THOMSON
DELMAR LEARNING

Embedded C Programming and the Atmel AVR

2nd Edition

Barnett, Cox and O'Cull

CD INSIDE

CONTENTS

PREFACE xiii

INTRODUCTION xxi

CHAPTER 1 EMBEDDED C LANGUAGE TUTORIAL

1.1 OBJECTIVES 1
1.2 INTRODUCTION 1
1.3 BEGINNING CONCEPTS 2
1.4 VARIABLES AND CONSTANTS 4
1.4.1 Variable Types 4
1.4.2 Variable Scope 5
Local Variables 5
Global Variables 5
1.4.3 Constants 6
Numeric Constants 7
Character Constants 7
1.4.4 Enumerations and Definitions 7
1.4.5 Storage Classes 9
Automatic 9
Static 9
Register 9
1.4.6 Type Casting 9
1.5 I/O OPERATIONS 11
1.6 OPERATORS AND EXPRESSIONS 12
1.6.1 Assignment and Arithmetic Operators 12
Bitwise Operators 13
1.6.2 Logical and Relational Operators 14
Logical Operators 15
Relational Operators 15
1.6.3 Increment, Decrement, and Compound Assignment 16
Increment Operators 16

Decrement Operators 17

Compound Assignment Operators 17

1.6.4 The Conditional Expression 17

1.6.5 Operator Precedence 18

1.7 CONTROL STATEMENTS 19

1.7.1 While Loop 19

1.7.2 Do/While Loop 21

1.7.3 For Loop 22

1.7.4 If/Else 23

If Statement 23

If/Else Statement 23

Conditional Expression 26

1.7.5 Switch/Case 26

1.7.6 Break, Continue, and Goto 28

Break 28

Continue 28

Goto 29

1.8 FUNCTIONS 33

1.8.1 Prototyping and Function Organization 34

1.8.2 Functions that Return Values 36

1.8.3 Recursion 37

1.9 POINTERS AND ARRAYS 41

1.9.1 Pointers 41

1.9.2 Arrays 45

1.9.3 Multidimensional Arrays 47

1.9.4 Pointers to Functions 49

1.10 STRUCTURES AND UNIONS 54

1.10.1 Structures 54

1.10.2 Arrays of Structures 56

1.10.3 Pointers to Structures 57

1.10.4 Unions 58

1.10.5 Typedef Operator 60

1.10.6 Bits and Bitfields 61

1.10.7 Sizeof Operator 62

1.11 MEMORY TYPES 63

1.11.1 Constants and Variables 63

1.11.2 Pointers 65

1.11.3 Register Variables 65

sfrb and sfrw 66

1.12 REAL-TIME METHODS 69

1.12.1 Using Interrupts 69

1.12.2 Real-Time Executives 72

1.12.3 State Machines 75

1.13 PROGRAMMING STYLE, STANDARDS, AND GUIDELINES 80

1.14 CHAPTER SUMMARY 81

1.15 EXERCISES 81
1.16 LABORATORY ACTIVITIES 83

CHAPTER 2 THE ATMEL RISC PROCESSORS

2.1 OBJECTIVES 87
2.2 INTRODUCTION 87
2.3 ARCHITECTURAL OVERVIEW 88
2.4 MEMORY 89
2.4.1 FLASH Code Memory 89
2.4.2 Data Memory 89
Registers 90
I/O Registers 90
SRAM 92
2.4.3 EEPROM Memory 94
2.5 RESET AND INTERRUPT FUNCTIONS 97
2.5.1 Interrupts 98
2.5.2 Reset 101
Watchdog Timer and Reset 102
2.6 PARALLEL I/O PORTS 105
2.7 TIMER/COUNTERS 109
2.7.1 Timer/Counter Prescalers and Input Selectors 110
2.7.2 Timer 0 110
2.7.3 Timer 1 114
Timer 1 Prescaler and Selector 115
Timer 1 Input Capture Mode 115
Timer 1 Output Compare Mode 119
Timer 1 Pulse Width Modulator Mode 123
2.7.4 Timer 2 128
One–second recording interval using Timer 0 129
Engine rpm measurement using Timer 1 130
Drive shaft rpm measurement using Timer 1 131
2.8 SERIAL COMMUNICATION USING THE USART 132
2.9 ANALOG INTERFACES 141
2.9.1 Analog-to-Digital Background 141
2.9.2 Analog-to-Digital Converter Peripheral 142
2.9.3 Analog Comparator Peripheral 146
Measuring engine temperature using the analog-to-digital converter (ADC) 149
Sending collected data to the PC 150
2.10 SERIAL COMMUNICATION USING THE SPI 151
2.11 SERIAL COMMUNICATION USING I^{2}C 158
2.12 THE AVR RISC ASSEMBLY LANGUAGE INSTRUCTION SET 160
2.13 CHAPTER SUMMARY 163

2.14 EXERCISES 167
2.15 LABORATORY ACTIVITIES 168

CHAPTER 3 STANDARD I/O AND PREPROCESSOR FUNCTIONS

3.1 OBJECTIVES 171
3.2 INTRODUCTION 171
3.3 CHARACTER INPUT/OUTPUT FUNCTIONS – *getchar()* AND *putchar()* 172
3.4 STANDARD OUTPUT FUNCTIONS 178
3.4.1 Put String—*puts()* 178
3.4.2 Put String FLASH—*putsf()* 179
3.4.3 Print Formatted—*printf()* 180
3.4.4 String Print Formatted—*sprintf()* 182
3.5 STANDARD INPUT FUNCTIONS....183
3.5.1 Get String—*gets()* 184
3.5.2 Scan Formatted—*scanf()* 185
3.5.3 Scan String Formatted—*sscanf()* 187
3.6 PREPROCESSOR DIRECTIVES 188
3.6.1 The #include Directive 188
3.6.2 The #define Directive 189
3.6.3 The #ifdef, #ifndef, #else, and #endif Directives 191
3.6.4 The #pragma Directive 196
#pragma warn 197
#pragma opt 197
#pragma optsize 197
#pragma savereg 198
#pragma regalloc 199
#pragma promotechar 199
#pragma uchar 199
#pragma library 200
3.6.5 Other Macros and Directives 200
3.7 CHAPTER SUMMARY 201
3.8 EXERCISES 202
3.9 LABORATORY ACTIVITIES 202

CHAPTER 4 THE CODEVISIONAVR C COMPILER AND IDE

4.1 OBJECTIVES 205
4.2 INTRODUCTION 205
4.3 IDE OPERATION 206
4.3.1 Projects 206
Open Existing Projects 206
Create New Projects 207
Configure Projects 208
Close Project 209

4.3.2 Source Files 209
Open an Existing Source File 209
Create a New Source File 209
Add an Existing File to the Project 209
4.3.3 Edit Files 211
4.3.4 Print Files 214
4.3.5 The File Navigator 214
4.4 C COMPILER OPTIONS 216
4.4.1 Memory Model 217
4.4.2 Optimize For 218
4.4.3 Optimization Level 218
4.4.4 Program Type 218
4.4.5 (s)printf Features and (s)scanf Features 218
4.4.6 SRAM 219
4.4.7 Compilation 219
4.4.8 Messages Tab 220
4.5 COMPILE AND MAKE PROJECTS 220
4.5.1 Compile a Project 220
4.5.2 Make a Project 221
4.6 PROGRAM THE TARGET DEVICE 222
4.6.1 Chip 223
4.6.2 FLASH and EEPROM 224
4.6.3 FLASH Lock Bits 226
4.6.4 Fuse Bits 226
4.6.5 Boot Lock Bit 0 and Boot Lock Bit 1 226
4.6.6 Signature 226
4.6.7 Chip Erase 227
4.6.8 Programming Speed 227
4.6.9 Program All 227
4.6.10 Other Programmers 228
4.7 CODEWIZARDAVR CODE GENERATOR 229
4.7.1 Chip Tab 232
4.7.2 Ports Tab 233
4.7.3 External IRQ Tab 234
4.7.4 Timers Tab 235
4.7.5 USART Tab 236
4.7.6 ADC Tab 237
4.7.7 Project Information Tab 238
4.7.8 Generate Source Code 239
4.8 TERMINAL TOOL 247
4.9 THE ATMEL AVR STUDIO DEBUGGER 249
4.9.1 Create a COFF File for AVR Studio 250
4.9.2 Launch AVR Studio from CodeVisionAVR 250
4.9.3 Open a File for Debug 250
4.9.4 Start, Stop, and Step 250

4.9.5 Set and Clear Breakpoints 251
4.9.6 View and Modify Registers and Variables 252
4.9.7 View and Modify the Machine State 252
4.10 CHAPTER SUMMARY 253
4.11 EXERCISES 253
4.12 LABORATORY ACTIVITIES 255

CHAPTER 5 PROJECT DEVELOPMENT

5.1 OBJECTIVES 257
5.2 INTRODUCTION 257
5.3 CONCEPT DEVELOPMENT PHASE 257
5.4 PROJECT DEVELOPMENT PROCESS STEPS 257
5.4.1 Definition Phase 258
5.4.2 Design Phase 260
5.4.3 Test Definition Phase 261
5.4.4 Build and Test the Prototype Hardware Phase 262
5.4.5 System Integration and Software Development Phase 262
5.4.6 System Test Phase 263
5.4.7 Celebration Phase 263
5.5 PROJECT DEVELOPMENT PROCESS SUMMARY 263
5.6 EXAMPLE PROJECT: A WEATHER MONITOR 263
5.6.1 Concept Phase 263
5.6.2 Definition Phase 264
5.6.2.1 Electrical Specification 266
5.6.2.2 Operational Specification 266
5.6.2.3 Basic Block Diagrams 267
5.6.3 Measurement Considerations for the Design 269
5.6.3.1 Temperature 270
5.6.3.2 Barometric Pressure 272
5.6.3.3 Humidity 273
5.6.3.4 Wind Speed 274
5.6.3.5 Wind Direction 277
5.6.3.6 Rainfall 278
5.6.3.7 Dew Point Computation 281
5.6.3.8 Wind Chill Computation 282
5.6.3.9 Battery Health 283
5.6.3.10 Real Time 283
5.6.4 Hardware Design, Outdoor Unit 284
Wind Speed Input 284
Rain Gauge Input 286
900 MHz Transmitter 286
Power Supply 286
5.6.5 Software Design, Outdoor Unit 286
5.6.6 Hardware Design, Indoor Unit 287
900 MHz Receiver 287
Power Supply 290

5.6.7 Software Design, Indoor Unit 290
5.6.8 Test Definition Phase 292
Wind Direction 293
Wind Speed 293
Rain Gauge 293
Air Temperature 293
Barometric Pressure 294
Relative Humidity 294
System Test for Complete Project 294
5.6.9 Build and Test Prototype Hardware Phase 294
Outdoor Unit Checkout 295
Indoor Unit Checkout 297
5.6.10 System Integration and Software Development Phase, Outdoor Unit 301
Temperature, Humidity, Wind Direction, and Battery Health 305
Rainfall 305
Wind Speed 306
RF Telemetry 306
5.6.11 System Integration and Software Development Phase, Indoor Unit 312
Keeping Time 312
Low-Battery Indication 314
The Buttons and the Beeper 316
Decoding the RF Telemetry 318
Collecting and Protecting Rainfall Data 321
Converting from Counts to Real Units 324
Routines for Controlling the LCD 325
Keeping the Display Up to Date 331
Editing the Time and Date 335
5.6.12 System Test Phase 339
5.6.13 Changing It Up 343
Picking a Part for a Better Fit 343
Changes to the Schematic 344
Changes to I/O Mapping 344
Other Considerations 348
5.7 CHALLENGES 349
5.8 CHAPTER SUMMARY 350
5.9 EXERCISES 350
5.10 LABORATORY ACTIVITY 351

APPENDIX A LIBRARY FUNCTIONS REFERENCE 353

APPENDIX B GETTING STARTED WITH CODEVISIONAVR AND THE STK500 453

APPENDIX C PROGRAMMING THE AVR MICROCONTROLLERS 471

APPENDIX D INSTALLING AND USING THECABLEAVR 475

APPENDIX E THE MEGAAVR-DEV DEVELOPMENT BOARD 489

APPENDIX F ASCII TABLE 493

APPENDIX G AVR INSTRUCTION SET SUMMARY 497

APPENDIX H ANSWERS TO SELECTED EXERCISES 503

APPENDIX I A "FAST START" TO EMBEDDED C PROGRAMMING AND THE AVR 509

INDEX 519