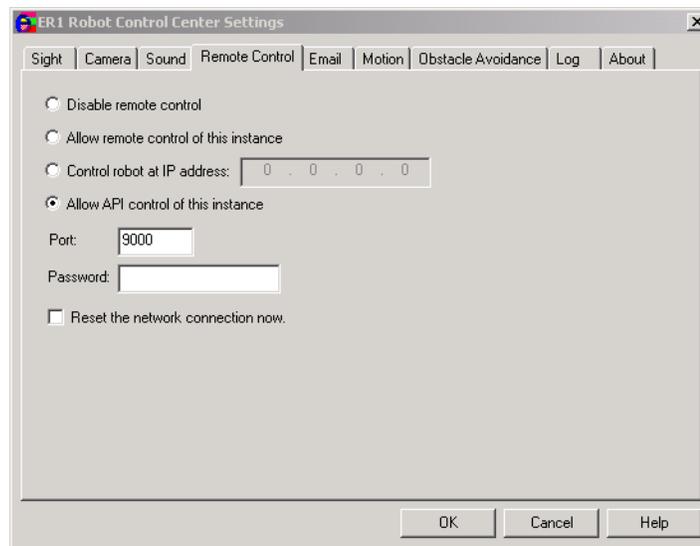
Because the APIs are written in a language neutral way, they can be used by most programming and scripting languages (e.g. C, C++, Perl, Java, Visual Basic, etc.). They also work equally well when issued from the laptop controlling the robot and from another machine via a (wireless) network.

## Accessing the API

The API is accessed by opening a TCP/IP socket to `er1.exe` and sending text commands along the socket.

Activating the feature is done via the **Remote Control** tab of the **Settings** dialog box. There is a new remote control option, labeled **Allow API control of this instance**. To activate:

1. On the robot's laptop, with ER1 running, choose the **Settings** button, select the **Sight** tab and click on the **Allow API Control of this instance** checkbox.



2. On the remote client running ER1, click on the **Settings** button, and then the **Sight** tab and click on the **Disable Remote Control** checkbox.

This makes it such that you cannot control a robot with both remote control and the API at the same time.

Also, once the API connection is established and the user has "logged in" (see below), a modal dialog box pops up on the robot's system. This prevents any GUI interaction with the robot until the API connection is closed, or the button in the dialog box which allows the connection to be terminated is pressed.

There is also a checkbox in the dialog which allows the user to turn off the API interface without having to bring up the Settings dialog box. This prevents a malicious user, or a wayward program, from continually reconnecting to the robot and locking out the user.

## Telnet Windows XP Setup

This interface allows the telnet program to be used to connect to the robot and try out the commands.

Telnet programs vary, but here's a description of how you would connect to the robot using the one which comes with Windows XP:

1. Start the ER, click on the **Settings** button, select the **Remote Control** tab, and click on the robot's laptop **Allow API control in this instance**. (You can use the same port and password which you would normally use for remote control. The password can also be empty.)
2. On the remote client PC, click the **Start** button on the taskbar. Click **Run**, and type the following in the **Open** field:

```
telnet <ip address> <port>
```

e.g.:

```
telnet localhost 9000
```

or

```
telnet 10.0.0.2 9000
```

substituting the IP address of the system running `er1.exe` (localhost if it is the same machine that you are going to run telnet on), and the port number being used in the **Remote Control** tab.

3. Since local echo is off, by default, you will not be able to see what you type as you type it. To correct this, type:  

```
Ctrl+]
```
4. This will take you to command mode for telnet. Type:  

```
set LOCAL_ECHO
```
5. Then, hit the enter key twice. The first time turns on local echo mode, and the second takes you back to your telnet session to the robot, blanking the screen.

## Older Windows Telnet Setup

For older Windows telnet programs, e.g. Win98 and NT4:

1. Follow steps 1 and 2 above.
2. To turn on local echo, use the **Terminal / Preferences...** menu item in telnet, click the **Local Echo** box, then click OK.

## Using the Telnet Connection

Now you can type commands to the ER1 API and see what you are typing.

1. Make sure you are connected by hitting the Enter key once. You should see the word:

```
OK
```

appear on the next line.

2. You can now type just about anything and it will show you a list of valid commands. If you have set a password using the **Remote Control** tab, the only valid command initially is `login`. Type:

```
login <password>
```

where `<password>` is the correct password (case sensitive) followed by the Enter key.

3. If you see OK, you are logged in and can begin issuing commands to the robot from the remote client.
4. Also, at this point, you will see a dialog box on the robot pop up, preventing any interaction with the robot's GUI until the API connection is closed, either by you or by telnet.
5. Test out your setup by typing the following, then pressing ENTER:

```
play phrase "hello"
```

The robot should speak the word hello.

6. If you type an incorrect or incomplete command, you will be prompted with a list of correct commands. For example:

```
?  
Error: '?(*)', Expected 'clear', 'events', 'input', 'move', 'output',  
'play', 'sense', 'set', or 'stop'.
```

```
set  
Error: 'set(*)', Expected 'voice', 'linear velocity' (or 'v'),  
'angular velocity' (or 'w'), 'power stopped', 'power moving', 'collision detection', 'obstacle  
avoidance', 'confidence threshold', 'color tolerance', or 'color percentage'.
```

```
move  
Error: 'move(*)', Expected 'rotate toward', 'drive toward', 'x', 'y', or a distance.
```

7. To quit telnet, either click on the close button at the top right, or (XP version) press:

```
Press Ctrl+]
```

```
Type quit
```

```
Press Enter.
```

## Commands

Here's a description of the commands that are currently implemented. In general, commands and their parameters are not case sensitive, though object names are case sensitive.

Commands to the robot are always of the form:

`<command><Enter>`

Every command is acknowledged with either an:

`OK<Enter>`

or, if something goes wrong:

`Error: <error message><Enter>`

## Examples

Here are some examples of API commands:

```

move 6 i
play phrase "I am moving 6 inches forward"
move -20 c
play phrase "I am moving 20 centimeters backwards"
move 90 d
play phrase "I am moving 90 degrees to the left"
move -180 d
play phrase "I am moving 180 degrees to the right"
play phrase "I will now detect sound levels"
sense sound level on
events
set voice "Microsoft Mary"
play phrase "I am Microsoft Mary"
set voice "Microsoft Mike"
play phrase "I am Microsoft Mike"
set voice "Microsoft Sam"
play phrase "I am Microsoft Sam"
set v 10
move 2 f
play phrase "I am moving at 10 centimeters per second"
set v 50
move 3 f
play phrase "I am moving at 50 centimeters per second"
set w 10
move 270 d
play phrase "I am turning at 10 degrees per second"
set w 50
move -270 d
play phrase "I am turning at 50 degrees per second"

```

## Command Quick Reference

Here's a list of the commands. See later in the chapter for descriptions, examples, etc.:

```

move <distance> <units>
move rotate toward object <object name>
move rotate toward color <r> <g> <b>
move drive toward object <object name> {stop <distance> <units>}
move drive toward color <r> <g> <b> {stop <percentage> {}}
play file <file_name>

```

```
play phrase <phrase to speak> {voice <voice name>}
stop
sense <sensor type> {on}
sense <sensor type> off
clear
events
set voice <voice name>
set linear velocity <velocity>
set angular velocity <velocity>
set power stopped <power>
set power moving <power>
set collision detection off
set collision detection on {sensitivity <sensitivity percentage>}
set obstacle avoidance off
set obstacle avoidance on {method <method>}
{tolerance <tolerance>}
{disable distance <distance>}{disable percentage <percent>}
set confidence threshold <percentage>
set color tolerance <tolerance>
set color percentage <percent>
input digital
output digital <hex number>
input analog {<input number> <input number> ...}
```

## Move Command

The `move` command makes the robot move.

The API responds to the command after it has been verified, but before the robot has completed the motion. If another `move` command is issued before the robot completes the previous one, the prior `move` command is interrupted. See the `events` command for how to determine when the `move` command is complete.

There are quite a few different forms of the `move` command:

### Move Distance

**Format:**

```
move <distance> <units>
```

**Parameters:**

<distance> is a positive or negative floating point number

<units> is inches, centimeters, meters, feet, or degrees

Since the first letter of each of these is unique, anything which begins with that first letter is assumed to be the correct word. This allows abbreviations, such as in:

```
move 30.5 i
```

```
move -2 ft
```

---

**Examples:**

```
move 90 degrees
move 50 feet
move -5.3 cm
```

**Move Rotate Toward Object****Format:**

```
move rotate toward object <object name>
```

**Parameter:**

<object name> is the name of an object in the library. It is case-sensitive. If it contains spaces, it must be surrounded by single or double quotes.

**Example:**

```
move rotate toward "Spot Page 1"
```

**Move Rotate Toward Color****Format:**

```
move rotate toward color <r> <g> <b>
```

**Parameters:**

<r> is the red color value (0 to 255)

<g> is the green color value (0 to 255)

<b> is the blue color value (0 to 255)

**Example:**

```
move rotate toward color 255 0 0
which rotates the robot towards the color red.
```

*Important Note:* Are you not sure how to determine the RGB values of an object? Try downloading and installing GIMP (Gnu Image Manipulation Program) from [www.gimp.org](http://www.gimp.org). It is a free program. Using the ER1 camera, take a snapshot of the object in question. Open the file in GIMP. Use the dropper tool to select the color you want to analyze. The RGB values will be displayed.

You can also use Microsoft® Paint to analyze the colors. Again capture the image using the ER1. Open the file in Paint. Select the desired color in the image using the dropper tool. Choose **Colors>Edit Colors** on the toolbar menu and click on the **Define Custom Colors** button. The RGB values will be displayed.

## **Move Drive Toward Object**

***Format:***

```
move drive toward object <object name> {stop <distance> <units>}
```

***Parameters:***

*object name* specifies the filename of the object to move toward

{ } The curly braces indicate the use of the word 'stop'

The stop distance is optional. There is a built-in default of 2 feet.

***Examples:***

```
move drive toward object Coke stop 6 in
move drive toward object 'back door'
```

## **Move Drive Toward Color**

***Format:***

```
move drive toward color <r> <g> <b> {stop <percentage> {%}}
```

This is similar to the `rotate toward color` command, but you can optionally specify a percentage value to make the robot stop when that percentage of the screen is filled by the chosen color.

***Examples:***

```
move drive toward color 0 255 0 stop 10 %
move drive toward color 0 255 0 stop 10
move drive toward color 128 128 128
```

## **Objects Command**

### **Objects**

***Format***

```
objects
```

It returns the list of object in the library in the following format:

```
OK
<object name 1>
<object name 2>
<...>
<object name N>
<blank line>
```

Since each object is listed on a separate line, no quotes are used around the object name. The last object is followed by a blank line.

---

## Play Command

The `play` command allows the robot to play sounds.

Like the `move` command, the API's response comes before the sound has completed playing and playing another sound interrupts the previous sound. The `events` command is used to determine when the command completes. There are two forms to the `play` command:

### Play File

#### *Format:*

```
play file <file_name>
```

#### *Parameters*

<file\_name> is the name of the file to play.

The file must exist on the robot's system at the specified path. Quotes (single or double) are required if there are spaces in the name of the file.

#### *Examples:*

```
play file sounds\Laugh.wav  
play file "c:\mp3\Linkin Park\Hybrid Theory\In The End.mp3"
```

### Play Phrase

#### *Format:*

```
play phrase <phrase to speak> {voice <voice name>}
```

#### *Parameters*

<phrase to speak> is what you want the robot to say. Quotes are required if there are spaces in it.

You can also specify a voice for this particular phrase. The name of the voice is assigned by the system (e.g. "Microsoft Sam"). If the voice is not found, there is no error indication, and the phrase is played using the system default. Quotes are required if there are spaces in the voice name.

#### *Examples:*

```
play phrase Hello voice "Microsoft Mary"  
play phrase 'You have mail.'
```

## Position Command

### Position

#### *Format*

```
position
```

This returns the robot's current x,y position (relative to the position where the robot started) and rotation angle.

*Example*

```
OK 3.5 4.6 90.1
```

The first value is the X position (in centimeters), the second is the Y position (in centimeters), and the third is the angle (in degrees).

## **Stop Command**

The stop command stops any robot motion or sounds which are in progress.

*Format:*

```
stop
```

## **Sense Command**

The sense command lets you turn on and off the robot's sensors.

The format of the command is:

```
sense <sensor type> {on}
sense <sensor type> off
```

*Parameters:*

<sensor type> is one of:

```
objects
sound level
speech
```

When you turn a sensor on, that allows messages from that sensor to appear when the `events` command is issued.

*Note:* Quotes should not be used around the sensor name since the name is a part of the command, not a parameter.

Examples:

```
sense objects
sense objects off
sense sound level on
sense speech off
```

## **Clear Command**

*Format:*

```
clear
```

The clear command throws away all events which have not yet been sent to the user.

---

## Events Command

*Format:*

```
events
```

The `events` command allows the robot to send back information from its sensors (if enabled) as well as sound and move completion notices.

The command is acknowledged with an OK message, then events are sent as they happen.

Events are terminated by the receipt of any command, even a blank line. It is recommended that a blank line be used to terminate the sending of events because otherwise a programmer may confuse an event message for a status return from a command being issued.

In other words, the user should:

1. Issue the `events` command
2. Look for the OK
3. Parse the event messages as they come down
4. Then, when ready to act on one of the events, send a blank line and wait for the OK from that.
5. Event messages will be turned off, though they may still accumulate.

The issuing of new `move` and `play` commands clear any pending completion messages from the corresponding command. That makes it easy to see that a message is from the most recently issued command. Also, only the most recent sound level event is kept.

Here are examples of event messages.

```
speech hello
sound level 8
sound level 24
object spot new (23%) (117,16,19.876 Inches)
move done
play done
play error Sound file c:\fubar.txt not found or don't know
how to play it
move error Cannot connect to the robot hardware.
```

Here is what happens when the robot recognizes an object:

```
object <"name"> <features matched> \
<total features> <x> <y> <distance in cm>
```

The object name is always surrounded by double quotes. The distance is always in centimeters and has one decimal place.

## **Set Command**

The `set` command is used to adjust settings which affect other commands.

These settings are 'sticky' in that they apply until reset to other values or until the API session ends. The limits are generally the same as for the corresponding commands in the GUI.

### **Set Voice**

***Format:***

```
set voice <voice name>
```

This overrides the default voice for all subsequent play phrase and play file commands for .txt files.

***Example:***

```
set voice 'Microsoft Mary'
```

### **Set Linear Velocity**

***Format:***

```
set linear velocity <velocity>
```

***Alternate format:***

```
set v <velocity>
```

This command sets the linear velocity, in cm/sec for all motion types.

If a motion doesn't support the velocity, it will be minimized or maximized against the limit. For example, the `move drive toward <object>` command is limited to 30 cm/sec but the `'move 30 inches'` command allows a maximum velocity of 50 cm/sec. If the linear velocity is set to 50 cm/sec, and you issue a `'move drive toward'` command, the velocity will be limited to 30 cm/sec automatically.

**Examples:**

```
set linear velocity 10
set v 47
```

### **Set Angular Velocity**

***Format:***

```
set angular velocity <velocity>
```

***Alternate form:***

```
set w <velocity>
```

The angular (rotational) velocity is set in degrees/sec. Otherwise, this command is similar to `set linear velocity` in that it affects all motion types.

*Examples:*

```
set angular velocity 10
set w 20
```

**Set Power Stopped***Format:*

```
set power stopped <power>
```

Sets the robot's power level when stopped. The <power> value is the percentage of full power.

*Example:*

```
set power stopped 20
```

**Set Power Moving***Format:*

```
set power moving <power>
```

Sets the robot's power level when moving. Otherwise identical to the above command.

**Set Collision Detection***Format:*

```
set collision detection off
```

And:

```
set collision detection on {sensitivity <sensitivity percentage>}
```

These turn collision (bump) detection off or on, respectively. If turning it on, you can optionally specify a sensitivity.

*Examples:*

```
set collision detection on
set collision detection on sensitivity 30
set collision detection off
```

**Set Obstacle Avoidance***Format:*

```
set obstacle avoidance off
```

And:

```
set obstacle avoidance on {method <method>}
{tolerance <tolerance>}
{disable distance <distance>}
{disable percentage <percent>}
```

These control obstacle avoidance and, if turning it on, adjust the various settings.

*Parameters:*

<method> is either 'intensity' or 'color'

*<tolerance>* is a number from 10 to 80

*<distance>* is the standoff distance at which it is turned off when approaching an object, in cm.

*<percent>* is the fraction of the stop percentage for turning it off when moving toward a color.

The optional parameters can appear in any order.

***Examples:***

```
set collision detection off
set collision detection on
set collision detection on tolerance 20 method intensity
disable distance 40
set collision detection on disable percentage 50
```

**Set Confidence Threshold**

***Format:***

```
set confidence threshold <percentage>
```

Sets the object recognition feature threshold, expressed as a percentage of the features which must be recognized for the object to be recognized.

***Example:***

```
set confidence threshold 20
```

**Set Color Tolerance**

***Format:***

```
set color tolerance <tolerance>
```

Sets the color recognizer's color tolerance.

***Example:***

```
set color tolerance 10
```

**Set Color Percentage**

***Format:***

```
set color percentage <percentage>
```

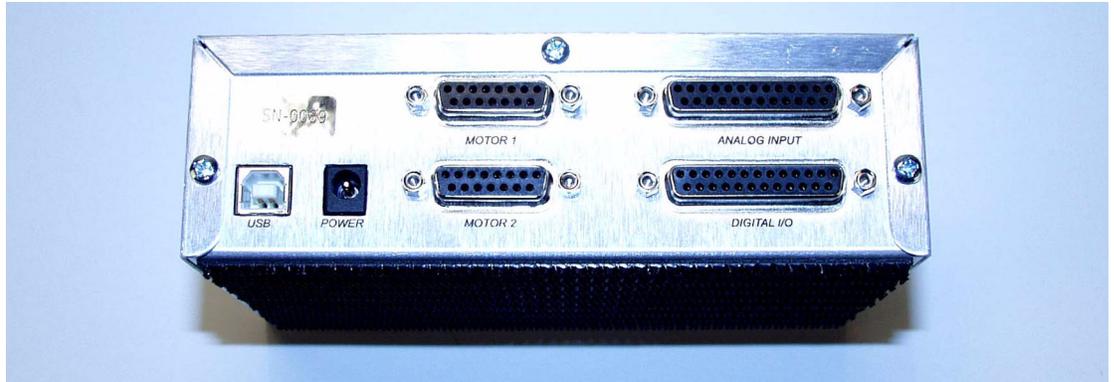
Sets the color recognizer's screen percentage for a match.

***Example:***

```
set color percentage 30
```

## Digital Input/Output Commands

These commands allow accessing of the digital input and output ports of the ER1's Robot Control Module and the analog input ports. The pin outs for these ports are in subsequent sections.



### Electrical Characteristics

The following summarizes the electrical characteristics of the ports.

#### Absolute Maximums

- Maximum input voltage on any pin (digital or analog) 5.3V
- Minimum input voltage on any pin (digital or analog) -0.3V
- Maximum total current sourced from VCC outputs 500mA

#### Input Logic Thresholds

- For Logic 1: min 2.0V -> max 5.3V
- For Logic 0: min -0.3V -> max 0.8V

#### Output Logic Thresholds

- For Logic 1: min 2.4V@Io = -300uA
- For Logic 0: max 0.33V@Io = 2mA
- Input Current: min -50uA max 50uA

#### Analog Input Source Impedance

Analog input source impedance = 9kOhms

## **Input Digital**

***Format:***

```
input digital
```

If the robot is properly connected, the response is:

```
OK <hex number>
```

***Example:***

```
OK ff
```

Each input port is represented as one bit of the result code, where digital input (DI) 0 is the least significant bit and DI 7 is the most significant bit, so `ff` indicates that all the inputs are high. `OK fe` would mean that all but input 0 was high.

## **Output Digital**

This command sets the digital outputs (DO) to the 8 bit value represented by `<hex number>`. Note that digital output (DO) 0 is the least significant bit and DO 7 is the most significant bit.

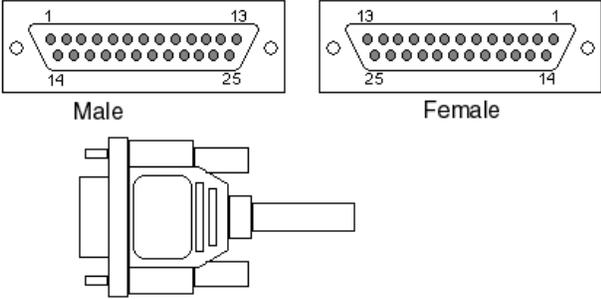
***Format:***

```
output digital <hex number>
```

***Example:***

```
output digital 80
```

This example sets digital output 7 high and all the others low.

<b>Hobby Digital I/O</b>	<b>J5</b>
<p>Pin 1 – VCC +5v Pin 2 – DO 0 Pin 3 – DO 2 Pin 4 – DO 4 Pin 5 – DO 6 Pin 6 – GND Pin 7 – GND Pin 8 – VCC +5v Pin 9 – DI 1 Pin 10 – DI 3 Pin 11 – DI 5 Pin 12 – DI 7 Pin 13 – GND Pin 14 – VCC +5v Pin 15 – DO 1 Pin 16 – DO 3 Pin 17 – DO 5 Pin 18 – DO 7 Pin 19 – GND Pin 20 – VCC +5v Pin 21 – DI 0 Pin 22 – DI 2 Pin 23 – DI 4 Pin 24 – DI 6 Pin 25 – GND</p>	 <p>The diagrams show the pin configurations for the J5 connector. The Male connector has pins 1, 13, 14, and 25 labeled. The Female connector has pins 13, 1, 25, and 14 labeled. The 3D view shows the connector's physical structure with a central pin and surrounding housing.</p>

## **Input Analog**

This command displays the current value of one of the 16 analog inputs. (The 16th input is tied to the battery. This uses the 0-5V scale as well, but it maps to a 0-15.6V scale on the battery.)

### ***Format:***

```
input analog {<input number> <input number> ...}
```

### ***Parameter***

*<input number>* This specifies the number of the input to query. See the Analog Pin Out that follows. The input numbers are the pins labelled A *<input number>* on the diagram. You will only have access to inputs 0-14, although pin 15 is shown in the output display. The resulting output values range from 0 to 65535, where:

- 0 represents 0 volts
- 65535 represents 5 volts

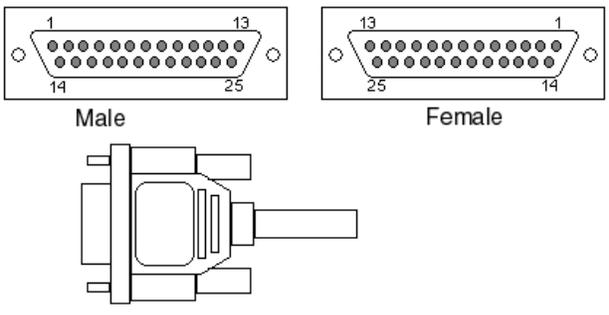
If the command is given with no arguments, each of the inputs, from 0 to 15, is listed in the response. For example:

```
input analog
OK 3328 3328 3328 3392 3328 3328 3328 3264 3328 58432 3328
55040 3328 51648 3328 61568
```

If the command is issued with arguments, the value of each listed input, from 0 to 15 is given in the response. For example:

```
input analog 0 9
OK 3328 58432
```

The arguments do not have to appear in any particular order, and repeated arguments are allowed.

Hobby Analog Input		J6	
Pin 1 – VCC +5v	Pin 16 – A1	 <p>The diagram shows three views of the J6 connector. At the top left is a 'Male' view of a 25-pin D-sub connector with pins numbered 1, 13, 14, and 25. At the top right is a 'Female' view of a 25-pin D-sub connector with pins numbered 13, 1, 25, and 14. Below these is a perspective view of a physical connector with a central pin and four side pins.</p>	
Pin 2 – VCC +5v	Pin 17 – A3		
Pin 3 – A0	Pin 18 – A5		
Pin 4 – A2	Pin 19 – A7		
Pin 5 – A4	Pin 20 – A9		
Pin 6 – A6	Pin 21 – A11		
Pin 7 – A8	Pin 22 – A13		
Pin 8 – A10	Pin 23 – GND		
Pin 9 – A12	Pin 24 – GND		
Pin 10 – A14	Pin 25 – GND		
Pin 11 – GND			
Pin 12 – GND			
Pin 13 – GND			
Pin 14 – VCC +5v			
Pin 15 – VCC +5v			

## Building ER1 Scripts

You can use your favorite programming language to build ER1 scripts using our APIs. The following is an example of how to use Java to create API programs. We've included a few files with the ER1 release to help you get started. They are:

- `ER1Client.java`, which handles the interface between your client PC and the robot.
- `TestClient.java`, which is a sample user program that presently issues a few simple move commands. You can run it as is, add your own commands to it, or use it as a template to create your own scripts.

The `TestClient.java` script also has some built-in command line help, which is displayed by typing `-h`:

```
$ java TestClient -h
usage: java TestClient [options]
```

Options:

- a *<address>* Specify the address of ER1 RCC. Defaults to 127.0.0.1.
- p *<port>* The port number of the ER1 RCC. Defaults to 9000.
- q Quiet mode. Only errors are printed out.
- h This message.

## Download Java SDK

To compile the Client java script and the Test script (as well as any scripts you write yourself), you need the latest Java SDK. Here is the URL where you can download what you need to run your java scripts in Windows:

<http://java.sun.com/downloads/index.html>

This provides a Java compiler for \*.java text program files (TestClient.java and ER1Client.java). The compiler takes the text program files as input and creates \*.class files. You can then use the Java runtime environment to execute the files.

Each time you edit your \*.java file, you must use the compiler to recompile it.

## Compiling

After downloading and installing the latest Java SDK, you can compile the Client script and the Test script by typing the following at the C:\ command line (don't forget to take directory paths into account):

```
javac ER1Client.java
javac TestClient.java
```

For example:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>cd ER1
C:\ER1>dir *Client*
10/07/2002  08:58a                8,115 ER1Client.java
10/07/2002  08:58a                1,496 TestClient.java
C:\ER1>c:\j2sdk1.4.1\bin\javac ER1Client.java
C:\ER1>c:\j2sdk1.4.1\bin\javac TestClient.java
C:\ER1>dir *Client*
10/07/2002  05:04p                4,569 ER1Client.class
10/07/2002  08:58a                8,115 ER1Client.java
10/07/2002  06:10p                 719 TestClient.class
10/07/2002  08:58a                1,496 TestClient.java
```

## Running the Script

To run the compiled TestClient Java script:

1. Put the robot's laptop into remote API mode as described earlier.
2. On your Client machine, use telnet to log into the robot's laptop.
3. Type the following at the command line:

```
java TestClient -a XX.X.XX.XXX
```

where XX.X.XX.XXX is the robot laptop's IP address, for example:

```
java TestClient -a 10.1.3.108
```

4. You should see:

```
C:\ER1>c:\j2sdk1.4.1\bin\java TestClient -a 10.1.3.111
Sending command: move 10 i
Response: OK
Sending command: events
Response: OK
```

```

Sending command: events
Response: OK
Sending command: events
Response: move done
Sending command: move -90 d
Response: OK
Sending command: events
Response: OK
move done
Sending command: move 50 c
Response: OK
Sending command: events
Response: OK
Sending command: events
Response: OK
Sending command: events
Response: OK
move done

C:\ER1>

```

## TestClient.java

Here is the TestClient.java file:

```

/**
 * @file    TestClient.java
 * @author  www.evolution.com
 * @author  Copyright (C) 2002, Evolution Robotics, Inc.
 */

import java.io.DataOutputStream;
import java.io.BufferedReader;
import java.net.Socket;

class TestClient
{
    ////////////////////////////////////////////////////////////////////
    // userProgram.
    ////////////////////////////////////////////////////////////////////
    /**
     * @brief
     * Users put their own commands here.
     */
    public static void userProgram()
    {
        // Send the first move command, 10 inches forward.
        String recvMsg = client.sendCommand("move 10 i");

        client.waitFor("move");

        // Send the turn command, 90 degrees to left.
        recvMsg = client.sendCommand("move -90 d");

        client.waitFor("move");

        // Send the next move command, 50 cm forward.
        recvMsg = client.sendCommand("move 50 c");

        client.waitFor("move");
    }
}

```

```
////////////////////////////////////  
// Application Entry Point  
////////////////////////////////////  
/**  
 * @brief  
 * Program entry point.  
 *  
 * @param args Command-line arguments.  
 */  
  
static ER1Client client = null;  
  
public static void main(String[] args) {  
    // Create a new client to communicate with ER1.  
    client = new ER1Client(args);  
  
    // Connect to the ER1 RCC.  
    if (client.connect()) {  
        // Run the user program.  
        userProgram();  
    }  
}
```

## Java API Example Programs

To change what commands you send it:

1. Edit the `TestClient.java` file. make changes to this section:

```
/**  
 * @brief  
 * Users put their own commands here.  
 */  
public static void userProgram()  
{  
    // Send the first move command, 10 inches forward.  
    String recvMsg = client.sendCommand("move 10 i\n");  
    client.waitFor("move");  
  
    // Send the turn command, 90 degrees to left.  
    recvMsg = client.sendCommand("move -90 d\n");  
    client.waitFor("move");  
  
    // Send the next move command, 50 cm forward.  
    recvMsg = client.sendCommand("move 50 c\n");  
    client.waitFor("move");  
}
```

2. Just add your API command in quotes to `recvMsg` line
3. Change the client. Wait for the line to reflect what is output at the end of your action in the `events` command.
4. For example to have the robot speak the sentence "This is a test", you would add this to the bottom:

```
recvMsg = client.sendCommand(play phrase "This is a test.\n");  
client.waitFor("play");
```

Then you must recompile everything:

```
javac ER1test.java  
javac TestClient.java
```

```
java TestClient -a <robot IP address>
```

Here are the command line flags:

```
> java TestClient -h
```

Usage:

```
java TestClient [options]
-a <address> Specify the address of ER1 RCC. Defaults to 127.0.0.1.
-p <port> The port number of the ER1 RCC. Defaults to 9000.
-q          Quiet mode. Only errors are printed out.
-h          This message.
```

## ER1Client.java

Here is the ER1Client.java file:

```
/**
 * @file    ER1Client.java
 * @author  www.evolution.com
 * @author  Copyright (C) 2002, Evolution Robotics, Inc.
 */
import java.io.DataOutputStream;
import java.io.BufferedInputStream;
import java.net.Socket;

/**
 * Generic socket client class that takes a data listener
 * and a protocol handler to process socket data.
 */
public class ER1Client {
    String      _address      = "127.0.0.1";
    int         _port         = 9000;
    Socket      _socket;
    boolean     _verbose     = true;
    DataOutputStream _write_stream = null;
    BufferedInputStream _read_stream = null;
    Thread      _connect_wait = null;
    boolean     _more_data   = false;

    public ER1Client(String[] args) {
        // Parse command line parameters.
        parseCommandLine(args);
    }

    public Socket getSocket()
    {
        return _socket;
    }

    public boolean connect()
    {
        return _connect(_address, _port);
    }

    public boolean _connect(String address, int port)
    {
```

```
.....

if (_socket != null) { disconnect(); }
_address = "";
_port = 0;
try {
    // Create a new socket connection.
    _socket = new Socket(address, port);
    try {
        // Set the output stream.
        _write_stream = new DataOutputStream(_socket.getOutputStream());
        _read_stream = new BufferedInputStream(_socket.getInputStream());
    }
    catch (Exception ex) {
        System.out.println(ex.toString());
    }
    _address = address;
    _port = port;
    return true;
}

catch (Exception ex) {
    System.out.println(ex.toString());
    //throw ex;
    return false;
}
}

public String readString()
{
    try
    {
        if (_read_stream == null) {
            System.out.println("Error: _read_stream is null.");
            return null;
        }
        try {
            int size;
            String msg;
            do {
                size = _read_stream.available();
                if (size != 0) {
                    byte[] buffer = new byte[size];
                    _read_stream.read(buffer, 0, size);
                    msg = new String(buffer);
                    return msg;
                }
                Thread.sleep(50);
            } while (size == 0);
        }
        catch (Exception ex) {
            System.out.println(ex.toString());
            return null;
        }
    }
    catch (RuntimeException ex)
    {
        System.out.println(ex.toString());
        return null;
        //Global.applet.showDebug(ex.toString());
    }
    return null;
}

public boolean sendString(String s) {
.....
```

```

.....

        return sendBuffer(s.getBytes(), s.length());
    }

    boolean sendBuffer(byte[] bytes, int size) {
        if (_write_stream == null) {
            //Global.applet.showDebug("No server connection so cannot send message!");
            return false;
        }

        try {
            _write_stream.write(bytes, 0, size);
            _write_stream.flush();
        }
        catch (Exception ex) {
            disconnect();
            System.out.println(ex.toString());
            return false;
        }
        return true;
    }

    public void disconnect() {
        try {
            if (_socket != null) {
                _socket.close();
                _socket = null;
                _write_stream = null;
            }
        }
        catch (Exception ex) {
            System.out.println(ex.toString());
        }
    }

    ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    // Helpers
    ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    public void parseCommandLine(String[] args)
    {
        // Parse command line parameters.
        int paramCount = args.length;
        for (int i = 0; i < paramCount; ++i) {
            // Check for quiet mode.
            if (args[i].compareTo("-q") == 0) {
                // quiet mode!
                _verbose = false;
            }
            // Check for new host address.
            else if (args[i].compareTo("-a") == 0) {
                if (i >= paramCount - 1) {
                    System.out.println("ignoring -a flag with no address.");
                }
                else {
                    i++;
                    _address = args[i];
                }
            }
            // Check for new host address.
            else if (args[i].compareTo("-p") == 0) {
                if (i >= paramCount - 1) {
                    System.out.println("ignoring -p flag with no port number.");
                }
                else {

```

```

        i++;
        _port = Integer.valueOf(args[i]).intValue();
    }
}
// Help.
else if (args[i].compareTo("-h") == 0) {
    usage();
}
else {
    usage();
}
}
}

public void usage()
{
    System.out.println("usage:  java TestClient [options]");
    System.out.println("  -a <address>  Specify the address of ER1 RCC.  Defaults to
127.0.0.1.");
    System.out.println("  -p <port>    The port number of the ER1 RCC.  Defaults to 9000.");
    System.out.println("  -q          Quiet mode.  Only errors are printed out.");
    System.out.println("  -h          This message.");
    System.exit(0);
}

public void waitFor(String command)
{
    boolean moveDone = false;
    // Wait until the move command is done by continuously sending
    // events command until "move done" is include in the response.
    while (!moveDone) {
        try {
            Thread.sleep(500);
        }
        catch (Exception ex) {
            System.out.println(ex.toString());
        }
        String recvMsg = sendCommand("events

// Check to see if "[command] done" is part of the response.
String doneString = command + " done";

moveDone = (recvMsg.indexOf(doneString) >= 0);
if (!moveDone) {
    // Check for error.
    String errorString = command + " error";
    int errorIndex = recvMsg.indexOf(errorString);
    moveDone = errorIndex >= 0;
    if (moveDone && !_verbose) {
        // print error message if in quiet mode.  If not
        // in quiet mode, error message is already printed.
        System.out.print(recvMsg.substring(errorIndex));
    }
}
}
}

public String sendCommand(String command)
{
    if (_verbose) {
        System.out.print("Sending command: " + command);
    }
    if (!sendString(command)) {

```

```
.....  
  
        return null;  
    }  
    String recvMsg = readString();  
    if (!checkResponse(recvMsg, command))  
    {  
        return null;  
    }  
    else {  
        return recvMsg;  
    }  
}  
  
public boolean checkResponse(String response, String command)  
{  
    if (response == null) {  
        System.out.println("Error sending command " + command + "!");  
        return false;  
    }  
    else {  
        if (_verbose) {  
            System.out.print("Response: " + response);  
        }  
        return true;  
    }  
}  
}
```



## ■ Chapter 11

---

# Tricks of the Trade

The following are some tricks and tips to help you operate your robot.

*Important Note:* Visit the ER1 community for up to the minute tricks and tips from other ER1 owners. If you have registered, go to [www.evolution.com](http://www.evolution.com) and join your fellow robot enthusiasts in discovering the capabilities of the ER1.

### Robot Search Pattern

This version of the RCC software improves the way the robot searches for an object or color to follow. Once you click the **Run it** button, the robot will pause for five seconds, while attempting to locate the object or color. If it does locate the object or color, it will begin the follow behavior. If it does not see the object or color after five seconds, it will begin a search spin. The robot will turn 360° unless it finds the object or color. If it does not find the object or color, it will stop after turning 360°.

Also, if after seeing the object, the robot loses sight of it, the robot will stop for one or two seconds to try to reacquire the object before starting a spin search.

## **Motor Noise**

If your robot's motors seem to be trying to move and are creating a disturbing noise, check the battery meter in the lower left corner of the RCC screen. If the battery is low, turn the robot off and charge the battery.

## **Wireless Network Disconnect**

If your wireless link between your robot and your remote computer has disconnected do the following:

1. Click on the **Settings** button.
2. Select the **Remote Control** tab.
3. Either the **Allow remote control of this instance** or the **Control robot at this IP address** option must be selected.
4. Click on the **Reset the network connection now** button.
5. Click on the **Ok** button. This signals the two computers to reestablish the connection. You may have to repeat this process more than once to get a connection.

This problem is generally an issue with your wireless network, not your robot. If you continue to have problems, contact your wireless card manufacturer to troubleshoot your system.

## **Frozen Robot**

Rarely the ER1 module will freeze, and you will need to reset the internal memory in the module in order to make the robot move. To do this, follow these steps:

1. Make sure the laptop is on, and the software is not running.
2. Turn off the ER1 Battery.
3. Unplug the USB cable from the laptop.
4. Turn on the ER1 battery.
5. Plug the USB cable back into the laptop.
6. The internal memory should now be reset and the robot should function properly.

## **Camera Shuts Off When Launching RCC/Unplugging Power Supply**

This issue has been documented on Hewlett-Packard laptops. This seems to happen due to the HP laptop's handling of power management. When you unplug the laptop from the wall outlet and Windows' power management scheme is set to 'Laptop/Portable', power to the USB bus (and all devices on the system) gets reduced significantly. Since the Kritter Cam requires the full 500ma of power, there is not enough to keep it functioning.

Here's what to do: Set your computer's power management scheme to 'Always On' as follows:

1. Click **Start>Control Panel**.
2. Double-click the **Power Options** icon.
3. On the General tab, select **Always On** on the **Power Scheme** menu.
4. Click **Apply** and then click **Ok**.

Note that there are some side effects. The camera now works when the computer is unplugged from AC power. However, you will notice a reduction in battery life.

## **Textbox Input Size**

Each of the textboxes in RCC have an input limit of 100,000 characters. Additional limits may be set by the various functions. If you need to input more than 100,000 characters in a textbox, breakup the text into pieces and input it into consecutive behaviors.



## ■ Appendix A

---

# Assembly Instructions

### Getting to Know the Parts

Have you ever asked yourself “What is a XBeam? Which one is an Allen screw?”. You will if you don’t read this section. The following explains all of the parts involved in assembling your ER1. You may refer back to this section throughout the assembly process to verify that you are using the right part in the right location.

*Important Note:* If you are missing any parts, contact Customer Support at [customerservice@evolution.com](mailto:customerservice@evolution.com) for replacements.



### **90° Connector**

This connector is used to form the laptop cradle. Remember to use set screws to tighten the connection.



The 90° connectors come as two halves that need to be pressed together to form a whole unit before they are inserted into an aluminum beam. There are four 90° connector halves in your kit.



### **135° Connector**

The 135° connectors are used to attach the top of the robot to the bottom.



The 135° connectors come as halves. You must press the two halves together before using the connector. There are four 135° connector halves in your kit.



### **Allen Screw**

The Allen screws are used for such things as attaching the motors and the struts. There are two sizes of Allen screws used in the ER1: 3/8" and 1/2". Use the two larger Allen wrenches that come with your kit to tighten the Allen screws. The Allen screws are already screwed into the T-nuts. (See the T-Nuts section for more information.) All of the Allen screws you will need are attached to the subassemblies, so you won't find any of these loose in the box.



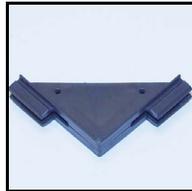
### **Camera Assembly**

The camera assembly is designed to be installed at the topmost part of the robot. The robot uses the camera for object recognition, and to take pictures and video. It provides the video feed for teleoperation, and can be used for rudimentary obstacle avoidance. (For better obstacle avoidance, a second camera (not included) is recommended.) There is one camera assembly in your kit.



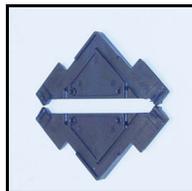
### End Cap

The end caps are used to cover the ends of the aluminum beams on your ER1. These will help protect you from any sharp edges. Be very careful attaching them to the aluminum beams. You may need a small hammer to get these to seat properly. There are four endcaps in your kit.



### Gusset

The gussets are used to help support the camera arm. Remember to use set screws to tighten the connection.



Gussets come in halves. Press the two pieces together before inserting them into the aluminum beam. There are four gusset halves in your kit.



### Hinge

This is a fully assembled hinge. A locking washer is located between the two halves to prevent unwanted rotation. A hinge is used in the camera arm. Remember to tighten the connection between these and the XBeam using set screws.



The hinge halves, when paired, make a hinge. These can also be used separately to attach an XBeam to the robot. There are four hinge halves in your kit.



### Locking Washer

The locking washer is used to lock joints into a certain position. This is particularly desirable in such cases as positioning the camera. There is one locking washer in your kit.



### Logo Plate

The logo plate goes on the front of your ER1. There is one logo plate in your kit.

---



### **Mounting Plates**

Mounting plates come in two varieties: foam (left) and mushroom headed fastener tape (right). Mushroom headed fastener tape is similar to Velcro®, but is designed for heavy-duty, industrial uses. There are seven mounting plates in your kit.



### **Philips Screw**

The Philips screws are used to connect a variety of components to each other. There are three Philips screws in your kit.



### **Power Cord**

The power cord connects the power module to the Robot Control Module. There is 1 power cord in your kit.

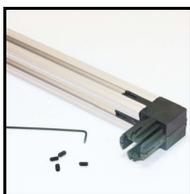


### **Power Module (Battery Pack)**

The power module is connected by the power cord to the Robot Control Module (RCM). It provides power to the robot's wheels and any robot peripherals. Note the on/off switch on the top. This is how you will turn off the ER1's motors. There is one power module in your kit. To order an additional power module, go to our website at [www.evolution.com](http://www.evolution.com).



This is the back view of the Power Module. The cable from the RCM should be plugged into Input 1. Input 2 is for later accessories and is not used by the robot at this time.



### **Set Screws**

Set screws should be added to all of the plastic connectors once they are in place. If you look carefully, all of the connectors have small holes in the sides. Once a connector has been placed in the XBeam, screw the set screw into the hole until it securely touches the beam. The setscrew, when installed correctly, should be below the surface of the connector, not flush with the surface. Do not overtighten. Until you do this, you will notice that the connection will be very loose. The smallest Allen wrench is used to screw in the set screws. There are 100 set screws included in your kit. This is more than you will need, so don't worry about counting them.



### Struts

The Struts are used to attach the laptop cradle to the caster wheel. There are two struts in your kit, one right and one left.



### T-nuts

T-nuts are used with the Allen screws to secure such things as the struts. All the T-nuts you will need are already attached to the subassemblies.



### Thumbnuts

The thumbnuts are paired with the Philips screws. Together, they secure joints in the robot. There are three thumbnuts in your kit.



### U-clips

The U-clips are used to connect the beams together. Remember to use the sets screws to tighten the connections.



U-clips, like the rest of the plastic connectors, come as halves. Press the halves together to create a whole U-clip. There are eighteen U-clip halves in your kit.



### USB Cord

The USB cord connect the Robot Control Module to your laptop. There is 1 USB cord in your kit.



### **Wheel Assembly**

The wheel assembly contains the wheel and a motor to move the robot. There are 2 wheel assemblies in your kit.



### **XBeams**

The XBeams are used to create the robot's body. You can use the ruler on the previous page to estimate the different sizes of XBeams mentioned in the Assembly Instructions, or you can use your own ruler or tape measure. Make sure that you measure the XBeams before adding them to the ER1's structure. There are three 12", three 10", two 8", two 7" and two 2" XBeams included in your kit. We recommend that you line them up from biggest to smallest to lessen the chances of using the wrong beam.

## **Things You Need**

- A ruler or tape measure
- A rubber mallet or a small hammer

## **Charging the ER1**

Charging your power module will take 4-5 hours, so start charging the power module as soon as possible. You don't have to wait until the power module is attached to the robot.

1. Plug in your charger to a wall socket. The ER1 charger is meant to use a 110V outlet. If you do not have this, you will need a voltage converter. Do not attempt to use any other voltage as this may cause damage to the unit and / or bodily injury.
2. Connect the other end of the charger to the power module in the input labelled Charger.
3. You will know when your power module is completely charged when the light on the charger is green.

A complete charge will allow the robot to run for about 3 hours (driving). Never fully drain the charge from your power module as this will significantly reduce the life of your power module. To check the level of charge in your robot, check the battery meter in the lower left of the Robot Control Center (RCC) software screen. You will install this software in a subsequent step.

If you would like to purchase an additional power module, contact Evolution Robotics at [www.evolution.com](http://www.evolution.com).

## Assembling Your “Assembled” ER1

This section describes the procedure to assemble your pre-assembled robot. Because of the configuration of the robot, it must be shipped in several pieces to avoid damage to the components. Remember to check your parts against your packing list to make sure you have received all of the parts you will need to build your ER1. Note that the list of parts for the assembled robot is slightly different than that of the unassembled robot.

**STOP!!!** Have you read the [Getting to Know the Parts](#) section? This section gives you the necessary information on the robot parts. Remember to refer to this section if you need further clarification on the assembly instructions.

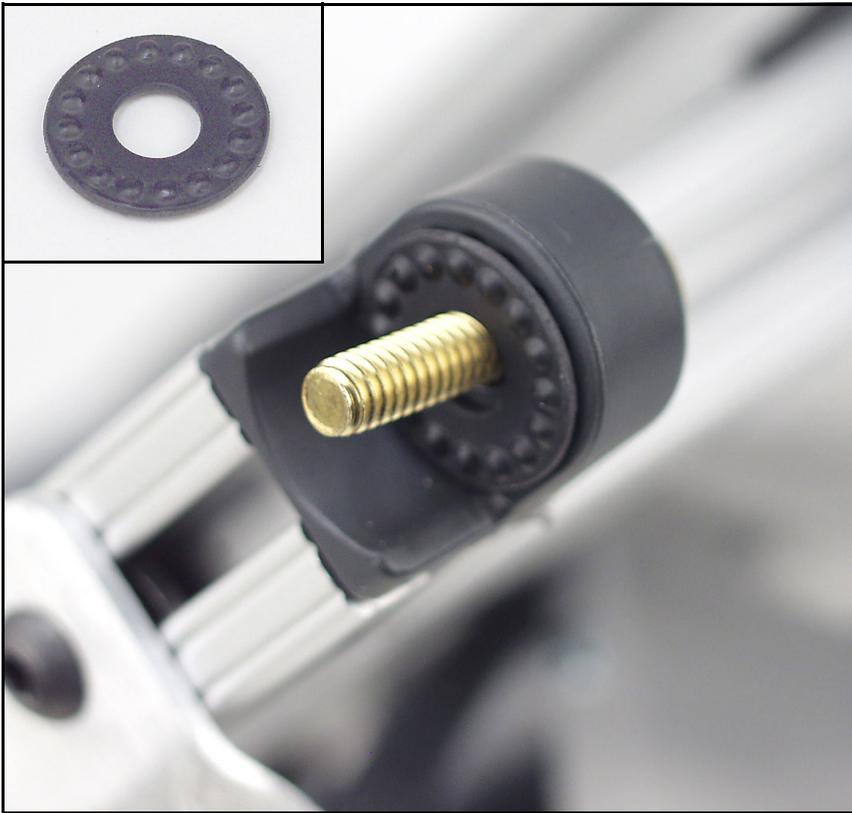


### Step 1.

You need:
<ul style="list-style-type: none"><li>• All of the robot parts from the box.</li></ul>



Instructions:
<ul style="list-style-type: none"><li>• You should have everything shown here.</li></ul>



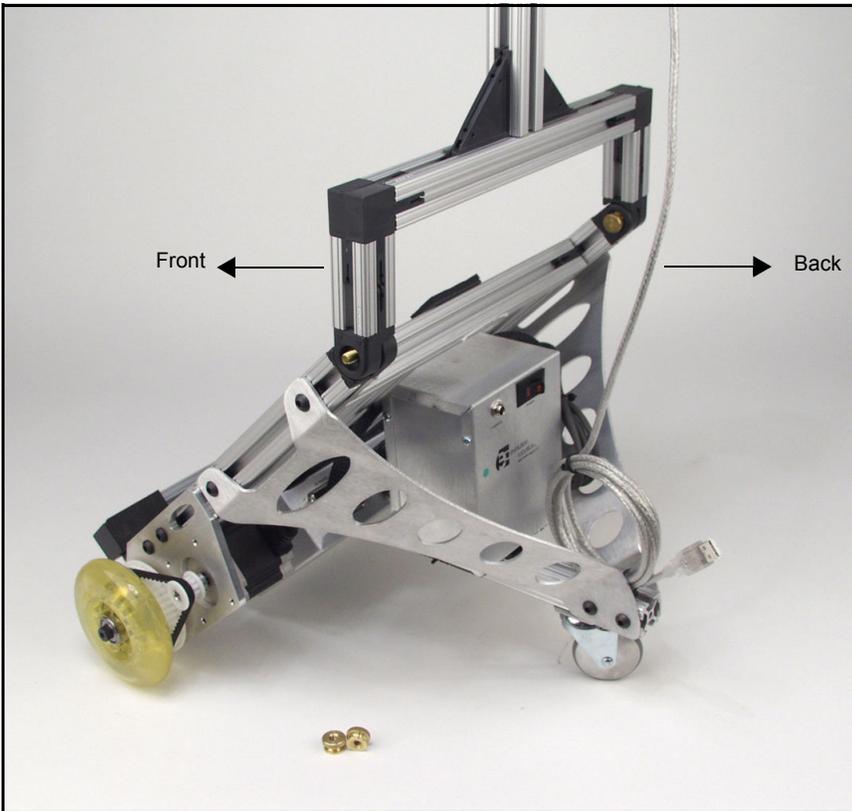
**Step 2.**

**You need:**

- 2 locking washers

**Instructions:**

- Place one locking washer on each of the screw posts.



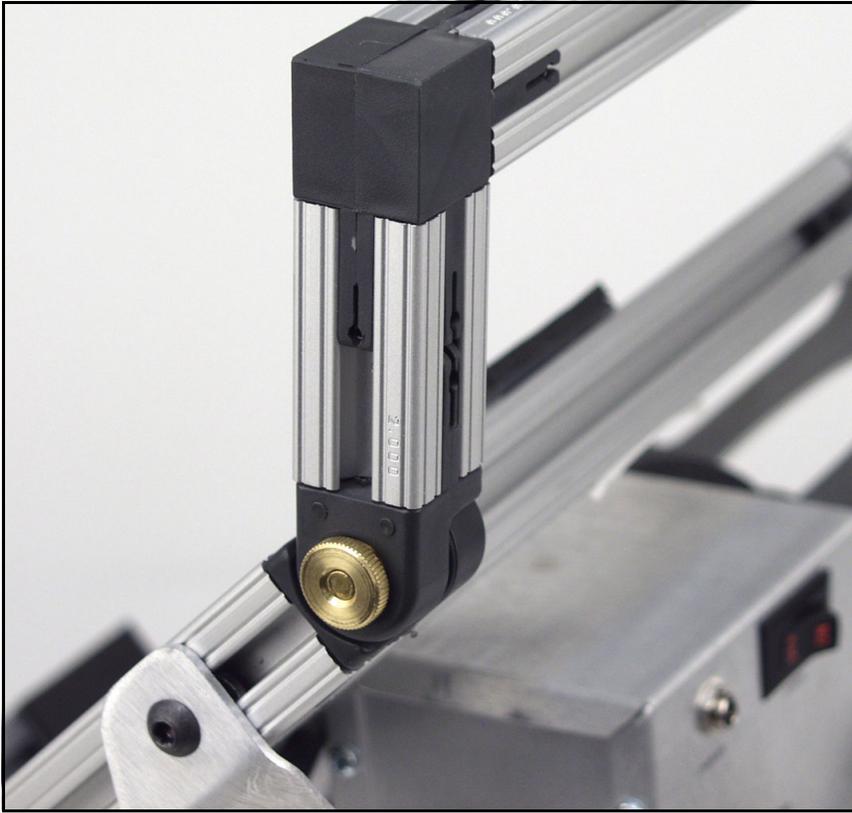
**Step 3.**

**You need:**

- The top of the robot (with the camera head).

**Instructions:**

- Attach the top of the robot to the screwposts with the locking washers as shown. You will need to pull the screws back through the holes about 1/2" and then thread them through the hinge halves on the top of the robot.
- Make sure that the camera is pointing toward the front of the robot.



**Step 4.**

**You need:**

- 2 Thumbnuts

**Instructions:**

- Screw the thumbnuts onto the screwposts.
- Make sure that they are firmly secured.



**Step 5.**

**You need:**

- 1 Laptop

**Instructions:**

- Plug the webcam into one of the USB ports on the back of your laptop (different for every laptop).
- Plug the RCM into the another USB port.



**Step 6.**

<b>You need:</b>
<ul style="list-style-type: none"><li>• Nothing</li></ul>



<b>Instructions:</b>
<ul style="list-style-type: none"><li>• You're done building your robot.</li><li>• Next, turn to the <a href="#">Installing the Software</a> section.</li></ul>

## Assembling Your Unassembled ER1

**STOP!!!** Have you read the [Getting to Know the Parts](#) section? This section gives you the necessary information on the robot parts. Remember to refer to this section if you need further clarification on the assembly instructions. Remember to check your parts against the [Getting to Know the Parts](#) section to make sure you have received all of the parts you will need to build your ER1.



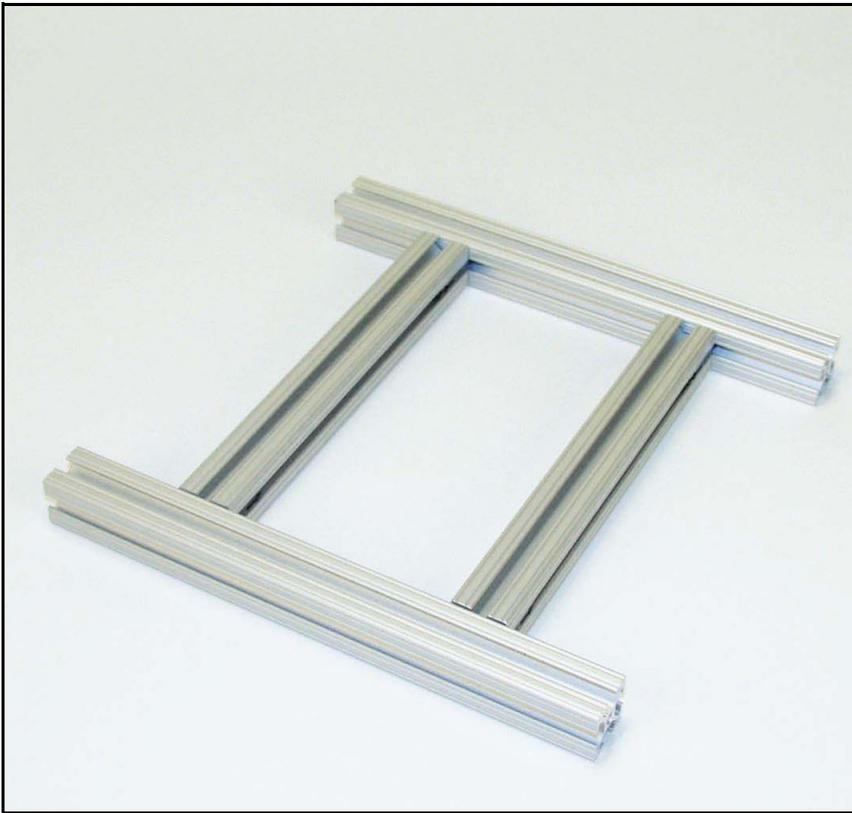
### Step 1.

#### You need:

- Two 8" XBeams
- 4 U-clips (8 half U-clips)
- Set screws

#### Instructions:

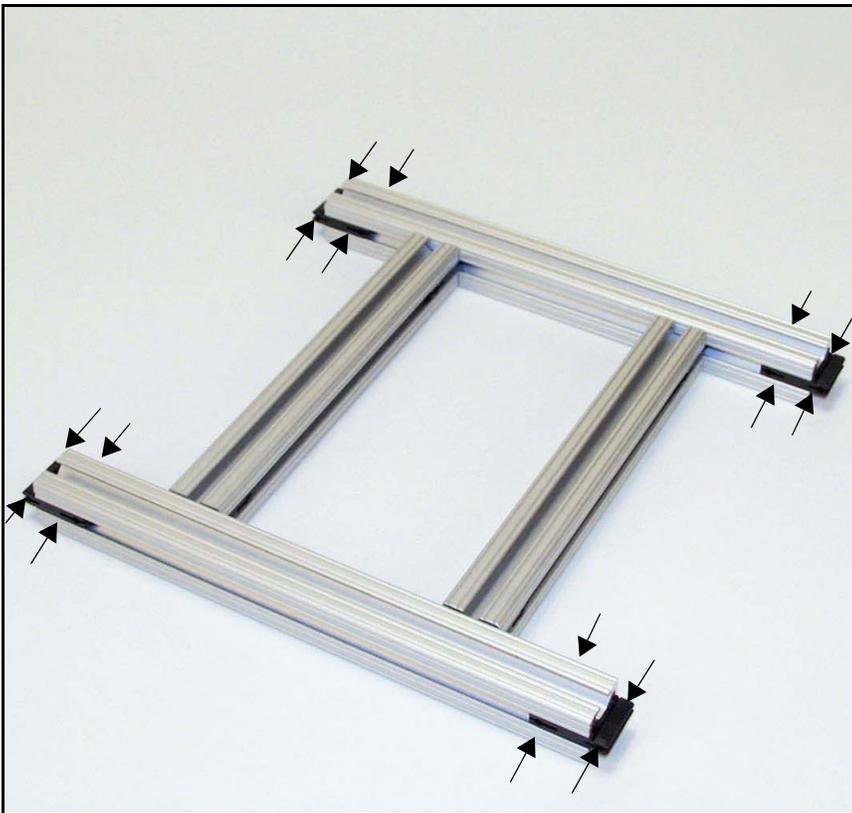
- Snap the U-clips halves together.
- Slide the U-clips into both ends of the two beams. Make sure that the U-clips are oriented in the same plane as shown (see black arrows).
- Screw in the set screws in the positions indicated by the gray arrows.



**Step 2.**

<b>You need:</b>
<ul style="list-style-type: none"><li>• Two 10" beams</li></ul>

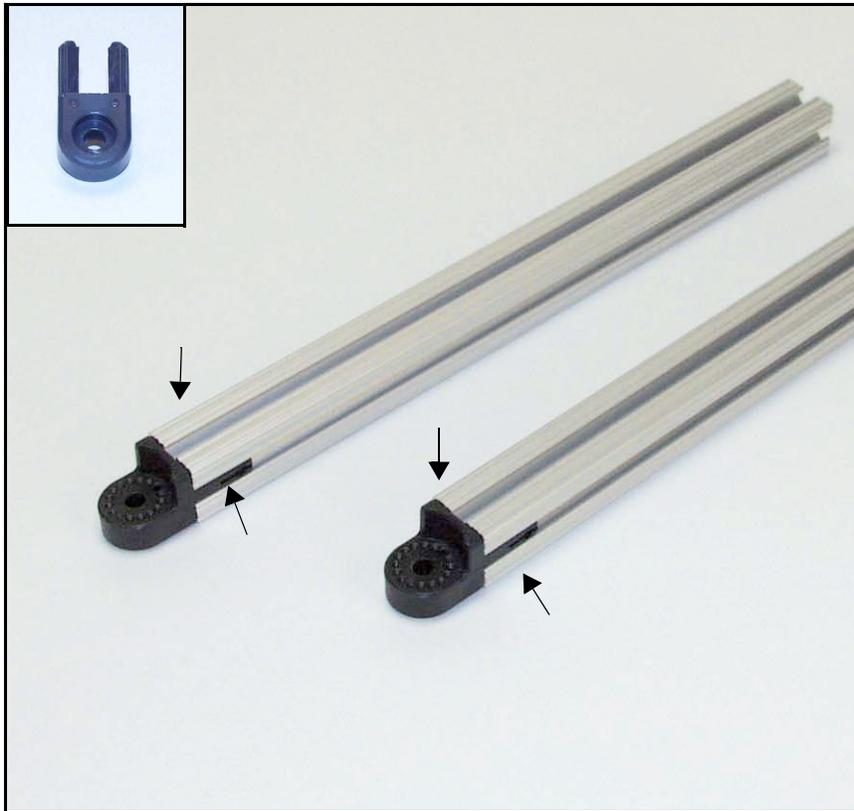
<b>Instructions:</b>
<ul style="list-style-type: none"><li>• Attach the 10" beams to the 8" beams as shown. The easiest way to do this is by sliding the 10" beams onto the U-clips.</li><li>• The 8" beams should be placed 2" from the ends of the 10" beams. You can use the 2" beams to measure this distance.</li><li>• Tighten the set screws as needed.</li></ul>



**Step 3.**

<b>You need:</b>
<ul style="list-style-type: none"><li>• 4 U-clips (8 half U-clips)</li><li>• Set screws</li></ul>

<b>Instructions:</b>
<ul style="list-style-type: none"><li>• Press the U-clip halves together to form 4 U-clips.</li><li>• Slide the U-clips into the ends of the beams.</li><li>• Screw in the set screws in the positions shown. (See the arrows.)</li></ul>



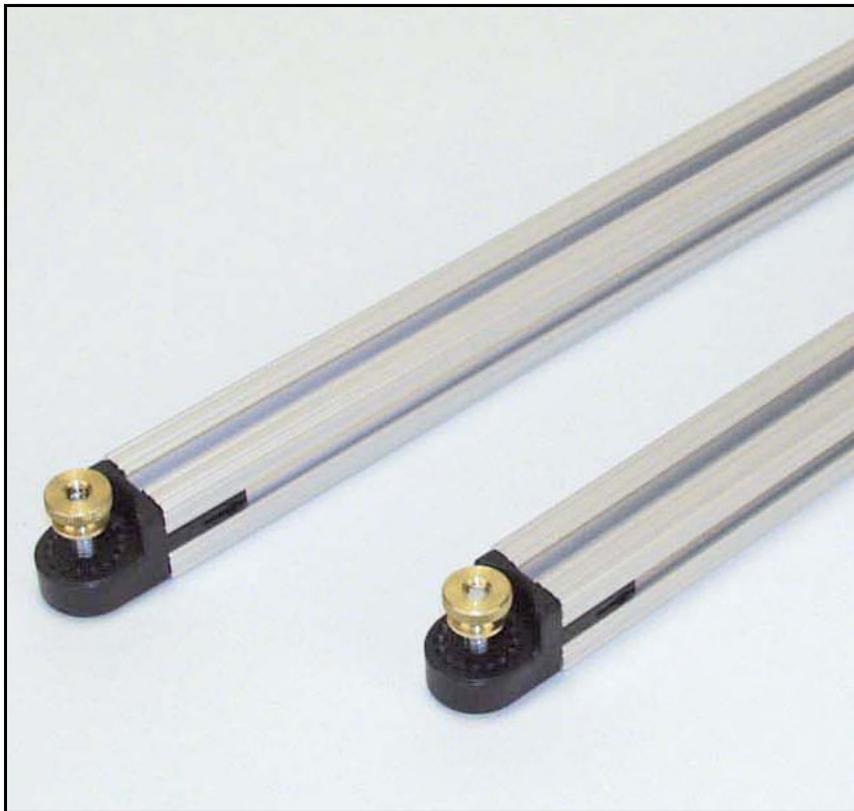
**Step 4.**

**You need:**

- Two 12" beams
- Two hinge halves
- Set screws

**Instructions:**

- Slide the hinge halves into one end of each beam.
- Remember to add the set screws to both sides of each of the hinge halves (see the arrows).



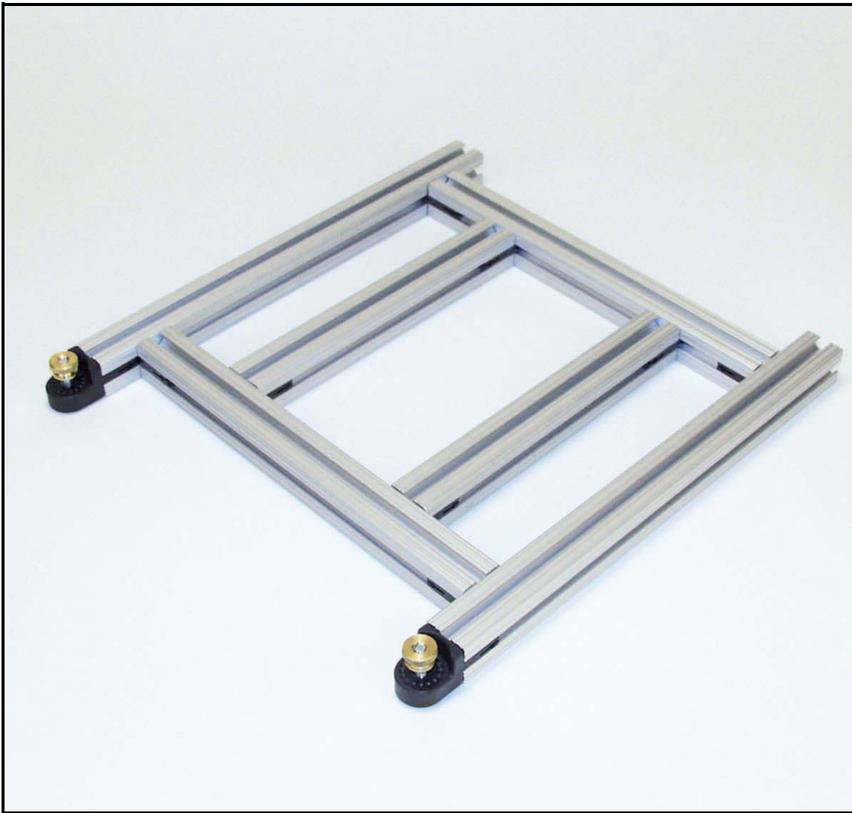
**Step 5.**

**You need:**

- Two Philips screws with thumbnuts

**Instructions:**

- Thread the Philips screws up through the bottom into the holes in the hinge halves on the end of each XBeam.
- Screw the thumbnuts (larger end up) onto the end of each screw.



**Step 6.**

**You need:**

- The 12" beams from the previous step.

**Instructions:**

- Attach the 12" beams to the 10" beams by sliding them onto the U-clips on the ends of the 10" beams.
- Make sure there is 1" clearance on the ends of the 12" beams.
- Make sure the thumbnuts are facing up as shown.



**Step 7.**

**You need:**

- 2 Wheel assemblies with 4 Allen screws with 2 one-holed T-nuts attached.

**Instructions:**

- Loosen the T-nuts on the screws.
- Slide the T-nuts down the track of the 12" XBeam to the bottom making sure that the motors are oriented as shown.
- Be sure to slide the wheel assembly all the way down to the hinge and tighten the screws. Make sure that the motor mounting plate is flush with the beams.
- Repeat for opposite side.



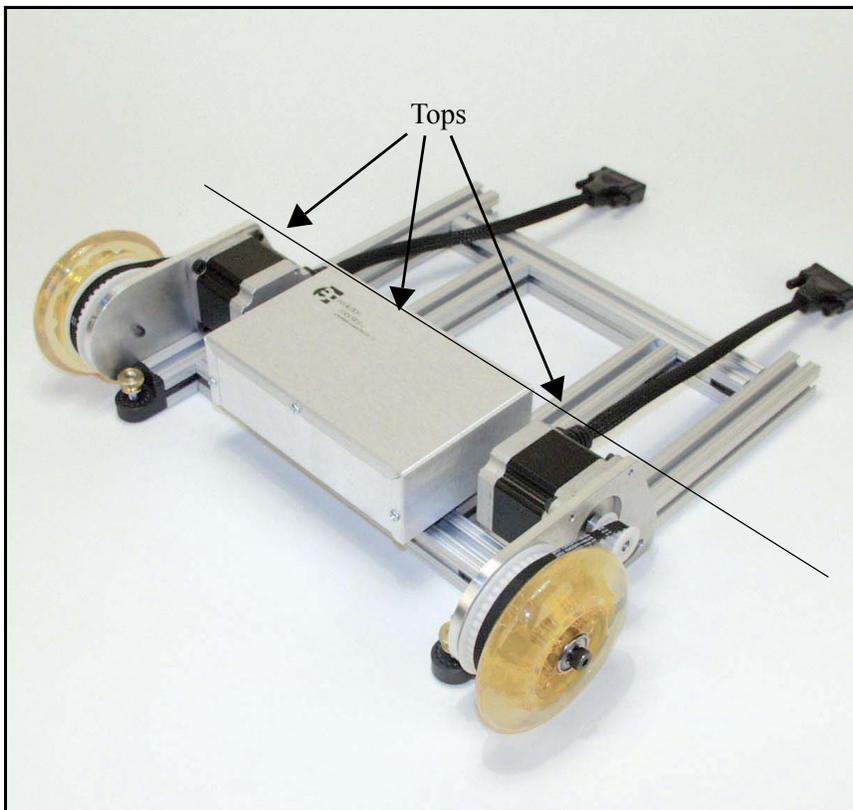
**Step 8.**

**You need:**

- 3 1x2 mounting plates with mushroom headed fastner tape

**Instructions:**

- Attach the mounting plates as shown.



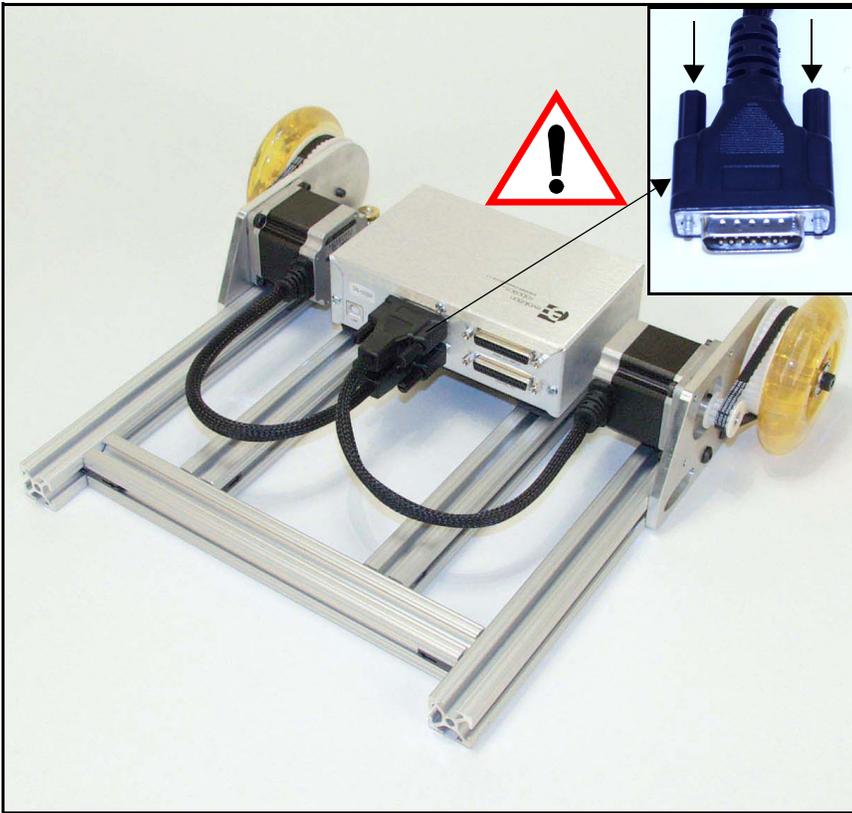
**Step 9.**

**You need:**

- 1 Robot Control Module (RCM)

**Instructions:**

- Use the mounting plates on the assembly and the mushroom headed fastner to attach the RCM.
- The top of the RCM should be level with the tops of the motors.
- Make sure the RCM is secure by pushing down firmly.



**Step 10.**

**You need:**

- The parts in front of you

**Instructions:**

- Plug the right (as seen here) motor cable into the slot labelled Motor 1. Plug the left (as seen here) motor cable into the slot labelled Motor2.
- **Tighten the screws** on either side of each motor cable connector (see arrows).  
**WARNING: If the motor cable connectors should become loose, they will cause SEVERE damage to the RCM, so double check that the screws are completely secure.**



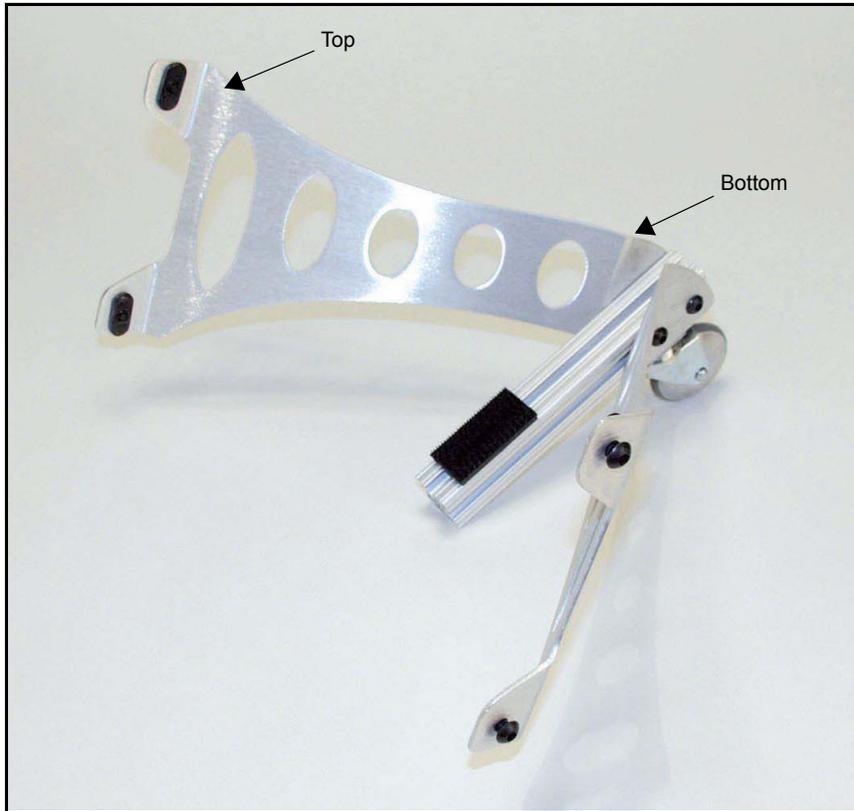
**Step 11.**

**You need:**

- 7" XBeam
- 1 Castor wheel with two 3/8" Allen screws and T-nuts attached
- 1x2 mounting plate with mushroom headed fastner tape

**Instructions:**

- Slide the 1x2 mounting plate onto the beam.
- Loosen the T-nuts on the castor and slide them into the track on the X-beam.
- Position the castor mount plate ~1/2" from the end of the beam as shown. Tighten the screws.



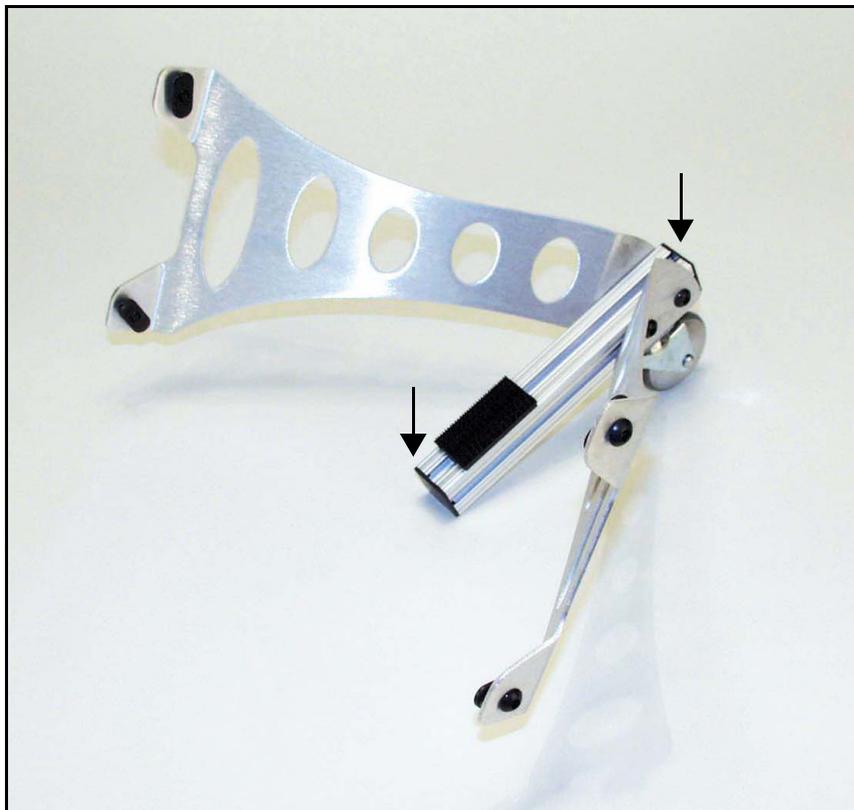
**Step 12.**

**You need:**

- 2 struts, one right, one left, each with 4 Allen screws, 2 one-holed and 1 two-holed T-nuts.

**Instructions:**

- Slide the T-nut on the bottom of the strut down the track of the XBeam flush with the end of the beam as shown. Note that the struts are mirror images of each other. Position them as shown.
- Tighten the screws.
- Repeat for the other side, making sure to position as shown.



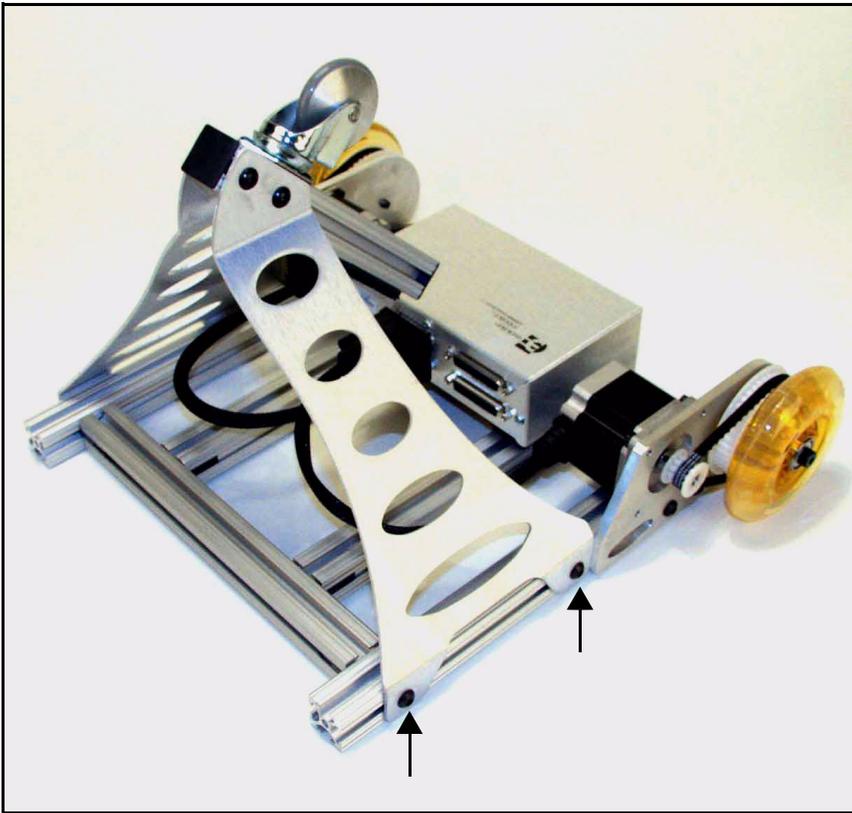
**Step 13.**

**You need:**

- 2 endcaps

**Instructions:**

- Place endcaps on both ends of the 7" beam. You may need to tap these with a small hammer to get them to seat properly.



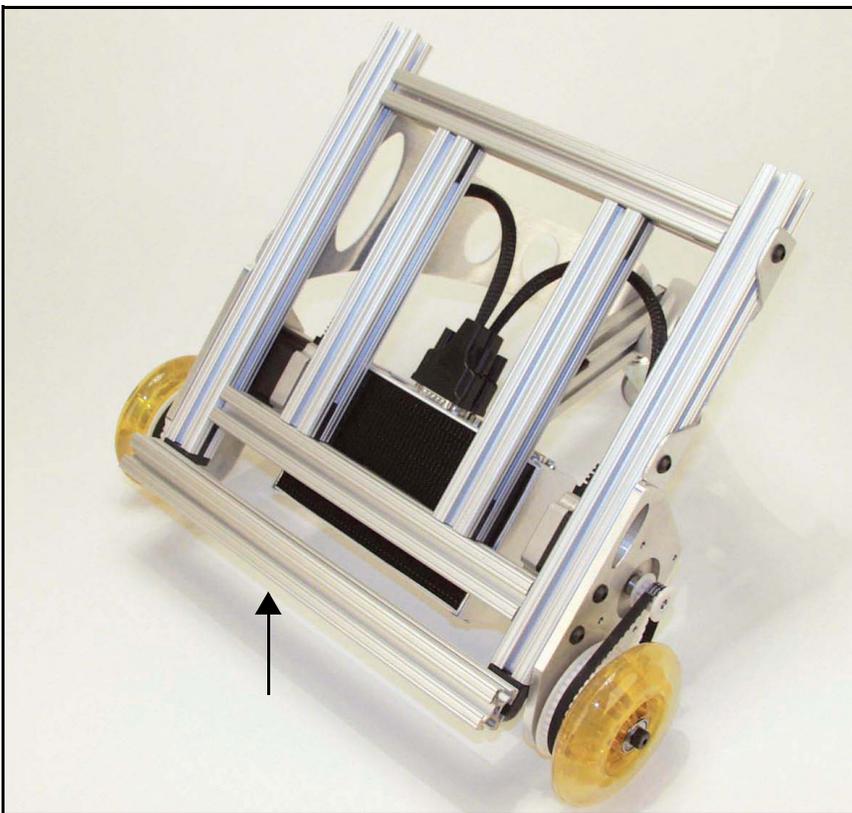
**Step 14.**

**You need:**

- 4 Allen screws with 2 one-holed T-nuts already attached to the struts.

**Instructions:**

- Slide the T-nuts at the top of the struts down the tracks of the 12" beams on both sides to the end.
- Tighten the screws (see arrows - only one side shown).



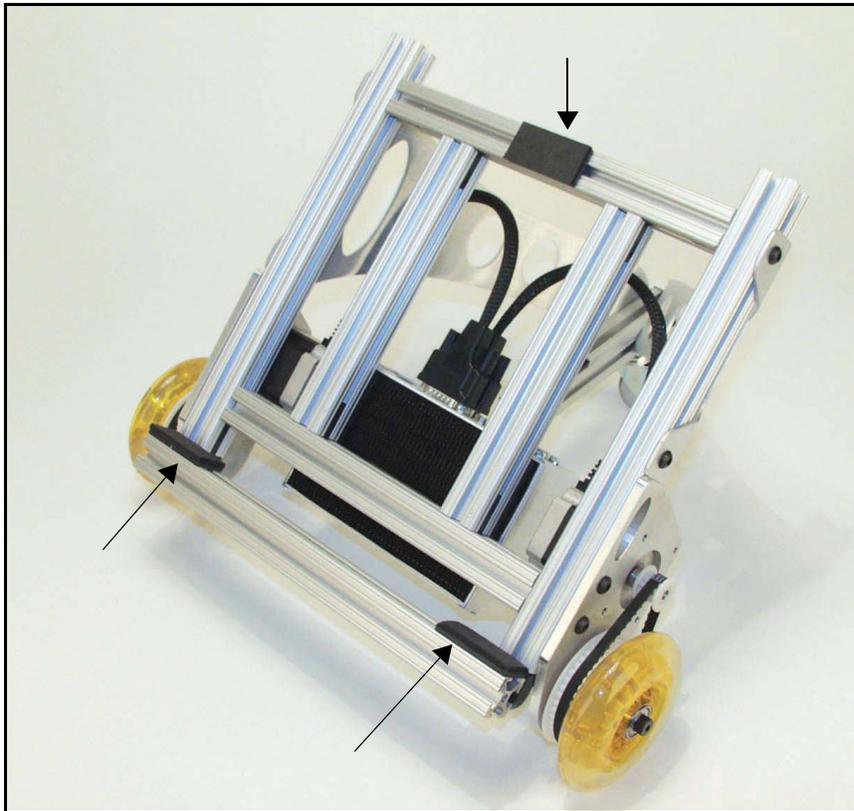
**Step 15.**

**You need:**

- 1 12" beam

**Instructions:**

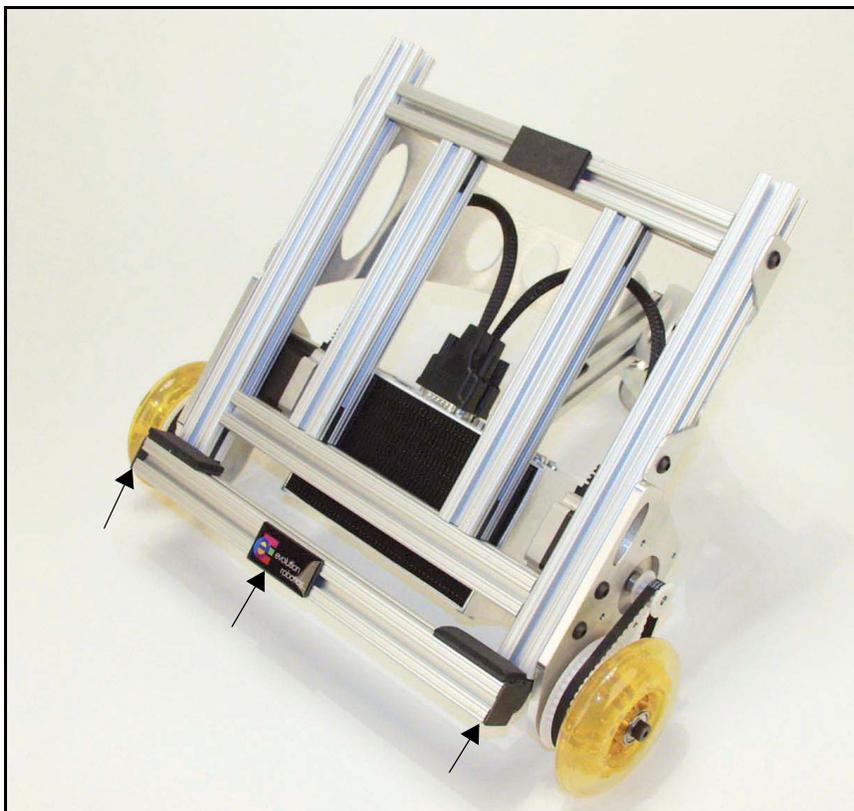
- Attach the 12" beam to the bottom front of the robot by sliding the screw heads in the hinges halves through the XBeams channel.
- Slide the XBeam on until centered on the robot.
- Tighten the thumbnuts - pliers are helpful here.



**Step 16.**

- | You need:  |
|--|
| <ul style="list-style-type: none"><li>• 3 1x2 mount plates with foam</li></ul> |

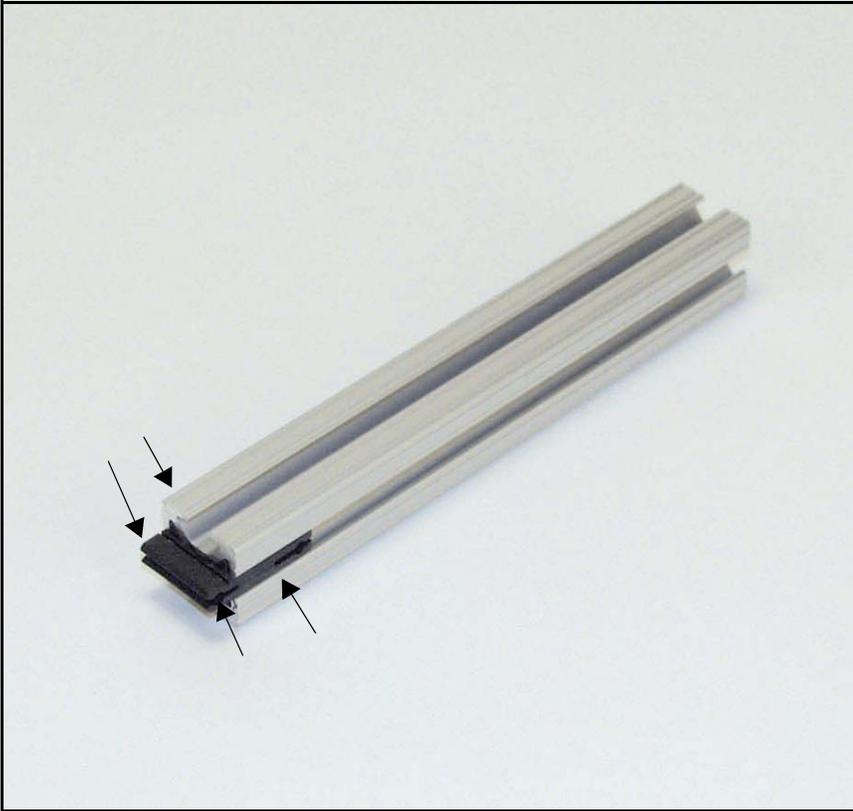
- | Instructions:   |
|---|
| <ul style="list-style-type: none"><li>• Apply the mount plates by sliding them into the beams in the positions shown.</li></ul> |



**Step 17.**

- | You need:  |
|--|
| <ul style="list-style-type: none"><li>• 2 endcaps</li><li>• Logo plate</li></ul> |

- | Instructions:   |
|---|
| <ul style="list-style-type: none"><li>• Slide the logo plate onto the middle of the beam.</li><li>• Attach the endcaps to the ends of the 12" beam. You may need to tap these with a small hammer to get them to seat properly.</li></ul> |



**Step 18.**

**You need:**

- 1 7" beam
- 1 U-clip (2 U-clip halves)
- Set screws

**Instructions:**

- Snap the U-clip together and slide it into the end of the beam.
- Add the set screws as shown (see arrows).



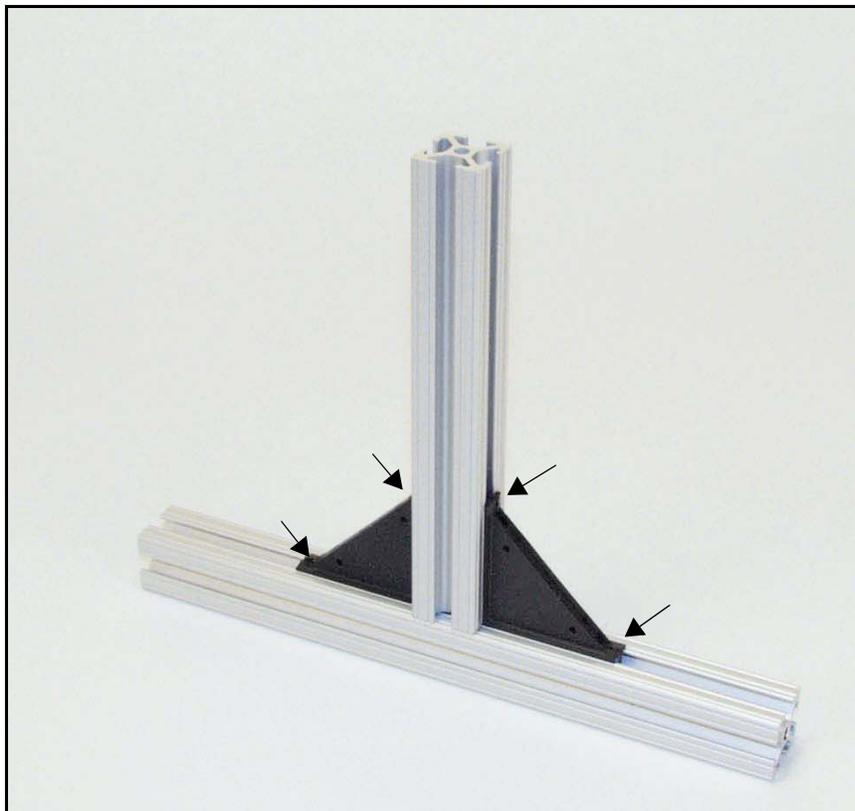
**Step 19.**

**You need:**

- 1 10" beam

**Instructions:**

- Slide the 10" beam onto the end of the 7" beam using the U-clip and center it as shown.



**Step 20.**

You need:
<ul style="list-style-type: none"> <li>• 2 whole gussets (4 halves)</li> <li>• Set screws</li> </ul>

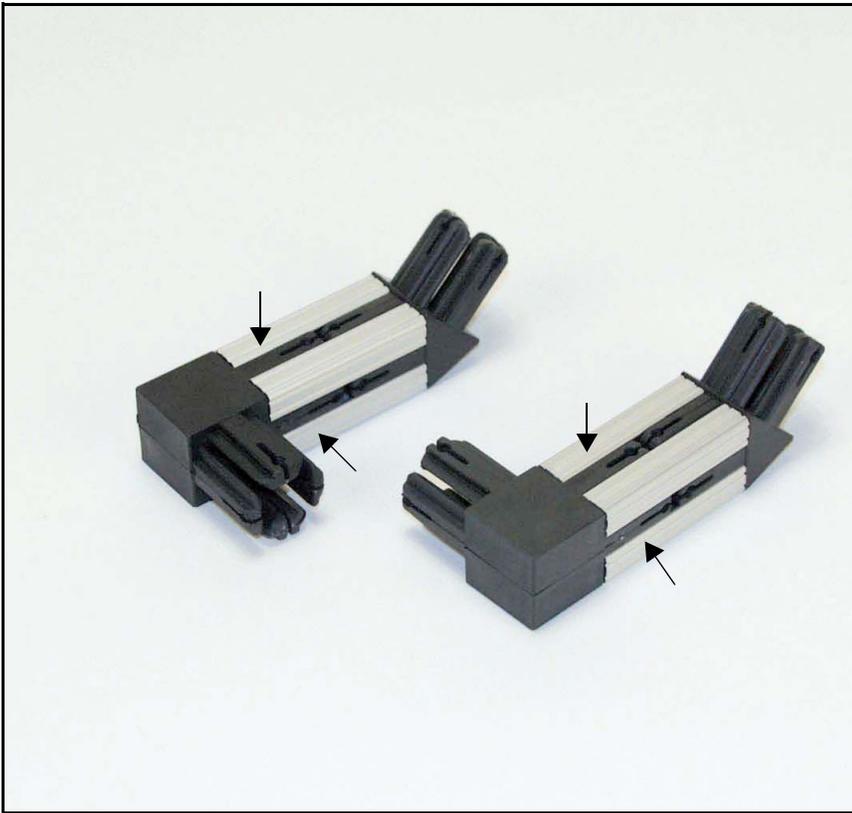
Instructions:
<ul style="list-style-type: none"> <li>• Snap the gussets halves together</li> <li>• Snap the gussets into place.</li> <li>• Remember to add the set screws! There will be some very small holes toward the center of the gusset. Do not put set screws in these holes. Only place set screws as shown (see arrows).</li> </ul>



**Step 21.**

You need:
<ul style="list-style-type: none"> <li>• 2 2" beams</li> <li>• 2 whole 135° connectors (4 halves)</li> <li>• Set screws</li> </ul>

Instructions:
<ul style="list-style-type: none"> <li>• Press the 135° connector halves together.</li> <li>• Slide the 135° connectors onto the ends of the 2" beams.</li> <li>• Add the set screws to all four sides of both of the connectors (see arrows - not all set screws are shown).</li> </ul>



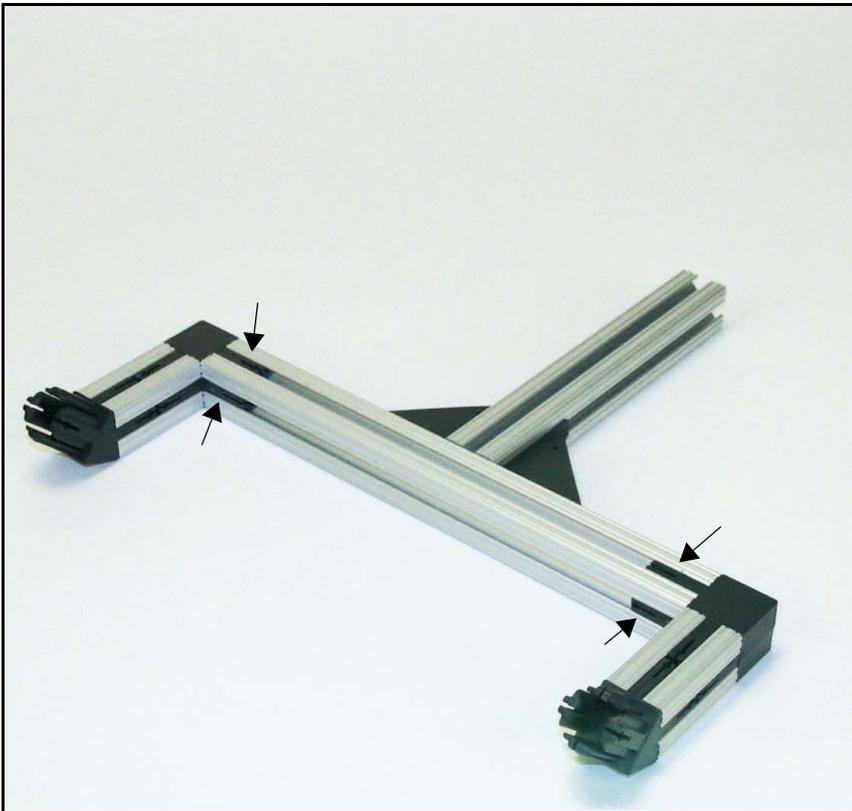
**Step 22.**

**You need:**

- 2 whole 90° connectors (4 halves)
- Set screws

**Instructions:**

- Press the 90° connector halves together
- Attach them to the end of the 2" pieces as shown.
- Add the set screws to all four sides of the 90° connectors (see arrows - not all set screws shown).



**Step 23.**

**You need:**

- Subassemblies from previous steps
- Set screws

**Instructions:**

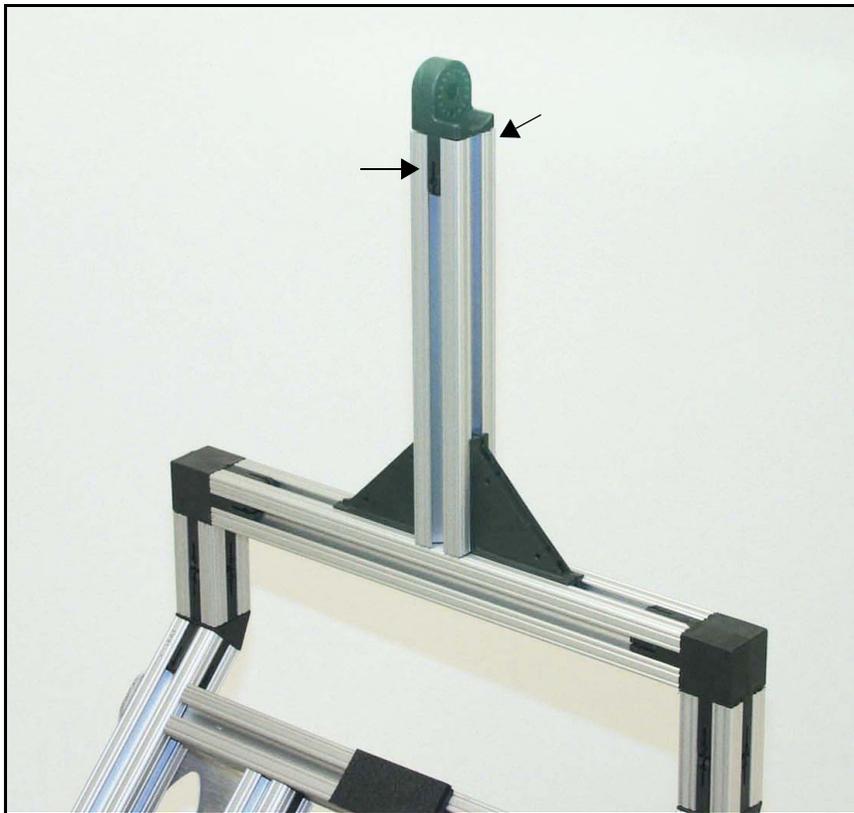
- Connect the subassemblies by attaching the 2" XBeams either end of to the 10" XBeam using the 90° connectors.
- Add set screws to all four sides of the 90° connectors where they attach to the 10" beam (see arrows - not all set screws shown).



**Step 24.**

You need:
<ul style="list-style-type: none"> <li>• Previous subassemblies</li> <li>• Setscrews</li> </ul>

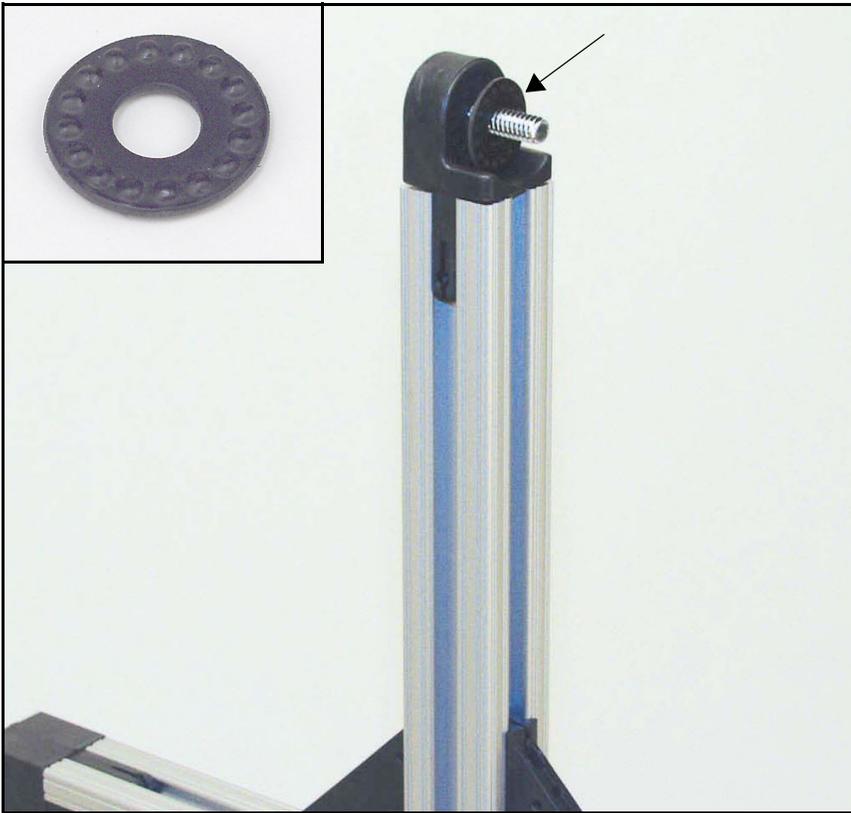
Instructions:
<ul style="list-style-type: none"> <li>• Attach the subassembly from the last step to the robot body using the 135° connectors.</li> <li>• Add set screws on all four sides of the 135° connectors that attach to the body of the robot (see arrows - not all set screws shown).</li> <li>• <b>WARNING:</b> Do not pick up the robot by the crossbar (circled).</li> </ul>



**Step 25.**

You need:
<ul style="list-style-type: none"> <li>• 1 hinge half</li> <li>• Set screws</li> </ul>

Instructions:
<ul style="list-style-type: none"> <li>• Attach a hinge half to the 7" beam.</li> <li>• Add set screws on both sides of the hinge (see arrows).</li> </ul>



**Step 26.**

- | You need:  |
|--|
| <ul style="list-style-type: none"><li>• 1 Philips screw</li><li>• 1 locking washer</li></ul> |

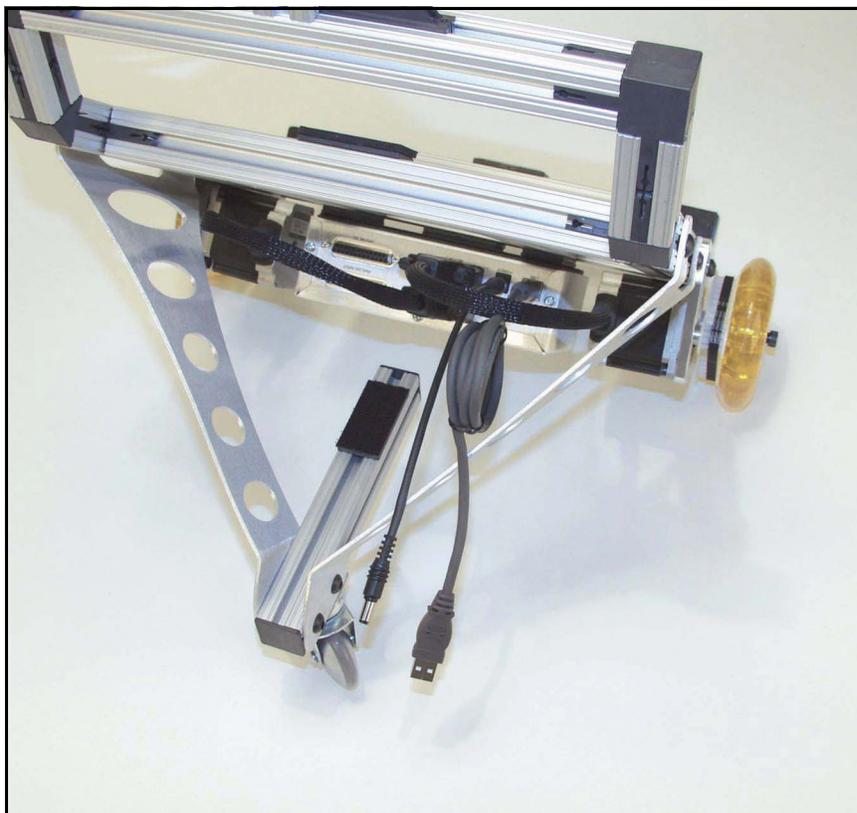
- | Instructions:  |
|--|
| <ul style="list-style-type: none"><li>• Thread a Philips screw through the hole in the hinge half.</li><li>• Place the locking washer on the screw as shown (see arrow).</li></ul> |



**Step 27.**

- | You need:   |
|---|
| <ul style="list-style-type: none"><li>• 1 camera subassembly</li><li>• 1 thumbnut</li></ul> |

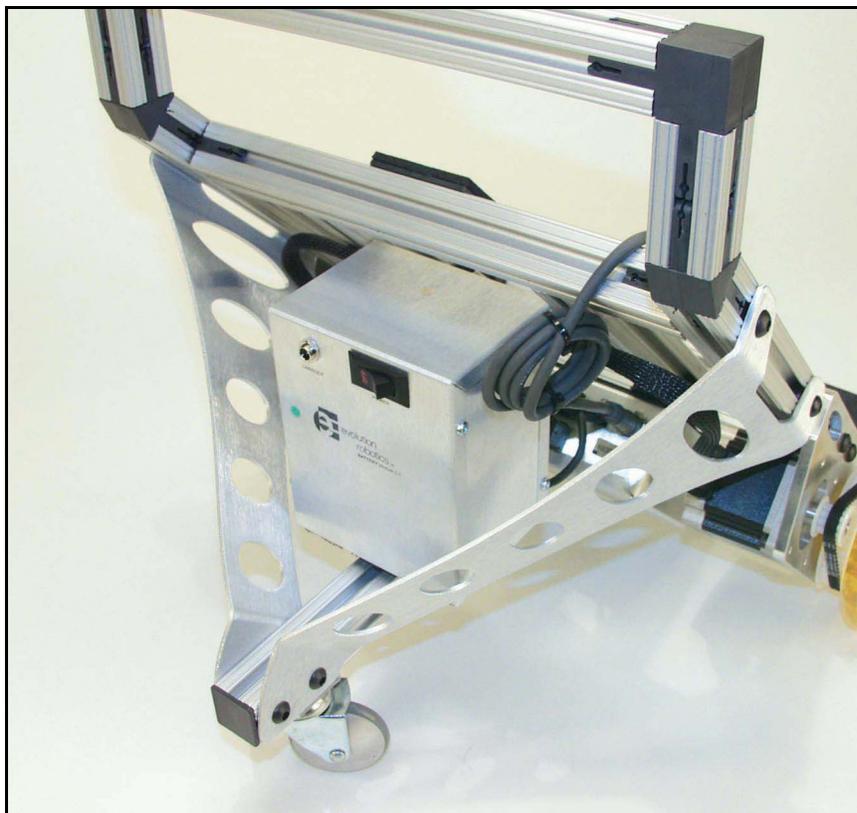
- | Instructions:  |
|--|
| <ul style="list-style-type: none"><li>• Slide the hinge half on the bottom of the camera subassembly onto the screw post.</li><li>• Make sure to position camera subassembly facing forward (see picture in upper left).</li><li>• Screw the thumbnut onto the screw post and tighten.</li></ul> |



**Step 28.**

You need:
<ul style="list-style-type: none"><li>• 1 power cord</li><li>• 1 USB cable</li></ul>

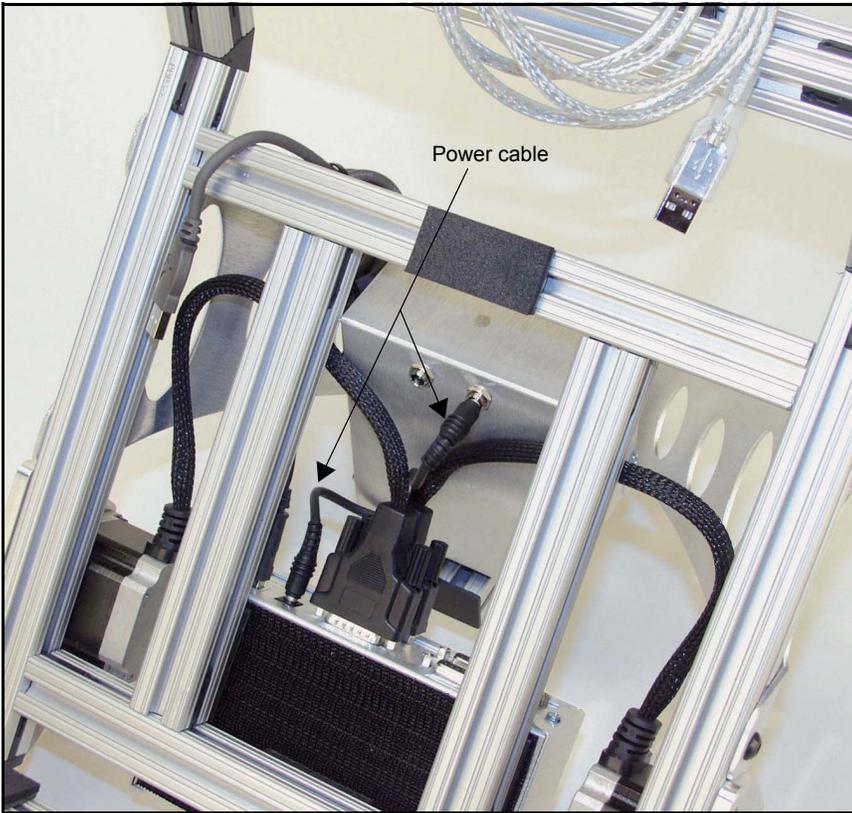
Instructions:
<ul style="list-style-type: none"><li>• Attach the power cord to the slot in the RCM marked power.</li><li>• Attach the USB cord to the slot in the RCM marked USB.</li><li>• Tidy up and tuck away all of your cables.</li></ul>



**Step 29.**

You need:
<ul style="list-style-type: none"><li>• 1 power module</li></ul>

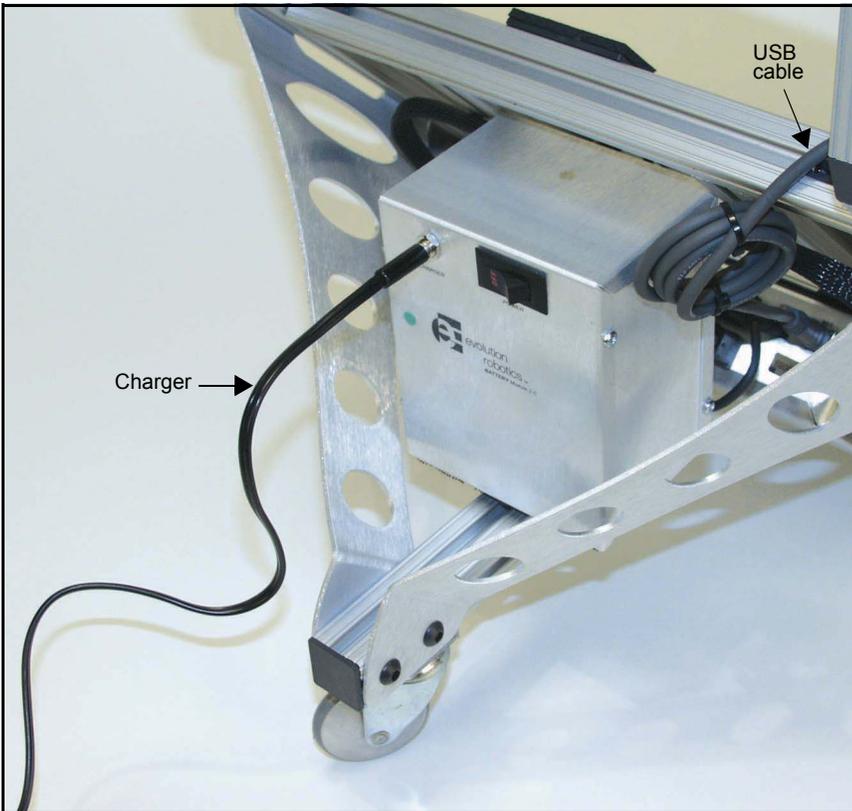
Instructions:
<ul style="list-style-type: none"><li>• Secure the power module to the robot by pressing down until the the mushroom headed fastner tape on the 1x2 mounting plate and the mushroom headed fastner tape on the bottom of the battery are interlocked.</li></ul>



**Step 30.**

- |   |
|---|
| <b>You need:</b>  |
| <ul style="list-style-type: none"><li>• Power cable</li></ul> |

- |   |
|---|
| <b>Instructions:</b>  |
| <ul style="list-style-type: none"><li>• Plug the RCM into the Power 1 slot on the power module.</li></ul> |



**Step 31.**

- |   |
|---|
| <b>You need:</b>  |
| <ul style="list-style-type: none"><li>• 1 Charger</li></ul> |

- |  |
|--|
| <b>Instructions:</b>   |
| <ul style="list-style-type: none"><li>• Plug the power module into the charger as shown (if you haven't completely charged your power module already).</li></ul> |

**Step 32.****You need:**

- Nothing

**Instructions:**

- You're done building your robot!
- The next section will walk you through installing your software.

## Installing the Software

*Important Note:* For Microsoft® Windows Me users: you may be prompted to replace the `kstvtuner.ax` (WDM streaming TvTuner), `kswdmcap.ax` (WDM streaming video capture), and `ksxbar.ax` (WDM streaming crossbar) files with older versions of the same files. You should decline to do this and you should keep the newer versions of the files.

Here's how to install the software on your laptop:

1. Close all programs.
2. Put the CD in the laptop's drive.
3. If the Robot Control Center (RCC) software installation does not start automatically, you can start it from the CD. Different Windows platforms let the user do this in different ways. Here's how to do it in a Windows 2000 environment:
  - On your Windows desktop, click on the **Start** button.
  - Choose **Run**.
  - In the pop-up window, use the **Browse** button to navigate to the directory for the CD-ROM drive.
  - In the top-level directory, double-click on the file `setup.exe`.
4. There may be a dialog box telling you to reboot before the installation. Click Yes and the installation should automatically continue after the reboot.

5. The software installation should start automatically, if your computer uses Microsoft's autorun feature. Click **Next** to proceed through the installation. You may click on the **Cancel** button at any time to stop the installation.
6. The ER1 Robot Control Center licensing agreement will be displayed and you will be prompted to accept/reject it. Click **Next**.
7. The Release Notes will be displayed. Make sure to print these out and read them. They contain information that cannot be found in the *ER1 User Guide*. Click **Next**.
8. You will be prompted to accept/reject the usual setup options for destination folder. Click **Next**.
9. Typical/custom install - You will also be prompted to choose a type of installation. Usually you should select **typical**. Click **Next**.
10. A message is displayed: **Ready to Install**. Click **Next**. Wait while the system copies files for installation.
11. The software will confirm that your computer has the latest version of DirectX installed. The ER1 software requires version 8.1 or newer.
12. You will then be prompted to install DirectX 8.1. Click **Yes**.
13. You will then be shown the Microsoft license agreement for DirectX and you will be prompted to accept/reject it. Click **Yes**.
14. A popup dialog may be displayed asking you to reboot your computer. If so, select **Ok** to reboot and then click **Finish**.
15. Reboot your computer.
16. After the installation is complete, a shortcut icon for the ER1 Robot Control Center software will appear on your desktop. Double-click on it to start the ER1 application.

*Important Note:* You should always start the software from the RCC icon or from the installed destination directory. Otherwise, the software may not be able to find all the associated files it needs to operate correctly.

## **Connecting the Laptop**

The following is an example of how to connect your laptop if you have Windows 2000. Other versions of Windows will be similar.

1. Place your laptop in the cradle of the robot.
2. Connect the Robot Control Module (RCM) to the laptop's USB port using the USB cable provided with the robot.
3. When you plug the RCM's USB cable into the laptop, its Windows should display a prompt for hard drive locations to search for drivers for the new hardware.
4. Choose **Search for a suitable driver for my device**. Select **Next**. *Windows 98 users:* You may be prompted for the Windows 98 CD.

5. Specify the CD-ROM drive. *Windows 98 users:* You may also need to specify the ER1 Robot Control Module folder in order to specify the correct driver. Select **Next>Next>Finish**.
6. After installing the RCM drivers, you will notice that Windows will find 'New Hardware' which is the USB Serial Port. A "Found New Hardware Wizard" will pop up.
7. Select the default 'recommended' search for drivers.
8. Click on **Next**.
9. Click on **Continue Anyway** in regards to the not passing Windows logo testing. Click on **Finish**. Windows 98 users may not see this dialog box.
10. Plug the Webcam's USB cable into another port on the laptop.
11. Again, you will see the New hardware found message, this time for the Kritter USB camera A/V.
12. Select to **Search removable media** and specify the CD-ROM. Windows 98 users may have to browse for the specific location of the driver on the CD. It is in a folder named `Camera_98` on the RCC CD.
13. Detects driver and gives "Digital Signature Not Found" message. Select **Yes**.
14. If you have purchased a microphone, connect the robot's microphone to a USB port on the laptop using the microphone's cable. Configure your laptop to use that microphone.
15. Go through the same process to add any new hardware, such as an additional camera. (Note that you will may a USB hub).
16. Power cycle the robot power, reseal both the USB cables, and launch the ER1 software on the laptop. *Windows 98 users:* You may need to reboot your laptop.

*Important Note:* UNPLUG THE LAPTOP'S CHARGER BEFORE INSTRUCTING THE ROBOT TO MOVE. The robot should detect whether or not the laptop is plugged into a wall outlet and should not move until the laptop is unplugged. However, take care to ensure that the laptop is unplugged, because if the robot fails to detect that the laptop is plugged in, it will move in response to your commands until it reaches the limit of the laptop's power cord, then will continue to try to move, potentially damaging the laptop, the laptop's charger, the robot, and the electrical outlet. This could start an electrical fire.

## Testing the Robot

The ER1 has a behavior file included to help you test your ER1 robot and software.

Before we start with an example, try moving your robot with the arrow keys on your keyboard. If the robot does not move, recheck your cable connections.

The following example will teach your robot to follow the box it came in.

1. Click on the ER1 icon to start the application on your laptop.
2. Click on the **Open** button and the **Open File** dialog should come up, showing the contents of the ER1 program directory.

3. Double click on the `behaviors` subdirectory to open it. This will display a list of the `.rbt` files in this directory.
4. Select the `ER1box.rbt` file and click on the **Open** button to load it.
5. The `ER1box.rbt` behavior set will load.
6. Click on the green **Run It** button at the top right of the RCC program screen.
7. Pick up the box that the robot was packed in, and show it to the robot (the front side of the box with the multicolored ER1s should be facing the robot's camera).
8. The robot should tell you that it sees the box, and that it will move towards the box if you move the box farther away from it. Try moving the box various distances from the robot.
9. When you're done with the robot behavior, click **Stop**.

*Important Note:* If you didn't get a chance while you were installing the software, take a minute to look at the Release Notes in the `docs` subdirectory of the ER1 program directory. You will find important information that is not included in this manual.

## **Uninstalling**

If you use the Control Panel's Add/Remove Software Programs utility, make sure that you also delete the entire installation folder, which will contain user data files, log files, and captured images.

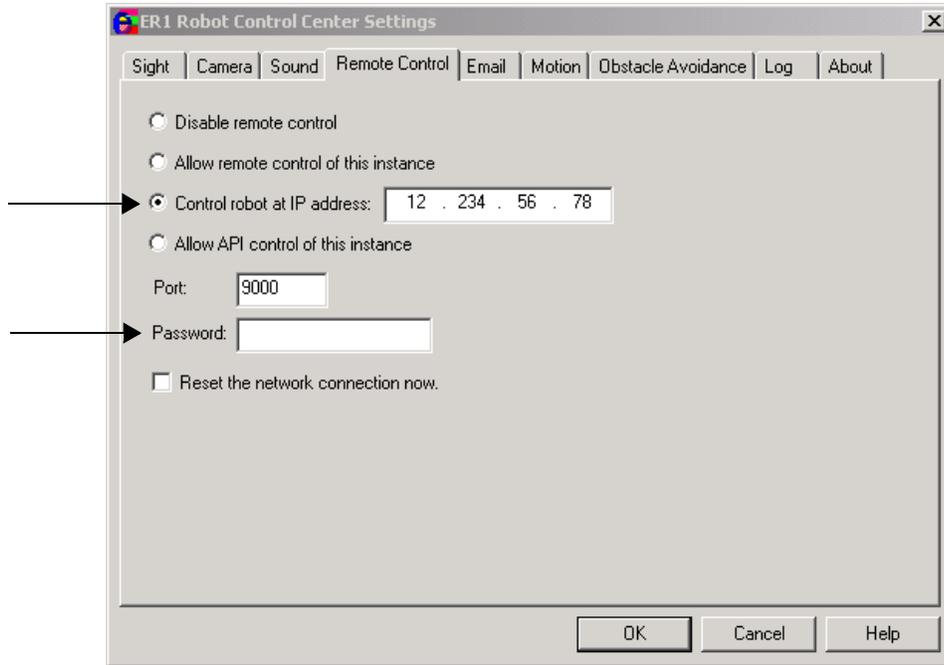
## ■ Appendix B

---

# Telepresence Over the Internet

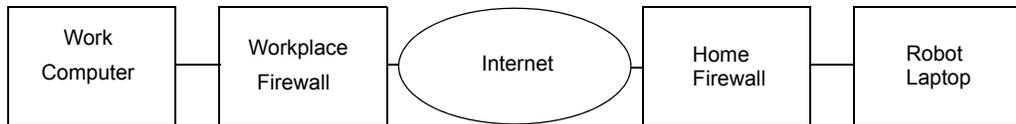
In order for the user to teleoperate a home robot from work, the IP address and the port number of the laptop controlling the robot must be known. An IP address is a number of the form  $x.x.x.x$ , where  $x$  is between 0 and 255. The port number is an integer between 0 and 65535. The port number of ER1 teleoperation defaults to 9000, but the user can change this in the remote settings page of the ER1 Robot Control Center software running on the robot laptop.

The IP address of the robot is the IP address assigned to the user's DSL or cable connection by the user's broadband provider. Typically, a new dynamic IP address is given to the user on each new connection. However, DSL and cable connections are always on and can persist for days, if not weeks, so the dynamic IP address is fairly stable over time. To figure out what the current assigned IP address is, the user can use a tool such as MyWanIP, freely downloadable from <http://www.mywanip.com/>. The assigned IP address is also available in the control software for the user's home router. The documentation for the various cable / DSL routers should have information on where this IP address can be found.



Once the IP address and teleoperation port number are identified, they can be entered into the remote control settings page on the ER1 Robot Control Center running on the user's work computer. Click on the **Settings** button and then the remote control tab. Click on the **Control robot at this IP address** checkbox and enter the IP address. The port number should not need to be changed. (Remember to have the **Allow remote control of this instance** set on your robot's laptop.)

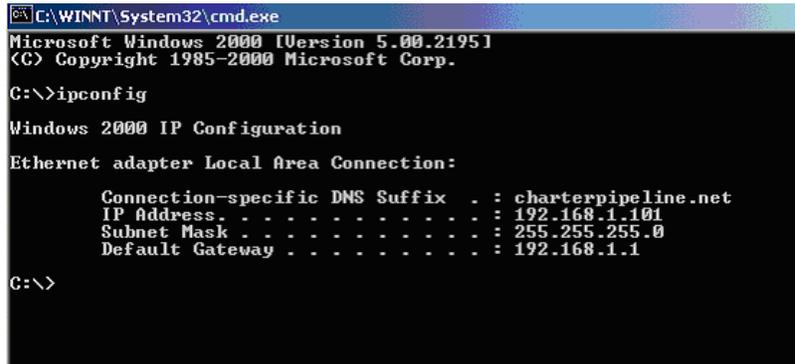
After the IP address and port for teleoperation are located and entered into the RCC software, there is one last step: setting up port-forwarding to allow the work computer to connect to the robot laptop at home through the your home firewall. The most common network setup between a user at work and the robot at home is a double firewall situation. At work, you are behind your workplace's firewall. At home, the robot is on a wireless network connected to a combination of wireless access point, router, and cable or DSL modem that also functions as a firewall.



In order to control the robot via teleoperation, you must be able to connect from the work computer to the robot through the home firewall. Most cable or DSL routers have a service called port forwarding. This service forwards communication on certain ports to certain specified computers behind the firewall. Assuming that you opt to use the default port of 9000 for teleoperation, you will need to configure the home router to forward port 9000 to the robot laptop's IP address on your home network. Note that this IP address is not the same as the external IP address assigned to you by the broadband provider. It is typically of the form 192.168.x.x or 10.x.x.x, and can be obtained by typing in `ipconfig`

.....

in a DOS command window on the robot laptop. The command window can be opened by selecting **Start>Run** and typing in `cmd` on Windows XP or 2000, or `command` on Windows 98 or Me. It should look like this:



```
C:\WINNT\System32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : charterpipeline.net
    IP Address. . . . . : 192.168.1.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

C:\>
```

In the previous case shown, the IP address of the robot laptop is 192.168.1.101. The user will need to specify that port 9000 be forwarded to this IP address in the port forwarding configuration page of the home router. Once port forwarding of the teleoperation port is set up, you should be able to connect to the robot laptop and remotely operate the robot from work. Some home routers block all connections from the Internet by default as a safeguard against hacking. This feature will need to be disabled to allow the teleoperation client to connect to the robot. You should consult your hardware's documentation to see whether the hardware you are using has this safeguard and, if so, how to turn it off.



## ■ Appendix C

---

# 50 Ways to Love Your Robot

Ok, maybe not quite 50, but here are some interesting tutorials to help you learn more about your ER1. Some of these are very simplistic, but you can combine lots of simple behaviors like these to form some very complex behaviors.

### Easy

#### Take a video of the kids and send it to Grandma

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to shoot the video. It'll take a few minutes to round up everybody, so let's set the time to five minutes from now (let's say that's 2:00 P.M.), then set the frequency to **Once**.
2. Under **Then**, click on **Record**, select **Shoot Video** and select how many seconds of video you want ER1 to shoot. (Let's say 5 seconds).

3. Under **Record**, checkmark **Then Name It** and give your video a name. (Let's call it "OurKids".) Also, checkmark the **Then Email It** to button and type the recipient's email address. (Say, grandma@evolution.com.)
4. Click ER1's **Run it** button and gather everybody to wave for the video.
5. At 2:00 P.M. ER1 should take 5 seconds of video, name the video "OurKids," attach it to an email and send it to Grandma.
6. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Take a picture of the family and email it to Uncle Fred**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to shoot the picture. Everybody here? Ok, then let's set the time for one minute from now, and the frequency to **Once**.
2. Under **Then**, click on **Record**, select **Take Photo**.
3. Under **Record**, checkmark **Then Name It** and give your video a name. (Let's call it "TheCrew".) Also, checkmark the **Then Email It** to button and type the recipient's email address. (Say, unclesfred@evolution.com.)
4. Click ER1's **Run it** button and start smiling.
5. In one minute ER1 should take a picture, name it "TheCrew," attach it to an email and send it to Uncle Fred.
6. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Send an email if ER1 hears a loud sound while you are away**

1. Under **If** on a blank behavior tab, checkmark **Sound** and select a volume level from the **Level** checkboxes (volume increases from left to right).
2. Under **Then**, click on **Message Send** and type the email address you use at work in the **To** field.
3. On the Subject line type, "Loud Noise".
4. In the Body field, type, "Oh Nancy, I heard a loud noise at home."
5. Click on the **Behavior Settings** button in the upper right of the screen.
6. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
7. Click on the **Ok** button.
8. Click ER1's **Run it** button and walk out the door.
9. Whenever ER1 hears a loud noise, it will send you an email letting you know.
10. Save the behavior and then click on **Open** to get a new, blank behavior set.

*Important Note:* Experiment a bit to get a feeling for what level to set the volume sensitivity.

---

### Remind you to take the muffins out of the oven

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to be reminded to remove the muffins from the oven. If it is 6:30 A.M. now, maybe you want to be reminded in 15 minutes, so you would set the time to 6:45 A.M., and the frequency to **Once**.
2. Under **Then**, click on **Play a Sound** and type “Take the muffins out of the oven” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. ER1 should remind you at 6:45 A.M. to “Take the muffins out of the oven.”
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### Remind you to call home if you are running late at the office

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to be reminded to call home. So, if you want to be reminded daily at 5:30 P.M., enter 5:30 P.M. and select **Daily**.
2. Under **Then**, click on **Play a Sound** and type, “Remember to call home” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. ER1 will remind you every day at 5:30 P.M. to “Remember to call home.”
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### Remind you to pick up the dry cleaning

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to be reminded to pick up the dry cleaning. If you want to be reminded at 4:30 P.M., type 4:30 P.M.
2. Select **Once** from the third pulldown menu in the **Time** condition area.
3. Under **Then**, click on **Play a Sound** and type “Don’t forget to pick up the dry cleaning” in the **Phrase to Speak** field.
4. Click ER1’s **Run it** button.
5. At 4:30 P.M. ER1 will remind you to pick up your dry cleaning.
6. Save the behavior and then click on **Open** to get a new, blank behavior set.

### Remind you to feed the fish every day

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to feed the fish. So, if you want to be reminded daily at 9:00 A.M., enter 9:00 A.M. and select **Daily**.
  2. Under **Then**, click on **Play a Sound** and type, “Remember to feed the fish” in the **Phrase to Speak** field.
  3. Click ER1’s **Run it** button.
-

4. ER1 should remind you every day at 9:00 A.M. to “Remember to feed the fish.”
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Remind you to call your Mom every Wednesday night**

1. On Wednesday, select a blank behavior tab, then under **If** checkmark **Time** and set the time when you want to call your Mom. Let’s wait until the rates go down, so make it at 8:30 P.M., and select **Weekly**.
2. Under **Then**, click on **Play a Sound** and type, “Call Mom” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. ER1 should remind you every Wednesday at 8:30 P.M. to give Mom a ring.
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Give you five-minute, advance notice before your next class**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to be reminded to hurry off to class. If your class begins at 10 A.M. and you only want to be reminded today, set the time to 9:55 A.M. and select **Once**.
2. Under **Then**, click on **Play a Sound** and type “You better get moving. You’re going to be late for class” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. ER1 should remind you to go to class at 9:55 A.M.
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Play your favorite song at the same time every day**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to hear your favorite song. So, if you just adore hearing “Santa Lucia” at precisely 6:30 P.M. every day, enter 6:30 P.M. and select **Daily**.
2. Under **Then**, click on **Play a Sound**, browse to “Santa Lucia” on your hard drive (You *do* have Santa Lucia on your hard drive, don’t you?), click on it and press the **Ok** button.
3. Click ER1’s **Run it** button.
4. ER1 should play “Santa Lucia” every day at 6:30 P.M.
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Play your favorite music with a voice command**

1. Under **If** on a blank behavior tab, checkmark **Sound** and type “favorite song” in the **Listen for** field.
  2. Under **Then**, click on **Play a Sound** and browse for your favorite song on your hard drive (I know you like “Santa Lucia”). Click on it and press the **Ok** button.
-

- 
3. Click on the **Behavior Settings** button in the upper right of the screen.
  4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
  5. Click on the **Ok** button.
  6. Click ER1's **Run it** button.
  7. Each time you say "favorite song" clearly so ER1 can hear, ER1 should play your favorite song.
  8. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Have ER1 play your favorite 20 songs**

1. Under **If**, on the **Behavior 1** tab, checkmark **Sound** and type "jukebox" in the Listen for field.
2. Under **Then**, click on **Play a Sound** and browse your hard drive for the first song you want to hear. Click on it and press the **Ok** button.
3. Click on the **Behavior 2** tab, checkmark **Sequence**, and select **Behavior 1**. (This means Behavior 2 will run immediately after Behavior 1.)
4. Under **Then**, click on **Play a Sound** and browse your hard drive for the second song you want to hear. Click on it and press the **Ok** button.
5. Click on the **Behavior 3** tab, checkmark **Sequence**, select **Behavior 2** and select the third song you want to hear just as you did in the previous step.
6. Continue using Behavior tabs, one per song, until you are done. Use the **Sequence** option to link each behavior to the previous behavior.
7. When you have finished linking all 20 of your favorite songs together, click ER1's Run it button.
8. Say "jukebox" clearly so ER1 can hear.
9. ER1 should play all your favorite songs in order.
10. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Tell your dog to "Be quiet!" if it barks too loud**

1. Under **If** on a blank behavior tab, checkmark **Sound** and select a volume level from the Level checkboxes (volume increases from left to right).
  2. Under **Then**, click on **Play a Sound** and type, "Be quiet!" in the **Phrase to Speak** field.
  3. Click on the **Behavior Settings** button in the upper right of the screen.
  4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
  5. Click on the **Ok** button.
  6. Click ER1's **Run it** button.
-

7. Wait until your dog barks loudly. (No taunting the dog to get her to bark!)
8. ER1 should tell Spot to “Be quiet!”
9. Save the behavior and then click on **Open** to get a new, blank behavior set.

*Important Note:* You may need to experiment with the volume level. Wait until everybody is gone, lock the door, and practice barking at various volumes into the microphone.

### **Teach your parrot some new words**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time to when you’ll be out of the room (say, 2:00 P.M.), then select **Every minute**.
2. Under **Then**, click on **Play a Sound** and type, “Pretty bird” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. At 2:00 P.M. ER1 should start saying, “Pretty bird” to your parrot every minute.
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Stand outside your door and tell anyone who knocks you are in a meeting**

1. Under **If** on a blank behavior tab, checkmark **Sound** and select a volume level from the **Level** checkboxes (volume increases from left to right).
2. Under **Then**, click on **Play a Sound** and type, “Rhonda is in a meeting right now. Please try back at 2P.M.” in the **Phrase to Speak** field. (If by some strange circumstance your name is not “Rhonda,” substitute your name.)
3. Click on the **Behavior Settings** button in the upper right of the screen.
4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
5. Click on the **Ok** button.
6. Click ER1’s **Run it** button.
7. Whenever ER1 hears a loud sound, such as knocking, it will say, “Rhonda is in a meeting right now. Please try back next at 2 P.M.”
8. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Play “You’re Free!” at quitting time every Friday**

1. On Friday, select a blank behavior tab, then under **If** checkmark **Time** and enter your quitting time. Select **weekly**.
  2. Under **Then**, click on **Play a Sound** and type, “You’re free! Free!” in the **Phrase to Speak** field.
  3. Click ER1’s **Run it** button.
  4. ER1 should remind you every Friday at quitting time that you’re free to take off.
  5. Save the behavior and then click on **Open** to get a new, blank behavior set.
-

---

**Play “Shush!” when it hears a loud sound**

1. Under **If** on a blank behavior tab, checkmark **Sound** and select a volume level from the **Level** checkboxes (volume increases from left to right).
2. Under **Then**, click on **Play a Sound** and type, “Shush!” in the **Phrase to Speak** field.
3. Click on the **Behavior Settings** button in the upper right of the screen.
4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
5. Click on the **Ok** button.
6. Click ER1’s **Run it** button.
7. Whenever ER1 hears a loud sound, it will say, “Shush!”
8. Save the behavior and then click on **Open** to get a new, blank behavior set.

**Give homework reminders every day before the end of class**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want ER1 to remind the class to do its homework. So, if the class ends at 4:00 P.M. every day, enter 3:59 P.M. and select **Daily**.
2. Under **Then**, click on **Play a Sound** and type, “Class! Don’t forget to do your homework.” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. ER1 should remind the class every day at 3:59 P.M.: “Class! Don’t forget to do your homework.”
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

**Tell people when it’s time to cut the cake at your child’s birthday party**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want ER1 to tell people it’s time to cut the cake. If you want to cut the cake at 2:00 P.M., enter 2:00 P.M. and select **Once**.
2. Under **Then**, click on **Play a Sound** and type, “Hey everybody! It’s time to cut the cake.” in the **Phrase to Speak** field.
3. Click ER1’s **Run it** button.
4. At 2:00 P.M. ER1 should say, “Hey everybody! It’s time to cut the cake.”
5. Save the behavior and then click on **Open** to get a new, blank behavior set.

**Email video of the party every hour to a sick friend who couldn’t make it**

1. Under **If** on a blank behavior tab, checkmark **Time** and set the time when you want to shoot the video. It’ll take a few minutes to round up everybody, so let’s set the time to ten minutes from now (let’s say that’s 4:00 P.M.), then set the frequency to “Once.”

2. Under **Then**, click on **Record**, select **Shoot Video** and select how many seconds of video you want ER1 to shoot. (Let's say 15 seconds).
3. Under **Record**, checkmark **Then Name It** and give your video a name. (Let's call it "Party".) Also, checkmark the **Then Email It to** button and type the recipient's email address. (Say, sickbuddy@evolution.com.)
4. Click ER1's **Run it** button and party away.
5. At 4:00 P.M. ER1 should take 15 seconds of video, name the video "Party," attach it to an email and send it to your sick buddy.
6. Save the behavior and then click on **Open** to get a new, blank behavior set.

## Medium

### Teach the ER1 to dance and entertain your friends

1. Under **If**, on the Behavior 1 tab, checkmark **Sound** and type "dance" in the **Listen for** field.
2. Under **Then**, checkmark **Move**, select **Distance** or **Angle**, then in **This Amount** select enter 45 and select Degrees. Then in **This Direction**, select Right.
3. Click on the **Behavior Settings** button in the upper right of the screen.
4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
5. Click on the **Apply to all behaviors** button.
6. Click on the **Ok** button.
7. Click on the **Behavior 2** tab, checkmark **Sequence**, and select **Behavior 1**. (This means Behavior 2 will run immediately after Behavior 1.)
8. Under **Then**, checkmark **Move**, select **Distance or Angle**, then in **This Amount** select enter 1 and select Feet. Then in **This Direction**, select Backward.
9. Click on the **Behavior 3** tab, checkmark **Sequence**, select **Behavior 2** and under **Move**, describe another movement you want the robot to make, just as you did in the previous step.
10. Continue programming Behavior tabs, one per movement, until you are done. Remember to use the **Sequence** option to link each behavior to the previous behavior.
11. When you are finished, position ER1 so it has room to move, then click the **Run it** button.
12. Whenever someone says "dance" clearly so ER1 can hear, ER1 should dance away.
13. Save the behavior and then click on **Open** to get a new, blank behavior set.

---

### Read books out loud to your child (while under your supervision)

1. First, teach ER1 to recognize the pages of the book. To do so, checkmark **Sight** under **If**, then:
  - Hold up a page of the book to ER1's camera (having a plain background helps).
  - Click on the **Capture** button.
  - In the **Capture Object** dialog box, give the picture a unique name and indicate the approximate distance it was from the camera.
  - Click on the **Ok** button in the **Capture Object** dialog box.
  - Repeat these steps for each page of the book.
2. Next, from the **Object** option in **Sight**, under **If** on a blank behavior tab, select one of the images you just saved.
3. Under **Then**, checkmark **Play a Sound** and either select a sound file that contains the words on the page, or type the words on the page into the **Phrase to Speak** field.
4. Click on the **Behavior Settings** button in the upper right of the screen.
5. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
6. Click on the **Apply to all behaviors** button.
7. Click on the **Ok** button.
8. Do the previous step for each page of the book, selecting a new Behavior tab for each page.
9. When you are done, position ER1 so the child can show ER1's camera the pages.
10. Click on ER1's **Run it** button.
11. As the child shows the pages of the book to ER1's camera, ER1 will say the words on the page.
12. Save the behavior and then click on **Open** to get a new, blank behavior set.

### Teach your child the names of things (while under your supervision)

1. You might want to collect pictures of numbers, letters, dinosaurs, or whatever. Or, you may simply draw pictures on sheets of white paper.
  2. Next, teach ER1 to recognize the names of the objects you would like to teach your child. To do so, checkmark **Sight** under **If** on a blank behavior tab, then:
    - Hold up a drawing to ER1's camera
    - Click on the **Capture** button
    - In the **Capture Object** dialog box, give the picture a name and indicate the approximate distance it was from the camera
    - Click on the **Ok** button in the **Capture Object** dialog box.
-

- Repeat these steps for each picture.
3. Next, from the **Object** option in **Sight**, under **If**, select one of the images you just saved.
  4. Under **Then**, checkmark **Play a Sound** and either select a sound file that contains the name of the object, or type the object's name into the **Phrase to Speak** field.
  5. Click on the **Behavior Settings** button in the upper right of the screen.
  6. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
  7. Click on the **Ok** button.
  8. Do the previous steps for each picture, selecting a new Behavior tab for each object.
  9. When you are done, position ER1 so the child can show ER1's camera the pictures.
  10. Click on ER1's **Run it** button.
  11. As the child shows the pictures to ER1's camera, ER1 will name the object.
  12. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Play a lullaby (under your supervision) if the baby cries loudly**

1. Under **If** on a blank behavior tab, checkmark **Sound** and select a volume level from the **Level** checkboxes (volume increases from left to right).
2. Under **Then**, click on **Play a Sound**, browse to a lullaby which you saved to your hard drive, click on it and press the **Ok** button.
3. Click on the **Behavior Settings** button in the upper right of the screen.
4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
5. Click on the **Ok** button.
6. Click the **Run it** button.
7. Whenever the baby cries loudly, ER1 should play the lullaby.
8. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Help you recognize familiar objects if you have lost your glasses**

1. If you have difficulty telling your coffee cup from a ballpoint pen without your glasses, round up your coffee cup and ballpoint pen (with your glasses on) and teach ER1 to recognize them. To do so, checkmark **Sight** under **If** on a blank behavior tab, then:
  - Hold up a the ballpoint pen to ER1's camera (a plain background is helpful).
  - Click on the **Capture** button.
  - In the **Capture Object** dialog box, give the picture a name (ballpoint pen) and indicate the approximate distance it was from the camera.

- 
- Click on the **Ok** button in the **Capture Object** dialog box.
  - Do this again for your coffee cup.
2. Next, from the **Object** option in **Sight**, under **If**, select the ballpoint pen image.
  3. Under **Then**, checkmark **Play a Sound** and type “ballpoint pen” into the **Phrase to Speak** field.
  4. Select a new behavior and do the previous step for the coffee cup.
  5. Click on ER1’s **Run it** button.
  6. Next time you lose your glasses, just hold up the coffee cup or pen and ER will tell you which is which.
  7. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Tell the kids a bedtime story in your voice while you entertain nearby**

1. Record a bedtime story using any standard audio recording system on your computer, then save the bedtime story file on the robot’s laptop.
2. Then, under **If** on a blank behavior tab, checkmark **Sound** and type “story” in the **Listen for** field.
3. Under **Then**, click on **Play a Sound** and browse for the bedtime story file on ER1’s harddrive. Click on it and press the **Ok** button.
4. Click ER1’s **Run it** button.
5. Say “story” clearly so ER1 can hear.
6. ER1 will begin telling a bedtime story.
7. Save the behavior and then click on **Open** to get a new, blank behavior set.

## **Hard**

### **Play a disco song every time somebody says the word “party”**

1. Under **If** on a blank behavior tab, checkmark **Sound** and type “party” in the Listen for field.
  2. Under **Then**, click on **Play a Sound** and browse for a disco song on ER1’s hard disk. Note that the song must be in .wav format. Click on it and press the **Ok** button.
  3. Click on the **Behavior Settings** button in the upper right of the screen.
  4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
  5. Click on the **Ok** button.
  6. Click ER1’s **Run it** button.
  7. Whenever someone says “party” clearly so ER1 can hear, ER1 should begin playing the disco song.
-

8. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Take a photograph of anyone who says “Hello, robot” at a party**

1. Under **If** on a blank behavior tab, checkmark **Sound** and type “Hello, robot” in the Listen for field.
2. Under **Then**, click on **Record**, select **Take Photo**.
3. Click on the **Behavior Settings** button in the upper right of the screen.
4. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
5. Click on the **Video** tab and make sure the **Add sequence number to each video** is set.
6. Click on the **Ok** button.
7. Click ER1’s **Run it** button.
8. Whenever someone says “Hello, robot” clearly so ER1 can hear, ER1 should take a picture, which it will save in the ER1 folder at c:\Program Files\ER1.
9. Save the behavior and then click on **Open** to get a new, blank behavior set.

### **Remind your guest holding a beer not to drink and drive**

1. First, teach ER1 to recognize various brands of beer you will be serving. To do so, checkmark **Sight** under **If**, then:
  - Hold up a beer to ER1’s camera (a plain background is helpful).
  - Click on the **Capture** button.
  - In the **Capture Object** dialog box, give the picture a name and indicate the approximate distance it was from the camera.
  - Click on the **Ok** button in the **Capture Object** dialog box.
  - Repeat these steps for each beer.
2. Next, from the **Object** option in **Sight**, under **If** on a blank behavior tab, select one of the beer images you just saved.
3. Under **Then**, checkmark **Play a Sound** and type, “Please do not drink and drive” into the **Phrase to Speak** field.
4. Click on the **Behavior Settings** button in the upper right of the screen.
5. Click on the **Run** tab and checkmark the box labeled **Let behavior run more than once**.
6. Click on the **Ok** button.
7. Repeat steps 2-6 for each beer picture, selecting a new Behavior tab for each object.
8. When you are done, position ER1 so it can watch your guests.
9. Click on ER1’s Run it button.

## ■ Index

---

### A

- About tab
  - using [8-15](#)
- Adding images [6-5](#)
- API
  - accessing [10-2](#)
  - clear command [10-10](#)
  - command structure [10-5](#)
  - digital input/output commands [10-15](#)
    - input analog [10-18](#)
    - input digital [10-16](#)
    - output digital [10-16](#)
  - events command [10-11](#)
  - examples [10-5](#)
  - move command [10-6](#)
    - move distance [10-6](#)
    - move drive toward color [10-8](#)
    - move drive toward object [10-8](#)
    - move rotate toward color [10-7](#)
    - move rotate toward object [10-7](#)
    - move X,Y distance [10-9](#)
  - objects command [10-8](#)
  - play command [10-9](#)
    - play file [10-9](#)
    - play phrase [10-9](#)
  - position command [10-9](#)
  - sense command [10-10](#)
  - set command [10-12](#)
    - set angular velocity [10-12](#)

- set collision detection [10-13](#)
- set color tolerance [10-14](#)
- set confidence threshold [10-14](#)
- set linear velocity [10-12](#)
- set obstacle avoidance [10-13](#)
- set power moving [10-13](#)
- set power stopped [10-13](#)
- set voice [10-12](#)
- stop command [10-10](#)
- API control [8-8](#)

## **B**

- Battery [4-3](#)
- Behavior
  - execution [3-5](#)
  - files [3-2](#)
  - tabs [3-2](#)
- Best match [4-2](#), [8-2](#)

## **C**

- Camera tab
  - using [8-3](#)
- Capture button
  - using [6-2](#)
- Capturing images
  - how to [6-3](#)
- Collision detection [9-7](#)
- Color detection [1-3](#)
- Color recognition [8-2](#), [9-2](#)
- Color selection tool
  - using [6-2](#)
- Commands
  - move [7-2](#)
  - navigate [4-2](#)
- Conditions
  - message receive [6-8](#)
  - message send [7-4](#)
  - sequence [6-9](#)
  - sight [6-2](#)
  - sound [6-6](#)
  - time [6-6](#)
- Customer support [1-4](#)

## **D**

- Deleting images [6-5](#)
- Device [8-3](#)
- Dropper tool [6-2](#)

---

**E**

Email [1-3](#)  
    optional [2-4](#)  
Email tab  
    using [8-9](#)  
ER1 community [1-4](#)  
Example application [2-5](#)

**F**

Files  
    saving [5-2](#)

**H**

Help button [5-4](#)

**I**

Images  
    adding [6-5](#)  
    deleting [6-5](#)  
    how to capture [6-3](#)  
    modifying files [6-3](#)

**L**

Laptop requirements [2-3](#)  
Log tab  
    using [8-14](#)  
Logging [3-7](#)

**M**

Message receive condition [3-4](#), [6-8](#)  
Message send [3-5](#)  
Message send condition [7-4](#)  
Modify  
    image file information [6-3](#)  
Motion tab  
    using [8-11](#), [9-5](#)  
Motor power [8-11](#), [9-7](#)  
Move [3-5](#), [7-2](#)  
Move command [7-2](#)  
Mute sound checkbox [4-2](#)

**N**

Navigate command [4-2](#)  
New button [5-1](#)

New files  
opening [5-1](#)

## O

Object recognition [1-3](#), [2-3](#), [4-1](#), [6-2](#), [6-4](#), [8-2](#), [8-3](#), [9-1](#)  
stopping [4-3](#)  
Obstacle avoidance [1-4](#), [2-3](#), [4-1](#), [4-2](#), [8-3](#), [8-12](#)  
camera [8-12](#)  
calibration [8-13](#)  
view [8-13](#)  
Obstacle camera [4-2](#)  
Open [5-2](#)  
Opening new files [5-1](#)  
Optional  
email [2-4](#)  
remote control [2-4](#)  
second camera [2-3](#)

## P

Photo [1-3](#)  
Photo file formats [9-3](#)  
Photo tab  
using [9-3](#)  
Play a sound [3-5](#), [7-3](#)  
Programs  
running [7-3](#)

## R

Recognized objects [4-2](#)  
Record [3-5](#)  
Recording [7-4](#)  
Registration [1-4](#)  
Remote control [1-4](#)  
optional [2-4](#)  
Remote control tab  
using [8-6](#)  
Reset button [7-1](#)  
Resolution [8-4](#)  
RGB values  
determining [10-7](#)  
Robot specifications [2-3](#)  
RobotMail [2-4](#), [8-9](#)  
opening an account [6-8](#)  
Run a program [3-5](#), [7-3](#)  
Run tab  
using [9-9](#)

---

**S**

- Save button 5-2
- Saving files 5-2
- Search pattern 11-1
- Second camera
  - optional 2-3
- Sequence condition 3-4, 6-9
- Settings
  - about 8-15
  - camera 8-3
  - device 8-3
  - email 8-9
  - log tab 8-14
  - motion 8-11
  - remote control 8-6
  - sight 8-2
  - sound 8-5
- Settings button 5-3
- Sight condition 3-3, 6-2
- Sight tab
  - using 8-2, 9-1
- Sound
  - condition 6-6
  - muting 4-2
  - playing 7-3
  - tab
    - using 8-5, 9-3
    - turning down 4-2
- Sound condition 3-3
- Speech recognition 6-6
- Speed
  - linear velocity 9-6
  - rotation angular velocity 9-6

**T**

- Text-to-Speech 1-3, 9-3
- Time condition 3-4, 6-6
- Try It button 7-1
- Turning sound down 4-2
- Tutorial 2-5

## U

### Using

- about tab [8-15](#)
- camera tab [8-3](#)
- email tab [8-9](#)
- motion tab [8-11](#), [9-5](#)
- obstacle camera [4-2](#)
- photo tab [9-3](#)
- remote control tab [8-6](#)
- run tab [9-9](#)
- sight tab [8-2](#), [9-1](#)
- sound tab [8-5](#), [9-3](#)
- video [4-1](#)
- video tab [9-4](#)

## V

### Video [1-3](#)

- compression [9-5](#)
- display [4-1](#)
- tab
  - using [9-4](#)

### Voice recognition [1-3](#)