

Beginning C for Arduino, Second Edition

Learn C Programming
for the Arduino



Jack Purdum, Ph.D.

Apress®

Beginning C for Arduino, Second Edition: Learn C Programming for the Arduino

Jack Purdum
Ecosoft, Inc.
Cincinnati, Ohio, USA

ISBN-13 (pbk): 978-1-4842-0941-7
DOI 10.1007/978-1-4842-0940-0

ISBN-13 (electronic): 978-1-4842-0940-0

Library of Congress Control Number: 2015944814

Copyright © 2015 by Jack Purdum, Ph.D.

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. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

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: Welmoed Spahr

Lead Editor: Michelle Lowman

Technical Reviewer: Terry King

Editorial Board: Steve Anglin, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Michelle Lowman,
James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick,
Ben Renow-Clarke, Gwenan Spearing, Steve Weiss

Coordinating Editor: Kevin Walter

Copy Editor: Kimberly Burton-Weisman

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

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 www.apress.com.

Apress and friends of ED books 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 Special Bulk Sales-eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com. For additional information about how to locate and download your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

To my children: Katie and John

This page intentionally left blank

Contents at a Glance

About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Introduction	xxv
■ Chapter 1: Introduction	1
■ Chapter 2: Arduino C	23
■ Chapter 3: Arduino C Data Types	45
■ Chapter 4: Decision Making in C	69
■ Chapter 5: Program Loops in C	97
■ Chapter 6: Functions in C	119
■ Chapter 7: Storage Classes and Scope	143
■ Chapter 8: Introduction to Pointers	165
■ Chapter 9: Using Pointers Effectively	197
■ Chapter 10: Structures, Unions, and Data Storage	219
■ Chapter 11: The C Preprocessor and Bitwise Operations	253
■ Chapter 12: Arduino Libraries	277
■ Chapter 13: Interfacing to the Outside World	299
■ Chapter 14: A Gentle Introduction to Object-Oriented Programming and C++	321

Appendix A: Suppliers and Sources..... 339

Appendix B: Electronic Components for Experiments 349

Index..... 353

Contents

About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Introduction	xxv
■ Chapter 1: Introduction	1
Why Choose This Book?	1
Assumptions About You	2
What You Need	3
An Atmel-Based Microcontroller Card	3
Types of Memory	3
Making the Choice.....	4
Board Size	5
Input/Output (I/O) Pins	6
Breadboard.....	6
Miscellaneous Parts	8
Installing and Verifying the Software	8
Verifying the Hardware.....	11
Attaching the USB Cable.....	11
Selecting Your μ c Board in the Integrated Development Environment.....	12
Port Selection	12

Loading and Running Your First Program.....	16
Writing Your First Program.....	16
Compiling and Uploading a Program	19
Summary.....	22
■ Chapter 2: Arduino C	23
The Building Blocks of All Programming Languages.....	23
Expressions	24
Statements	25
Statement Blocks	26
Function Blocks	27
The Five Program Steps	28
1. Initialization Step.....	28
2. Input Step	29
3. Process Step.....	29
4. Output Step.....	29
5. Termination Step.....	30
The Purpose of the Five Program Steps	30
A Revisit to Your First Program.....	30
The setup() Function.....	31
The loop() Function	32
Arduino Program Requirements	34
The Blink Program.....	34
Program Comments.....	35
The setup() Function in Blink.....	37
The loop() Function	40
delay(): Good News, Bad News	41
Summary	42

■ Chapter 3: Arduino C Data Types	45
Keywords in C	46
Variable Names in C	47
The boolean Data Type	47
Walking Through the Function Call to ReadSwitchState ()	48
Binary Numbers.....	48
The char Data Type and Character Sets	49
Generating a Table of ASCII Characters	50
The byte Data Type	51
The int Data Type.....	52
The word Data Type.....	52
The long Data Type	52
The float and double Data Types	53
Floating Point Precision	53
The string Data Type.....	53
String Data Type	55
Which Is Better: String or strings Built from char Arrays?	56
The void Data Type	57
The array Data Type.....	58
Array Generalizations	58
Defining vs. Declaring Variables.....	59
Language Errors	59
Symbol Tables.....	59
lvalues and rvalues.....	60
Understanding an Assignment Statement	61
The Bucket Analogy	62
Using the cast Operator.....	64
The Cast Rule.....	65
Summary.....	66

- **Chapter 4: Decision Making in C 69**
 - Relational Operators..... 69
 - The if Statement 70
 - What if Expression1 Is Logic True? 71
 - What if Expression1 Is Logic False? 71
 - Braces or No Braces? 72
 - A Modified Blink Program..... 72
 - The Circuit 73
 - Circuit Resistor Values..... 74
 - The Modified Blink Program 75
 - Software Modifications to the Alternate Blink Program 78
 - The if-else Statement Block..... 79
 - Cascading if statements..... 80
 - The Increment and Decrement Operators 82
 - Two Types of Increment Operators (++) 82
 - The switch statement..... 84
 - A switch Variation, the Ellipsis Operator (...)..... 87
 - Which to Use: Cascading if-else or switch? 88
 - The goto Statement 88
 - Getting Rid of Magic Numbers 88
 - The C Preprocessor 88
 - Heads or Tails 91
 - Initialization Step 91
 - Input Step 91
 - Process Step..... 91
 - Output Step..... 91
 - Termination Step 92
 - Summary..... 94

■ Chapter 5: Program Loops in C.....	97
The Characteristics of Well-Behaved Loops.....	97
Condition 1: Initialization of Loop Control Variable	97
Condition 2: Loop Control Test	98
Condition 3: Changing the Loop Control Variable's State.....	98
Using a for Loop	98
Program to Show Expression Evaluation.....	100
When to Use a <i>for</i> Loop	103
The while Loop	103
When to Use a while Loop	104
The sizeof() Operator	105
The do-while Loop.....	106
Why a do-while is Different from a while Loop	107
The break and continue Keywords.....	107
The break Statement.....	108
The continue Statement	109
A Complete Code Example	109
Step 1. Initialization.....	110
Step 2. Input	110
Step 3. Process.....	110
Step 4. Output.....	110
Step 5. Termination.....	110
Listing 5-5 Is SDC.....	112
Getting Rid of a Magic Number	113
Loops and Coding Style.....	114
Portability and Extensibility.....	115
Summary.....	116

- **Chapter 6: Functions in C 119**
 - The Anatomy of a Function..... 120
 - Function Type Specifier 120
 - Function Name 120
 - Function Arguments 121
 - Function Signatures and Function Prototypes..... 123
 - Function Body..... 124
 - Overloaded Functions 125
 - What Makes a “Good” Function 126
 - Good Functions Use Task-Oriented Names..... 126
 - Good Functions Are Cohesive 126
 - Good Functions Avoid Coupling 126
 - Writing Your Own Functions 127
 - Function Design Considerations 127
 - Function Name 128
 - Argument List 129
 - Function Body..... 129
 - Logical Operators 129
 - Logical AND Operator (&&)..... 130
 - Logical OR (||)..... 131
 - Logical NOT (!) 131
 - Writing Your Own Function 132
 - The IsLeapYear() Function and Coding Style 133
 - Why Use a Specific Function Style? 134
 - Leap Year Calculation Program 134
 - Passing Data into and Back from a Function 137
 - Pass-by-Value..... 137
 - Summary 140

■ Chapter 7: Storage Classes and Scope	143
Hiding Your Program Data	143
The Three Scope Levels	143
Statement Block Scope	144
Why Use Statement Block Scope?	146
Function Block Scope	146
Name Collisions and Scope	147
Global Scope	150
Trade-offs	151
Global Scope and Name Conflicts	151
Scope and Storage Classes	152
The auto Storage Class	152
The register Storage Class	152
The static Storage Class	153
The Effect of the static Storage Class	153
The extern Storage Class	154
Adding a Second Source Code File to a Project	154
Function Prototypes	158
#include Preprocessor Directive	158
A common #include Idiom	159
Where Are the Header Files Stored?	160
The volatile keyword	160
Summary	160
■ Chapter 8: Introduction to Pointers	165
Defining a Pointer	165
Pointer Name	166
Asterisk (*)	166
Pointer Type Specifiers and Pointer Scalars	166
Why All Arduino Pointers Use Two Bytes for Storage	168
Pointer Initialization	169

Using the Address-Of Operator	170
The Indirection Operator (*)	171
Why Are Pointers Useful?	175
Modified Blink Program.....	179
Pointers and Arrays	180
The Importance of Scalars.....	183
Pass-by-Value vs. Pass-by-Reference.....	185
Your Turn	188
One Approach	189
One Solution	189
Debug Statements Using the Serial Object.....	192
Summary.....	193
■ Chapter 9: Using Pointers Effectively	197
Relational Operations and Test for Equality Using Pointers.....	197
Pointer Comparisons Must Be Between Pointers to the Same Data	198
Pointer Arithmetic.....	198
Constant lvalues	203
Two-Dimensional Arrays.....	203
A Small Improvement	206
How Many Dimensions?	206
Two-Dimensional Arrays and Pointers.....	207
Treating the Two-Dimensional Array of chars As a String.....	209
Pointers to Functions	209
Arrays of Pointers to Functions	211
enum Data Type	212
The Right-Left Rule	216
Summary.....	217

■ Chapter 10: Structures, Unions, and Data Storage	219
Structures.....	219
Declaring a Structure.....	220
Defining a Structure	221
Accessing Structure Members	222
Escape Sequences	224
Memory Requirements for a Structure	225
Returning a Structure from a Function Call	226
Using Structure Pointers.....	228
Initializing a Structure	231
Arrays of Structures.....	231
Unions	232
EEPROM Memory	233
Using EEPROM.....	234
Other Storage Alternatives	242
Shields.....	242
typedef	247
Summary	248
■ Chapter 11: The C Preprocessor and Bitwise Operations	253
Preprocessor Directives	253
#undef	255
#line.....	256
#if, Conditional Directives	257
#include.....	258
Parameterized Macros	259
Decimal to Binary Converter	261
Bitwise Operators	263
Bitwise Shift Operators.....	267

One More Example	268
Using Different Bases for Integer Constants.....	269
Parameterized Macros ... Continued	269
Summary.....	270
■ Chapter 12: Arduino Libraries	277
The Linker	277
Libraries	278
Arduino Libraries	278
The Arduino Core Libraries	280
Other Libraries.....	285
Writing Your Own Library.....	287
The Library Header File	288
The Library Code File (Dates.cpp).....	289
Setting the Arduino IDE to Use Your Library	292
A Sample Program Using the Dates Library	292
Adding the Easter Program As Part of the Library	294
The keywords.txt File	295
Keyword Coloring (theme.txt)	295
Summary	297
■ Chapter 13: Interfacing to the Outside World	299
The Serial Peripheral Interface (SPI)	300
An SPI Program	300
Interrupts and Interrupt Service Routines (ISR).....	305
Interrupt Details.....	307
An External Interrupt Program	308
An Alternative Interrupt Program.....	310

Ultrasonic Sensor Program	311
A Programming Problem	314
My Solution.....	314
Conclusion.....	318
■ Chapter 14: A Gentle Introduction to Object-Oriented Programming and C++	321
The OOP Trilogy	321
Encapsulation	321
Inheritance.....	322
Polymorphism.....	323
The OOP Class	323
Inside an OOP Class.....	324
OOP and Class Objects	325
public vs. private in a Class.....	325
The EEPROM.cpp File	327
Add julian() to Dates	329
Adding a private Class Member	330
Constructors and Destructors	332
Conclusion.....	336
Appendix A: Suppliers and Sources.....	339
Starter Kits	339
Shields, Boards, Sensors.....	342
Specific Parts Sources	347
Bezels	347
Jumper Wires.....	348
Project Cases.....	348
Domestic Parts Suppliers	348

Appendix B: Electronic Components for Experiments 349

 Microcontroller Board..... 349

 Solderless Breadboard 349

 Electronic Components 349

 Online Component Purchases 350

 Experiment! 351

Index..... 353