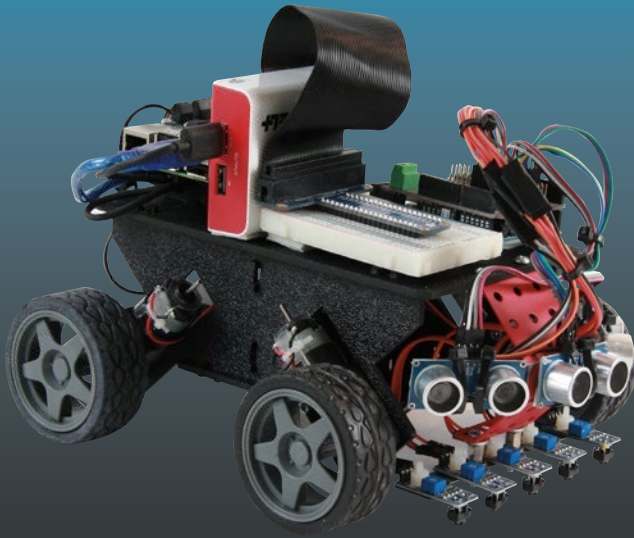


TECHNOLOGY IN ACTION™



Beginning Robotics with Raspberry Pi and Arduino



Using Python and OpenCV

—
Second Edition

—
Jeff Cicolani

Apress®

Beginning Robotics with Raspberry Pi and Arduino

Using Python and OpenCV

Second Edition

Jeff Cicolani

Apress®

Beginning Robotics with Raspberry Pi and Arduino: Using Python and OpenCV

Jeff Cicolani
Pflugerville, TX, USA

ISBN-13 (pbk): 978-1-4842-6890-2
<https://doi.org/10.1007/978-1-4842-6891-9>

ISBN-13 (electronic): 978-1-4842-6891-9

Copyright © 2021 by Jeff Cicolani

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Aaron Black
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 NY Plaza, New York, NY 10014. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6890-2. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*For Martha, my beautiful and patient wife,
for putting up with random robot parts strewn
about the house, pretty much constantly*

Table of Contents

About the Author	xi
About the Technical Reviewer	xiii
Introduction	xv
Chapter 1: Introduction to Robotics	1
Robotics Basics.....	2
Linux and Robotics	3
Sensors and GPIO	4
Motion and Control	5
Raspberry Pi and Arduino	5
Project Overview	7
The Robot	8
Bill of Materials (BOM).....	9
Summary.....	14
Chapter 2: An Introduction to Raspberry Pi.....	15
Downloading and Installing Raspberry Pi OS.....	15
Raspberry Pi OS with OpenCV	16
The “Hard” Way	17
The “Easy” Way	20
Connecting Raspberry Pi.....	21

TABLE OF CONTENTS

Configuring Your Pi.....	23
Users	28
Going Headless	30
Remote Access	30
Summary.....	38
Chapter 3: A Crash Course in Python.....	39
Python Overview	41
Downloading and Installing Python.....	41
Python Tools	42
The Python Shell.....	43
The Python Editor	44
The Zen of Python.....	47
Writing and Running a Python Program	48
Hello World	49
Basic Structure.....	50
Running a Program.....	53
Programming in Python	54
Variables.....	54
Data Types	55
A Final Note on Variables.....	66
Control Structures	67
Functions	73
Adding Functionality Through Modules	77
Classes	83
Styling	91
Summary.....	93

Chapter 4: Raspberry Pi GPIO	95
Raspberry Pi GPIO	95
Pin Numbering.....	97
Connecting to the Raspberry Pi	98
Limitations of Raspberry Pi's GPIO	99
Accessing GPIO with Python.....	100
Simple Output: LED Example	102
Simple Input	108
Summary.....	118
Chapter 5: Raspberry Pi and Arduino	119
Raspberry Pi's GPIO in Review	120
Real-Time or Near-Real-Time Processing	120
Analog Input	121
Analog Output.....	121
Arduino to the Rescue.....	122
Using Arduino	124
Installing the Arduino IDE	125
Connecting an Arduino	125
Programming Arduino.....	126
Sketches.....	135
A Brief Introduction to the Arduino Language	139
Including Other Files.....	140
Variables and Data Types.....	140
Control Structures	145
Working with Pins.....	153

TABLE OF CONTENTS

Objects and Classes	157
Serial	158
Arduino to Pi and Back Again	162
Pinguino	172
Setting Up the Circuit	173
Summary.....	176
Chapter 6: Driving Motors	179
Motors and Drivers.....	180
Types of Motors	180
Motor Properties.....	184
Motor Drivers.....	187
Working with Motor Controllers	187
Adafruit DC & Stepper Motor HAT	188
L298N Generic Motor Driver	208
Summary.....	219
Chapter 7: Assembling the Robot	221
Assembling the Chassis	222
Choosing a Material.....	222
The Whippersnapper.....	223
Mounting the Electronics	231
Wiring.....	238
Mounting Sensors	241
The Finished Robot	244
Making the Robot Mobile.....	245
Summary.....	258

Chapter 8: Working with Infrared Sensors	259
Infrared Sensors.....	259
Types of IR Sensors	260
Working with IR Sensors.....	264
Connecting an IR Sensor	264
Mounting the IR Sensors	267
The Code.....	269
Understanding PID Control.....	280
Control Loops.....	280
Implementing the PID Controller	283
Summary.....	287
Chapter 9: An Introduction to OpenCV	289
Computer Vision.....	289
OpenCV	291
Selecting a Camera	297
Installing the Camera	298
OpenCV Basics	299
Working with Images.....	300
Capturing Images	302
Image Transformations.....	311
Working with Color	314
Blobs and Blob Detection	320
Ball-Chasing Bot	328
Summary.....	336

TABLE OF CONTENTS

Chapter 10: Conclusion.....337

Types of Robotics 338

Tools..... 338

 Software 339

 Hardware 345

Summary..... 348

Index.....349

About the Author



Jeff Cicolani currently lives in the Austin, Texas, area with his wife, two dogs, and dozen or so robots. He is currently working as an embedded systems engineer, building robotic and automated platforms for an AI (artificial intelligence) company in Austin. His journey to robotics was circuitous, taking him through an odd career path that included systems analysis and design and database programming. In 2012, he joined The Robot Group in Austin, where he joined a group of robotics enthusiasts and began building robots as a hobby. In 2016, he became president of The Robot Group. In this role, he leads the group in their mission to promote STEM (science, technology, engineering, and mathematics) education through robotics. He is currently working to develop a better understanding of advanced robotics through ROS (Robot Operating System) and machine learning.

About the Technical Reviewer



Massimo Nardone has more than 22 years of experience in security, web/mobile development, cloud, and IT architecture. His true IT passions are security and Android.

He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years.

He holds a Master of Science degree in Computing Science from the University of Salerno, Italy.

He has worked as a project manager, software engineer, research engineer, chief security architect, information security manager, PCI/SCADA auditor, and senior lead IT security/cloud/SCADA architect for many years.

Introduction

Robotics does not have to be difficult. In this book, I introduce you to the field of robotics. The journey will be challenging; it's intended to be. But by the end of the book, you will have hands-on exposure to many of the fundamental—and not so fundamental—aspects of robotics. You will work with hardware, assemble and solder a circuit board, write code in two programming languages, install and configure a Linux environment, and work with computer vision. Everything else you do with robots will be an extension of the lessons learned in this book.

Whom This Book Is For

This book is for those who are new to electronics and IoT, those who have never used a Raspberry Pi or Arduino separately, let alone together.

This book is for the hobbyist who is interested in learning a little more about working with robots. Perhaps you've built a few circuits with an Arduino or a custom home entertainment system with a Raspberry Pi, and now you are curious about what goes into building a robot. You will learn how these two devices work together to provide very powerful capabilities.

This book is for the entrepreneur who needs to learn more about technology; someone who doesn't necessarily have the time to read through many different books on Arduino, Raspberry Pi, electronics, or programming; someone who is looking for a broad yet condensed introduction to some of the fundamentals.

This book is also for the student who wants to take their robot-building experience beyond bricks and puzzle-piece programming, someone who wants to work with hardware and software that more closely resembles what they might see in college or in the professional world.

No assumptions are made about experience or background in technology. As you go through the chapters, you may find parts that you are already familiar with and you can skip ahead. But if you are new to these topics, I try to provide you with a quick but easy introduction.

Chapter Overview

You start by learning about the Raspberry Pi and how to work with it. You download and install the Raspbian operating system (OS) and then configure the Pi for our project. The goal is to set up your system to be able to easily access your robot and write your code directly on it.

Once you are able to access your Pi remotely, in Chapter 3, you delve into programming with Python. I show you how to write simple programs on the Raspberry Pi. I also take you beyond the basics and cover some intermediate topics, such as modules and classes. This is one of the longest chapters since there is a lot of material to cover.

From there, you learn how to interface the Raspberry Pi with external electronics, such as sensors and LEDs, through the Pi's GPIO (general-purpose input/output) header. Chapter 4 discusses the different ways of addressing the pins on the header, some of the functionality exposed through the header, and how to use an ultrasonic rangefinder to detect objects. This gets you ready for the next chapter, which introduces the Arduino.

In Chapter 5, you connect the Arduino to the Raspberry Pi. I discuss some of the reasons you want to do this. I show you how to work with the Arduino IDE (integrated development environment) to write programs.

I cover serial communication between the two boards and how to pass information back and forth between them. We do this using the same ultrasonic rangefinder used in the previous chapter.

Chapter 6 has you turning motors with your Raspberry Pi. You use a special board called a *hat*, or *plate*, to control the motors. This is where I introduce another skill that you will inevitably need in robotics: soldering. The header and terminals need to be soldered onto the board that was selected for this purpose. The nice thing about soldering headers and terminal blocks is that it's hard to damage anything, and you will get plenty of practice.

Chapter 7 is where we bring it all together. You build the robot, and I discuss some of the physical characteristics of robotics. I cover some of the design considerations that you will need to keep in mind when you design your own chassis. Although I am listing a specific chassis kit for this project, you do not need to use the same one. In fact, I encourage you to explore other options to find the one that is right for you.

In Chapter 8, I introduce another type of sensor—the IR sensor—and I show you how to use a very common control algorithm called a PID (proportional, integral, and derivative) controller. I talk about the various types of IR sensors and where you want to use them. This chapter also discusses what PID control is and why you want to use it.

Chapter 9 is about computer vision, where you see the true power of the Raspberry Pi. In this chapter, I cover an open source package called OpenCV. By the end of this chapter, your little robot will be chasing a ball around the table.

I leave you with some parting thoughts in Chapter 10. I provide a few tips that I picked up, and I give you a glimpse into my workflow and tools. After that, you will be ready to begin your own adventures in robotics.

Second Edition Notes

This book has been updated for the Raspberry Pi 4. In updating the board, I also updated to the most recent version of the OS. Since the writing of the first edition, there were several changes in downloading the OS, now called Raspberry Pi OS, as well as changes in the installation.

The motor controller libraries have been updated to the newer Adafruit MotorKit libraries. This means there have been updates to the motor driver code. These changes do make the board easier to use.

CHAPTER 1

Introduction to Robotics

The word *robotics* can mean a lot of things. For some people, it is anything that moves by itself; kinetic art is robotics. To other people, robotics means something that is mobile or something that can move itself from place to place. There is actually a field called mobile robotics; automatic vacuum cleaners, such as a Roomba or a Neato, fall into this category. To me, robotics falls somewhere in between kinetic art and mobile robotics.

A *robot* is technology that applies logic to perform a task in an automated manner. This is a fairly broad definition, but robotics is a fairly broad field. It can cover everything from a child's toy to the automatic parallel parking capabilities in some automobiles. We build a small mobile robot in this book.

Many of the principles that you are exposed to in this book are easily transferable to other areas. In fact, we will go through the entire process of building a robot from beginning to end. A little later in this chapter, I go over the project that we will build. At that time, I will provide a list of the parts used in this book. These parts include sensors, drivers, motors, and so forth. You are welcome to use whatever you have on hand because, for the most part, everything we go through in this book can be applied to other projects.

Robotics Basics

I like to tell people who are new to robotics, or are just robotics curious, that a robot consists of three elements:

- The ability to gather data
- The ability to process or do something with the gathered data
- The ability to interact with the environment

In the following chapters, we apply this principle to build a small mobile robot. We will use ultrasonic rangefinders and infrared sensors to gather data about the environment. Specifically, we will identify when there is an object to be avoided, when we are about to drive off the edge of a table, and the contrast between the table and the line that we will follow. Once we have this data, we will apply logic to determine the appropriate response.

We will use Python in a Linux environment to process the information and send commands to our motors. I chose Python as the programming language because it is easy to learn and you don't have to have a complex development environment to build some pretty complex applications.

Our interaction with the environment will be simply to control the speed and direction of motors. This will allow our robot to move about freely on the table or floor. There really isn't much to driving a motor. We will look at two ways of doing it: with a motor driver made for the Raspberry Pi and with a common motor controller.

This book is intended to be challenging. I cover some pretty complex material and I do it quickly. There is no way that I can provide detailed coverage on any of these topics, but I hope to get you to a functional robot by the end of the book. In each chapter, I try to provide you with more resources to follow up on the topics discussed. You will struggle at times; I did and I frequently still do.

Not everyone will be interested in all the subjects. The expectation is that you will expand on the areas that interest you the most outside of this book. Persistence pays off.

At the end of the book, I add a little more challenge. In Chapter 9, we begin leveraging the real power of the Raspberry Pi. We look at computer vision. Specifically, we look at an open source package called OpenCV (CV stands for *computer vision*). It is a common and very powerful collection of utilities that make working with images and video streams very easy. It's also a six-hour build on the most recent version of the Raspberry Pi. To make things a little easier and a lot less time-consuming, I have available for download a version of the operating system with OpenCV already installed. I discuss this more in Chapter 2.

Linux and Robotics

Linux is a Unix-based operating system. It is very popular with programmers and computer scientists because it's simple and straightforward. They seem to enjoy the text-based interface of the terminal. Yet, for many others, including me, Linux can be very challenging. So why in the world would I choose this environment for an introduction-to-robotics book? The answer to that question is threefold.

First, when you work with robotics, you eventually have to confront Linux. That's just a fact. You can do a lot without ever typing a single `sudo` command, but you will have limited capabilities. The `sudo` command stands for *super user do* in Linux. This tells the operating system that you are about to perform a protected function that requires more than general user access. You will learn more about this when we begin working with the Raspberry Pi.

Second, Linux is challenging. As I stated before, this book will challenge you. If you have worked in Linux before, then this reason doesn't apply to you. However, if you are new to Linux, the Raspberry Pi,

or working in a command line, then some of the things that we do will be challenging. And that's good. You're learning something new and it should be a challenge.

Third, and this is by far the most important, the Raspberry Pi uses Linux. Yes, you can install other operating systems on the Pi, but it was designed and intended to use Linux. In fact, the Raspberry Pi has its own flavor of Linux called Raspbian. This is the recommended operating system, so it is what we'll use. One of the nice things about using a prebuilt operating system, besides its ease of use, is many of the tools are already installed and ready to go.

Since we are using Linux, we will use command-line instructions extensively. This is where most new users have problems. Command-line code is entered via a terminal. Raspbian has a Windows-style interface that we will use, but much of it uses the terminal. A terminal window is available in the graphical user interface (GUI), so we will use that. However, when we set up the Pi, we will set it up to boot into terminal mode by default. Getting to the GUI is only a simple `startx` command. All of this is covered in Chapter 2.

Sensors and GPIO

GPIO stands for *general-purpose input/output*. It represents all the various connections to devices. The Raspberry Pi has a lot of GPIO options: HDMI, USB, audio, and so forth. However, when I talk about GPIO in this book, I'm generally referring to the 40-pin GPIO header. This header provides direct access to most of the board's functionality. I discuss this in Chapter 2.

Arduino also has GPIO. In fact, one could argue that Arduino is all GPIO and nothing else. This isn't far from the truth given that all the other connections are there to allow you to communicate with and power the AVR chip at the heart of the Arduino.

All of these headers and GPIO connections are there so we can access sensors outside the boards themselves. A *sensor* is a device that gathers data. There are many different types of sensors, and all serve a purpose. Sensors can be used for detecting light levels, the range to an object, temperature, speed, and so forth. In particular, we will use GPIO headers with an ultrasonic rangefinder and an IR detector.

Motion and Control

One thing that most definitions of a robot have in common is that it needs to be able to move. Sure, you can have a robot that doesn't actually move, but this type of device generally falls under the moniker of *IoT*, the Internet of Things.

There are many ways to add motion to your project. The most common is the use of motors. But you can also use solenoids, air, or water pressure. I discuss motors more in Chapter 6.

Although it is possible to drive a motor directly off a Raspberry Pi or an Arduino board, it is strongly discouraged. Motors tend to draw more current than the processors on the boards can handle. Instead, it is recommended that you use a motor controller. Like motors, motor controllers come in many forms. The motor control board that we will use is accessed through the Raspberry Pi's header. I also discuss how to drive motors with an L298N dual motor controller.

Raspberry Pi and Arduino

We will use a Raspberry Pi (see Figure 1-1) in conjunction with an Arduino (see Figure 1-2) as our robot's processing platform.



Figure 1-1. *Raspberry Pi 3 B+*



Figure 1-2. *Arduino Uno*

The Raspberry Pi is a single-board computer that is about the size of a credit card. Despite its small size, it is a very capable device. The Pi runs a version of Linux that was customized to work on the ARM processor that drives it. This puts a lot of functionality into a small device that is easy to embed into things like robots. But, although it is a great computer,

there are a few places where it does not excel. One area is interfacing with external devices. It can work with sensors and external devices, but the Arduino does this much better.

Arduino is another small processing device that is readily available and easy to use. Unlike a Raspberry Pi, however, it does not have the capacity for a full operating system. Rather than running a microprocessor like the ARM, it uses a different type of chip called a *microcontroller*. The difference is that a microcontroller is specifically designed to interact with sensors, motors, lights, and all kinds of devices. It directly interacts with these external devices. The Pi works through many layers of processing before it ever reaches the pins that a device is connected to.

By combining the Raspberry Pi and the Arduino, we are able to leverage what each does best. The Raspberry Pi offers the high-level processing power of a full computer. Arduino provides the raw control over external devices. The Pi allows us to process a video stream from a simple USB camera, whereas the Arduino allows us to gather the information from the various sensors and apply logic to make sense of all that data and then return concise findings to the Pi.

You will learn more about the Raspberry Pi in Chapter 2. Later on, you will connect an Arduino to the Pi and learn about programming it, as well as how to pass information back and forth between the Arduino and the Pi.

Project Overview

In this book, we will build a small mobile robot. The robot is designed to demonstrate the lessons that you learn in each chapter. However, before we can actually build the robot, we need to cover a lot of material and lay the foundation for future lessons.

The Robot

The robot that we will build is a small two- or four-wheeled autonomous rover. It will be able to detect obstacles and the edge of a table and to follow a line. The chassis that I selected is a four-wheeled robot, but there are other designs suitable for this project (see Figures 1-3 and 1-4).



Figure 1-3. *The front of our robot shows the ultrasonic sensors and Pi T-Cobbler on a breadboard*



Figure 1-4. *The back of our robot shows the Raspberry Pi and motor control board*

Although I provide a list of the parts that I used for the project, you are welcome to use whatever parts you wish. The important thing is that they behave in a similar manner as those I have listed.

Bill of Materials (BOM)

For the most part, I tried to keep the list of materials as generic as possible. There are a couple of items that are vendor specific. I chose them because they provide a lot of functionality and convenience. The DC & Stepper motor controller and the Pi T-Cobbler are from an online retailer called Adafruit, which is a great resource for parts, tutorials, and inspiration. The chassis kit is from an online retailer called ServoCity, which produces many mechanical parts for robotics.

The following are the specialty parts (shown in Figure 1-5) that we use in this book:

- Junior Runt Rover robot chassis from ServoCity.com
- Adafruit DC & Stepper Motor HAT for Raspberry Pi – Mini Kit, PID 2348
- GPIO Stacking Header for Pi A+/B+, Pi 2, Pi 3 – Extra-long 2×20 pins, PID 2223 (allows the use of additional plates and the Cobbler to attach to the breadboard)
- Assembled Pi T-Cobbler Plus (GPIO Breakout) – Pi A+/B+, Pi 2, Pi 3, Pi Zero; PID 2028



Figure 1-5. *Runt Rover chassis parts and the Pi T-Cobbler, ribbon cable, motor control hat, and extended header*

The following parts (shown in Figure 1-6) are fairly generic and can be purchased from most vendors:

- Raspberry Pi 4 Model B – 4 G RAM
- Arduino Uno
- 4 × AA battery holder with on/off switch (powers the motors)
- USB battery pack – 2200 mAh capacity, 5 V 1 A output, PID 1959 (powers the Raspberry Pi)
- Half-size breadboard
- Ultrasonic sensors – HC-SR04

You may want to get a few of these. As you will discover, ultrasonic sensors are unreliable at angles, and it is good to have an array of them. I use at least three on most of my projects.

- A collection of jumper wires (see Figure 1-7)

You need both male-to-male jumpers and male-to-female jumpers. It is a good idea to get them in a number of colors. Black and red are used for powering your devices. A collection of other colors helps you make sense of your circuits. Fortunately, you can get jumpers of all types made out of a multicolored ribbon cable.

- USB cables for your Arduino
- A micro USB cable for your Raspberry Pi
- A common USB phone charger, preferably one for a modern smartphone or tablet that can provide 2 amps of power
- An HDMI TV or computer monitor

Most computer monitors do not have HDMI ports on them. You can get HDMI-to-DVI converters that allow you to use your existing monitor, however.

- A USB keyboard and mouse (I like the Logitech K400 wireless keyboard and touchpad combination, but there are countless options out there)
- A network-connected computer
- Wi-Fi or Ethernet cable for the Pi

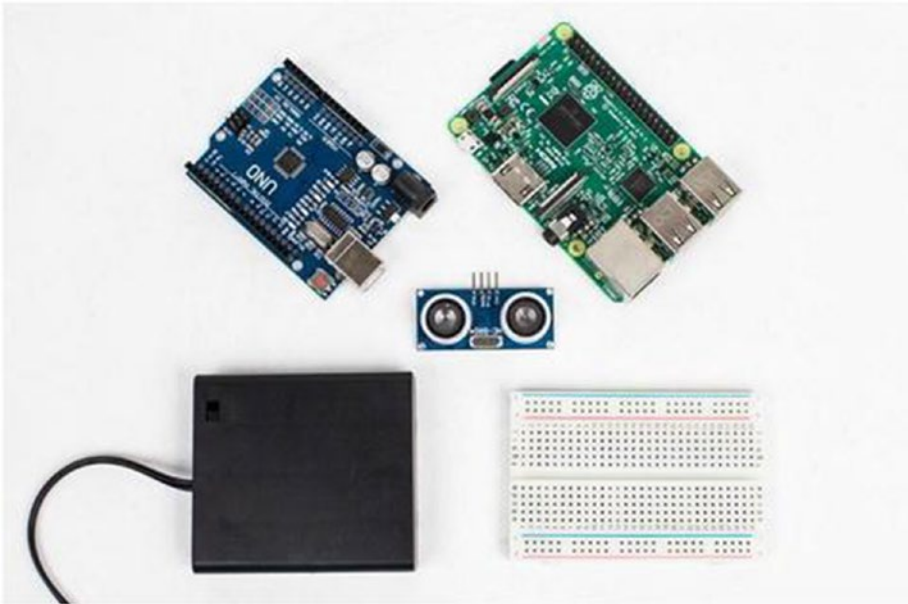


Figure 1-6. Common parts: Raspberry Pi, Arduino Uno, ultrasonic sensor, battery holder, and breadboard



Figure 1-7. Jumpers in ribbon cable form. Pull off what you need

You don't need to get fancy with the monitor and keyboard. Once you read Chapter 2, where we install and configure the Raspberry Pi, you no longer need them. I have a couple of the wireless keyboards because I usually have several projects going at once. For a monitor, I simply use one of my computer monitors with an HDMI-to-DVI adapter.

If you are not using a chassis kit with motors and wheels included, you also need the following parts (see Figure 1-8):

- Hobby gearmotor – 200 RPM (pair)
- Wheel – 65 mm (rubber tire, pair)



Figure 1-8. *DC geared motor and wheels*

If you do not want to use the Adafruit DC & Stepper Motor HAT, you can also use virtually any motor controller, although each one has a different interface and code. A common and fairly popular option is the L298N dual motor controller (see Figure 1-9).

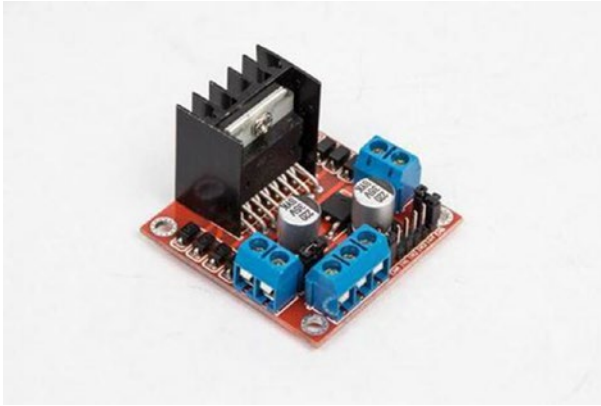


Figure 1-9. *The L298N dual motor controller module comes in numerous varieties, but essentially they work the same*

There are a few other supplies that I keep around because they are used in virtually every project. In Chapter 7, we assemble the robot; you'll also need double-sided foam mounting tape, 4-inch zip ties, and self-adhesive Velcro. As you continue in robotics, you'll find yourself turning to these items a lot. In fact, you may want to stock up on various sizes of zip ties. Trust me.

Summary

Getting started in robotics does not need to be difficult. It is challenging, however. This book is an introduction to a few of the skills that you need to develop if you are to succeed in this field. The robot that we build introduces you to the Raspberry Pi, Linux, Arduino, sensors, and computer vision. These skills easily scale into larger robot and other similar projects.