



Science and Engineering Projects Using the Arduino and Raspberry Pi



Explore STEM Concepts with
Microcomputers

Paul Bradt
David Bradt

apress®

Science and Engineering Projects Using the Arduino and Raspberry Pi

**Explore STEM Concepts
with Microcomputers**

**Paul Bradt
David Bradt**

Apress®

Science and Engineering Projects Using the Arduino and Raspberry Pi: Explore STEM Concepts with Microcomputers

Paul Bradt
Houston, TX, USA

David Bradt
Houston, USA

ISBN-13 (pbk): 978-1-4842-5810-1
<https://doi.org/10.1007/978-1-4842-5811-8>

ISBN-13 (electronic): 978-1-4842-5811-8

Copyright © 2020 by Paul Bradt and David Bradt

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, 233 Spring Street, 6th Floor, New York, NY 10013. 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 rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

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/978-1-4842-5810-1. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*The authors dedicate this book to
all of the Science, Technology, Engineering,
Math (STEM) teachers who guide and
shape the paths of many young minds (including ours)
to question, learn, and utilize new technology to
solve problems. Without these unsung heroes,
the world would not have powerful cell phones,
highly reliable cars, the Internet, and many other
amazing things we routinely take for granted.*

Table of Contents

| | |
|--|---------------|
| About the Authors..... | xi |
| About the Technical Reviewer | xiii |
| Acknowledgments | xv |
| Introduction | xvii |
| Chapter 1: Key Technology Tools | 1 |
| Arduino Basics | 1 |
| Arduino Setup | 2 |
| Ports and Interfaces | 3 |
| Lessons Learned About the Arduino | 6 |
| Raspberry Pi Basics | 8 |
| Raspberry Pi Setup | 10 |
| Lessons Learned About the Raspberry Pi | 15 |
| Basic Electronics Definitions..... | 18 |
| Summary..... | 18 |
| Chapter 2: Data Logging Basics..... | 19 |
| Data Logging with the Arduino..... | 20 |
| Data Logging with the Raspberry Pi | 26 |
| Summary..... | 35 |

TABLE OF CONTENTS

Chapter 3: Physics and Mathematics Basics37

 Temperature..... 37

 Force 38

 Pressure..... 39

 Basic Concept of Algebra 40

 Statistical Concepts 41

 Direct Compared to Inferred Measurements..... 41

 Summary..... 42

Chapter 4: Simple Science and Engineering Projects.....43

 Buoyancy of Air 43

 Arduino Buoyancy of Air Version..... 45

 Raspberry Pi Buoyancy of Air Version..... 52

 Buoyancy Recap 59

 Demonstrating Pressure 59

 Pressure/Force Recap 65

 Capturing Counts 65

 Counts Recap 70

 Summary..... 70

Chapter 5: Advanced Physics and Mathematics for Science and Engineering.....71

 Basics Terms of Calculus 72

 How Heat Transfer Works..... 72

 Conduction Heat Transfer 72

 Convection Heat Transfer..... 74

 Radiation Heat Transfer 76

 All Three Heat Transfer Mechanisms Work Together! 77

| | |
|---|-----------|
| Mass | 78 |
| Velocity and Acceleration | 78 |
| Inertia | 81 |
| Momentum | 81 |
| Friction | 81 |
| More Advanced Aspects of Calculus | 83 |
| Summary | 84 |
| Chapter 6: Time/Condition-Dependent Projects | 85 |
| Conduction Heat Transfer Through an Aluminum Rod | 85 |
| Ensure Consistency in Temperature Sensor Readings | 90 |
| Aluminum Rod Conduction Heat Transfer Recap | 94 |
| Conduction Heat Transfer Through a Window | 94 |
| Window Conduction Heat Transfer Recap | 103 |
| Convection Heat Transfer | 103 |
| Convection Heat Transfer Recap | 108 |
| Zero Gravity Demonstration | 108 |
| Zero Gravity Recap | 118 |
| Measuring Frictional Force Projects | 118 |
| Arduino Frictional Force Project | 119 |
| Operational Schematic | 121 |
| Arduino Frictional Force Recap | 124 |
| Raspberry Pi Frictional Force Project | 124 |
| Raspberry Pi Frictional Force Recap | 131 |
| Acceleration Projects | 131 |
| Acceleration Direct to Computer | 131 |

TABLE OF CONTENTS

| | |
|---|------------|
| Acceleration with Computer Recap..... | 139 |
| Acceleration Measurement Without a Computer | 139 |
| Acceleration Without Computer Recap | 148 |
| Summary..... | 148 |
| Chapter 7: Light and Imaging Projects | 149 |
| Radiation Heat Transfer | 149 |
| Analysis of Heat Transfer | 161 |
| Radiation Heat Transfer Recap | 162 |
| Astrophotography with the Raspberry Pi Camera..... | 162 |
| Assembling the Meade ETX-60AT and Raspberry Pi | 165 |
| Astrophotography Meade ETX-60AT Setup Recap..... | 169 |
| Assembling the 4 1/2-Inch Reflector Telescope and the Raspberry Pi | 169 |
| Components Needed to Assemble the Raspberry Pi 3 Mounting System to the 4 1/2-Inch Telescope | 172 |
| Reflector Telescope Setup Recap | 175 |
| Basic Raspistill Previewing an Image with the Terminal Command Line..... | 175 |
| Using Raspistill to Capture an Image..... | 177 |
| More Advanced Raspistill Input Without a Keyboard | 177 |
| Raspistill Image Capture Recap..... | 179 |
| Astrophotography Raspberry Pi Python GUI..... | 179 |
| Initiating the GUI | 183 |
| PI_SN003 Raspberry PI GUI Recap | 184 |
| Assembling the Raspberry Pi and Touchscreen in the Case | 184 |
| Raspberry Pi, Touchscreen, and Case..... | 184 |
| Modification of the Case and Assembly..... | 185 |
| Components and Assembly of the Raspberry Pi Case Recap | 188 |

TABLE OF CONTENTS

| | |
|--|------------|
| Camera Modifications, Camera Case, and Power Cables..... | 188 |
| Camera Modifications..... | 188 |
| Building the Camera Case | 189 |
| Final Assembly of the Camera in the Case | 196 |
| Power Cord Combination | 197 |
| Camera, Camera Case, and Power Cord Assembly Recap..... | 197 |
| Building the Shelf for the Meade ETX-60AT..... | 197 |
| Shelf Components and Assembly Recap | 203 |
| Helpful Hints Using the Telescope and Raspberry Pi | 203 |
| Lessons Learned Recap | 207 |
| Example Images and Enhancing Them Using a Video Capture GUI..... | 207 |
| Example Images Taken with the Upgraded Meade ETX-60AT Astrophotography System | 208 |
| Recap of Example Images and Enhancement Techniques | 217 |
| Summary..... | 217 |
| Appendix: Reference Material | 219 |
| Soldering Safety..... | 219 |
| General Shop Safety | 220 |
| Manufacturing Techniques..... | 220 |
| Soldering | 221 |
| Basic Arduino and Raspberry Pi Python Commands | 222 |
| 3D Printing..... | 224 |
| Computer-Aided Design Options | 225 |
| Project Management for Engineering | 226 |
| Decision Analysis for Engineering..... | 226 |
| Thermal Conductivity Coefficients | 227 |
| Coefficients of Friction | 228 |

TABLE OF CONTENTS

Astronomy Terms228

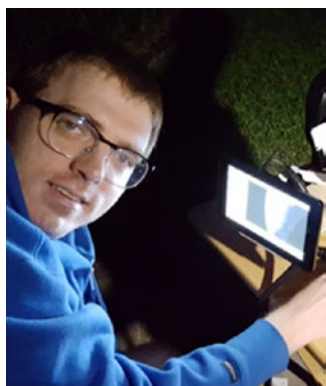
Specifications of the Meade ETX-60AT229

 Setup, Updates, and Repairs231

Helpful Books.....232

Index.....233

About the Authors



Paul Bradt has a BS in Computer Science from the University of Houston–Clear Lake. He currently owns a small business and writes books, develops code, and does IT support work. He has experimented with the Arduino and Raspberry Pi system and believes them to be excellent tools for developing an understanding of electronic components and hardware interaction in integrated systems. He believes they are very useful as a teaching aid in learning computer programming, science, and engineering. He likes to perform sophisticated troubleshooting of computer problems and has found that online resources can be a great help for novice users to get their experiments operating quickly and effectively.



David Bradt has a BS in Mechanical Engineering from New Mexico State University with many years of experience in the aerospace industry and in the petrochemical industry. He enjoys building and designing devices to measure and control systems. He has found the Arduino and Raspberry Pi to be incredibly powerful little devices that with a little bit of work can do many different tasks. He is a big fan of *Star Trek: The Original Series* and astronomy.

About the Technical Reviewer

Sri Manikanta Palakollu is an undergraduate student pursuing his bachelor's degree in Computer Science and Engineering at SICET under JNTUH. He is a founder of the OpenStack Developer Community in his college. He started his journey as a competitive programmer. He always loves to solve problems that are related to the data science field. His interests include data science, app development, web development, cybersecurity, and technical writing. He has published many articles on data science, machine learning, programming, and cybersecurity in publications like Hacker Noon, freeCodeCamp, Noteworthy, and DDI through the Medium platform.

Acknowledgments

This book would not be possible without the authors' gaining early technical insight regarding the Raspberry Pi and Arduino from others. Jared Brank and Dennis Pate provided a lot of basic information, key insights, and Arduino hardware early in the process. The authors thank the following individuals who listened to them on many occasions and provided help, insight, and inspiration with their own experiences with the Raspberry Pi and other projects: Jeff Duneheew, Todd Franke, and Fitz Walker. Additionally, significant assistance with 3D printing was provided by Mitch Long and David Thoerig.

Producing this book would not have been possible without the excellent help and guidance regarding scope and early editorial reviews by Joanna Opaskar and Ed Weisblatt. The authors also utilized many ideas from Andrew Bradt and Laura Brank's science fair experience. Most important was the support and advice from Andrea Bradt.

Introduction

The authors' journey developing this book started in 2013 when they discovered the Arduino microcontroller. It is interesting how something big really starts with one step as they found the Arduino incredibly powerful. Users are able to program it with computer code, and then it executes its instructions for as long as it has power. The authors started evaluating various applications of the Arduino around the house and in their hobby endeavors. In 2017, they started experimenting with the Raspberry Pi minicomputer which enables users to take projects to a whole new level with a low-cost computer that interfaces with sensors. Since a Raspberry Pi is very affordable, a real computer can now be dedicated to operating a system permanently. While requiring some technical steps to set up, both of these tools can be used to gather data, automate tasks, and provide a lot of fun. The authors found it very satisfying to watch a device do several tasks, especially when they set it up. This book chronicles some science and engineering projects the authors developed over the past few years and provides helpful hints, along with a few things to avoid.

There are two primary areas of focus or goals of this book. The first goal is to help the reader explore the Arduino and Raspberry Pi. The second goal is exploring science and engineering in interesting and fun ways.

The projects and concepts in this book are meant to accomplish the first goal by providing information to get an Arduino or Raspberry Pi system set up, running, and ready to capture data. The text provides enough detail for users with average assembly or electrical skills to complete them. Additionally, the goals of learning are to gain knowledge and skills. When the reader engages in a project that requires them to try new things, it reinforces how they learn and gain confidence and encourages them to try even more complex tools and techniques.

INTRODUCTION

The second goal is exploring concepts of STEM (Science, Technology, Engineering, Mathematics) and working through examples to demonstrate basic scientific and engineering concepts. Finally, the authors provide some detail on the mathematics needed to understand and explain the science demonstrated.

Science and engineering provide critical skill sets for the modern world that can be used in everyday life. People use these skills to develop the technology that the modern world relies on. This book can establish these skill sets for a fruitful and rewarding career.

The authors hope this book inspires the reader to expand and explore their own STEM projects by including a wide range from beginner to advanced. From these examples, the reader can learn many techniques, tools, and technologies and apply them beyond the ones listed here; but first, the authors introduce STEM.

What Is STEM?

STEM (Science, Technology, Engineering, Mathematics) is a program based on educating students in science, technology, engineering, and mathematics in an integrated, interdisciplinary approach to learning.

School systems today strive to improve education in STEM. This goal is an area where educators can use outside help developing and improving students' knowledge when they actively contribute, design, and build hands-on projects. In many ways, the young mind is excited and motivated building projects. They develop an in-depth understanding of what is required and how it works. The authors believe this is the best way to learn and remember these concepts, which results in a solid STEM foundation for students.

A question not often understood is how the scientific method is different from an engineering approach. Understanding the difference between science and engineering can be seen in the original *Star Trek* series.

Mr. Spock was the science officer, and Montgomery Scott (Scotty) was the chief engineer. Their jobs and how they approached new scenarios or problems really provide a great explanation about the differences and similarities between science and engineering. Let's examine some examples.

Mr. Spock used the term fascinating when describing a new event or phenomenon. The role of science is to expand knowledge and investigate new events. This fascination with new and unique areas is key for a scientist. Scotty, the engineer, on the other hand always had to fix the warp engines, the transporter, or some other critical system. The normal role of an engineer is to develop and implement solutions to problems. In one of the episodes, Scotty indicated he would rather read his engineering journals to learn about how others solved problems than go on shore leave!

Science

Researchers use the scientific method as a tool to understand questions in their area of interest. Based on the information they have initially, they develop a hypothesis and then methods to test the validity of the hypothesis. When sufficient test data are gathered and analyzed, the researcher either accepts or rejects the hypothesis. In many cases, positive or negative results point to the next step or direction of exploration and contribute to the general body of scientific and engineering knowledge.

Engineering

The primary goal of engineering is to evaluate alternatives and choose the optimal solution to minimize or eliminate specific problems or issues. Solutions are not necessarily new, but may be repurposed concepts applied to different problem areas. Other aspects of engineering include

INTRODUCTION

planning the work, selecting components to meet requirements, and following through on managing and completing a project. Often projects or systems fail because the planning, scheduling, and logistics of activities are not adequately engineered for an optimum solution. These skills are important and necessary in any job.

Science and engineering use many of the same tools and techniques, but it is important to understand the distinction between scientific experimentation and the engineering process of developing optimal solutions. For one thing, they both use the language of mathematics to describe percentages, results, probability, and other physical parameters. However, science's goal is to expand knowledge which is different than engineering's goal of selecting an optimum solution and proceeding with solving the problem. One other difference is a scientific test often gains new knowledge, whereas an engineering test often demonstrates how a system performs a function. In many ways, they are synergistic as science often provides new tools and ideas for engineers to use to solve problems.

In the authors' minds, the roots of some key technological advancements that exist now can be traced back 50 years to the original *Star Trek* TV show. For example, in the show, they used tricorders to gather data about aliens, equipment failures, medical problems, and a host of other out of this world challenges. They had communicators that allowed them to contact crewmembers all over alien worlds. Finally, they had the replicators that allowed them to produce any type of food they desired. Today we don't have tricorders, but we do have some examples that 50 years ago would have been amazing. Today there are personal computers, cell phones, 3D printing, and incredible sensors based on the early transistors of the 1960s. The Arduino and the Raspberry Pi, two examples of new technology, can be built into devices similar to the incredibly versatile *Star Trek* tricorders.

Both Mr. Spock and Scotty realized they needed each other (science and engineering) to accomplish the goals of exploration and keep the Enterprise flying safely through space. In today's complex world, integrating science and engineering is key to researching problems and developing solutions.

In the following chapters, the authors will demonstrate all of the components of STEM needed to research scientific questions, use new technology (Arduino and Raspberry Pi), employ engineering techniques, and use mathematics to quantify the scientific data. As *Star Trek* boldly went forth to explore new worlds, the authors hope the students of today do the same!

CHAPTER 1

Key Technology Tools

This chapter will highlight some of the basics about the Arduino and the Raspberry Pi. It will help the reader get started if they are unfamiliar with these powerful devices. It is amazing what these devices can do and this chapter provides some basic aspects for getting them set up to run.

Arduino Basics

The Arduino is a powerful microcontroller that is ready to program and acts as an intermediary device between a personal computer and various sensors. It is relatively new technology that is a great tool for gaining insight into physical properties and other scientific parameters.

The Arduino board was first developed in Italy in 2004 as a tool to help train students in programming. It is an open source tool and as such has developed a large base of helpful web sites and user groups. It represents a breakthrough as an easy-to-use, relatively inexpensive, programmable interface between a computer and various sensors. The software development package and all of the online resources help make this an ideal data logging tool for science fair/college projects.

The Arduino, Adafruit, SparkFun, Hacktronics, and other web sites are great places to start. There are also several introductory books to help the researcher get started using this device. *Getting started with Arduino* by Banzi is a very good beginner's book on Arduino.

Other sources of information for the Arduino novice are maker faires and user group activities.

There are several versions and sizes, but for the projects in this book, the Arduino Uno and the Integrated Development Environment (IDE) version 1.89 were utilized. Figure 1-1 shows an example of the Arduino Uno. The authors recommend for the person unfamiliar with Arduinos to use an official version and not a clone. The authors have never experienced a problem with an official Arduino, but there are many clones, and the authors have experienced problems with one of them.



Figure 1-1. *Arduino Uno*

Arduino Setup

Setting up an Arduino is relatively straightforward; the reader should follow these basic steps to get the device running:

1. The Arduino is connected to a computer via a USB connection to the input port (see Figure 1-3).
2. Load code using the IDE (see Figure 1-2).
3. Open the serial monitor to get data.

These steps sound basic, and after the reader completes these steps a few times, they will see how easy it is to connect and run an Arduino. In many cases,

the challenges occur with the code. If the reader is copying code from a source, it is important to type it in exactly as it looks. Even then there could be errors, but that is part of the adventure, and it's very rewarding when the code runs.



Figure 1-2. *Arduino IDE*

Ports and Interfaces

Figure 1-3 shows the main ports of the Arduino Uno.

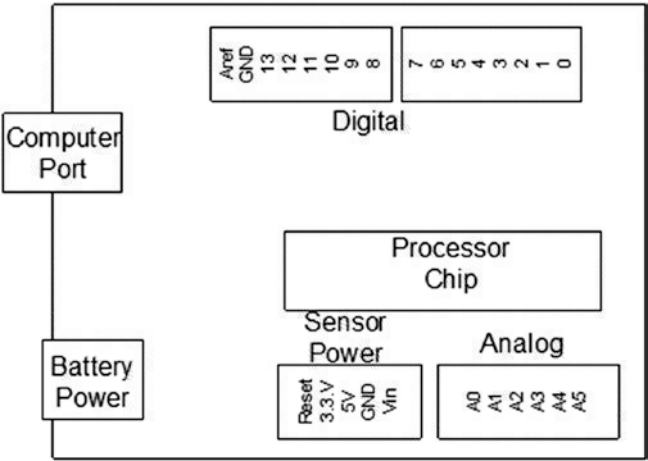


Figure 1-3. *Arduino Ports*

There are five primary port groupings that are used to connect to the Arduino:

Computer port: This is the primary port that is directly connected to the computer. It is a micro-USB port that powers and enables the user to upload the sketches or programs to the Arduino.

Battery power port: This port allows an Arduino to be unplugged from a computer and use battery power to operate. A standard wall power supply that provides 9–12 V DC can also be used.

Sensor power ports: These plug connections provide 3.3 V and 5 V DC power. There is also a reset connection and input voltage connection.

Analog device ports: These connections are for analog inputs.

Digital device ports: These are for digital inputs and outputs.

IDE (Integrated Development Environment):

The IDE is the program that is used to develop the code. It is the programming tool that runs on a computer and has features to help the developer write code. The IDE tool must be downloaded from the Arduino web site.

Sketch: The code that runs on an Arduino is called a sketch. Once the code is developed in the IDE, it is uploaded to the Arduino.

Libraries: These are code modules that are installed on the Arduino and called up by the program when needed. Libraries add a lot of functionality and do not require any additional coding.

There are other components and hardware that can be used with the Arduino:

Shields/breakout boards: These are add-on boards that are either inserted into the standard Arduino board ports or connected via wires.

Sensors: A sensor is a device that senses some type of data. It can be used to directly measure a physical aspect, or it can be used with some mathematics to infer a physical measurement.

Effectors: An effector imparts some change in the physical world when activated. Motors, solenoids, and servos are some examples.

LCDs: Liquid Crystal Display can be used to show data.

LEDs: Light-Emitting Diodes or other incandescent lights can also indicate an event has occurred.

Lessons Learned About the Arduino

The Arduino is relatively easy to use, but the authors found there are a few key points that will help when using this powerful device:

- Each Arduino attaches to a specific com port. The port may have to be changed or selected in the tools tab under “port” to get the IDE to recognize the Arduino.
- If the code is being pasted into the IDE, do not copy from Microsoft Word or another word processor. First, put it in a text editor such as Notepad, Notepad++, or some other C/C++ IDE editor and then copy it from there. Important note: Notepad and Notepad++ are not development tools like the IDE. One other very important item of note is when the code was transcribed into the book format some of the code text that must be on one line may show up on two lines in this book. The authors have tried their best to highlight the code that should be on one line in the IDE by bolding it in the Listing. Please contact the authors if there are questions at contact@pdanalytic.com.
- It is a good idea to test the devices with a basic program to be sure they work, before moving to a more complex program.
- If the final code is complex, get each piece of code working before adding more modules. This way, it is easier to find the module where the problem is located.
- The authors recommend for long timing events or complex programs to not use the “delay command,” because it locks the Arduino and prevents it from doing anything else. Instead, use the “milli command” that

tracks time intervals between events and still allows other actions to occur. The `milli` code might be a little more complex, but it allows the Arduino to perform other functions simultaneously. Using the `delay` command for short events or simple programs like the ones in this book, such as a switch debounce, is recommended.

- A feature built into the Arduino IDE is the “auto-format command.” It can be found under the tools tab or using “Alt+T.” This command helps identify missing items and also helps organize the code for improved readability.
- One more key aspect of Arduino coding is the “loop command.” There are a few different types, but common ones such as “void loop” and the “for command” perform several operations and then repeat them.
- Check the wiring twice before applying power. It can be difficult to see which port a wire is plugged into when there are several wires.
- It is hard to know what code is on an Arduino. One easy way that helps determine what is loaded on an Arduino is saving code with a descriptive name, date, and even time information. This helps programmers who may need to go back to a previous code version.
- One other very helpful trick is to put the descriptive name of the code on a piece of tape and stick it on top of the computer port. This helps when working on, or programing, several different Arduinos.
- One of the advantages of the Arduino is that once it is programmed, it remembers the code. When a power source is plugged into the battery power port, it will

operate the Arduino. According to the Arduino web site, any power source that can supply 9–12 V DC, 250 mA, or more will work. The plug must be 2.1 mm with the center pin providing positive voltage and the exterior of the plug the negative terminal. Some power supplies do not deliver enough current or do not provide stable power. If an Arduino is behaving strangely, try a different power supply.

- Some programs need special ways to use and communicate with the Arduino. To do this, the reader should understand these special connection ports on the Arduino Uno: analog A5 is the SCL (Clock port) and A4 is the SDA (Data port).

Raspberry Pi Basics

The Raspberry Pi 3 is a powerful minicomputer. This piece of technology comes with a lot of features like any other modern computer. It is an experimental/hobbyist device developed around 2011 in the United Kingdom to teach programming. For its low cost, it has many capabilities and allows the user to configure it in many ways. There are several models on the market. For this book, the authors choose the Raspberry Pi 3 Model B V1.2 (Figures 1-4 and 1-5). There is a new Raspberry Pi 4 that was recently released that has more features. The authors researched the setup and use of the Raspberry Pi 4, and it appears to be the same as the Raspberry Pi 3. We believe these projects will work the same if you have a Raspberry Pi 4.