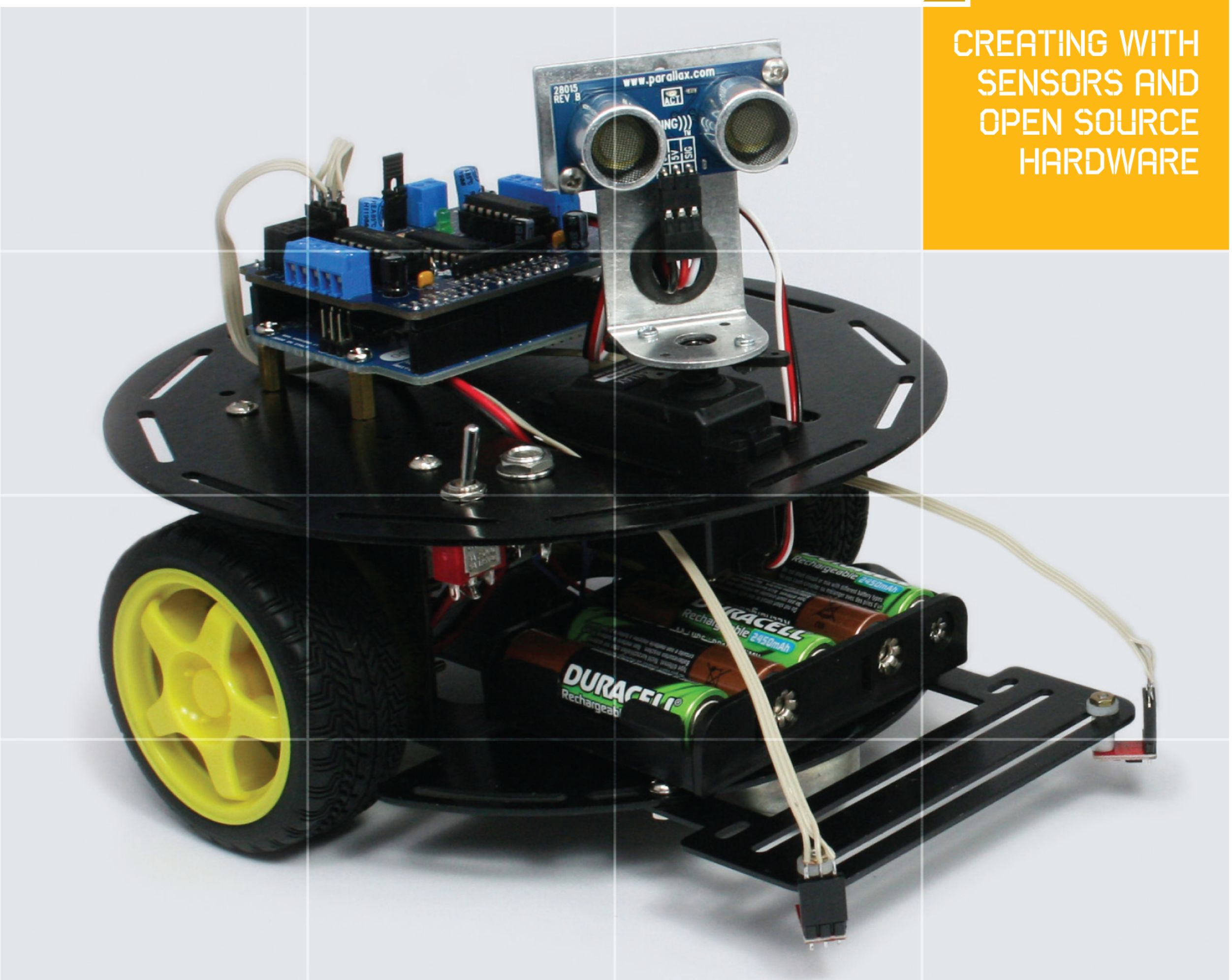
Michael Margolis

Make an Arduino-Controlled Robot

Learn by Discovery

Autonomous and Remote-Controlled Bots on Wheels

CREATING WITH SENSORS AND OPEN SOURCE HARDWARE



O'REILLY®

Make:
makezine.com

Make an Arduino-Controlled Robot

Michael Margolis

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Make an Arduino-Controlled Robot

by Michael Margolis

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Brian Jepson

Production Editor: Rachel Steely
Interior Designers: Nellie McKesson and Edie Freedman

October 2012: First Edition

Revision History for the First Edition:

2012-09-12 First release

2012-10-03 Second release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449344375* for release details.

ISBN: 978-1-449-34437-5

[LSI]

Table of Contents

Preface vii

1. Introduction to Robot Building 1
Why Build a Robot? 4
How Robots Move 5
Tools 6

2. Building the Electronics 9
Hardware Required 9
Construction Techniques 10
Soldering 10
Building the Motor Controller 10
Soldering the Reflectance Sensors 17
Making a Line Sensor Mount 17
Next Steps 20

3. Building the Two-Wheeled Mobile Platform . . 21
Hardware Required 22
Mechanical Assembly 23
Lay Out the Chassis Parts 23
Motor Assembly 24
Assemble the Chassis Components 26
Attaching the Control Electronics 37
Mounting the IR sensors 38
Mounting the IR Sensors for Edge Detection 39
Mounting the IR Sensors for Line Following 41

Next Steps . . . 43

4. Building the Four-Wheeled Mobile Platform . . . 45
Hardware Required . . . 46
Mechanical Assembly . . . 47
Lay Out the Chassis Parts . . . 47
Motor Assembly . . . 49
Assemble the Chassis Components . . . 51
Solder the Power and Motor Connections . . . 54
Connecting the Battery Pack and Power Switch . . . 55
Building the Optional Trickle Charger . . . 56
Assemble the Chassis . . . 57
Mounting Arduino and Connecting Wires to the Shield . . . 58
Mounting the IR sensors . . . 65
Mounting the IR Sensors for Edge Detection . . . 65
Mounting the IR Sensors for Line Following . . . 67
Next Steps . . . 68

5. Tutorial: Getting Started with Arduino . . . 71
Hardware Required . . . 72
Arduino Software . . . 72
Arduino Hardware . . . 72
Installing the Integrated Development Environment (IDE) . . . 74
Installing Arduino on Windows . . . 74
Installing Arduino on OS X . . . 75
Installing Arduino on Linux . . . 76
Driver Installation . . . 76
Connecting the Arduino Board . . . 78
Using the IDE . . . 78
Uploading and Running the Blink Sketch . . . 81
Using Tabs . . . 82
Installing Third-Party Libraries . . . 83

6. Testing the Robot's Basic Functions . . . 85
Hardware Required . . . 85
Software Prerequisites . . . 86
Sketches Used in This Chapter . . . 87
Load and Run helloRobot.ino . . . 88
About the Sketch . . . 95
Troubleshooting . . . 98
Making the Sketch Easy to Enhance . . . 99

7. Controlling Speed and Direction . . . 103
Hardware Required . . . 103

Sketches Used in This Chapter ... 103
Types of Motors ... 104
Motor Controllers ... 106
Controlling Motor Speed ... 109
How Motor Speed Is Controlled ... 109
Code for Motor Control ... 110
Calibrating Rotation and Tracking ... 116
Software Architecture for Robot Mobility ... 119
Functions to Encapsulate Robot Movements ... 123
Core Movement Code ... 124
Additional Core Functions ... 126
Functions to Rotate the Robot ... 127
Higher-Level Movement Functions ... 130

8. Tutorial: Introduction to Sensors ... 133
Hardware Discussed ... 133
Software ... 134
Infrared Reflectance Sensors ... 134
Sonar Distance Sensors ... 137
Maxbotix EZ1 Sonar Distance Sensor ... 139
Sharp IR Distance Sensor ... 141
Proximity Sensor ... 142
Sound Sensor ... 143
Arduino Cookbook ... 146

9. Modifying the Robot to React to Edges and Lines ... 147
Hardware Required ... 147
Sketches Used in This Chapter ... 148
The Look Code ... 149
Edge Detection ... 150
Line Following ... 154
Seeing Sketch Data ... 160

10. Autonomous Movement ... 163
Hardware Required ... 163
Sketches Used in This Chapter ... 164
Mounting a Ping Distance Sensor ... 165
Making a Mount for the Ping Sensor ... 166
Mounting the Ping Sensor in a Fixed Position ... 168
Mounting the Ping Sensor on a Servo ... 168
Letting the Robot Wander ... 170

Adding Scanning 178

11. **Remote Control** 185
Hardware Required 185
Sketches Used in This Chapter 186
Design of the Remote Control Code 186
Controlling the Robot with a TV Type IR Remote 190
Installing the IR Decoder Chip 190
The IR Remote Software 192

Appendix A. **Enhancing Your Robot** 201

Appendix B. **Using Other Hardware with Your Robot** 205

Appendix C. **Debugging Your Robot** 211

Appendix D. **Power Sources** 221

Appendix E. **Programming Constructs** 231

Appendix F. **Arduino Pin and Timer Usage** 235

Preface

Building a robot and enabling it to sense its environment is a wonderful way to take your Arduino knowledge to the next level. In writing this book, I have brought together my love for invention and my experience with electronics, robotics and microcontrollers. I hope you have as much pleasure building and enhancing your robot as I did developing the techniques contained in this book.

Arduino is a family of microcontrollers (tiny computers) and a software creation environment that makes it easy for you to create programs (called *sketches*) that can interact with the physical world. Arduino enables your robot to sense the environment and respond in a rich variety of ways. This book helps you to build a robot that is capable of performing a wide variety of tasks. It explains how to assemble two of the most popular mobile platforms, a robot with two wheels and a caster (for stability, since it's hard to balance on two wheels), and a robot with four wheels and motors. If you want your robot up and running quickly, choosing one of the kits detailed in this book should speed you through the build process and get you going with the robot projects. But whether you prefer to design and build a platform of your own construction or build from a kit, you will find the projects that comprise the core of this book a practical and fun introduction to Arduino robots.

Who This Book Is For

This book is for people who want to explore robotics concepts like: movement, obstacle detection, handling sensors, remote control, and all kinds of real world physical computing challenges. It is for people who want to understand how these concepts can be used to build, expand and customize your robot. See "What Was Left Out" (page xi) for some general references for those with limited programming or electronics experience.

How This Book Is Organized

The book contains information that covers a broad range of robotics tasks. The hardware and software is built up stage by stage, with each chapter using concepts explained in earlier chapters. A simple "Hello Robot" sketch is introduced in Chapter 6, *Testing the Robot's Basic Functions* and extended in subsequent chapters. Each chapter introduces sketches that add new capabilities to the robot. Experienced users can skip directly to the chapters of interest—full source code for every sketch in this book is available online. However, users who want to learn all about the techniques covered will benefit and hopefully enjoy working with all the sketches presented in the book, as each sketch enables the robot to perform increasingly complex tasks.

The sketches are built using functional modules. The modules are stored using Arduino IDE tabs (see Chapter 5). Modules described in early chapters are reused later and to avoid printing the same code over and over in the book, only code that is new or changed is printed. Figure P-1 illustrates how the code is enhanced from sketch to sketch. The horizontal bars represent the sketches, the vertical bars represent functional modules that are included in the sketches. The initial 'helloRobot' sketch is transformed into the 'myRobot' sketch by the moving the code for program definitions into a module named *robotDefines.ino* and reflectance sensors into a module named *IrSensors.ino*. These module are included as tabs in the 'myRobot' sketch. Each subsequent sketch is enhanced by adding code to an existing module or creating a new module as a tab.

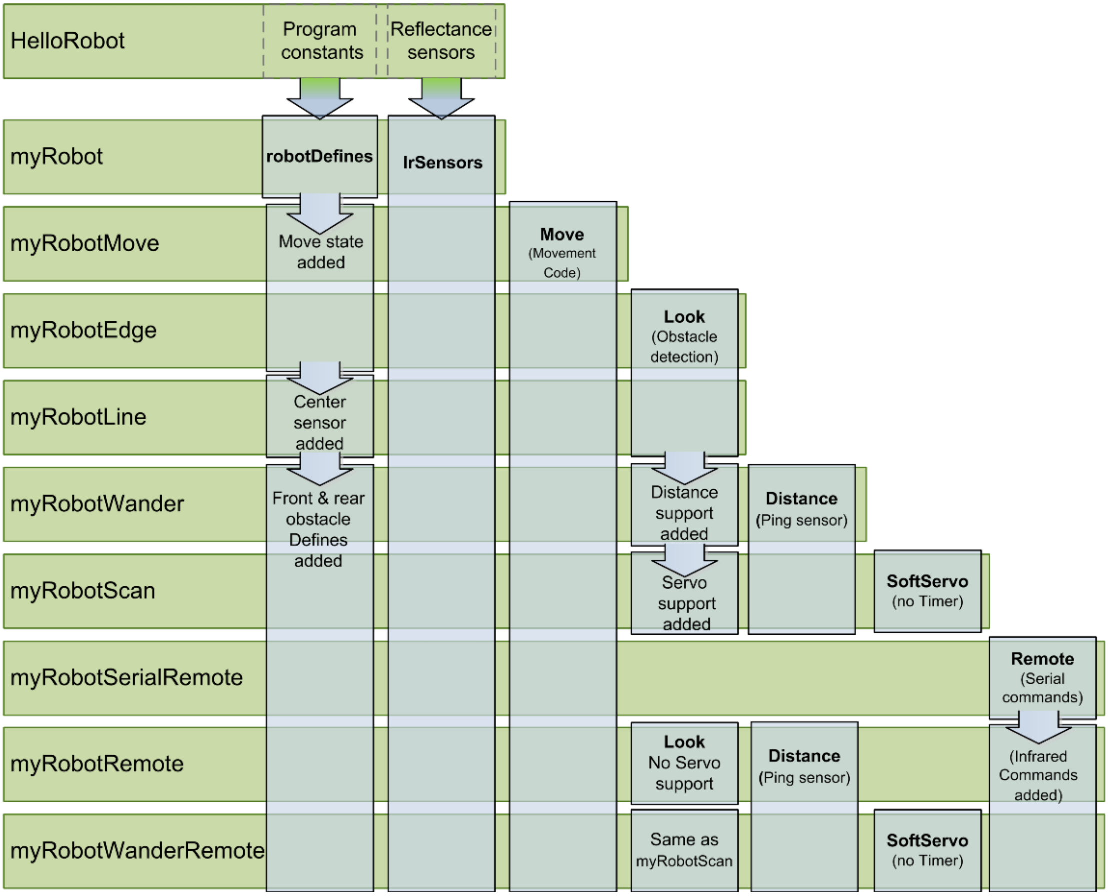


Figure P-1. *Sketch and module family tree*

All code for every sketch is available in the download for this book and you can load the sketch being discussed into your IDE if you want a complete view of all the code.

Chapter 1, *Introduction to Robot Building* provides a brief introduction to robot hardware and software.

Chapter 2, *Building the Electronics* describes how to prepare the electronics for use with the robot.

Chapter 3, *Building the Two-Wheeled Mobile Platform* describes how to assemble the 2 Wheel Drive (2WD) mobile platform.

Chapter 4, *Building the Four-Wheeled Mobile Platform* describes how to assemble the 4 Wheel Drive (4WD) mobile platform.

Chapter 5, *Tutorial: Getting Started with Arduino* introduces the Arduino environment and provides help getting the development environment and hardware installed and working.

Chapter 6, *Testing the Robot's Basic Functions* explains the first robotics sketch. It is used to test the robot. The code covered in this chapter is the basis of all other sketches in the book:

- *HelloRobot.ino* (Arduino sketch) — Brings the robot to life so you can test your build.
- *myRobot.ino* — Same functionality as above but structured into modules to make it easy to enhance.

Chapter 7, *Controlling Speed and Direction* explains how you make the robot move:

- *myRobotMove.ino* — Adds higher level movement capability.
- *myRobotCalibrateRotation.ino* — A sketch for running the robot through a range of speeds to calibrate the robot.

Chapter 8, *Tutorial: Introduction to Sensors* introduces the most popular sensors used with the 2WD and 4WD robots.

Chapter 9, *Modifying the Robot to React to Edges and Lines* describes techniques for using reflectance sensors to enable your robot to gain awareness of its environment. The robot will be able to follow lines or to avoid edges.

- *myRobotEdge.ino* — The robot will move about in an area bound by a non-reflective surface (a large sheet of white paper placed on a non-reflective surface).
- *myRobotLine.ino* — Repositions the sensors used above to allow the robot to follow black lines painted or taped to a white surface. A variant of this sketch that sends data over serial for display on an external serial device is named `myRobotLineDisplay` and is included in the download code.

Chapter 10, *Autonomous Movement* describes how to use distance sensors to enable the robot to see and avoid obstacles encountered as it moves around.

- *myRobotWander.ino* — Adds 'eyes' to give the robot the ability to look around and avoid obstacles.
- *myRobotScan.ino* — Adds a servo so robot 'eyes' can scan independent of robot movement.

Chapter 11, *Remote Control* describes techniques for remotely controlling the robot. Wired and wireless serial commands and using a TV type infrared remote control are covered.

- *myRobotSerialRemote.ino* — Controls the robot using serial commands.
- *myRobotRemote.ino* — Controls the robot using an IR remote controller.
- *LearningRemote.ino* — Captures key codes from your remote control to enable these to be added to the myRobotRemote sketch.
- *myRobotWanderRemote.ino* — Combines remote control with autonomous movement.

Appendix A, *Enhancing Your Robot* provides tips and techniques for designing and building complex projects.

Appendix B, *Using Other Hardware with Your Robot* describes some alternative solutions for motor control.

Appendix C, *Debugging Your Robot* has hardware and software debugging tips. This sections includes Arduino and Processing source code to enable real time graphical display of robot parameters on a computer screen.

- *myRobotDebug.ino* — Arduino example showing how to send data to your computer.
- *ArduinoDataDisplay.pde* (Processing sketch) — graphs data received from Arduino in real time.

Appendix D, *Power Sources* introduces some alternatives for powering your robot.

Appendix E, *Programming Constructs* provides a brief introduction to some of the programming constructs used in the sketches for this book that may not be familiar to some Arduino users.

Appendix F, *Arduino Pin and Timer Usage* summarizes the pins and Arduino resources used by the robot.

What Was Left Out

This book explains all the code used for the robot, but it is not an introduction to programming. If you want to learn more about programming with Arduino, you may want to refer to the Internet or to one of the following books:

- *Getting Started with Arduino, 2nd Edition* by Massimo Banzi (O'Reilly)
- *Arduino Cookbook, 2nd Edition* by Michael Margolis (O'Reilly)

A good book for inspiration on more robotics projects is:

- *Make: Arduino Bots and Gadgets* by Tero Karvinen, Kimmo Karvinen (O'Reilly)

Code Style (About the Code)

The code used throughout this book has been tailored to clearly illustrate the topic covered in each chapter. As a consequence, some common coding shortcuts have been avoided. Experienced C programmers often use rich but terse expressions that are efficient but can be a little difficult for beginners to read. For example, code that returns boolean values uses the somewhat verbose explicit expressions because they are easier for beginner programmers to read, see the example that follows, which returns true if no reflection was detected by the robot's sensor:

```
return irSensorDetect(sensor) == false;
```

Here is the terse version that returns the same thing (note the negation operator before the function call):

```
return !irSensorDetect(sensor);
```

Feel free to substitute your preferred style. Beginners should be reassured that there is no benefit in performance or code size in using the terse form.

One or two more advanced programming concepts have been used where this makes the code easier to enhance. For example, long lists of sequential constants use the `enum` declaration.

The `enum` keyword creates an enumeration; a list of constant integer values. All the enums in this book start from 0 and increase sequentially by one.

For example, the list of constants associated with movement directions could be expressed as:

```
const int MOV_LEFT = 0
const int MOV_RIGHT = 1;
const int MOV_FORWARD = 2;
const int MOV_BACK = 3;
const int MOV_ROTATE = 4;
const int MOV_STOP = 5;
```

The following declares the same constants with the identical values:

```
enum {MOV_LEFT, MOV_RIGHT, MOV_FORWARD,
      MOV_BACK, MOV_ROTATE, MOV_STOP};
```

In addition to brevity, there are many advantages to the `enum` version of the code. If you want to know more about `enum`, an online search for `c++ enum` should tell you all you need to know and more.

Good programming practice involves ensuring that values used are valid (garbage in equals garbage out) by checking them before using them in calculations. However, to keep the code focused on the topic, error-checking code has been kept to a minimum. If you expand the code, you are encouraged to add error-checking where needed.

Arduino Hardware and Software

The examples in this book were built using the Arduino Leonardo and Uno boards (see Chapter 5). The code has been tested with Arduino release 1.0.1 (the first release that fully supports the Leonardo board). Although many of the sketches will run on earlier Arduino releases, this has not been tested. If you really want to use a release older than 1.0, you need to change the extension from *.ino* to *.pde* to load the sketch into a pre-1.0 IDE.

There is a website for this book where you can download code for this book; see "How to Contact Us" (page xv).

There is also a link to errata on that site. Errata give readers a way to let us know about typos, errors, and other problems with the book. Errata will be visible on the page immediately, and we'll confirm them after checking them out. O'Reilly can also fix errata in future printings of the book in electronic books, and on Safari® Books Online, making for a better reader experience pretty quickly.

If you have problems getting the code to work, check the web link to see if the code has been updated. The Arduino forum is a good place to post a question if you need more help: *http://www.arduino.cc*.

If you like—or don't like—this book, by all means, please let people know. Amazon reviews are one popular way to share your happiness or other comments. You can also leave reviews at the O'Reilly site for the book.

Conventions Used in This Book

The following font conventions are used in this book:

Italic
: Indicates pathnames, filenames, and program names; Internet addresses, such as domain names and URLs; and new items where they are defined

`Constant width`
: Indicates command lines and options that should be typed verbatim; names and keywords in programs, including method names, variable names, and class names; and HTML element tags

`Constant width bold`
: Indicates emphasis in program code lines

`Constant width italic`
: Indicates text that should be replaced with user-supplied values

This icon signifies a tip, suggestion, or general note.

Using Code Examples

This book is here to help you make things with Arduino. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from this book *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Make an Arduino Controlled Robot* by Michael Margolis (O'Reilly). Copyright 2013 Michael Margolis, ISBN (978-1-4493-4437-5)."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at *http://my.safaribooksonline.com*.

How to Contact Us

We have tested and verified the information in this book to the best of our ability, but you may find that features have changed (or even that we have made a few mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

> O'Reilly Media, Inc.
> 1005 Gravenstein Highway North
> Sebastopol, CA 95472
> 800-998-9938 (in the United States or Canada)
> 707-829-0515 (international/local)
> 707-829-0104 (fax)

We have a web page for this book, where we list errata, example code, and any additional information. You can access this page at:

> *http://shop.oreilly.com/product/0636920028024.do*

To comment or ask technical questions about this book, send email to:

> *bookquestions@oreilly.com*

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

Rob DeMartin, the business manager at Maker Media, was the driving force behind the botkits, which inspired the writing of this book. Isaac Alexander and Eric Weinhoffer at Maker Media ran with the concept to make it a product. I thank them for testing the content of the book to ensure that the projects and the hardware worked well together.

I am grateful to the Arduino community for contributing a wealth of free software, in particular, the IrRemote library from Ken Sherriff that is used in the remote control chapter. I would also like to express my appreciation to Limor Fried (Ladyada) for creating the hardware, software and online build notes for the motor shield used in this book.

Thanks also to DFRobot, the innovative company that designed the robot platforms and provided the exploded view drawings used in the build chapters.

Mat Fordy at Cool Components (coolcomponents.co.uk) organized the robotics workshop that provided a testing ground for the book's projects. It was helpful and rewarding to work with the participants, each with a different level of experience, to build the robots and see their pleasure in bringing their creations to life. Their feedback helped make the book content clear, practical and fun.

If I have achieved my goal of making the rich variety of technical topics in this book accessible to readers with limited electronics or programming experience, then much of the credit goes to Brian Jepson. Brian, who was also my editor for the Arduino Cookbook, was with me every step of the way. I thank him for his guidance: from his support and passion in beginning the project, to his editorial expertise and application of his masterful communications skills right through to using his technical knowledge to test all the projects in the book.

I would like to thank my entire family for listening to me explain the finer points of robotics during a week- long vacation in the early stages of preparing this book. Four generations of my family were patient and constructive at times when they would have preferred to be boating on the lake or walking in the woods.

Finally, this book would not be what it is without the contributions made by my wife, Barbara Faden. Her feedback on early drafts of the manuscript helped shape the content. I am especially grateful for her support and patience in the wake of disruption created as I wrangled with these two little robots to meet the book's deadline.

Introduction to Robot Building

1

This book takes you through the steps needed to build a robot capable of autonomous movement and remote control. Build instructions are provided for 2WD (two wheel drive) and 4WD (four wheel drive) platforms. The platforms shown in Figure 1-1 and Figure 1-2 will make the construction a snap, but you can build your own robot chassis if you prefer. The connection and use of the control electronics and sensors are fully explained and the source code is included in the book and available for download online (see "How to Contact Us" (page xv) for more information on downloading the sample code).

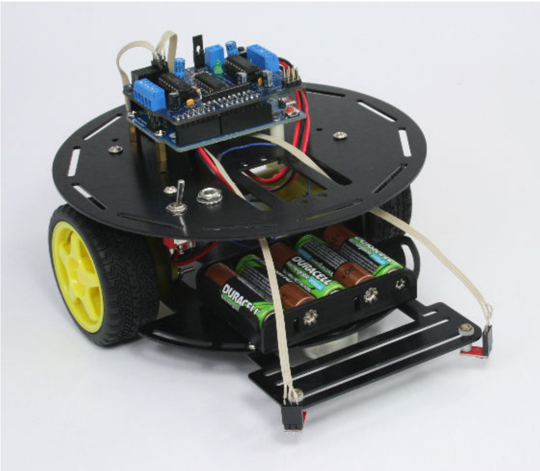


Figure 1-1. *The assembled two wheeled robot chassis*

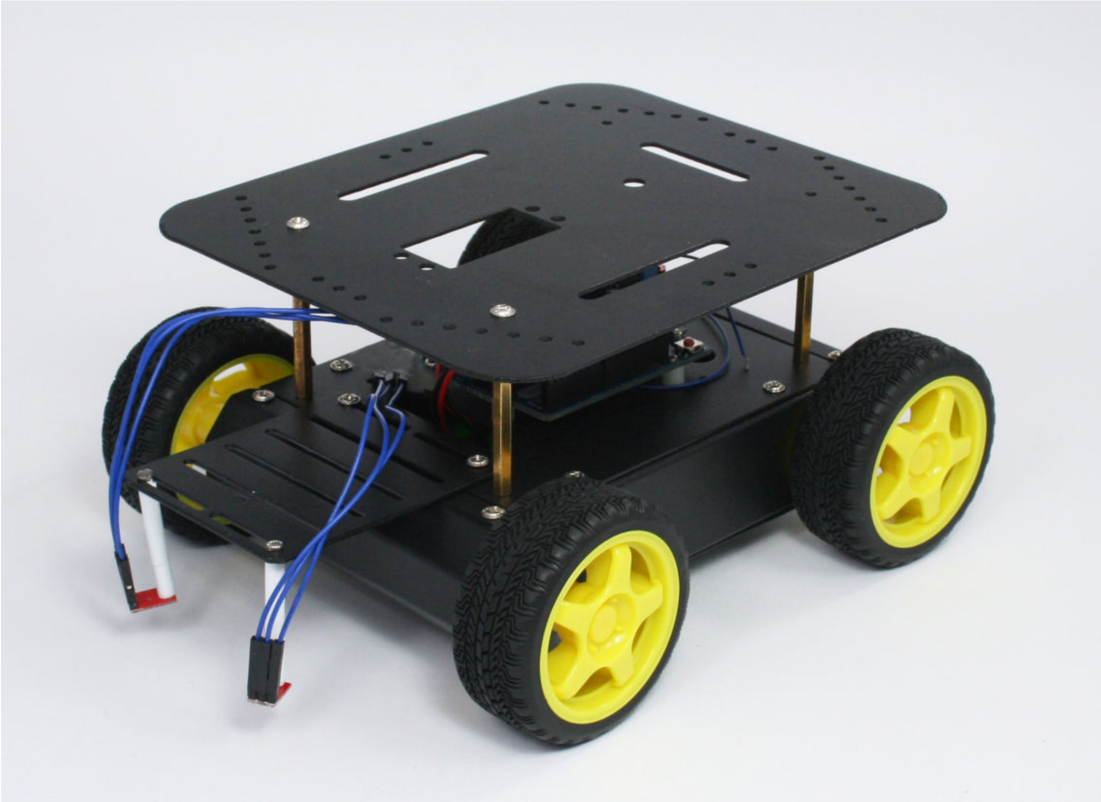


Figure 1-2. *The assembled four wheeled robot chassis*

Here is a preview of the projects you can build:

- Controlling speed and direction by adding high level movement capability.
- Enabling the robot to see the ground—using IR sensors for line and edge detection (see Figure 1-3 and Figure 1-4).
- Enabling the robot to look around—scanning using a servo so the robot can choose the best direction to move, as shown in Figure 1-5.
- Adding remote control using a TV remote control or a wired or wireless serial connection.

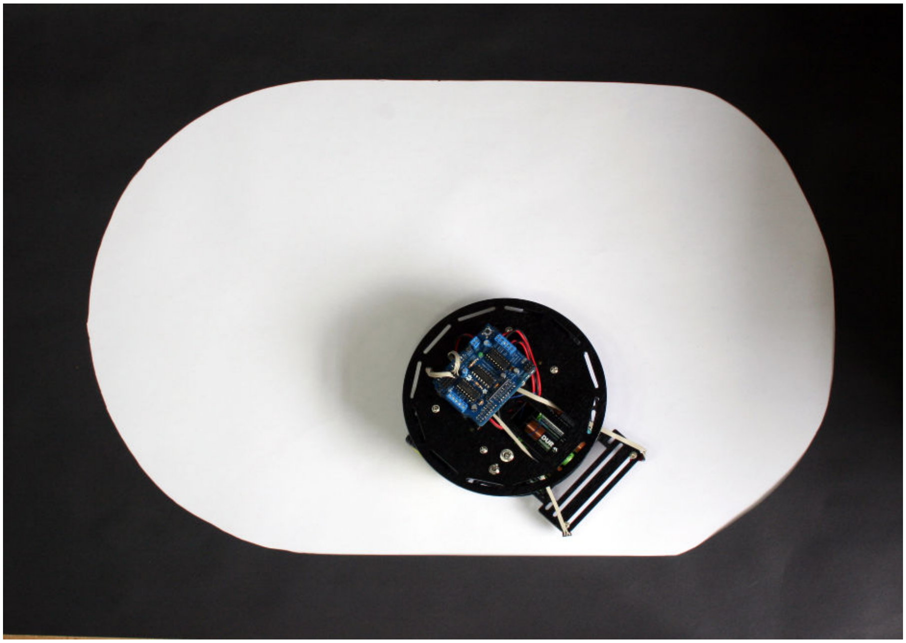


Figure 1-3. *Robot moves around but remains within the white area*

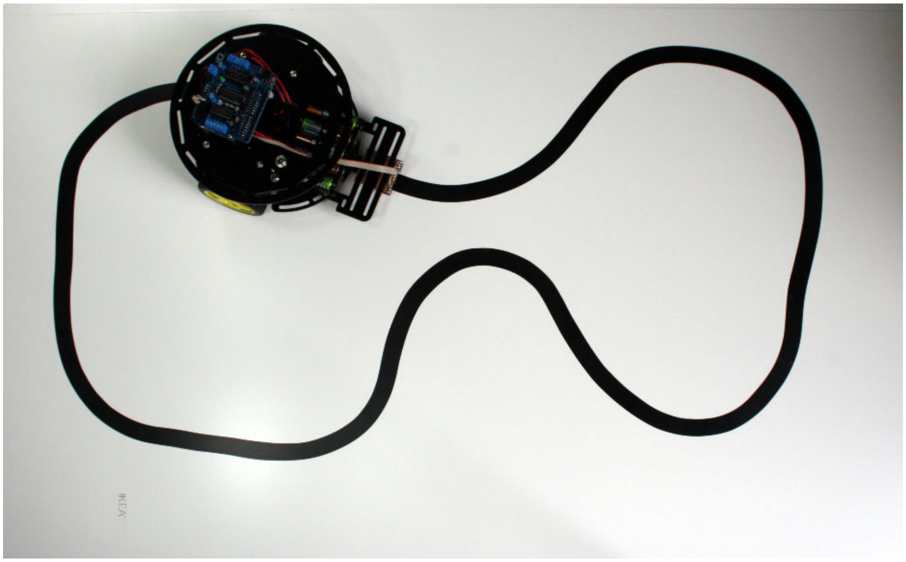


Figure 1-4. *Robot follows black line*

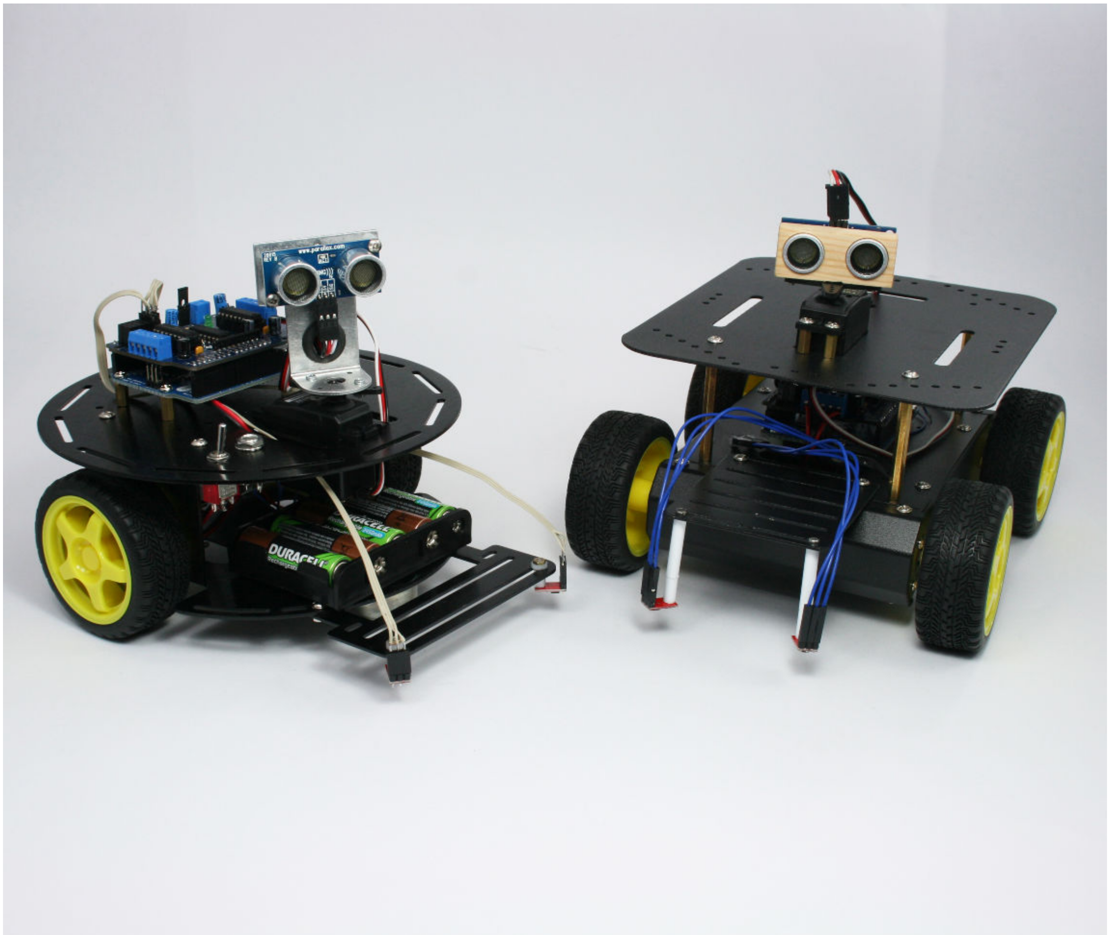


Figure 1-5. *Two wheeled and four wheeled robots with distance scanners*

Why Build a Robot?

Building a robot is different from any other project you can make with a microcontroller. A robot can move and respond to its environment and exhibit behaviors that mimic living creatures. Even though these behaviors may be simple, they convey a sense that your creation has a will and intent of its own. Building a machine that appears to have some spark of life has fascinated people throughout the ages. The robots built over 60 years ago by neurophysiologist W. Grey Walter (see *http://www.extremenxt.com/walter.htm*) explored ways that the rich connections between a small number of brain cells give rise to complex behaviors.

There are many different kinds of robots, some can crawl, or walk, or slither. The robots described in this book are the easiest and most popular; they use two or four wheels driven by motors.

Choosing Your Robot

The projects in this book can use either a two or four wheeled platform, but if you are still deciding which is right for you, here are some factors that will help you choose:

Two Wheeled Robot
: Light and very maneuverable, this is a good choice if you want to experiment with tasks such as line-following that require dexterous movement. However, the caster that balances the robot requires a relatively smooth surface.

Four Wheeled Robot
: This robot's four wheel drive makes this a good choice if you want it to roam over rougher surfaces. This platform has a large top plate that can be used to carry small objects. The robot is heavier and draws more current than the 2WD robot, so battery life is shorter.

How Robots Move

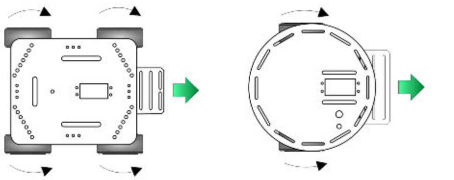


Figure 1-6. *Left and Right wheels turn forward, Robot moves Forward*

The robots covered in this book move forward, back, left and right much like a conventional car. Figure 1-6 shows the wheel motion to move the robot forward.

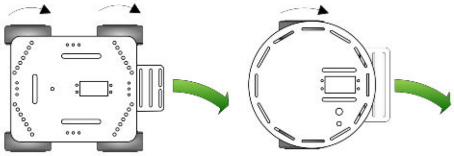


Figure 1-7. *Only Left wheels turn, Robot Turns Right*

If the wheels on one side are not driven (or are driven more slowly than the other side) the robot will turn, as in Figure 1-7.

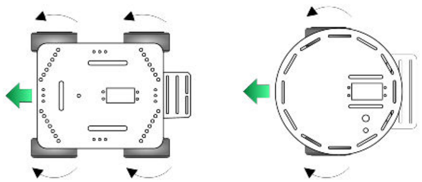


Figure 1-8. *Left and Right wheels turn backward, Robot moves Backward*

Figure 1-8 shows that reversing the wheel rotation drives the robot backward.

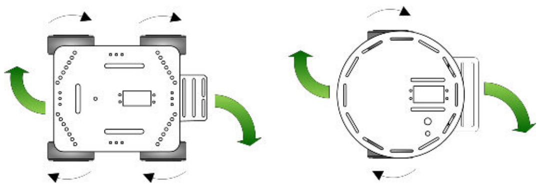


Figure 1-9. *Left wheels turn forward, Right wheels reverse, Robot rotates Clockwise*

Unlike a car (but a little like a tank), these robots can also rotate in place by driving the wheels on each side in different directions. If the wheels on each side are spinning in opposite directions, the robot will rotate. Figure 1-9 shows clockwise rotation.

Tools

These are the tools you need to assemble the robot chassis.

Phillips Screwdriver
: A small Phillips screwdriver from your local hardware store.

Small long-nose or needle-nose pliers
: For example, Radio Shack 4.5-inch mini long-nose pliers, part number 64-062 (see Figure 1-10) or Xcelite 4-inch mini long-nose pliers, model L4G.

Small wire cutters
: For example, Radio Shack 5” cutters, part number 64-064 (Figure 1-11) or Jameco 161411

Soldering iron
: For example, Radio Shack 640-2070 (Figure 1-12) or Jameco 2094143 are low cost irons suitable for beginners. But if you are serious about electronics, a good temperature controlled iron is worth the investment, such as Radio Shack 55027897 or Jameco 146595.

Solder 22 AWG (.6mm) or thinner
: For example, Radio Shack 640-0013 or Jameco 73605.

Figure 1-10. *Small Pliers*

Figure 1-11. *Wire Cutters (Side Cutters)*

Figure 1-12. *Soldering Iron*

Building the Electronics

2

This chapter guides you through the electronic systems that will control your robot. Both the two wheeled and four wheeled platforms use the same modules, a pre-built Arduino board (Arduino Uno or Leonardo), and a motor controller kit. The motor controller featured in this book is the AFMotor shield from Adafruit Industries. Although other motor controllers can be used (see Appendix B) the AFMotor shield provides convenient connections for the signals and power to all the sensors and devices covered in this book. It is also capable of driving four motors, which is required for the four wheel drive chassis.

Although the attachment of the boards to the robot differs somewhat depending on the chassis, the building of the AFMotor circuit board kit is the same for both. If you don't have much experience with soldering, you should practice soldering on some wires before tackling the circuit board (you can find soldering tutorials here: *http://www.ladyada.net/learn/soldering/thm.html*).

Hardware Required

See *http://shop.oreilly.com/product/0636920028024.do* for a detailed parts list.

- Tools listed in "Tools" (page 6)
- AFMotor shield kit
- Three 6 way 0.1" female headers
- Three QTR-1A reflectance sensors
- Stripboard, three 3 way 0.1" headers for line sensor mount
- Ribbon Cable, 11-way or wider, cut with a sharp knife as follows:
    - — One 10 inch length of 5 conductor ribbon cable for line sensors

  - — Two 10 inch lengths of 3 conductor ribbon cable for edge sensors
- Optional: 3 way 0.1″ female header for optional charging circuit
- Optional: 3 way 0.1″ female header for optional wireless connection

Construction Techniques

This section provides an overview of the motor controller shield construction.

Soldering

Soldering is easy to do if you understand the basic principles and have a little practice. The trick for making a good solder joint is to provide the right amount of heat to the parts to be soldered and use the right solder. 22 AWG solder (0.6mm or .025 inch) or thinner is a good choice for soldering printed circuit boards. A 25-watt to 40-watt iron, ideally with temperature control, is best. The components to be joined should be mechanically secure so they don't move while the solder is cooling—wires should be crimped around terminals (see Figure 4-11 and Figure 4-12). To make the joint, the tip of the iron should have good contact with all the components to be soldered. Feed a small amount of solder where the iron is touching the parts to be joined. When the solder flows around the joint, remove the solder first and then the iron. The connection should be mechanically secure and the joint shiny.

Building the Motor Controller

The motor controller shield is the heart of this robot. As well as controlling the motors, all the sensors are connected to Arduino through this board. The shield is provided as a kit and is the same for use with either the 2WD and 4WD robots, differing only in the method of connecting the motors and mounting to the chassis (both are detailed in later chapters).

The following is an overview of the construction with some tips that you should read through before starting to build the circuit board. You can find step by step construction details for the shield at this site: *http://ladyada.net/make/mshield/solder.html*

Figure 2-1 shows the components for the shield.

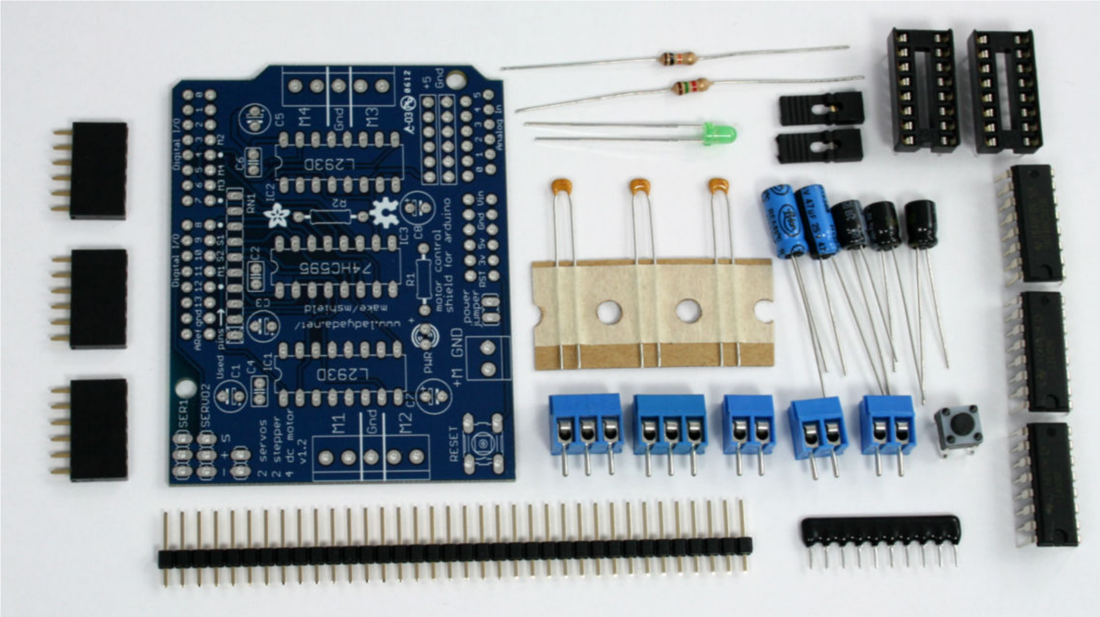


Figure 2-1. *Parts required to build the Motor Shield*

The parts to the right of (as well as below) the board are packed with the shield, but the three 6-pin headers on the left are not supplied with the standard shield. These headers are used to connect the sensors. These headers are *included with the Maker Shed companion kits that go along with this book. You can also purchase female headers from Adafruit and other suppliers.*

The two Maker Shed kits can be found at http://www.makershed.com/Bots_and_Bits_for_Bots_s/46.htm. Look for either the Rovera 2W (Arduino-Controlled 2 Wheel Robotics Platform) or Rovera 4W (Arduino Controlled 4 Wheel Robotics Platform).

Solder the smallest components first (Figure 2-2). The three small capacitors and two resistors are not polarized so you can insert them either way around.

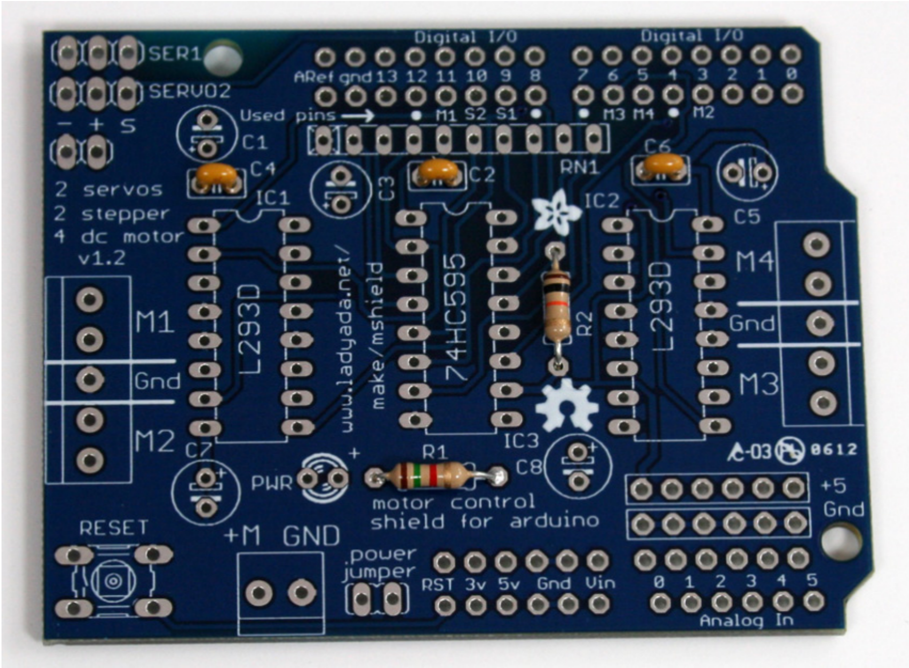


Figure 2-2. *Solder the Small Components*

The resistor network (the long thin component with ten pins) *is* polarized—the end with the white dot goes to the left of the PCB (nearest to C1) as shown in Figure 2-3.

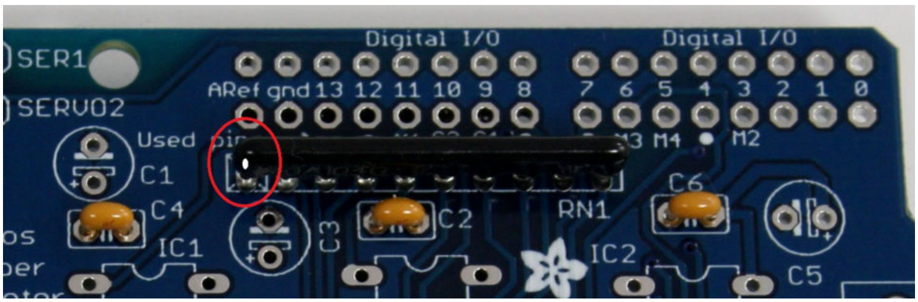


Figure 2-3. *Solder the resistor network - the marker (circled) indicates correct orientation*

The large capacitors, ICs, and LED are all polarized. The color of the components shown in the step-by-step assembly pictures on the Adafruit site (you can find the link at the beginning of these build notes) may not match the components or layout for the parts you received (particularly the capacitors) so carefully check that you have placed the correct value component in the correct orientation. Figure 2-4 shows the layout for version 1.2 of the shield PCB. The kit includes two IC sockets for the L293D chips. As mentioned in the assembly instructions on the Adafruit site, these are optional but if you like to play safe and want to use the sockets, solder them so the indent indicating pin 1 matches the outline printed on the PCB.

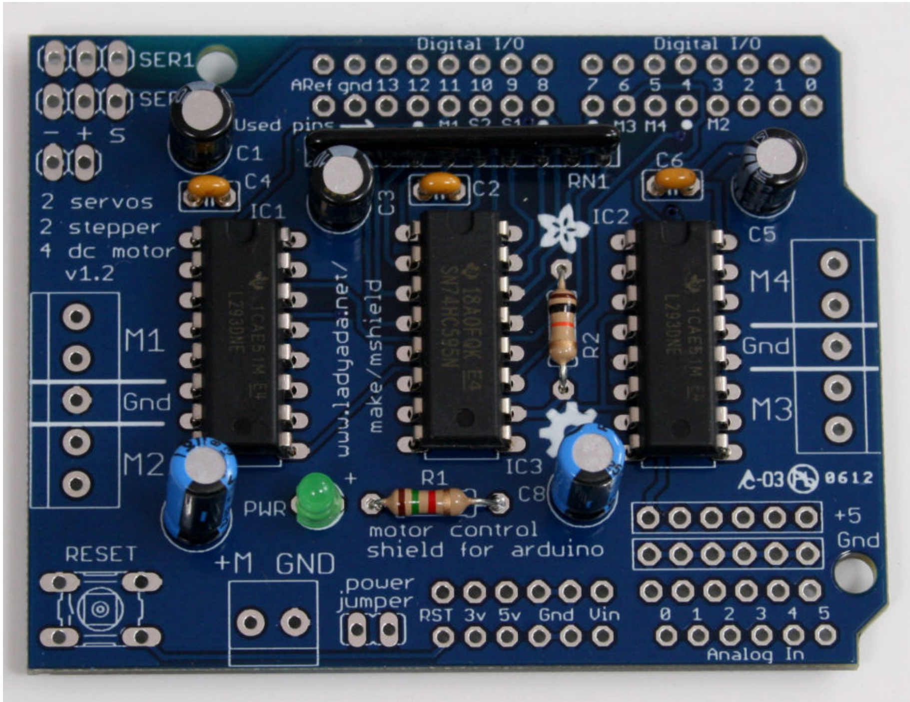


Figure 2-4. *Solder the rest of the polarized components*

Figure 2-5 shows the board with all of the standard shield components (pushbutton, headers, screw terminals) soldered. The final assembly step is to solder the three 6-pin female headers near the analog input pins. These headers are not included in the shield package or mentioned in Adafruit's step-by-step build instructions, but are included with the Maker Shed kits.

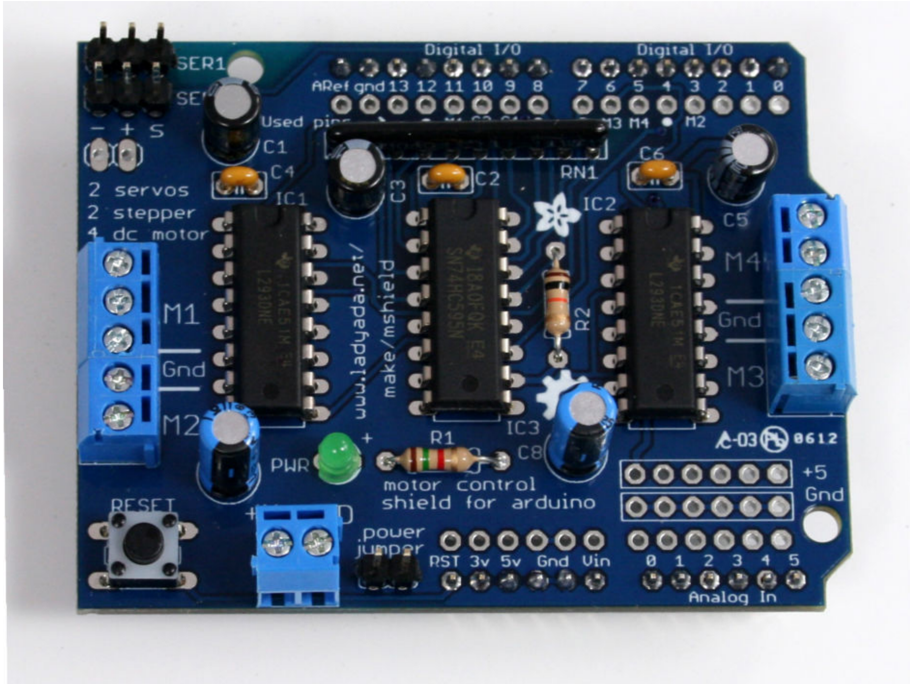


Figure 2-5. *Everything soldered except the sensor headers*

Figure 2-6 shows all components including the sensor headers soldered. Trim the component pins (except the header pins that connect the shield to the Arduino) on the underside of the board so they are clear of the Arduino when the shield is plugged onto the board. Locate one of the jumpers supplied with the shield and plug this onto the pins marked *power jumper*—this connects the motor power input and the Arduino VIN (power input) together so both are fed from the batteries that you will be wiring after you have built the robot chassis.

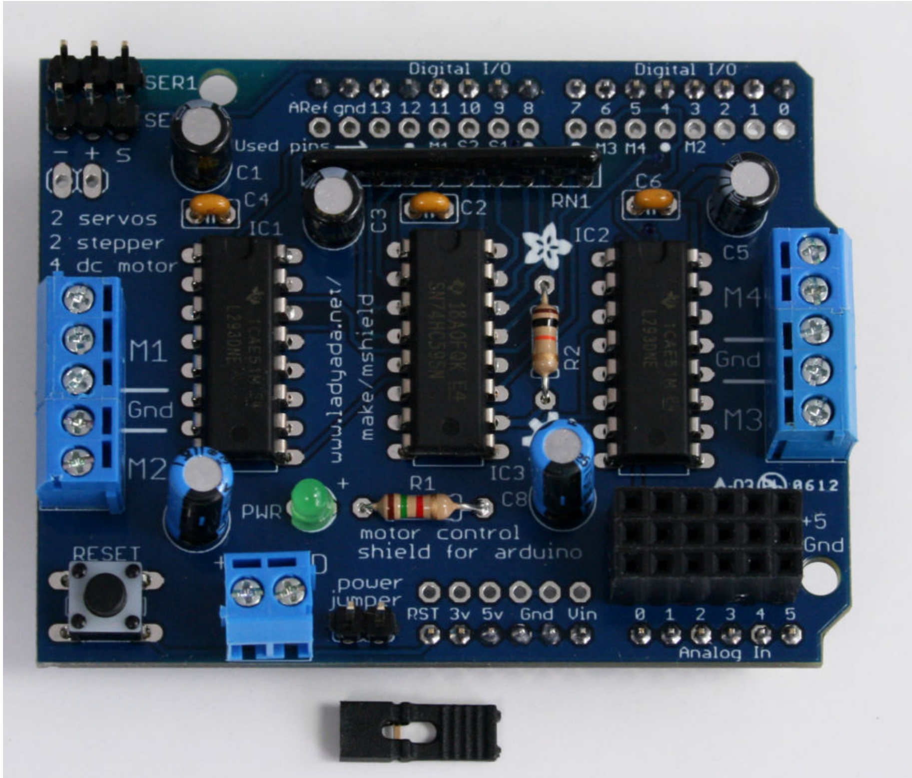


Figure 2-6. *Shield with sensor headers*

Figure 2-7 shows where all of the sensors and other external devices will be connected. The three pin female headers are not needed for some of the projects but you will find it convenient to solder these to the shield at this time.

Figure 2-8 shows two styles of connections. On the left, you'll find the stripboard-based wiring scheme as described in "Making a Line Sensor Mount" (page 17). As you'll see in later chapters, you can experiment with a variety of mounting methods, including the stripboard-based one. The right side of Figure 2-8 shows the wiring for separately connected sensors. As you read through the later chapters and experiment with various mounting techniques, you'll use one or the other wiring schemes. Because you'll be using sockets and ribbon connectors to hook up the sensors, you won't be locked into any particular connection scheme; you can mix and match.

The left and right designation in the diagram refers to left and right from the robot's perspective, and the later chapters will explain where to connect these.

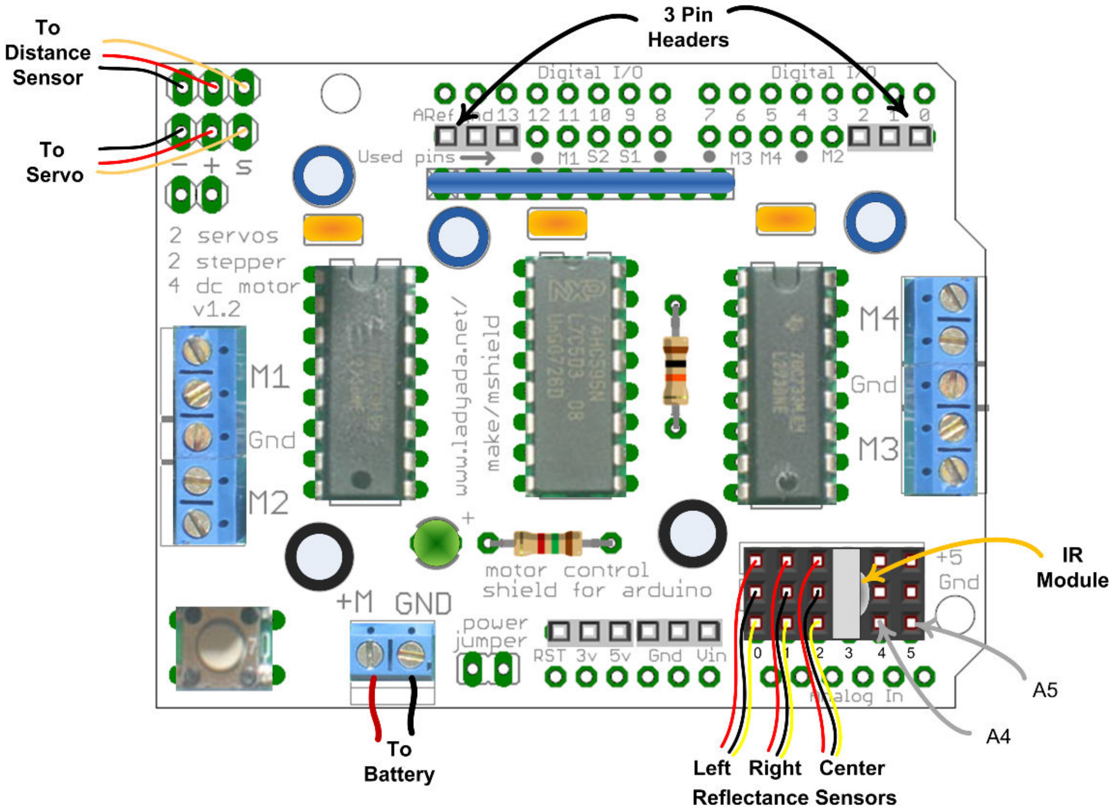


Figure 2-7. *Connections for devices covered in the chapters to come*

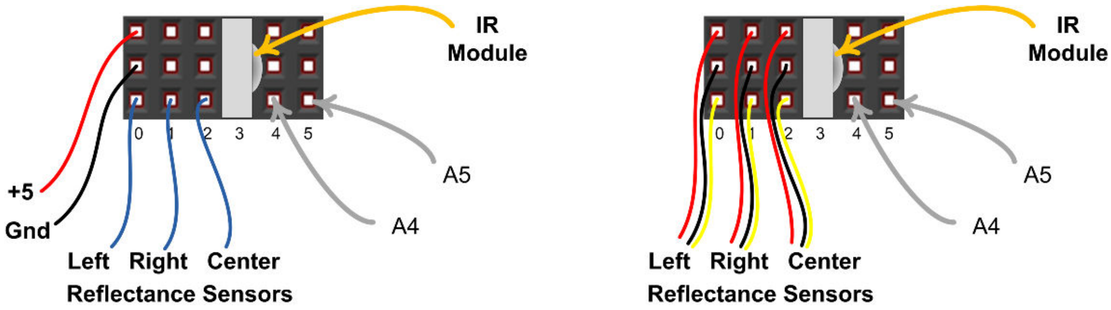


Figure 2-8. *Connection detail - stripboard wiring is shown on the left, individual jumpers shown on the right*

Soldering the Reflectance Sensors

Each sensor package contains a small PCB and a 3-pin header. Insert the header so the shorter length pins emerge on the side of the board with components already soldered, see Figure 2-9. After ensuring you have the header the right way around, solder the three pins.

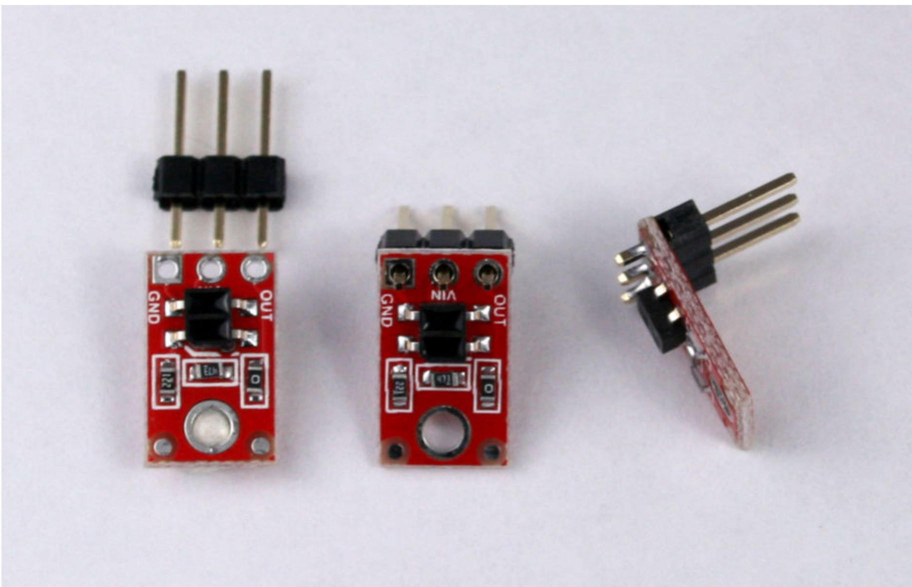


Figure 2-9. *QTR-1A Reflectance Sensors*

Making a Line Sensor Mount

The line sensing project in this book uses three reflectance sensors wired to analog inputs. Although it is possible to wire the three connections (+5V, Gnd, and Signal) using 9 jumpers, it is more convenient to use a small piece of stripboard to connect the power lines together. Header sockets soldered to the stripboard enable the sensor to be easily unplugged so you can change configuration if you want to swap back and forth between line and edge detection. Figure 2-10 shows the layout of the stripboard (note the five holes you'll need to drill out with a hand drill). Figure 2-12 shows the wires soldered directly to the stripboard pads. If you'd like to add some strain relief, you can drill out a few extra holes in an unused area of the stripboard. Next, divide the wire into two groups (one for positive and negative, and three for the analog pins), and feed the wires through large holes in the board before you solder them. That way, if you tug on the wires, they'll pull against the holes before they pull against your solder joints.

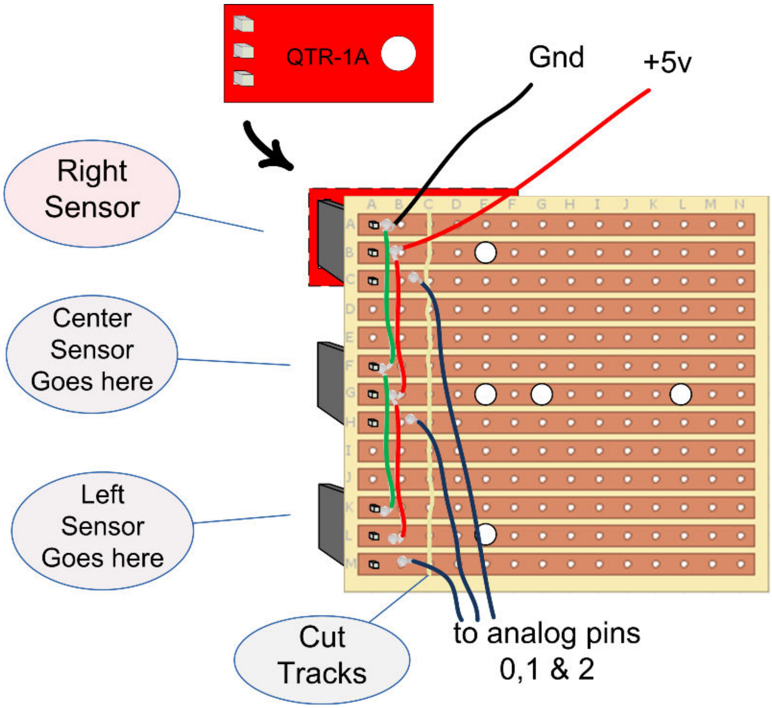


Figure 2-10. *Stripboard layout for mounting QTR-1A reflectance sensors for line following*

To ensure that the mounting bolts don't short the tracks, you can either cut the tracks as shown in Figure 2-10 (you will be cutting along the third column from the left, or the "C" column) or use insulated washers between the bolts heads and the tracks. Figure 2-11 shows how the header sockets are connected, and Figure 2-12 shows the completed stripboard, with the ribbon cable connected. A ten inch length of cable is more than ample. Figure 2-13 shows the other end of the ribbon connected to shield pins.

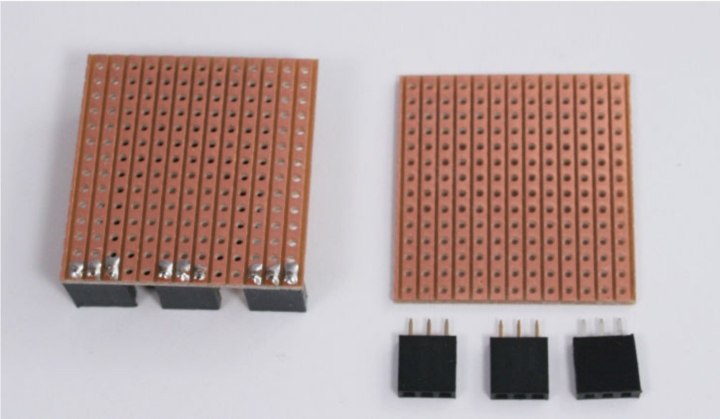


Figure 2-11. *Stripboard with three 3 pin header sockets*

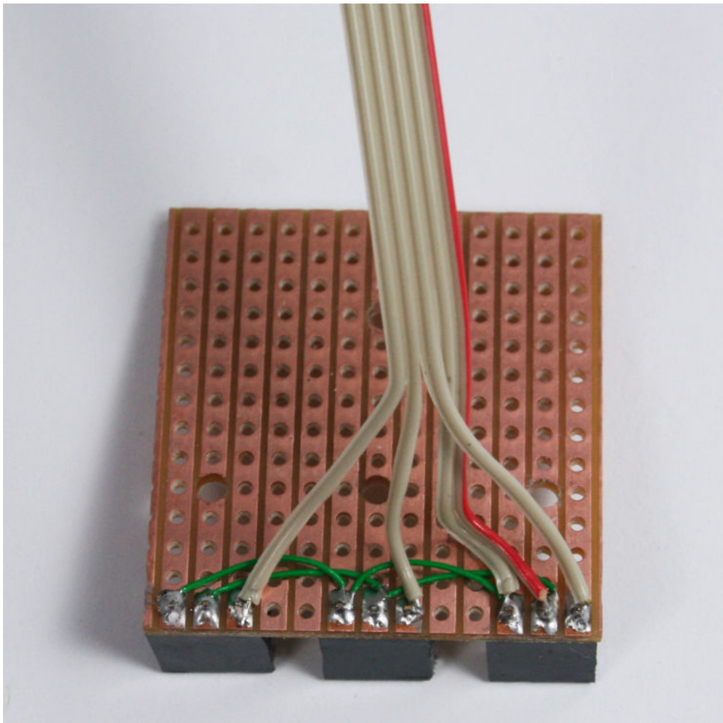


Figure 2-12. *Stripboard with all wires soldered*

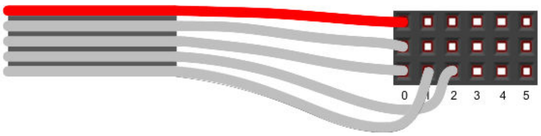


Figure 2-13. *Ribbon cable connections to shield pins*

The method of mounting the stripboard depends on the robot chassis; see Chapter 3, *Building the Two-Wheeled Mobile Platform* or Chapter 4, *Building the Four-Wheeled Mobile Platform*. The three holes shown will suit either chassis but you may prefer to wait until you have built the chassis and only drill the holes you need.

Next Steps

The next stage in building the robot is to assemble the chassis. Chapter 3 covers the two-wheeled robot and Chapter 4 is for the four-wheeled version.

Building the Two-Wheeled Mobile Platform

3

This chapter provides advice on the construction of a Two Wheel Drive (2WD) chassis with front caster, as shown in Figure 3-1. Construction is straightforward; you can follow the detailed steps or improvise if you want to customize your robot. The chapter also shows how you attach and connect sensors used in the projects covered in later chapters.

If you prefer to build a two wheeled robot of your own design, you should read the sections on attaching control electronics and sensors; this will prepare you to use the code for the projects in the chapters to come. Information in this chapter my also provide some ideas to help with the design of your own robot.

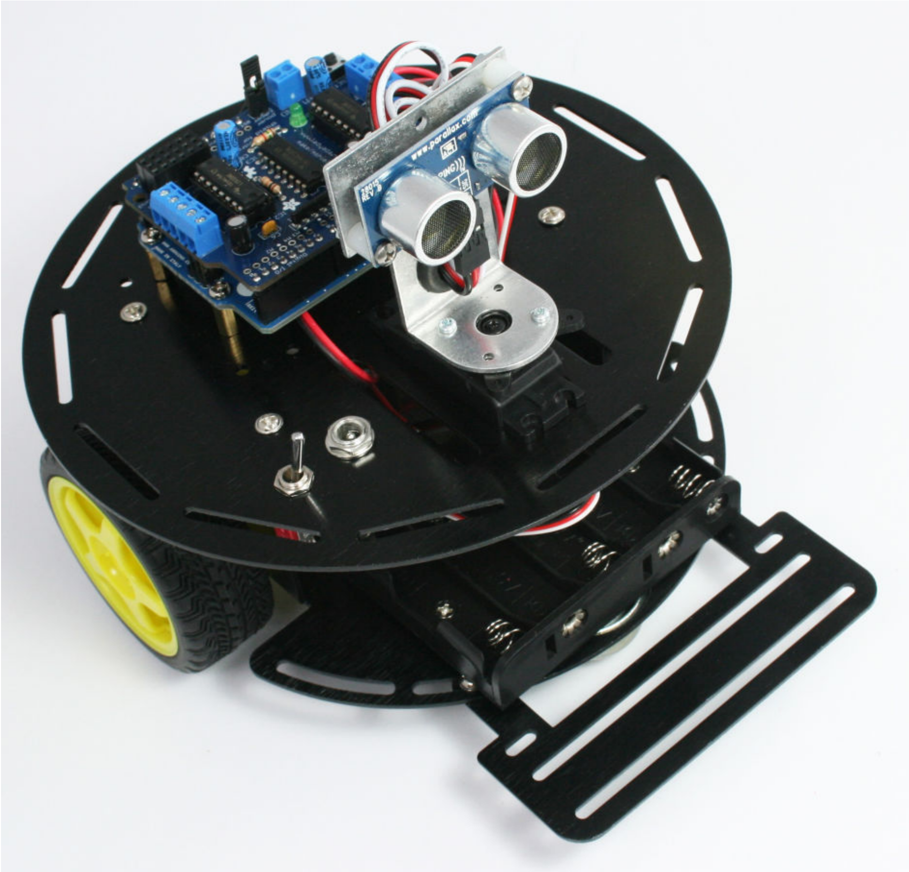


Figure 3-1. *2WD Robot Chassis*

Hardware Required

See *http://shop.oreilly.com/product/0636920028024.do* for a detailed parts list.

- Tools listed in "Tools" (page 6)
- The assembled electronics (see Chapter 2, *Building the Electronics*
- 2WD Mobile Platform (two wheeled robot kit made by DFRobot)
- Two 0.1uF ceramic capacitors
- Two lengths of 3 conductor ribbon cable, two 3 way 0.1" headers for edge sensors
- Optional: charging circuit resistors and diode, see detailed parts list

Mechanical Assembly

Lay Out the Chassis Parts

Figure 3-2 shows all of the parts contained in the 2WD chassis package. The three black brackets to the left of the figure are not needed for any of the projects in this book.

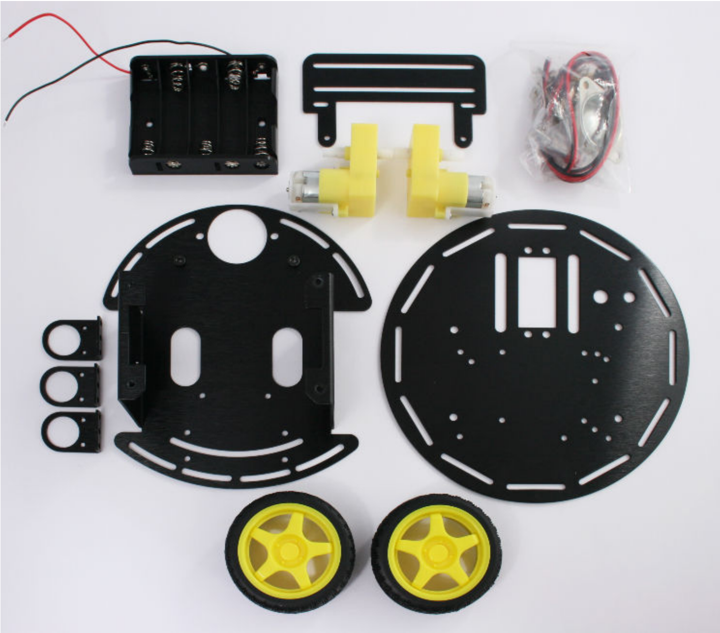


Figure 3-2. *2WD Chassis Parts*

Figure 3-3 shows the contents of the bag containing the mounting hardware. Locate the two bolts with the flat heads and put them aside for mounting the battery case. Also identify the two thicker (M4) bolts that will be used to attach the caster. The remaining short bolts in this pack are identical.

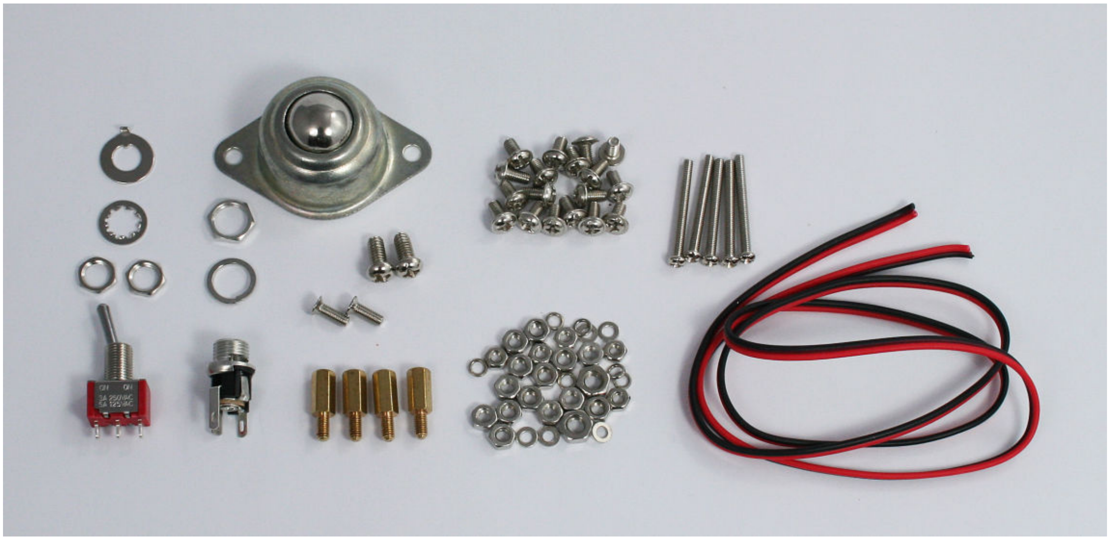


Figure 3-3. *2WD hardware pack contents*

Motor Assembly

Use two long bolts with lock washers and nuts, as shown in Figure 3-5, to attach each motor to the chassis lower plate. Tighten the nuts snugly but take care not to stress the plastic motor housing.

Lock washers are used to prevent a nut from accidentally coming lose due to vibration. This is particularly important for attaching the motor and switch. These washers have a split ring or serrations that apply extra friction when tightened.

If you find that things still come lose, don't overtighten the nuts; an effective solution is retighten the nut and apply a dab of nail polish to the point where the threads emerge from the nut.

Figure 3-4 shows the motors in place with the nut seen on the upper right ready to be tightened.

Figure 3-4. *Motors mounted on the chassis lower plate*

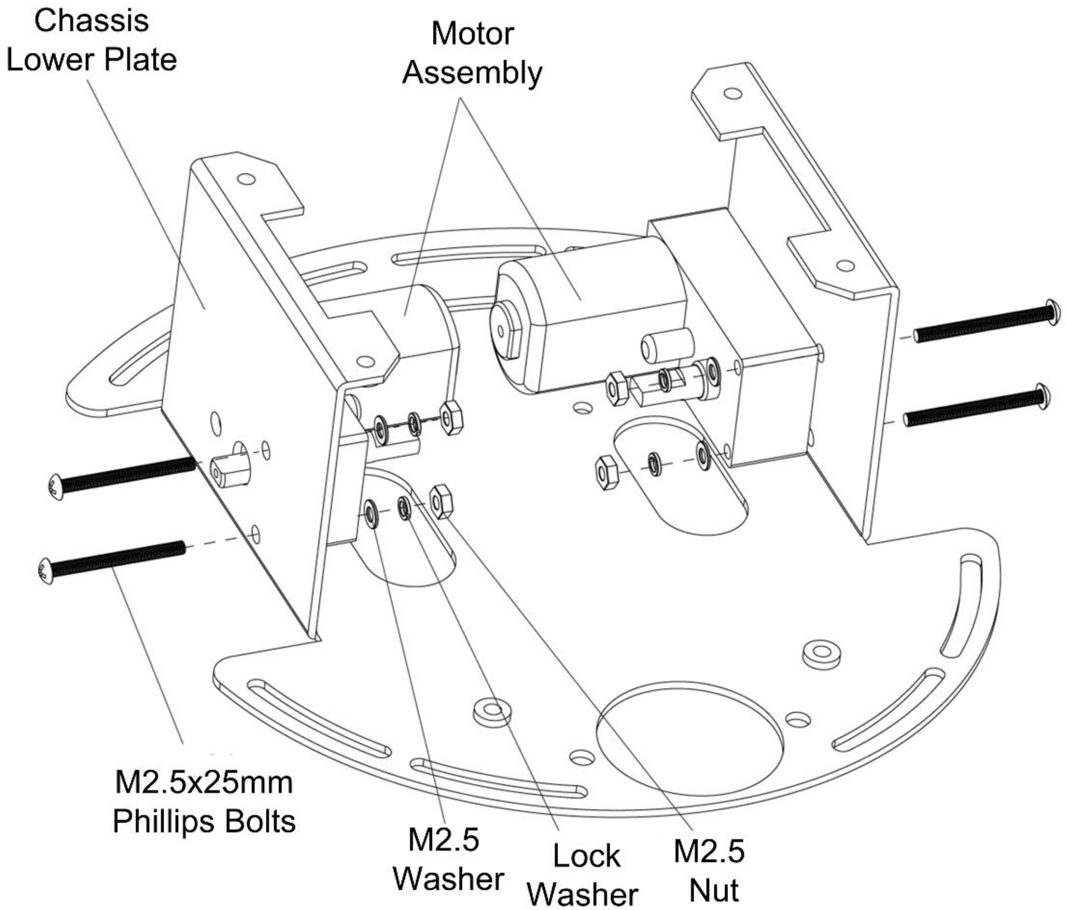


Figure 3-5. *Motor Assembly*

Assemble the Chassis Components

Push the wheels onto the motor assembly shafts, aligning the slots in the wheels with the flat section of the motor shaft. Attach the caster with two M4 bolts and nuts. Figure 3-6 and Figure 3-7 show this.

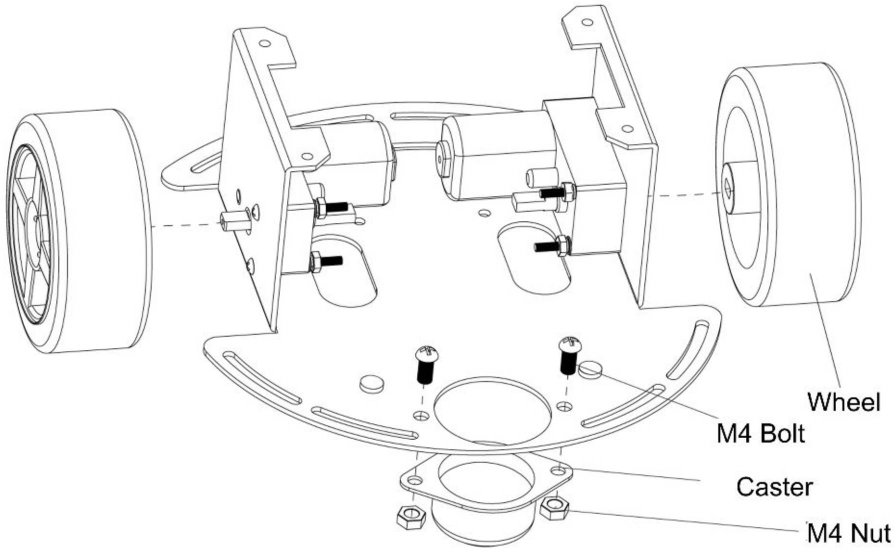


Figure 3-6. *Motor Assembly*

Figure 3-7. *Wheels and caster mounted*

Attach the sensor bracket to the underside of the lower chassis plate, as seen in Figure 3-8 and Figure 3-9.

This robot is sometimes built with the sensor plate mounted at the opposite end of the chassis (furthest from the caster). You can build yours however you like, but the orientation shown here enables the servo mounted distance scanner to be attached in the front of the robot. Also, the sensor bracket in this location maximizes the distance between the wheels and the line sensors and this improves line following sensitivity.

Figure 3-9 shows the underside of the chassis after mounting the sensor bracket. Note that the sensor bracket is attached to the bottom of the chassis plate.

Figure 3-8. *Sensor bracket viewed with the robot right side up*

Figure 3-9. *Sensor bracket viewed with the robot upside down*

The battery pack is bolted to the bottom base plate with two countersunk (flat headed) Phillips bolts as shown in Figure 3-10 and Figure 3-11. You may want to delay this step until after the battery leads have been soldered to make it easier to position all the wires.

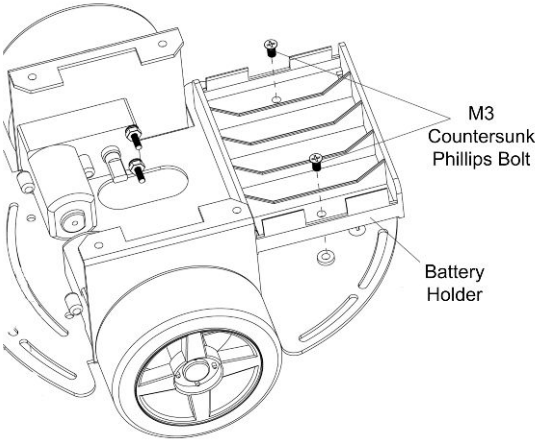


Figure 3-10. *Motor Assembly*

Figure 3-11. *Chassis with Battery Pack Attached*

Cut two pieces of red/black wire, each about 7 1/2 inches long. Strip to expose about 3/16 inch of bare wire at one end of the wires and attach to the motor terminals. Strip 1/4 inch off the other end of the pairs of wires; these will be connected to the motor shield. Connect a 0.1uF capacitor across each of the motor terminals, as shown in Figure 3-12. The capacitors suppress electrical spikes generated by the motor that could interfere with signals on the Arduino board.

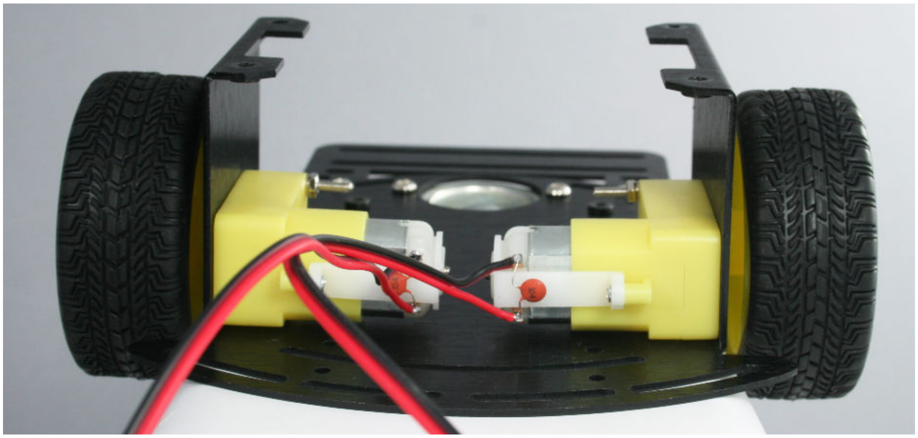


Figure 3-12. *Wires and capacitors soldered to Motors*

The DC power jack is bolted to the top plate using the large (M8) lock washer and nut. The switch is mounted using two (M6) nuts and a lock washer. Put one nut on the switch leaving around 3/16" of thread above the nut. Then place the lock washer on the thread and push this through the opening in the rear plate and secure with the second M6 nut.

Orient the switch so the toggle moves towards the jack, as shown in Figure 3-13 and Figure 3-14 (Figure 3-15 shows the view from beneath).

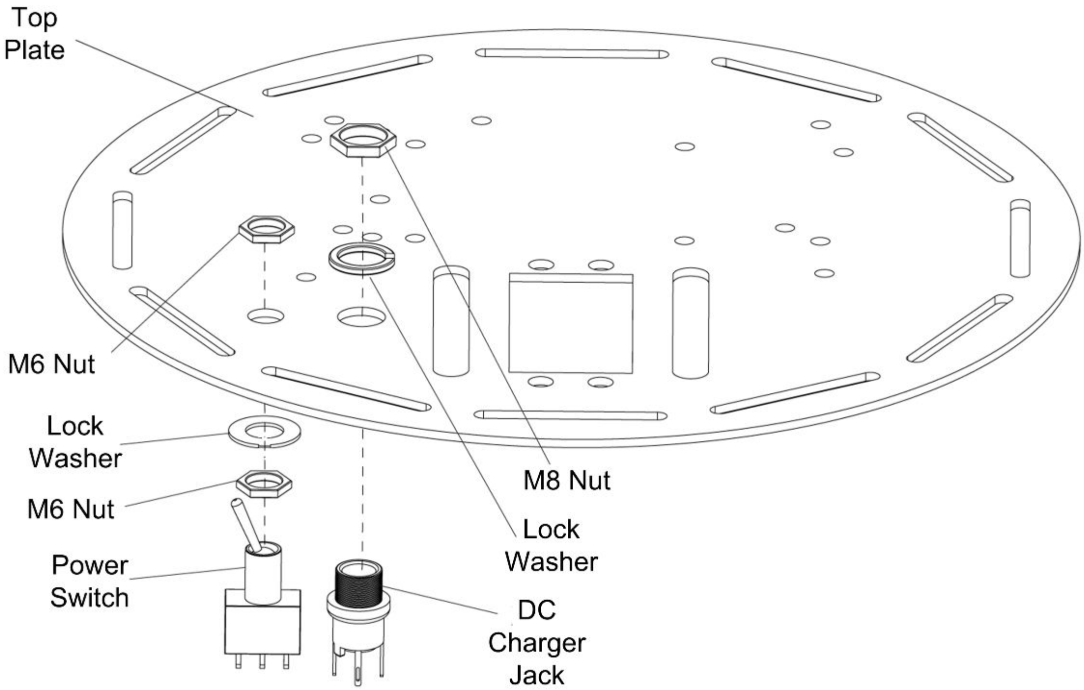


Figure 3-13. *Switch and Jack Assembly*

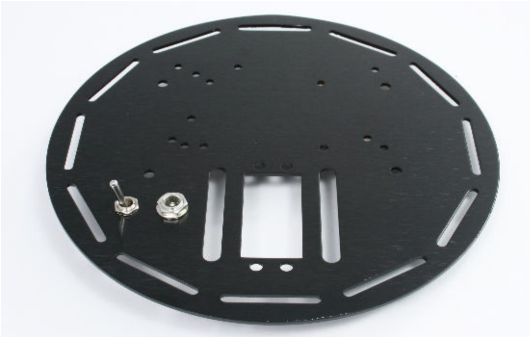


Figure 3-14. *Top panel showing location of switch and DC jack*

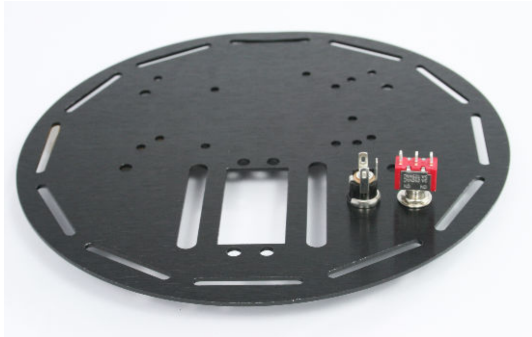


Figure 3-15. *Top panel underside showing orientation of switch and jack*

The battery can be wired as shown in Figure 3-16 and Figure 3-17. The power switch will disconnect the battery when the robot is not in use. The DC jack is not used in this configuration (other than as a junction point for the black ground wires). The switch is off when the toggle is closer to the DC jack as shown (the toggle is a lever; when the exposed end is up as seen in the figure, the contact at the bottom is connected and the contact wired to the shield is open).

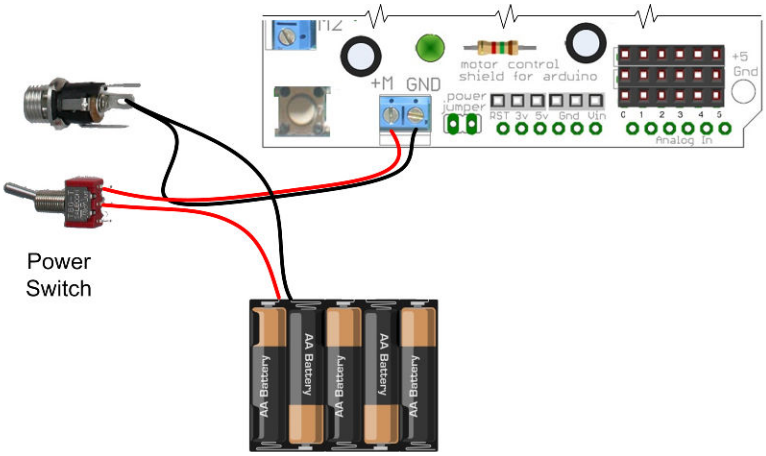


Figure 3-16. *Basic Switch Wiring (no trickle charger)*

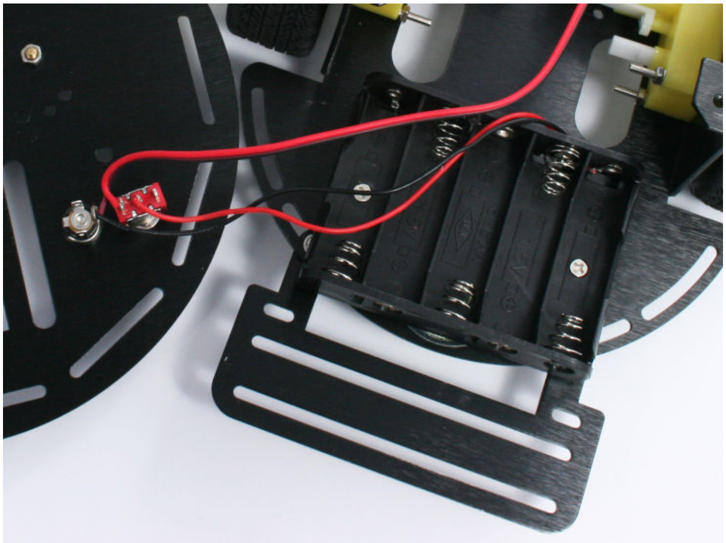


Figure 3-17. *Solder the battery wires to the switch*

You can build a simple trickle charger into the robot if you will be using rechargeable NiMH batteries. The charger can be built using the circuit shown in Figure 3-18 and Figure 3-19. See "Trickle Charging" (page 229) for information about using the charger.

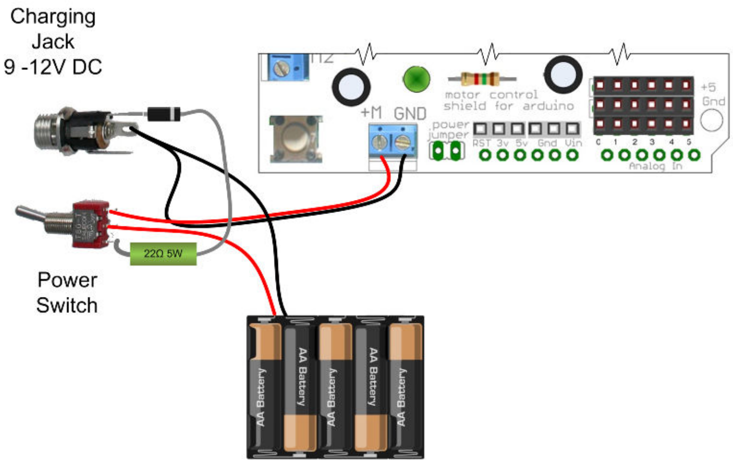


Figure 3-18. *Optional Trickle Charger Wiring*