

---

## Table of Contents

<b>Introduction . . . . .</b>	<b>17</b>
Learning C . . . . .	17
Target Audience . . . . .	17
AVR Microcontrollers . . . . .	17
The AVR Architecture. . . . .	17
Embedded Systems. . . . .	18
Choosing an Embedded System to Learn C On . . . . .	19
Why use the C Programming Language? . . . . .	20
Prerequisites . . . . .	20
Hardware Requirements. . . . .	21
Arduino Board . . . . .	21
Programmer . . . . .	22
The Approach Taken in this Book . . . . .	23
Some Good Advice . . . . .	23
Accompanying Files and Support Website. . . . .	23
<b>Chapter 1 Your First C Program. . . . .</b>	<b>25</b>
1.1 Download and Install Atmel Studio . . . . .	25
1.1.1 Downloading Atmel Studio . . . . .	25
1.1.2 Installing Atmel Studio . . . . .	26
1.2 Download and Install a Terminal Emulator. . . . .	26
1.3 Install Template Files . . . . .	26
1.4 Connecting the Hardware . . . . .	27
1.5 How C Programs are Created . . . . .	27
1.6 Start Programming. . . . .	27
1.6.1 Creating a New Project from the Template Files . . . . .	27
1.6.2 Building the Project. . . . .	31
1.6.3 Build Problems and Solutions . . . . .	31
1.6.4 Loading and Running the Program . . . . .	31
1.6.5 Program Loading and Running – Problems and Solutions. . . . .	33
1.7 About Your First C Program . . . . .	34

1.8	Analysing the Program . . . . .	35
1.8.1	Characters and Strings . . . . .	35
1.8.2	Functions . . . . .	36
1.8.3	Program Statements . . . . .	37
1.8.4	Preprocessor Directives . . . . .	37
1.8.5	Whitespace Characters . . . . .	37
1.8.6	Ending the Program . . . . .	38
1.9	Exercises . . . . .	38
1.9.1	Three Text Lines . . . . .	38
1.9.2	Find the Errors . . . . .	38
1.10	Solutions . . . . .	39
1.10.1	Solution to 1.9.1 . . . . .	39
1.10.2	Solution to 1.9.2 . . . . .	39
1.11	Summary . . . . .	39
<b>Chapter 2</b>	<b>C Basics . . . . .</b>	<b>41</b>
2.1	Input, Output and Variables . . . . .	41
2.1.1	Variables . . . . .	43
2.1.2	Getting Input from the User Using the scanf() Function . . . . .	44
2.2	Variable Types . . . . .	47
2.2.1	Floating Point Variables . . . . .	48
2.2.2	Character Variables . . . . .	48
2.3	Arithmetic Operators . . . . .	49
2.4	Field Width Specifiers . . . . .	51
2.5	Compiling and Linking . . . . .	52
2.6	Errors and Warnings . . . . .	53
2.6.1	Compile Errors . . . . .	54
2.6.2	Link Errors . . . . .	56
2.7	Exercises . . . . .	56
2.7.1	Ohm's Law Calculations . . . . .	56
2.7.2	Variable Names . . . . .	57
2.8	Solutions . . . . .	57
2.8.1	Solution to 2.7.1 . . . . .	57

---

2.8.2 Solution to 2.7.2. . . . .	58
2.9 Summary . . . . .	58
<b>Chapter 3 Comparative Operators and Decisions . . . . .</b>	<b>61</b>
3.1 Comparative Operators . . . . .	61
3.1.1 True and False . . . . .	61
3.1.2 C Comparative Operators. . . . .	62
3.2 Decisions. . . . .	62
3.2.1 Using if to Make a Decision . . . . .	63
3.2.2 Using else in Conjunction with if. . . . .	63
3.2.3 The = Operator and the == Operator . . . . .	65
3.2.4 Using else – if in Conjunction with if . . . . .	67
3.3 Exercises. . . . .	70
3.3.1 Variable Compare . . . . .	70
3.3.2 Variable Compare 2 . . . . .	70
3.4 Solutions. . . . .	70
3.4.1 Solution to 3.3.1. . . . .	70
3.4.2 Solution to 3.3.2. . . . .	71
3.5 Summary . . . . .	72
<b>Chapter 4 The while Loop and Commenting Code. . . . .</b>	<b>73</b>
4.1 The while Loop. . . . .	73
4.2 Using if Inside the while Loop . . . . .	75
4.3 The Guess My Number Game. . . . .	76
4.4 Back to the Temperature Controller Example . . . . .	78
4.5 Commenting Programs . . . . .	79
4.6 Programming Style. . . . .	81
4.6.1 Tab Settings in Atmel Studio . . . . .	81
4.6.2 Choosing Tab Width . . . . .	81
4.6.3 Code Indenting Examples . . . . .	82
4.6.4 Other Aspects of Programming Style. . . . .	84
4.7 Exercises. . . . .	84
4.7.1 Subtract . . . . .	84
4.7.2 Program Loop. . . . .	85

4.8 Solutions . . . . .	85
4.8.1 Solution to 4.7.1. . . . .	85
4.8.2 Solution to 4.7.2. . . . .	86
4.9 Summary . . . . .	86
<b>Chapter 5 Functions . . . . .</b>	<b>89</b>
5.1 Your Second Function . . . . .	89
5.2 Passing Data to a Function. . . . .	91
5.3 Passing More Than One Value to a Function . . . . .	92
5.4 Passing a Variable to a Function . . . . .	94
5.5 Getting a Value Back from a Function . . . . .	95
5.6 Passing Values to a Function and Receiving a Value Back . . . . .	96
5.7 Flashing LED Simulation Program . . . . .	97
5.7.1 The while(1) Loop. . . . .	98
5.7.2 The long Data Type. . . . .	98
5.8 Preprocessor Directives . . . . .	99
5.9 Functions Calling Functions . . . . .	103
5.10 Using Multiple Source Files. . . . .	103
5.10.1 Adding a New C Source File to an Atmel Studio Project . . . . .	104
5.10.2 Project with Two C Source Code Files . . . . .	104
5.11 Header Files. . . . .	106
5.12 How Functions Relate to Linking and Library Files . . . . .	107
5.13 Exercises . . . . .	107
5.13.1 Parallel Resistor Calculation . . . . .	107
5.13.2 Resistor Formula C File . . . . .	107
5.14 Solutions . . . . .	108
5.14.1 Solution to 5.13.1. . . . .	108
5.14.2 Solution to 5.13.2. . . . .	109
5.15 Summary . . . . .	110
<b>Chapter 6 Number Systems . . . . .</b>	<b>111</b>
6.1 Binary Basics . . . . .	111
6.2 The Need for Binary Numbers . . . . .	112
6.3 Numbering Systems . . . . .	112

---

6.3.1 A Quick Look at Decimal Numbers . . . . .	112
6.3.2 Binary Numbers . . . . .	113
6.3.3 Hexadecimal Numbers. . . . .	116
6.4 Working with Hexadecimal Numbers in C. . . . .	121
6.5 The ASCII Alphanumeric Code . . . . .	124
6.6 Exercises. . . . .	126
6.6.1 Hexadecimal to Port . . . . .	126
6.6.2 Print Number . . . . .	126
6.7 Solutions. . . . .	126
6.7.1 Solution to 6.6.1. . . . .	126
6.7.2 Solution to 6.6.2. . . . .	127
6.8 Summary . . . . .	127
<b>Chapter 7 Memory and Microcontrollers . . . . .</b>	<b>129</b>
7.1 Memory Basics. . . . .	129
7.1.1 ROM (Read Only Memory) . . . . .	129
7.1.2 RAM (Random Access Memory) . . . . .	129
7.1.3 Data Stored in Memory . . . . .	129
7.2 A Look at a Memory Chip. . . . .	130
7.3 How Microprocessors Access Memory and Peripherals . . . . .	133
7.4 Pointers. . . . .	134
7.5 More on C Data Types. . . . .	138
7.6 Microcontroller Memory Sizes. . . . .	142
7.7 Summary . . . . .	143
<b>Chapter 8 Accessing AVR Ports in C. . . . .</b>	<b>145</b>
8.1 AVR Pins and Ports . . . . .	145
8.2 Switching the On-board LED On and Off . . . . .	148
8.3 Controlling LEDs interfaced to a Port. . . . .	152
8.3.1 Connecting the Hardware . . . . .	152
8.3.2 Starting a New Atmel Studio Project from Scratch . . . . .	154
8.3.3 LED Port Control Program . . . . .	157
8.4 Increment and Decrement Operators . . . . .	159
8.4.1 Pre-increment and Pre-decrement . . . . .	159

---

8.4.2	Post-increment and Post-decrement . . . . .	159
8.5	Summary . . . . .	161
<b>Chapter 9</b>	<b>I/O and Memory Maps . . . . .</b>	<b>163</b>
9.1	Input Pins . . . . .	163
9.1.1	Connecting a Switch to a Port Pin . . . . .	163
9.1.2	Reading the State of a Switch in C . . . . .	164
9.2	Input and Output Pins on the Same Port . . . . .	166
9.2.1	Connecting a Switch and LED . . . . .	166
9.2.2	Programming a Switch and LED . . . . .	166
9.2.3	The AND Bitwise Operator . . . . .	167
9.3	AVR Memory Map . . . . .	169
9.3.1	Memory Architecture . . . . .	170
9.3.2	Flash Program Memory . . . . .	170
9.3.3	Data Memory . . . . .	170
9.4	Summary . . . . .	170
<b>Chapter 10</b>	<b>Previous C Topics Revisited . . . . .</b>	<b>173</b>
10.1	Format Specifiers . . . . .	173
10.2	Field Width Specifiers Revisited . . . . .	174
10.3	Escape Sequences . . . . .	176
10.4	Loops . . . . .	178
10.4.1	A while Loop that uses break and continue. . . . .	178
10.4.2	The do – while Loop . . . . .	180
10.4.3	The for Loop. . . . .	181
10.5	The Problem with printf() in Embedded Systems . . . . .	182
10.5.1	Programming the Serial Port . . . . .	182
10.5.2	Writing Serial Port Functions. . . . .	184
10.6	Nested Loops and Decisions . . . . .	188
10.6.1	Nested Loops . . . . .	188
10.6.2	Nesting Decisions . . . . .	189
10.7	Decision Making with the switch Statement . . . . .	190
10.8	The Conditional Operator. . . . .	193
10.9	Functions and Pointers . . . . .	194

---

10.9.1	Passing an Address to a Function . . . . .	194
10.9.2	Returning More Than One Value from a Function. . . . .	195
10.10	Variables and Scope . . . . .	196
10.10.1	Local Variables . . . . .	196
10.10.2	Global Variables . . . . .	197
10.11	Static Variables. . . . .	198
10.12	Floating Point Data Types. . . . .	199
10.13	Casts. . . . .	200
10.14	Exercises . . . . .	201
10.14.1	Count Down for Loop. . . . .	201
10.14.2	LED Control using a switch Construct. . . . .	201
10.15	Solutions . . . . .	201
10.15.1	Solution to 10.14.1 . . . . .	201
10.15.2	Solution to 10.14.2 . . . . .	202
10.16	Summary. . . . .	202
<b>Chapter 11</b>	<b>Arrays and Strings . . . . .</b>	<b>205</b>
11.1	Arrays . . . . .	205
11.2	Strings . . . . .	209
11.2.1	Writing to a String . . . . .	210
11.2.2	Initialising a String . . . . .	212
11.2.3	C Library String Functions . . . . .	213
11.3	Arrays and Addresses . . . . .	214
11.3.1	Arrays and Array Element Addresses. . . . .	214
11.3.2	Passing an Array to a Function . . . . .	216
11.4	Strings as Pointers . . . . .	218
11.5	Writing to a String using printf() . . . . .	219
11.6	Multidimensional Arrays. . . . .	221
11.7	Exercises . . . . .	226
11.7.1	Button Message . . . . .	226
11.7.2	Button Wait . . . . .	227
11.8	Solutions . . . . .	227
11.8.1	Solution to 11.7.1 . . . . .	227

11.8.2	Solution to 11.7.2 . . . . .	227
11.9	Summary . . . . .	228
<b>Chapter 12</b>	<b>Bit Manipulation and Logical Operators . . . . .</b>	<b>229</b>
12.1	Bit Manipulation with Bitwise Operators. . . . .	229
12.1.1	Why Do We Need Bitwise Operators?. . . . .	231
12.1.2	Using the Bitwise AND and OR Operators. . . . .	232
12.1.3	Using the Bitwise NOT Operator . . . . .	234
12.1.4	Using the Bitwise XOR Operator . . . . .	235
12.1.5	The Left and Right Shift Operators . . . . .	236
12.1.6	The C Assignment Operators . . . . .	238
12.2	Logical Operators . . . . .	238
12.3	Operator Precedence. . . . .	241
12.4	Using Pre-defined Bit Names with Shift Operators. . . . .	243
12.4.1	Pre-defined Bit Names in C Code . . . . .	243
12.4.2	The AVR _BV() Macro . . . . .	245
12.5	Exercises . . . . .	247
12.5.1	Print Binary Number . . . . .	247
12.6	Solution. . . . .	248
12.6.1	Solution to 12.5.1 . . . . .	248
12.7	Summary . . . . .	248
<b>Chapter 13</b>	<b>Programming Hardware . . . . .</b>	<b>251</b>
13.1	Timer Counter Hardware . . . . .	251
13.1.1	Polled Timer Time Delay . . . . .	252
13.1.2	Hardware Timer Function . . . . .	255
13.2	The Analogue to Digital Converter (ADC) . . . . .	256
13.3	The Watchdog Timer . . . . .	260
13.4	I/O Ports . . . . .	262
13.4.1	I/O Port Register Functionality . . . . .	262
13.4.2	Register Pin Toggle . . . . .	262
13.4.3	Enabling Pull-up Resistors . . . . .	263
13.5	Summary . . . . .	265
<b>Chapter 14</b>	<b>Debugging . . . . .</b>	<b>267</b>



---

14.1	Configuring Arduino Boards for Debugging . . . . .	267
14.2	Debugging Example Program . . . . .	267
14.3	Switching Compiler Optimisation Off . . . . .	269
14.4	Enabling debugWIRE on the AVR . . . . .	269
14.5	Using the Debugger . . . . .	271
14.5.1	Starting the Debugger . . . . .	271
14.5.2	Single Stepping and Adding Watches . . . . .	271
14.5.3	Viewing Hardware Registers . . . . .	271
14.5.4	Viewing and Modifying Variable and Register Values . . . . .	273
14.5.5	Viewing Variables in Memory . . . . .	273
14.5.6	Stopping Debugging and Setting Breakpoints . . . . .	274
14.5.7	Viewing Pin State Changes . . . . .	275
14.5.8	Viewing Variable Value Changes & Setting Conditional Breakpoints . . . . .	276
14.5.9	Stepping into a Function . . . . .	277
14.6	Enabling ISP on the AVR / Disabling debugWIRE . . . . .	277
14.7	Exercises . . . . .	277
14.8	Summary . . . . .	278
<b>Chapter 15</b>	<b>Interrupts . . . . .</b>	<b>279</b>
15.1	Handling Interrupts in C . . . . .	279
15.2	Using the Timer Interrupt . . . . .	279
15.3	One Millisecond Tick Interrupt . . . . .	281
15.4	Summary . . . . .	284
<b>Chapter 16</b>	<b>Wrapping Up the C Language. . . . .</b>	<b>285</b>
16.1	Structures . . . . .	285
16.1.1	Using Structures in C Programs . . . . .	285
16.1.2	Initialising Structures . . . . .	287
16.1.3	Pointers to Structures . . . . .	288
16.2	Unions . . . . .	288
16.3	Enumerated Type . . . . .	291
16.4	The typedef Declarator . . . . .	293
16.5	Storage Class Specifiers . . . . .	294
16.6	Type Qualifiers . . . . .	295

16.7	The goto Statement . . . . .	295
16.8	A List of All C Keywords. . . . .	296
16.9	More Preprocessor Directives . . . . .	296
16.10	C99 Variable Types . . . . .	298
16.11	Alternative Continuous Loop. . . . .	298
16.12	Summary. . . . .	299
<b>Chapter 17</b>	<b>C Projects . . . . .</b>	<b>301</b>
17.1	Serial Port Driver Functions . . . . .	301
17.1.1	Serial Port Driver Design Notes. . . . .	301
17.1.2	Serial Port Driver Code . . . . .	302
17.2	Converting Numbers to Strings. . . . .	306
17.2.1	Using AVR C Library Functions . . . . .	306
17.2.2	Writing Number to String Conversion Functions . . . . .	309
17.3	Voltmeter . . . . .	314
17.3.1	Voltmeter Project Features. . . . .	314
17.3.2	Voltmeter Project Code . . . . .	314
17.3.3	Enabling the printf Family of Functions to Work . . . . .	317
17.3.4	Principle of Operation of the Code. . . . .	317
17.3.5	Examining the Code . . . . .	318
17.3.6	Code Timing. . . . .	319
17.4	Stopwatch . . . . .	320
17.4.1	Stopwatch Project Hardware and Code . . . . .	320
17.4.2	Stopwatch Project Design . . . . .	323
17.4.3	Stopwatch Project Code Explanation . . . . .	323
17.5	Where to from here?. . . . .	327
<b>Appendix A</b>	<b>The ASCII Table . . . . .</b>	<b>329</b>
<b>Appendix B</b>	<b>Arduino – AVR Port Pin Mapping. . . . .</b>	<b>331</b>
B.1	Arduino Uno . . . . .	331
B.2	Arduino MEGA . . . . .	332
<b>Appendix C</b>	<b>Standard Arduino I/O Circuit. . . . .</b>	<b>335</b>
C.1	Arduino Uno . . . . .	335
C.2	Arduino MEGA . . . . .	336

---

<b>Appendix D References and Resources.</b> . . . . .	<b>337</b>
D.1 Book Website and Files . . . . .	337
D.2 Datasheets and Application Notes . . . . .	337
D.3 AVR Libc Reference. . . . .	337
D.4 AVR Forum . . . . .	337
D.5 Additional Resources. . . . .	337
<b>Index</b> . . . . .	<b>339</b>