
Table of Contents

Preface	xi
1. Getting Started	1
1.1 Installing the Integrated Development Environment (IDE)	4
1.2 Setting Up the Arduino Board	8
1.3 Using the Integrated Development Environment (IDE) to Prepare an Arduino Sketch	10
1.4 Uploading and Running the Blink Sketch	13
1.5 Creating and Saving a Sketch	14
1.6 Using Arduino	17
2. Making the Sketch Do Your Bidding	23
2.1 Structuring an Arduino Program	24
2.2 Using Simple Primitive Types (Variables)	25
2.3 Using Floating-Point Numbers	27
2.4 Working with Groups of Values	29
2.5 Using Arduino String Functionality	32
2.6 Using C Character Strings	37
2.7 Splitting Comma-Separated Text into Groups	38
2.8 Converting a Number to a String	41
2.9 Converting a String to a Number	43
2.10 Structuring Your Code into Functional Blocks	45
2.11 Returning More Than One Value from a Function	49
2.12 Taking Actions Based on Conditions	52
2.13 Repeating a Sequence of Statements	53
2.14 Repeating Statements with a Counter	55
2.15 Breaking Out of Loops	58
2.16 Taking a Variety of Actions Based on a Single Variable	59
2.17 Comparing Character and Numeric Values	61
2.18 Comparing Strings	63
2.19 Performing Logical Comparisons	64

2.20	Performing Bitwise Operations	65
2.21	Combining Operations and Assignment	68
3.	Using Mathematical Operators	69
3.1	Adding, Subtracting, Multiplying, and Dividing	69
3.2	Incrementing and Decrementing Values	70
3.3	Finding the Remainder After Dividing Two Values	71
3.4	Determining the Absolute Value	72
3.5	Constraining a Number to a Range of Values	73
3.6	Finding the Minimum or Maximum of Some Values	74
3.7	Raising a Number to a Power	75
3.8	Taking the Square Root	76
3.9	Rounding Floating-Point Numbers Up and Down	76
3.10	Using Trigonometric Functions	77
3.11	Generating Random Numbers	78
3.12	Setting and Reading Bits	80
3.13	Shifting Bits	84
3.14	Extracting High and Low Bytes in an int or long	85
3.15	Forming an int or long from High and Low Bytes	87
4.	Serial Communications	89
4.1	Sending Debug Information from Arduino to Your Computer	94
4.2	Sending Formatted Text and Numeric Data from Arduino	97
4.3	Receiving Serial Data in Arduino	100
4.4	Sending Multiple Text Fields from Arduino in a Single Message	105
4.5	Receiving Multiple Text Fields in a Single Message in Arduino	111
4.6	Sending Binary Data from Arduino	114
4.7	Receiving Binary Data from Arduino on a Computer	118
4.8	Sending Binary Values from Processing to Arduino	120
4.9	Sending the Value of Multiple Arduino Pins	122
4.10	How to Move the Mouse Cursor on a PC or Mac	125
4.11	Controlling Google Earth Using Arduino	130
4.12	Logging Arduino Data to a File on Your Computer	135
4.13	Sending Data to Two Serial Devices at the Same Time	138
4.14	Receiving Serial Data from Two Devices at the Same Time	141
4.15	Setting Up Processing on Your Computer to Send and Receive Serial Data	145
5.	Simple Digital and Analog Input	147
5.1	Using a Switch	150
5.2	Using a Switch Without External Resistors	154
5.3	Reliably Detecting the Closing of a Switch	155
5.4	Determining How Long a Switch Is Pressed	158

5.5	Reading a Keypad	163
5.6	Reading Analog Values	166
5.7	Changing the Range of Values	168
5.8	Reading More Than Six Analog Inputs	170
5.9	Displaying Voltages Up to 5V	173
5.10	Responding to Changes in Voltage	176
5.11	Measuring Voltages More Than 5V (Voltage Dividers)	177
6.	Getting Input from Sensors	181
6.1	Detecting Movement	183
6.2	Detecting Light	186
6.3	Detecting Motion (Integrating Passive Infrared Detectors)	187
6.4	Measuring Distance	189
6.5	Measuring Distance Accurately	193
6.6	Detecting Vibration	197
6.7	Detecting Sound	198
6.8	Measuring Temperature	202
6.9	Reading RFID Tags	206
6.10	Tracking Rotary Movement	208
6.11	Tracking the Movement of More Than One Rotary Encoder	211
6.12	Tracking Rotary Movement in a Busy Sketch	214
6.13	Using a Mouse	216
6.14	Getting Location from a GPS	220
6.15	Detecting Rotation Using a Gyroscope	225
6.16	Detecting Direction	230
6.17	Getting Input from a Game Control Pad (PlayStation)	235
6.18	Reading Acceleration	238
7.	Visual Output	241
7.1	Connecting and Using LEDs	245
7.2	Adjusting the Brightness of an LED	248
7.3	Driving High-Power LEDs	249
7.4	Adjusting the Color of an LED	252
7.5	Sequencing Multiple LEDs: Creating a Bar Graph	255
7.6	Sequencing Multiple LEDs: Making a Chase Sequence (Knight Rider)	258
7.7	Controlling an LED Matrix Using Multiplexing	259
7.8	Displaying Images on an LED Matrix	262
7.9	Controlling a Matrix of LEDs: Charlieplexing	265
7.10	Driving a 7-Segment LED Display	271
7.11	Driving Multidigit, 7-Segment LED Displays: Multiplexing	274
7.12	Driving Multidigit, 7-Segment LED Displays Using MAX7221 Shift Registers	276

7.13	Controlling an Array of LEDs by Using MAX72xx Shift Registers	279
7.14	Increasing the Number of Analog Outputs Using PWM Extender Chips (TLC5940)	281
7.15	Using an Analog Panel Meter as a Display	285
8.	Physical Output	289
8.1	Controlling the Position of a Servo	292
8.2	Controlling One or Two Servos with a Potentiometer or Sensor	294
8.3	Controlling the Speed of Continuous Rotation Servos	296
8.4	Controlling Servos Using Computer Commands	298
8.5	Driving a Brushless Motor (Using a Hobby Speed Controller)	299
8.6	Controlling Solenoids and Relays	300
8.7	Making an Object Vibrate	302
8.8	Driving a Brushed Motor Using a Transistor	304
8.9	Controlling the Direction of a Brushed Motor with an H-Bridge	306
8.10	Controlling the Direction and Speed of a Brushed Motor with an H-Bridge	309
8.11	Using Sensors to Control the Direction and Speed of Brushed Motors (L293 H-Bridge)	311
8.12	Driving a Bipolar Stepper Motor	317
8.13	Driving a Bipolar Stepper Motor (Using the EasyDriver Board)	320
8.14	Driving a Unipolar Stepper Motor (ULN2003A)	323
9.	Audio Output	327
9.1	Playing Tones	329
9.2	Playing a Simple Melody	331
9.3	Generating More Than One Simultaneous Tone	333
9.4	Generating Audio Tones and Fading an LED	335
9.5	Playing a WAV File	338
9.6	Controlling MIDI	341
9.7	Making an Audio Synthesizer	344
10.	Remotely Controlling External Devices	347
10.1	Responding to an Infrared Remote Control	348
10.2	Decoding Infrared Remote Control Signals	350
10.3	Imitating Remote Control Signals	354
10.4	Controlling a Digital Camera	357
10.5	Controlling AC Devices by Hacking a Remote-Controlled Switch	359
11.	Using Displays	363
11.1	Connecting and Using a Text LCD Display	364
11.2	Formatting Text	367
11.3	Turning the Cursor and Display On or Off	370

11.4	Scrolling Text	372
11.5	Displaying Special Symbols	375
11.6	Creating Custom Characters	377
11.7	Displaying Symbols Larger Than a Single Character	379
11.8	Displaying Pixels Smaller Than a Single Character	382
11.9	Connecting and Using a Graphical LCD Display	385
11.10	Creating Bitmaps for Use with a Graphical Display	389
11.11	Displaying Text on a TV	390
12.	Using Time and Dates	397
12.1	Creating Delays	397
12.2	Using millis to Determine Duration	398
12.3	More Precisely Measuring the Duration of a Pulse	402
12.4	Using Arduino as a Clock	404
12.5	Creating an Alarm to Periodically Call a Function	412
12.6	Using a Real-Time Clock	415
13.	Communicating Using I2C and SPI	421
13.1	Controlling an RGB LED Using the BlinkM Module	425
13.2	Using the Wii Nunchuck Accelerometer	430
13.3	Interfacing to an External Real-Time Clock	435
13.4	Adding External EEPROM Memory	437
13.5	Reading Temperature with a Digital Thermometer	441
13.6	Driving Four 7-Segment LEDs Using Only Two Wires	445
13.7	Integrating an I2C Port Expander	449
13.8	Driving Multidigit, 7-Segment Displays Using SPI	451
13.9	Communicating Between Two or More Arduino Boards	454
14.	Wireless Communication	459
14.1	Sending Messages Using Low-Cost Wireless Modules	459
14.2	Connecting Arduino to a ZigBee or 802.15.4 Network	465
14.3	Sending a Message to a Particular XBee	472
14.4	Sending Sensor Data Between XBees	475
14.5	Activating an Actuator Connected to an XBee	480
14.6	Sending Messages Using Low-Cost Transceivers	486
14.7	Communicating with Bluetooth Devices	491
15.	Ethernet and Networking	495
15.1	Setting Up the Ethernet Shield	498
15.2	Obtaining Your IP Address Automatically	500
15.3	Resolving Hostnames to IP Addresses (DNS)	502
15.4	Requesting Data from a Web Server	504
15.5	Requesting Data from a Web Server Using XML	508

15.6	Setting Up an Arduino to Be a Web Server	511
15.7	Handling Incoming Web Requests	514
15.8	Handling Incoming Requests for Specific Pages	517
15.9	Using HTML to Format Web Server Responses	521
15.10	Serving Web Pages Using Forms (POST)	525
15.11	Serving Web Pages Containing Large Amounts of Data	528
15.12	Sending Twitter Messages	535
15.13	Sending and Receiving Simple Messages (UDP)	539
15.14	Getting the Time from an Internet Time Server	545
15.15	Monitoring Pachube Feeds	550
15.16	Sending Information to Pachube	556
16.	Using, Modifying, and Creating Libraries	561
16.1	Using the Built-in Libraries	561
16.2	Installing Third-Party Libraries	563
16.3	Modifying a Library	565
16.4	Creating Your Own Library	568
16.5	Creating a Library That Uses Other Libraries	574
16.6	Updating Third-Party Libraries for Arduino 1.0	580
17.	Advanced Coding and Memory Handling	583
17.1	Understanding the Arduino Build Process	584
17.2	Determining the Amount of Free and Used RAM	587
17.3	Storing and Retrieving Numeric Values in Program Memory	589
17.4	Storing and Retrieving Strings in Program Memory	592
17.5	Using #define and const Instead of Integers	594
17.6	Using Conditional Compilations	595
18.	Using the Controller Chip Hardware	599
18.1	Storing Data in Permanent EEPROM Memory	603
18.2	Using Hardware Interrupts	606
18.3	Setting Timer Duration	609
18.4	Setting Timer Pulse Width and Duration	611
18.5	Creating a Pulse Generator	614
18.6	Changing a Timer's PWM Frequency	617
18.7	Counting Pulses	620
18.8	Measuring Pulses More Accurately	621
18.9	Measuring Analog Values Quickly	624
18.10	Reducing Battery Drain	626
18.11	Setting Digital Pins Quickly	628
18.12	Uploading Sketches Using a Programmer	631
18.13	Replacing the Arduino Bootloader	632
18.14	Reprogram the Uno to Emulate a Native USB device	633

A. Electronic Components	637
B. Using Schematic Diagrams and Data Sheets	643
C. Building and Connecting the Circuit	651
D. Tips on Troubleshooting Software Problems	655
E. Tips on Troubleshooting Hardware Problems	659
F. Digital and Analog Pins	663
G. ASCII and Extended Character Sets	667
H. Migrating to Arduino 1.0	671
Index	677

Who This Book Is For

Unlike in most technical cookbooks, experience with software and hardware is not assumed. This book is aimed at readers interested in using computer technology to interact with the environment. It is for people who want to quickly find the solution to hardware and software problems. The recipes provide the information you need to accomplish a broad range of tasks. It also has details to help you customize solutions to meet your specific needs. There is insufficient space in a book limited to 700 pages to cover general theoretical background, so links to external references are provided throughout the book. See “What Was Left Out” on page xiv for some general references for those with no programming or electronics experience.

If you have no programming experience—perhaps you have a great idea for an interactive project but don’t have the skills to develop it—this book will help you learn what you need to know to write code that works, using examples that cover over 200 common tasks.

If you have some programming experience but are new to Arduino, the book will help you become productive quickly by demonstrating how to implement specific Arduino capabilities for your project.