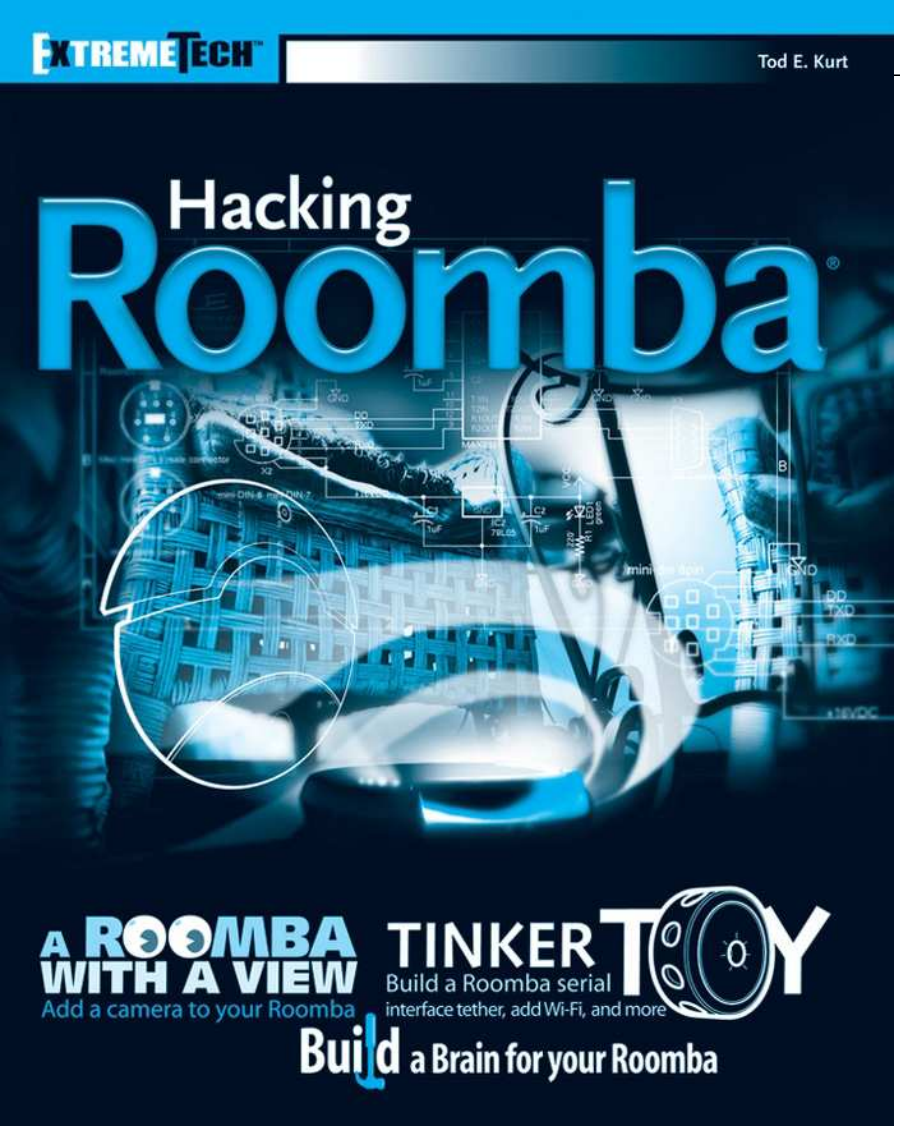
EXTREMETECH
Tod E. Kurt
Hacking
Roomba®
A ROOMBA WITH A VIEW
Add a camera to your Roomba
TINKER TOY
Build a Roomba serial interface tether, add Wi-Fi, and more
Build a Brain for your Roomba

Hacking Roomba®

Tod E. Kurt

Wiley Publishing, Inc.

Hacking Roomba®

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN-13: 978-0-470-07271-4
ISBN-10: 0-470-07271-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Kurt, Tod E., 1969–
Hacking Roomba / Tod E. Kurt.
p. cm.
ISBN-13: 978-0-470-07271-4 (paper/website)
ISBN-10: 0-470-07271-7 (paper/website)
1. Robots—Programming. 2. Roomba vacuum cleaner. 3. Autonomous robots. 4. Mobile robots. I. Title.
TJ211.45.K87 2007
629.8´9251—dc22
2006031031

About the Author

Tod E. Kurt has engineered the hardware and software for robotic camera systems that went to Mars. He was a founding developer and systems architect of Overture Systems, originally GoTo.com, later sold to Yahoo. Now as co-creator of ThingM.com, he's designing sketchable hardware and networked smart objects. He has degrees in electrical engineering from Caltech and physics from Occidental College. He started robotics hacking at the age of twelve when he took apart his BigTrak, RC car, and chemistry set to make an upright programmable robot.

Credits

Executive Editor
Chris Webb

Development Editor
Kelly Talbot

Copy Editor
Michael Koch

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Compositor
Happenstance Type-O-Rama

Proofreader
Nancy Riddiough

Indexer
Ted Laux

Anniversary Logo Design
Richard Pacifico

Cover Design
Anthony Bunyan

Contents at a Glance

Contents

Acknowledgments

Thanks to my friends and family for being there for me and understanding when I would disappear for days at a time to write this book and commune with the Roombas. Thanks again to them for providing encouragement and many great ideas for fun hacks. Particularly I'm grateful to Ben C, Ben F, Chris L, Liz C, John Joseph, Mike K, Phil T, Paul R, and Preston P for all their great ideas. I owe you all.

I'd like to thank everyone at the iRobot Corporation for providing the ROI specification to the world at large and supplying the assistance I needed for this book. Your company has done an amazing thing by being so open to its users. You're a leader in creating the new conversation between a company and its users.

This book wouldn't be possible without my editors: Kelly Talbot, who kept me going and added to this book in substantial ways, and Chris Webb, who believed in me and the book. Thanks to everyone at Wiley Publishing for doing the hard work of turning bits to atoms.

Thanks to the extended Roomba hacking community, especially those who provided information about their hacks for this book. Also thanks to everyone at Makezine, Roomba Review, and Hackaday for providing a medium for news and discussion about Roomba hacking and hacking in general.

And thanks to you, the reader. Hacking anything, including Roombas, can only get better with the addition of new people, fresh viewpoints, and sharing of ideas. Welcome to the Roomba hacking community and have fun with this book and your Roomba!

Introduction

The iRobot Roomba is perhaps the best example of mobile robotics entering the home. It performs a useful task and is relatively inexpensive, and while it doesn't look like Rosie the Robot on *The Jetsons*, it does have a charm of its own.

The purpose of this book is to introduce robot hacking to people who are interested in programming and modifying their own robot but who don't want to destroy a functioning Roomba. This "reversible hacking" is device modification that can be undone to return the device to its original state. In the case of the Roomba, the ROI connector is the gateway to reversible Roomba hacking. All manner of devices can be plugged into the ROI: a desktop computer, a microcontroller "brain" to replace the original, new sensors and actuators, or maybe just some snazzy running lights. Any of these modifications can be quickly removed, leaving the Roomba in the original state. All hacks presented in this book are reversible hacks. None of the projects in this book will damage your Roomba or even void its warranty. (However, a few potentially warranty-voiding options are included and explained as such, if you are intrepid enough to explore them.)

This book shows how to make the Roomba do more than be a cute vacuum cleaner. With this book, the reader can give the Roomba new or more complex behaviors, connect it to the Internet, control it from a computer, and literally make it dance and sing. This book is a way to learn the basics of robotics and have fun programming a robot without needing to build one. All of the projects can be done without breaking open the Roomba or even voiding its warranty. And like all good hacking books, this one shows how to install Linux on a Roomba. This book is a practical demonstration of several ways to create networked objects, normal devices with intelligence and Internet connectivity.

The History of Hacking the Roomba

Most people who purchased the first Roombas were early adopters of technology and liked the idea of a personal robot to do their bidding. To watch a Roomba roaming around their living room, cleaning up after a mess, was to experience in a small way life in the future.

Unfortunately, the Roomba wasn't very "hackable" by the normal gadgeteer. If you wanted to easily reprogram your Roomba to alter its behavior or make it do tricks, you were out of luck. At the least you had to take the Roomba apart, definitely voiding its warranty. Once inside perhaps you could reverse engineer the small computer (also known as the microcontroller) used as its brain, maybe replace it completely, and hook into the motors and sensors, effectively destroying it for its original purpose. Communities devoted to hacking the Roomba in this low-level way grew and flourished. The hacking section of Roomba Review (`http://roombareview.com/hack/`) is one of the most famous, and the accompanying forum is still the best place to go to discuss Roomba hacking. Other sites like Hackaday (`http://hackaday.com/`) and Makezine (`http://makezine.com/`) routinely featured projects that used stripped-down or heavily modified Roombas. But hacking the Roomba was a difficult and expensive task, only suitable for the most experienced engineers. Recently this has changed.

In December 2005, iRobot Corporation, the maker of the Roomba, recognized the growing hacking community and released documentation describing the Serial Command Interface (SCI) present on third-generation Roombas. In mid-2006 iRobot renamed the SCI to be the Roomba Open Interface (ROI), a name that better fits its role. The ROI allows you to take full control of the Roomba and its behavior. This is no simple remote control interface, but instead a protocol that allows complete sensor readout and full actuator control.

Since the release of the SCI/ROI specification, there has been an explosion of new Roomba hacks. The Roomba hacking community has blossomed to include not just professional hardware engineers, but people from all experience levels, from normal people looking to play with their Roomba in a new way to academics experimenting with low-cost robotics. The ROI turns the Roomba into a true robotics platform. And because these are all reversible hacks, it's easy to try someone else's hacks. No longer do you have to break a Roomba to try something out. To see some of the hacks people are working on and join in discussions about them with others, see the Roomba Review hacking site mentioned above, the accompanying forum at `http://www.roombareview.com/chat/`, and the Roomba hacking wiki at `http://roomba.pbwiki.com/`.

Whom This Book Is For

This book is for those who want to experience the fun of programming a robot without all the problems normally associated with building one.

Most of this book is designed for beginning hackers, those who know a bit of programming and a little bit of electronics. Familiarity but not expertise is assumed with soldering and schematics. No mechanical expertise is required, but if you have it, you can do even more impressive things than what is outlined in this book.

If your soldering and schematics skills are a little rusty, two useful appendixes are supplied to help get you back up to speed.

Most of the code examples are presented in Java. Java was chosen for its ubiquity and cross-platform capability and is used for all the PC-connected projects. Thus, knowing how to compile and run Java programs is required. The later, more advanced projects are programmed on microcontrollers in either PIC BASIC or AVR GCC.

What This Book Covers

The projects in this book are based around the Roomba and the Roomba Open Interface (ROI). The Roomba's capabilities as presented via the ROI are described and tested. The few capabilities that aren't accessible via the ROI are mentioned briefly. The Roomba's mechanical and electrical internals are also discussed, but since this book is about hacking the Roomba without taking it apart, they're mentioned only briefly.

The ROI protocol is covered in detail, with examples given as each part of the protocol is examined. Practical and fun examples of each ROI command are given. As a way of abstracting the rather low-level commands of the ROI, a code library of software routines is slowly built up to become the RoombaComm API with applications built using it.

Throughout this book, the ideas and practices of Network Objects are developed. As computing and networking become so cheap as to be effectively free, all objects will become network objects. The Roomba robot is already a computing object, a normal everyday device (a vacuum cleaner) that has been imbued with intelligence in the form of a small computer. This book describes several methods of extending the Roomba to become a networked object. These methods are similar to those used for current and future network objects that will exist throughout the home.

How This Book Is Structured

This book is designed mostly for the novice electronics hacker, but it contains several advanced projects toward the end. The book is divided into three parts. Each part is mostly self-contained, depending upon which shortcuts are taken, but knowing the concepts presented in earlier chapters helps in the later ones.

Part I: Interfacing

This part describes the Roomba, its history, and its model variations, to dispel the confusion regarding which Roombas are hackable via the ROI protocol. The ROI protocol is discussed in depth, showing exactly what bytes are sent and received to command the Roomba. To allow a PC to speak to the ROI, two simple hardware interface projects are shown — one wired, one wireless. With those created, a software library is given that provides an easy-to-use abstraction layer on the PC.

Part II: Fun Things to Do

Using the hardware and software infrastructure from the previous part, this part focuses on interesting, or just plain silly, things to do with a computer-controlled Roomba. Make it dance and sing, draw huge artwork on the ground, and create a complete dashboard/remote control PC application called RoombaView.

Part III: More Complex Interfacing

With experience from using a PC to control a Roomba, the focus now becomes making the Roomba a true Internet device and fully autonomous. The first few hacks are Internet versions of the initial interfaces. From there a fully reprogrammable replacement brain is added to the Roomba using microcontrollers like the PIC Basic Stamp or Arduino AVR. This part ends with adding a larger microcontroller board that can run Linux and use a webcam, microphone, or any other sensor imaginable.

Appendixes

If your electronics hacking skills are a little rusty, Appendix A covers the basics on how to solder circuits and work safely with electronics. Appendix B explains how to interpret common schematic circuit diagrams like the ones in many of the projects. Appendix C is a reprint of the ROI specification from iRobot. The ROI is what enables all the hacks in this book, and it is the authority on how the Roomba can be hacked.

What You Need to Use This Book

Of course you will need a Roomba, one with ROI capability. Chapter 1 describes which Roombas have ROI. To run the code, you will need a PC with USB and Java JDK 1.5 installed. Windows, Mac OS X, and Linux computers can all fit this requirement. For Windows and Linux, Java is not installed by default and can be obtained as a free download from `http://java.com/`. To write and compile programs, you'll need a text editor and knowledge of the command line or experience with a Java IDE. If you're unfamiliar with how to create and compile Java programs, there are many tutorials on the Net. This book assumes basic familiarity with programming and Java. Even so, all code presented in the book is available in ready-to-run form from `www.wiley.com/go/extremetech` and `http://roombahacking.com/`.

For projects that have circuits, a soldering iron and other tools are required, as well as basic knowledge of their use. Expect to have on hand a multimeter, wire cutters/strippers, test leads, and so on. Each chapter describes exactly which tools are required. There is an appendix that contains a basic overview on soldering, tool use, and electronics assembly. It also covers how to be safe around these somewhat dangerous tools. There are many good references on the Internet going into more depth on these topics than this book has room for, so some of my favorite electronics "how-to" sites will be listed in that appendix as well.

Many of the circuits presented in this book can be purchased as kits or fully assembled from various suppliers. Notably, `RoombaDevTools.com` provides fully assembled Roomba interfaces that are functionally identical to the interfaces provided in Chapters 3 and 4.

To build the circuits, various electronic components are required. Only a few components are more than a dollar or two. There are several suppliers for these components: Digikey (`http://digikey.com/`), Mouser (`http://mouser.com/`), Jameco (`http://jameco.com/`), and Radio Shack (`http://radioshack.com/`) are four of the more popular. Sparkfun (`http://sparkfun.com/`) is a great source for the specialized components used. Throughout this book, Jameco part numbers will be used when possible for the commonly available parts. Jameco is a great resource that is very popular. They stock almost anything an electronics hobbyist needs, at decent prices; they ship fast; and, most important, they have an easy-to-use web interface. Jameco also sells all the tools needed for the projects in this book.

The later projects assume some experience in microcontroller programming. While this book hasn't the space to go into how to do this, it's not that different from programming on a PC, and controlling the Roomba would be a great excuse to learn about it.

If you'd like to learn about microcontroller programming, a good starting resource is Parallax's *What Is a Microcontroller?* book available as a free PDF download from their website (`http://parallax.com/`) in their Documents/Tutorials section. It's focused on the Basic Stamp microcontroller, but the techniques and concepts are universal. A good repository for other microcontroller info is NYU's ITP program tutorial website (`http://itp.nyu.edu/physcomp/Tutorials/Tutorials`) and Tom Igoe's Physical Computing Site (`http://tigoe.net/pcomp/`).

Conventions Used in This Book

In this book, you'll find several notification icons — Note, Caution, Tip, and Cross-Reference — that point out important information. Here's what the three types of icons look like:

Notes provide you with additional information or resources.

A caution indicates that you should use extreme care to avoid a potential disaster.

A tip is advice that can save you time and energy.

A cross-reference directs you to more information elsewhere in the book.

Code lines are often longer than what will fit across a page. The symbol ↵ indicates that the following code line is actually a continuation of the current line. For example,

```
root@OpenWrt:~# wget http://roombahacking.com/software/openwrt/↵
roombacmd_1.0-1_mipsel.ipk
```

is really one line of code when you type it into your editor.

Code, functions, URLs, and so forth within the text of this book appear in a `monospaced font`, while content you will type appears either **bold** or `monospaced`.

What's on the Companion Website

On the companion website at `http://roombahacking.com/` and `www.wiley.com/go/extremetech`, you'll find source code and schematics for all the projects in this book. All the code and schematics are open source. At `http://roombahacking.com/`, they are improved and added to by the Roomba hacking community. You'll also find additional projects that expand upon the ideas presented in this book. The site also contains mirrors of important documents like the ROI specification and data sheets for useful electrical components. Galleries are available for Roomba hackers to upload and share information about their favorite Roomba hacks. Finally, the site contains links to other Roomba sites, tutorials about electronics assembly and microcontroller programming, and other useful hacking websites.

Interfacing

part

in this part

Getting Started with Roomba

chapter 1

iRobot has produced a dizzying variety of Roomba vacuuming robots since the original Roomba model was introduced in 2002. They now have even the Scooba, a robot that washes floors.

Compared to other robotic vacuum cleaners, the typical Roomba robotic vacuum cleaner is very inexpensive at under $300 for even the most expensive Roombas and $150 for the least expensive. The cheapest new Roombas can be found for around $100 on the Internet. For a vacuum cleaner that's a pretty good price. For a robot that's also a vacuum cleaner, that's an amazing price. And for a robotic vacuum cleaner that's hackable by design?

in this chapter

- ☑ Uncover how Roomba evolved
- ☑ Explore which Roomba models are hackable
- ☑ Examine the components of Roomba
- ☑ Learn about the OSMO//hacker module

Quick Start

If you're already familiar with Roomba, know it's compatible with the Roomba Open Interface (ROI), and you'd like to start hacking immediately, skip to Chapter 3 to begin building some hacks. If you're uncertain which Roomba you have, if it is hackable through the ROI, and want to learn the details on the ROI protocol that enables all these hacks, keep reading.

All projects in this book will utilize the Roomba Open Interface (ROI). It was previously known as the Roomba Serial Command Interface (SCI) and you'll find many references to the SCI on the Internet. It's exactly the same as the ROI; only the name has changed.

What Is Roomba?

Roomba is an autonomous robotic vacuum cleaner created by iRobot Corporation. To operate, Roomba requires no computer and no technical knowledge from its owner. It only needs a power outlet and occasional cleaning, like any vacuum cleaner.

Originally released as just Roomba in 2002, the Roomba design and functionality have evolved over the years. Currently there are five varieties of Roomba available with names like Roomba Discovery and Roomba Red. According to iRobot, with over 2 million units sold, not only is Roomba one of the most successful domestic robots to date, it is also one of the very few robots to have sold over a million units. This accomplishment is the result of a long evolutionary process of robotics design at the iRobot Corporation.

iRobot Corporation

The creators of Roomba have been making robots for over 15 years. iRobot was founded by Rodney Brooks, Colin Angle, and Helen Greiner. These three MIT alumni have been instrumental in guiding robotics research for many years, not only through their research but also through the practical application of their ideas through iRobot.

Subsumption Architecture

Rodney Brooks coined the term *subsumption architecture* in 1986 in his classic paper "Elephants Don't Play Chess." This paper began a shift in artificial intelligence research. At the time it was believed that to create a thinking machine, one needed to start with a symbolic representation of its world from some set of base principles. (For example, a robot butler having a built-in map of a house would be a kind of basic symbol.) This top-down view of cognition is opposite to how nature works: When we enter a new house, we must explore and build up our own unique perception of how to get from place to place. Brooks codified a bottom-up, behavior-based approach to robotics.

In subsumption architecture, increasingly complex behaviors arise from the combination of simple behaviors. The most basic simple behaviors are on the level of reflexes: "avoid an object," "go toward food if hungry," and "move randomly." A slightly less simple behavior that sits on top of the simplest may be "go across the room." The more complex behaviors subsume the less complex ones to accomplish their goal.

Genghis and PackBot

In 1990 while at MIT, Rodney Brooks and iRobot created the Genghis Robot, an insect-like robot with six legs and compound eyes. It was a research platform that bucked the trend in artificial intelligence at the time by using Brook's subsumption architecture. Genghis was designed from an evolutionary perspective instead of the normal high-level cognition perspective of traditional AI. It looked and acted like an insect. This behavior-based robotics architecture would inform the design of all future iRobot robots.

From Genghis, iRobot developed a few other research robots but quickly moved into developing robots for real-world use. iRobot has had great success with their PackBot, a series of ruggedized telepresence (able to withstand harsh outdoor environments and remotely controlled) and autonomous robots for the military and law enforcement. Instead of sending soldiers or a SWAT team into a dangerous area, the PackBot can be pulled from a backpack and thrown into the area. With its onboard video and audio sensors, the area can be inspected without risking a life. The PackBot can withstand 400+ *g*s of force. This makes it much

tougher than a human. One *g* is the force you feel every day from gravity. Three *g*s are what most roller coasters make you feel, and at five *g*s you black out. Although the Roomba isn't nearly so rugged, it definitely seems to have inherited some of its cousin's toughness.

Enter Roomba

The Roomba robotic vacuum cleaner is a physical embodiment of Brooks' subsumption architecture. Roomba has no room map or route plan. It has no overall view of what it is doing. Instead it functions much more like an insect: going toward things it likes (dirt, power) and away from things it dislikes (walls, stairs), moving in predefined movement routines while occasionally and randomly jumping out of a predefined routine.

This random walk feature of the Roomba algorithm is perhaps what confuses people the most at first. It will seem to be going along doing the right thing when it suddenly takes off in a different direction to do something else. But for every time it moves from the right place to the wrong place, it has moved from the wrong place to the right place. On average (and if left for a long enough time), Roomba covers the entire area. In terms of time efficiency, Roomba is not the most effective, as it takes several times longer for it to fully cover a region than it would for a person with a normal vacuum cleaner. But whose time is more valuable? Roomba can work while the person does something else.

Which Roomba Cleaners Are Hackable?

There is some confusion as to which Roomba cleaners are easily hackable through the ROI. This is complicated by the fact that iRobot doesn't make obvious the model numbers and firmware versions of the different Roomba cleaners.

All new Roomba cleaners currently have the ROI protocol built-in and ready to use. These are third-generation Roomba cleaners. The two most common Roomba cleaners, Roomba Discovery and Roomba Red, will be used in the examples in this book.

Following is a fairly comprehensive list of Roomba cleaners available in North America. International versions are functionally identical and named the same, with only small modifications to function on different mains voltages.

First Generation

The first generation of Roomba cleaners was astounding in the amount of capability they packed into a small, inexpensive package. This generation did not have any ROI capability. There was only one type of Roomba in the first generation:

- **Roomba:** The original Roomba model, shown in Figure 1-1, was released in 2002 and improved in 2003. It could clean small, medium, or large rooms when instructed