Contents

Introduction xix

Part I Arduino Engineering Basics 1

Chapter 1 Getting Up and Blinking with the Arduino 3
- Exploring the Arduino Ecosystem 4
  - Arduino Functionality 4
  - Atmel Microcontroller 6
  - Programming Interfaces 6
  - General I/O and ADCs 7
  - Power Supplies 7
- Arduino Boards 8
- Creating Your First Program 13
  - Downloading and Installing the Arduino IDE 13
  - Running the IDE and Connecting to the Arduino 14
  - Breaking Down Your First Program 16
- Summary 18

Chapter 2 Digital Inputs, Outputs, and Pulse-Width Modulation 19
- Digital Outputs 20
  - Wiring Up an LED and Using Breadboards 20
  - Working with Breadboards 21
  - Wiring LEDs 22
  - Programming Digital Outputs 24
  - Using For Loops 25
- Pulse-Width Modulation with analogWrite() 27
- Reading Digital Inputs 29
  - Reading Digital Inputs with Pulldown Resistors 29
  - Working with “Bouncy” Buttons 32
- Building a Controllable RGB LED Nightlight 35
- Summary 39
# Contents

## Chapter 3  Reading Analog Sensors  41
- Understanding Analog and Digital Signals  42
  - Comparing Analog and Digital Signals  43
  - Converting an Analog Signal to a Digital One  44
- Reading Analog Sensors with the Arduino: analogRead()  45
  - Reading a Potentiometer  45
  - Using Analog Sensors  50
    - Working with Analog Sensors to Sense Temperature  52
- Using Variable Resistors to Make Your Own Analog Sensors  54
  - Using Resistive Voltage Dividers  55
  - Using Analog Inputs to Control Analog Outputs  56
- Summary  59

## Part II  Controlling Your Environment  61

## Chapter 4  Using Transistors and Driving Motors  63
- Driving DC Motors  65
  - Handling High-Current Inductive Loads  65
    - Using Transistors as Switches  66
    - Using Protection Diodes  67
    - Using a Secondary Power Source  68
    - Wiring the Motor  68
  - Controlling Motor Speed with PWM  70
  - Using an H-Bridge to Control DC Motor Direction  72
    - Building an H-bridge Circuit  73
    - Operating an H-bridge Circuit  76
- Driving Servo Motors  80
  - Understanding the Difference Between Continuous Rotation and Standard Servos  80
  - Understanding Servo Control  80
  - Controlling a Servo  85
- Building a Sweeping Distance Sensor  86
- Summary  90

## Chapter 5  Making Sounds  91
- Understanding How Speakers Work  92
  - The Properties of Sound  92
  - How a Speaker Produces Sound  94
- Using tone() to Make Sounds  95
  - Including a Definition File  95
  - Wiring the Speaker  96
  - Making Sound Sequences  99
    - Using Arrays  99
    - Making Note and Duration Arrays  100
    - Completing the Program  101
  - Understanding the Limitations of the tone() Function  102
- Building a Micro Piano  102
- Summary  105
## Chapter 6  USB and Serial Communication  
### Understanding the Arduino's Serial Communication Capabilities  
- Arduino Boards with an Internal or External FTDI USB-to-Serial Converter  
- Arduino Boards with a Secondary USB-Capable ATmega MCU Emulating a Serial Converter  
- Arduino Boards with a Single USB-Capable MCU  
- Arduino Boards with USB-Host Capabilities  
- Listening to the Arduino  
  - Using print Statements  
  - Using Special Characters  
  - Changing Data Type Representations  
- Talking to the Arduino  
  - Reading Information from a Computer or Other Serial Device  
  - Telling the Arduino to Echo Incoming Data  
  - Understanding the Differences Between Chars and Ints  
  - Sending Single Characters to Control an LED  
  - Sending Lists of Values to Control an RGB LED  
- Talking to a Desktop App  
  - Talking to Processing  
  - Installing Processing  
  - Controlling a Processing Sketch from Your Arduino  
  - Sending Data from Processing to Your Arduino  
- Learning Special Tricks with the Arduino Leonardo (and Other 32U4-Based Arduinos)  
  - Emulating a Keyboard  
  - Typing Data into the Computer  
  - Commanding Your Computer to Do Your Bidding  
  - Emulating a Mouse  
- Summary

## Chapter 7  Shift Registers  
### Understanding Shift Registers  
- Sending Parallel and Serial Data  
- Working with the 74HC595 Shift Register  
  - Understanding the Shift Register Pin Functions  
  - Understanding How the Shift Register Works  
- Shifting Serial Data from the Arduino  
- Converting Between Binary and Decimal Formats  
- Controlling Light Animations with a Shift Register  
  - Building a “Light Rider”  
  - Responding to Inputs with an LED Bar Graph  
- Summary
Part III Communication Interfaces 161

Chapter 8 The I²C Bus 163
History of the I²C Bus 164
I²C Hardware Design 164
  Communication Scheme and ID Numbers 165
  Hardware Requirements and Pull-Up Resistors 167
Communicating with an I²C Temperature Probe 167
  Setting Up the Hardware 168
  Referencing the Datasheet 169
  Writing the Software 171
Combining Shift Registers, Serial Communication, and I²C Communications 173
  Building the Hardware for a Temperature Monitoring System 173
  Modifying the Embedded Program 174
  Writing the Processing Sketch 177
Summary 180

Chapter 9 The SPI Bus 181
Overview of the SPI Bus 182
SPI Hardware and Communication Design 183
  Hardware Configuration 184
  Communication Scheme 184
Comparing SPI to I²C 185
Communicating with an SPI Digital Potentiometer 185
  Gathering Information from the Datasheet 186
  Setting Up the Hardware 189
  Writing the Software 190
Creating an Audiovisual Display Using SPI Digital Potentiometers 193
  Setting Up the Hardware 194
  Modifying the Software 195
Summary 197

Chapter 10 Interfacing with Liquid Crystal Displays 199
Setting Up the LCD 200
Using the LiquidCrystal Library to Write to the LCD 203
  Adding Text to the Display 204
  Creating Special Characters and Animations 206
Building a Personal Thermostat 209
  Setting Up the Hardware 210
  Displaying Data on the LCD 211
  Adjusting the Set Point with a Button 213
  Adding an Audible Warning and a Fan 214
  Bringing It All Together: The Complete Program 215
  Taking This Project to the Next Level 219
Summary 219
## Chapter 11  Wireless Communication with XBee Radios

Understanding XBee Wireless Communication

XBee Radios

The XBee Radio Shield and Serial Connections

3.3V Regulator

Logic Level Shifting

Associate LED and RSSI LED

UART Selection Jumper or Switch

Hardware vs. Software Serial UART Connection Option

### Configuring Your XBees

Configuring via a Shield or a USB Adapter

Programming Option 1: Using the Uno as a Programmer (Not Recommended)

Programming Option 2: Using the SparkFun USB Explorer (Recommended)

Choosing Your XBee Settings and Connecting Your XBee to Your Host Computer

Configuring Your XBee with X-CTU

Configuring Your XBee with a Serial Terminal

### Talking with Your Computer Wirelessly

Powering Your Remote Arduino

USB with a Computer or a 5V Wall Adapter

Batteries

Wall Power Adapters

Revisiting the Serial Examples: Controlling Processing with a Potentiometer

Revisiting the Serial Examples: Controlling an RGB LED

### Talking with Another Arduino: Building a Wireless Doorbell

System Design

Transmitter Hardware

Receiver Hardware

Transmitter Software

Receiver Software

Summary

---

## Part IV  Advanced Topics and Projects

## Chapter 12  Hardware and Timer Interrupts

Using Hardware Interrupts

Knowing the Tradeoffs Between Polling and Interrupting

Ease of Implementation (Software)

Ease of Implementation (Hardware)

Multitasking

Acquisition Accuracy

Understanding the Arduino’s Hardware Interrupt Capabilities
Building and Testing a Hardware-Debounced Button Interrupt Circuit 262
Creating a Hardware-Debouncing Circuit 262
Assembling the Complete Test Circuit 267
Writing the Software 267
Using Timer Interrupts 270
Understanding Timer Interrupts 270
Getting the Library 270
Executing Two Tasks Simultaneously(ish) 271
Building an Interrupt-Driven Sound Machine 272
Sound Machine Hardware 272
Sound Machine Software 273
Summary 275

**Chapter 13  Data Logging with SD Cards** 277

Getting Ready for Data Logging 278
  Formatting Data with CSV Files 279
  Preparing an SD Card for Data Logging 279
Interfacing the Arduino with an SD Card 284
  SD Card Shields 284
  SD Card SPI Interface 288
  Writing to an SD Card 289
  Reading from an SD Card 293
Using a Real-Time Clock 297
  Understanding Real-Time Clocks 298
    Using the DS1307 Real-Time Clock 298
    Using the RTC Arduino Third-Party Library 299
  Using the Real-Time Clock 300
    Installing the RTC and SD Card Modules 300
    Updating the Software 301
Building an Entrance Logger 305
  Logger Hardware 306
  Logger Software 307
  Data Analysis 311
Summary 312

**Chapter 14  Connecting Your Arduino to the Internet** 313

The Web, the Arduino, and You 314
  Networking Lingo 314
    IP Address 314
    Network Address Translation 315
    MAC Address 316
    HTML 316
    HTTP 316
    GET/POST 316
    DHCP 316
    DNS 317
Contents xvii

Clients and Servers 317
Networking Your Arduino 317
Controlling Your Arduino from the Web 318
Setting Up the I/O Control Hardware 318
Designing a Simple Web Page 318
Writing an Arduino Server Sketch 320
    Connecting to the Network and Retrieving an IP via DHCP 321
    Replying to a Client Response 321
    Putting It Together: Web Server Sketch 322
Controlling Your Arduino via the Network 326
    Controlling Your Arduino over the Local Network 326
    Using Port Forwarding to Control your Arduino from Anywhere 327
Sending Live Data to a Graphing Service 329
    Building a Live Data Feed on Xively 330
    Creating a Xively Account 330
    Creating a Data Feed 330
    Installing the Xively and HttpClient Libraries 331
Wiring Up Your Arduino 332
    Configuring the Xively Sketch and Running the Code 332
Displaying Data on the Web 335
    Adding Feed Components 336
        Adding an Analog Temperature Sensor 336
        Adding Additional Sensor Readings to the Datastream 336
Summary 339

Appendix Deciphering the ATmega Datasheet and Arduino Schematics 341
Reading Datasheets 341
    Breaking Down a Datasheet 341
    Understanding Component Pin-outs 344
Understanding the Arduino Schematic 345

Index 349