

# Introduction to Embedded Systems

Manuel Jiménez · Rogelio Palomera  
Isidoro Couvertier

# Introduction to Embedded Systems

Using Microcontrollers and the MSP430

 Springer

Manuel Jiménez  
Rogelio Palomera  
Isidoro Couvertier  
University of Puerto Rico at Mayagüez  
Mayagüez, PR  
USA

ISBN 978-1-4614-3142-8                      ISBN 978-1-4614-3143-5 (eBook)  
DOI 10.1007/978-1-4614-3143-5  
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013939821

© Springer Science+Business Media New York 2014

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. The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

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.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

*To my family, for all your support and all we missed while  
I was working on this project*

Manuel Jiménez

*To my family*

Rogelio Palomera

*To my Father, in Whom I move and live and exist,  
and to my family*

*Microcontrollers*

*What is it that You seek which such tenderness to Thee?  
Microcontrollers I seek, microcontrollers belong to Me  
Why do you so earnestly seek microcontrollers to Thee?  
Microcontrollers I seek, microcontrollers to freed*

*Microcontrollers you did set free when for them you died  
Microcontrollers I did set free, microcontrollers are Mine  
Why do they still need to be free if they are already Thine?  
Microcontrollers are alienated still, alienated in their minds*

Isidoro Couvertier

# Preface

The first years of the decade of 1970 witnessed the development of the first microprocessor designs in the history of computing. Two remarkable events were the development of the 4004, the first commercial, single-chip microprocessor created by Intel Corporation and the TMS1000, the first single-chip microcontroller by Texas Instruments. Based on 4-bit architectures, these early designs opened a whole new era in terms of technological advances. Several companies such as Zilog, Motorola, Rockwell, and others soon realized the potential of microprocessors and joined the market with their own designs. Today, we can find microprocessors from dozens of manufacturers in almost any imaginable device in our daily lives: from simple and inexpensive toys to sophisticated space ships, passing through communication devices, life supporting and medical equipment, defense applications, and home appliances.

Being such a ubiquitous device, it is not surprising that more and more people today need to understand the basics of how microprocessors work in order to harness and exploit their capacity. Among these people we find enthusiastic hobbyists, engineers without previous experience, electronic technology students, and engineering students.

Throughout the years, we have encountered many good books on microprocessors and embedded systems. However, most of these books share the same basic type of problems: they were written either for one specific microprocessor and their usefulness was limited in scope to the device they presented, or were developed without a specific target processor in mind lacking the practical side of teaching embedded systems. Aware of these realities, we have developed an introductory-level book that falls in the middle of these two extremes.

This book has been written to introduce the reader to the subjects of microprocessors and embedded systems, covering architectural issues, programming fundamentals, and the basics of interfacing. As part of the architectural aspects, it covers topics in processor organization, emphasizing the typical structure of today's microcontrollers, processor models, and programming styles. It also covers fundamentals on computer data representations and operations, as a prelude to subjects in embedded software development. The presented material is rounded off with discussions that cover from the basics of input/output systems to using all

sorts of embedded peripherals and interfacing external loads for diverse applications.

Most practical examples use Texas Instruments MSP430 devices, an intuitive and affordable platform. But the book is not limited to the scope of this microcontroller. Each chapter has been filled with concepts and practices that set a solid design foundation in embedded systems which is independent from the specific device being used. This material is then followed by a discussion of how these concepts apply to particular MSP430 devices. This allows our readers to first build a solid foundation in the underlying functional and design concepts and then to put them in practice with a simple and yet powerful family of devices.

## Book Organization

The book contents are distributed across ten chapters as described below:

**Chapter 1:** *Introduction* brings to the reader the concept of an embedded system. The chapter provides a historical overview of the development of embedded systems, their structure, classification, and complete life cycle. The last section discusses typical constraints such as functionality, cost, power, and time to market, among others. More than just a mere introduction, the chapter provides a system-level treatment of the global issues affecting the design of embedded systems, bringing awareness to designers about the implications of their design decisions.

**Chapter 2:** *Number Systems and Data Formats* reviews basic concepts on number formats and representations for computing systems. The discussion includes subjects in number systems, base conversions, arithmetic operations, and numeric and nonnumeric representations. The purpose of dedicating a whole chapter to this subject is to reinforce in a comprehensive way the student background in the treatment given by computers to numbers and data.

**Chapter 3:** *Microcomputer Organization* covers the topics of architecture and organization of classical microcomputer models. It goes from the very basic concepts in organization and architecture, to models of processors and microcontrollers. The chapter also introduces the Texas Instruments MSP430 family of microcontrollers as the practical target for applications in the book. This chapter forms the ground knowledge for absolute newcomers to the field of microprocessor-based systems.

**Chapter 4:** *Assembly Language Programming* sets a solid base in microprocessor's programming at the most fundamental level: Assembly Language. The anatomy of an assembly program is discussed, covering programming techniques and tips, and illustrating the application cases where programming in assembly language becomes prevalent. The chapter is crowned with the assembly programmer's model of the MSP430 as a way to bring-in the practical side of the assembly process. We use the IAR assembler in this chapter. Yet, the discussion is

complemented by Appendix C with a tutorial on how to use Code Composer Studio<sup>1</sup> (CCS) for programming the MSP430 in assembly language.

**Chapter 5:** *C Language Programming* treats the subject of programming embedded systems using a high-level language. The chapter reviews fundamental programming concepts to then move to a focussed discussion on how to program the MSP430 in C language. Like in **Chap. 4**, the discussion makes use of IAR and other tools as programming and debugging environments.

**Chapter 6:** *Fundamentals of Interfacing* is structured to guide first time embedded systems designers through the process of making a microprocessor or microcontroller chip work from scratch. The chapter identifies the elements in the basic interface of a microprocessor-based system, defining the criteria to implement each of them. The discussion is exemplified with a treatment of the MSP430 as target device, with an in-depth treatment of its embedded modules and how they facilitate the basic interface in a wide range of applications.

**Chapter 7:** *Embedded Peripherals* immerses readers into the array of peripherals typically found in a microcontroller, while also discussing the concepts that allow for understanding how to use them in any MCU- or microprocessor-based system. The chapter begins by discussing how to use interrupts and timers in microcontrollers as support peripherals for other devices. The subjects of using embedded FLASH memory and direct memory access are also discussed in this chapter, with special attention to their use as a peripheral device supporting low-power operation in embedded systems. The MSP430 is used as the testbed to provide practical insight into the usage of these resources.

**Chapter 8:** *External World Interface* discusses one of the most valuable resources in embedded microcontrollers: general purpose I/O lines. Beginning with an analysis of the structure, operation, and configuration of GPIO ports the chapter expands into developing user interfaces using via GPIOs. Specific MSP430 GPIO features and limitations are discussed to create the basis for safe design practices. The discussion is next directed at how to design hardware and software modules for interfacing large DC and AC loads, and motors through GPIO lines.

**Chapter 9:** *Principles of Serial Communication* offers an in-depth discussion of how serial interfaces work for supporting asynchronous and synchronous communication modalities. Protocols and hardware for using UARTs, SPI, I<sup>2</sup>C, USB, and other protocols are studied in detail. Specific examples using the different serial modules sported in MSP430 devices provide a practical coverage of the subject.

**Chapter 10:** *The Analog Signal Chain* bridges the digital world of microcontrollers to the domain of analog signals. A thorough discussion of the fundamentals of sensor interfacing, signal conditioning, anti-aliasing filtering, analog-to-digital

---

<sup>1</sup> CCS is a freely available integrated development environment provided by Texas Instruments at that allows for programming and debugging all members of the MSP430 family in both assembly and C language.

and digital-to-analog conversion, and smoothing and driving techniques are included in this chapter. Concrete hardware and software examples are provided, including both, Nyquist and oversampled converters embedded in MSP430 MCUs are included, making this chapter an essential unit for mixed-signal interfaces in embedded applications.

Each chapter features examples and problems on the treated subjects that range from conceptual to practical. A list of selected bibliography is provided at the end of the book for those interested in reading further about the topics discussed in each chapter.

Appendix materials provide information complementing the book chapters in diverse ways. Appendix A provides a brief guide to the usage of flowcharting to plan the structure of embedded programs. Appendix B includes a detailed MSP430 instruction set with binary instruction encoding and cycle requirements.

A detailed tutorial on how to use Code Composer Essentials, the programming and debugging environment for MSP430 devices, is included in Appendix C. The extended MSP430X architecture and its advantages with respect to the standard CPU version of the controller is the subject of Appendix D.

## Suggested Book Usage

This book has been designed for use in different scenarios that include undergraduate and graduate treatment of the subject of microprocessors and embedded systems, and as a reference for industry practitioners.

In the academic environment, teaching alternatives include EE, CE, and CS study programs with either a single comprehensive course in microprocessors and embedded systems, or a sequence of two courses in the subject. Graduate programs could use the material for a first-year graduate course in embedded systems design.

For a semester-long introductory undergraduate course in microprocessors, a suggested sequence could use [Chaps. 1–5](#) and selected topics from [Chaps. 6 and 8](#). A two-course sequence for quarter-based programs could use topics in [Chaps. 1 through 4](#) for an architecture and assembly programming-focused first course. The second course would be more oriented towards high-level language programming, interfacing, and applications, discussing select topics from [Chaps. 1 and 5–10](#).

Besides these suggested sequences, other combinations are possible depending on the emphasis envisioned by the instructor and the program focus.

Many EE and CE programs offer a structured laboratory in microprocessors and embedded systems, typically aligned with the first course in the subject. For this scenario, the book offers a complementary laboratory manual with over a dozen experiments using the MSP430. Activities are developed around MSP430 launchpad development kits. Instructions, schematics, and layouts are provided for building a custom designed I/O board, the eZ-EXP. Experiments in the lab manual are structured in a progressive manner that can be easily synchronized with the subjects taught in an introductory microprocessors course. All experiments can be

completed using exclusively the inexpensive launchpad or TI eZ430 USB boards, IAR and the freely available IAR or CCS environments, and optionally the eZ-EXP attachment.

For industry practitioners, this book could be used as a refresher on the concepts on microprocessors and embedded systems, or as a documented reference to introduce the use of the MSP430 microcontroller and tools.

## **Supplemental Materials**

Instructor's supplemental materials available through the book web site include solutions to selected problems and exercises and power point slides for lectures. The site also includes materials for students that include links to application examples and to sites elsewhere in the Web with application notes, downloadable tools, and part suppliers.

We hope this book could result as a useful tool for your particular learning and/or teaching needs in embedded systems and microprocessors.

Enjoy the rest of the book!

# Acknowledgments

We would like to thank Texas Instruments, Inc. for giving us access to the MSP430 technology that brings up the practical side of this book. In particular, our gratitude goes to the people in the MSP430 Marketing and Applications Group for their help and support in providing application examples and documentation during the completion of this project.

Completing this work would have not been possible without the help of many students and colleagues in the Electrical and Computer Engineering Department of the University of Puerto Rico at Mayagüez who helped us verifying examples, revising material, and giving us feedback on how to improve the book contents. Special thanks to Jose Navarro, Edwin Delgado, Jose A. Rodriguez, Jose J. Rodriguez, Abdiel Avilés, Dalimar Vélez, Angie Córdoba, Javier Cardona, Roberto Arias, and all those who helped us, but our inability to remember their names at the time of this writing just evidences of how old we have grown while writing this book. You all know we will always be grateful for your valuable help.

Manuel  
Rogelio  
Isidoro

# Contents

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>Introduction</b> . . . . .                               | 1  |
| 1.1      | Embedded Systems: History and Overview . . . . .            | 1  |
| 1.1.1    | Early Forms of Embedded Systems. . . . .                    | 1  |
| 1.1.2    | Birth and Evolution of Modern<br>Embedded Systems . . . . . | 3  |
| 1.1.3    | Contemporary Embedded Systems . . . . .                     | 6  |
| 1.2      | Structure of an Embedded System . . . . .                   | 7  |
| 1.2.1    | Hardware Components . . . . .                               | 8  |
| 1.2.2    | Software Components . . . . .                               | 9  |
| 1.3      | Classification of Embedded Systems . . . . .                | 11 |
| 1.4      | The Life Cycle of Embedded Designs . . . . .                | 13 |
| 1.5      | Design Constraints . . . . .                                | 14 |
| 1.5.1    | Functionality . . . . .                                     | 15 |
| 1.5.2    | Cost . . . . .  | 17 |
| 1.5.3    | Performance. . . . .  | 19 |
| 1.5.4    | Power and Energy. . . . .                                   | 22 |
| 1.5.5    | Time-to-Market . . . . .                                    | 24 |
| 1.5.6    | Reliability and Maintainability . . . . .                   | 26 |
| 1.6      | Summary . . . . .   | 28 |
| 1.7      | Problems . . . . .  | 29 |
| <b>2</b> | <b>Number Systems and Data Formats</b> . . . . .            | 31 |
| 2.1      | Bits, Bytes, and Words . . . . .                            | 31 |
| 2.2      | Number Systems. . . . .                                     | 32 |
| 2.2.1    | Positional Systems . . . . .                                | 33 |
| 2.3      | Conversion Between Different Bases. . . . .                 | 34 |
| 2.3.1    | Conversion from Base . . . . .                              | 34 |
| 2.3.2    | Conversion from Decimal to Base . . . . .                   | 35 |
| 2.3.3    | Binary, Octal and Hexadecimal Systems . . . . .             | 38 |
| 2.4      | Hex Notation for $n$ -bit Words . . . . .                   | 39 |
| 2.5      | Unsigned Binary Arithmetic Operations. . . . .              | 41 |
| 2.5.1    | Addition . . . . .  | 41 |
| 2.5.2    | Subtraction. . . . .  | 42 |

|          |   |           |
|----------|---|-----------|
| 2.5.3    | Subtraction by Complement's Addition . . . . .                      | 43        |
| 2.5.4    | Multiplication and Division . . . . .                               | 46        |
| 2.5.5    | Operations in Hexadecimal System . . . . .                          | 46        |
| 2.6      | Representation of Numbers in Embedded Systems . . . . .             | 48        |
| 2.7      | Unsigned or Nonnegative Integer Representation . . . . .            | 49        |
| 2.7.1    | Normal Binary Representation . . . . .                              | 49        |
| 2.7.2    | Biased Binary Representation . . . . .                              | 50        |
| 2.7.3    | Binary Coded Decimal Representation . . . . .                       | 51        |
| 2.8      | Two's Complement Signed Integers Representation . . . . .           | 53        |
| 2.8.1    | Arithmetic Operations with Signed Numbers<br>and Overflow . . . . . | 57        |
| 2.8.2    | Subtraction as Addition of Two's Complement . . . . .               | 58        |
| 2.9      | Real Numbers . . . . .  | 58        |
| 2.9.1    | Fixed-Point Representation . . . . .                                | 59        |
| 2.9.2    | Floating-Point Representation . . . . .                             | 62        |
| 2.10     | Continuous Interval Representations . . . . .                       | 65        |
| 2.10.1   | Encoding with $\frac{1}{2}$ LSB Offset Subintervals . . . . .       | 68        |
| 2.10.2   | Two's Complement Encoding of $[-a, a]$ . . . . .                    | 69        |
| 2.11     | Non Numerical Data . . . . .  | 70        |
| 2.11.1   | ASCII Codes . . . . .   | 70        |
| 2.11.2   | Unicode . . . . .   | 72        |
| 2.11.3   | Arbitrary Custom Codes . . . . .                                    | 73        |
| 2.12     | Summary . . . . .   | 74        |
| 2.13     | Problems . . . . .  | 75        |
| <b>3</b> | <b>Microcomputer Organization . . . . .</b>                         | <b>81</b> |
| 3.1      | Base Microcomputer Structure . . . . .                              | 81        |
| 3.2      | Microcontrollers Versus Microprocessors . . . . .                   | 82        |
| 3.2.1    | Microprocessor Units . . . . .                                      | 83        |
| 3.2.2    | Microcontroller Units . . . . .                                     | 83        |
| 3.2.3    | RISC Versus CISC Architectures . . . . .                            | 84        |
| 3.2.4    | Programmer and Hardware Model . . . . .                             | 85        |
| 3.3      | Central Processing Unit . . . . .                                   | 86        |
| 3.3.1    | Control Unit . . . . .  | 87        |
| 3.3.2    | Arithmetic Logic Unit . . . . .                                     | 88        |
| 3.3.3    | Bus Interface Logic . . . . .                                       | 89        |
| 3.3.4    | Registers . . . . .   | 89        |
| 3.3.5    | MSP430 CPU Basic Hardware Structure . . . . .                       | 95        |
| 3.4      | System Buses . . . . .  | 96        |
| 3.4.1    | Data Bus . . . . .  | 96        |
| 3.4.2    | Address Bus . . . . .   | 97        |
| 3.4.3    | Control Bus . . . . .   | 98        |
| 3.5      | Memory Organization . . . . .                                       | 98        |
| 3.5.1    | Memory Types . . . . .  | 99        |

|          |   |            |
|----------|---|------------|
| 3.5.2    | Data Address: Little and Big Endian Conventions . . . | 100        |
| 3.5.3    | Program and Data Memory . . . . .                     | 103        |
| 3.5.4    | Von Neumann and Harvard Architectures . . . . .       | 104        |
| 3.5.5    | Memory and CPU Data Exchange: An Example. . . .       | 105        |
| 3.5.6    | Memory Map . . . . .                                  | 107        |
| 3.5.7    | MSP430 Memory Organization . . . . .                  | 108        |
| 3.6      | I/O Subsystem Organization . . . . .                  | 109        |
| 3.6.1    | Anatomy of an I/O Interface . . . . .                 | 111        |
| 3.6.2    | Parallel Versus Serial I/O Interfaces . . . . .       | 115        |
| 3.6.3    | I/O and CPU Interaction . . . . .                     | 116        |
| 3.6.4    | Timing Diagrams . . . . .                             | 117        |
| 3.6.5    | MSP430 I/O Subsystem and Peripherals . . . . .        | 118        |
| 3.7      | CPU Instruction Set . . . . .                         | 119        |
| 3.7.1    | Register Transfer Notation . . . . .                  | 120        |
| 3.7.2    | Machine Language and Assembly Instructions . . . . .  | 122        |
| 3.7.3    | Types of Instructions . . . . .                       | 124        |
| 3.7.4    | The Stack and the Stack Pointer . . . . .             | 127        |
| 3.7.5    | Addressing Modes . . . . .                            | 132        |
| 3.7.6    | Orthogonal CPU's . . . . .                            | 138        |
| 3.8      | MSP430 Instructions and Addressing Modes . . . . .    | 138        |
| 3.9      | Introduction to Interrupts . . . . .                  | 138        |
| 3.9.1    | Device Services: Polling Versus Interrupts . . . . .  | 139        |
| 3.9.2    | Interrupt Servicing Events . . . . .                  | 141        |
| 3.9.3    | What is Reset? . . . . .                              | 143        |
| 3.10     | Notes on Program Design . . . . .                     | 143        |
| 3.11     | The TI MSP430 Microcontroller Family . . . . .        | 145        |
| 3.11.1   | General Features . . . . .                            | 145        |
| 3.11.2   | Naming Conventions . . . . .                          | 147        |
| 3.12     | Development Tools . . . . .                           | 149        |
| 3.12.1   | Internet Sites . . . . .                              | 149        |
| 3.12.2   | Programming and Debugging Tools . . . . .             | 149        |
| 3.12.3   | Development Tools . . . . .                           | 150        |
| 3.12.4   | Other Resources . . . . .                             | 151        |
| 3.13     | Summary . . . . .                                     | 151        |
| 3.14     | Problems . . . . .                                    | 152        |
| <b>4</b> | <b>Assembly Language Programming . . . . .</b>        | <b>155</b> |
| 4.1      | An Overview of Programming Levels . . . . .           | 155        |
| 4.1.1    | High-Level Code . . . . .                             | 156        |
| 4.1.2    | Machine Language . . . . .                            | 157        |
| 4.1.3    | Assembly Language Code . . . . .                      | 158        |
| 4.2      | Assembly Programming: First Pass . . . . .            | 159        |
| 4.2.1    | Directives . . . . .                                  | 160        |
| 4.2.2    | Labels in Instructions . . . . .                      | 164        |

- 4.2.3 Source Executable Code . . . . . 165
- 4.2.4 Source File: First Pass . . . . . 167
- 4.2.5 Why Assembly? . . . . . 169
- 4.3 Assembly Instruction Characteristics . . . . . 169
  - 4.3.1 Instruction Operands . . . . . 170
  - 4.3.2 Word and Byte Instructions . . . . . 173
  - 4.3.3 Constant Generators . . . . . 173
- 4.4 Details on MSP430 Instruction Set . . . . . 175
  - 4.4.1 Data Transfer Instructions . . . . . 176
  - 4.4.2 Arithmetic Instructions . . . . . 178
  - 4.4.3 Logic and Register Control Instructions. . . . . 181
  - 4.4.4 Program Flow Instructions . . . . . 188
- 4.5 Multiplication and Division . . . . . 194
  - 4.5.1 16 Bit Hardware Multiplier . . . . . 195
  - 4.5.2 Remarks on Hardware Multiplier . . . . . 198
- 4.6 Macros . . . . . 198
- 4.7 Assembly Programming: Second Pass . . . . . 200
  - 4.7.1 Relocatable Source . . . . . 200
  - 4.7.2 Data and Variable Memory Assignment  
and Allocation . . . . . 202
  - 4.7.3 Interrupt and Reset Vector Allocation . . . . . 204
  - 4.7.4 Source Executable Code . . . . . 206
- 4.8 Modular and Interrupt Driven Programming. . . . . 207
  - 4.8.1 Subroutines . . . . . 207
  - 4.8.2 Placing Subroutines in Source File . . . . . 212
  - 4.8.3 Resets, Interrupts, and Interrupt Service Routines . . . . . 213
- 4.9 Summary . . . . . 215
- 4.10 Problems . . . . . 216
  
- 5 Embedded Programming Using C . . . . . 219**
  - 5.1 High-Level Embedded Programming. . . . . 219
    - 5.1.1 (C-Cross Compilers Versus C-Compilers)  
or (Remarks on Compilers)? . . . . . 220
  - 5.2 C Program Structure and Design. . . . . 222
    - 5.2.1 Preprocessor Directives . . . . . 224
    - 5.2.2 Constants, Variables, and Pointers . . . . . 224
    - 5.2.3 Program Statements . . . . . 229
    - 5.2.4 Struct . . . . . 233
  - 5.3 Some Examples . . . . . 233
  - 5.4 Mixing C and Assembly Code . . . . . 238
    - 5.4.1 Inlining Assembly Language Instructions. . . . . 239
    - 5.4.2 Incorporating with Intrinsic Functions . . . . . 239
    - 5.4.3 Mixed Functions. . . . . 242

- 5.4.4 Linking Assembled Code with Compiled C Programs . . . . . 243
- 5.4.5 Interrupt Function Support . . . . . 245
- 5.5 Using MSP430 C-Library Functions . . . . . 245
- 5.6 Summary . . . . . 246
- 5.7 Problems . . . . . 246
  
- 6 Fundamentals of Interfacing . . . . . 249**
  - 6.1 Elements in a Basic MCU Interface . . . . . 249
  - 6.2 Processors' Power Sources . . . . . 250
    - 6.2.1 Power Supply Structure . . . . . 252
    - 6.2.2 Power-Speed Tradeoff . . . . . 253
    - 6.2.3 Power Supply Capacity . . . . . 253
    - 6.2.4 Power Supply Distribution Tips . . . . . 256
  - 6.3 Clock Sources . . . . . 260
    - 6.3.1 Clock Parameters . . . . . 261
    - 6.3.2 Choosing a Clock Source . . . . . 264
    - 6.3.3 Internal Versus External Clock Sources . . . . . 265
    - 6.3.4 Quartz Crystal Oscillators . . . . . 266
  - 6.4 The MSP430 System Clock . . . . . 270
    - 6.4.1 The FLL and FLL+ Clock Modules . . . . . 271
    - 6.4.2 The Basic Clock Module and Module+ . . . . . 274
    - 6.4.3 The Unified Clock System Sources . . . . . 276
    - 6.4.4 Using the MSP430 Clock . . . . . 279
  - 6.5 Power-On Reset . . . . . 282
    - 6.5.1 Power-On Reset Specifications . . . . . 282
    - 6.5.2 Practical POR Circuits . . . . . 283
  - 6.6 Bootstrap Sequence . . . . . 287
    - 6.6.1 Boot Sequence Operation . . . . . 288
    - 6.6.2 Reset Address Identification . . . . . 288
  - 6.7 Reset Handling in MSP430 Devices . . . . . 290
    - 6.7.1 Brownout Reset . . . . . 291
    - 6.7.2 Initial Conditions . . . . . 292
    - 6.7.3 Reset Circuitry Evolution in MSP430 Devices . . . . . 293
  - 6.8 Summary . . . . . 295
  - 6.9 Problems . . . . . 295
  
- 7 Embedded Peripherals . . . . . 299**
  - 7.1 Fundamental Interrupt Concepts . . . . . 299
    - 7.1.1 Interrupts Triggers . . . . . 300
    - 7.1.2 Maskable Versus Non-Maskable Interrupts . . . . . 301
    - 7.1.3 Interrupt Service Sequence . . . . . 303
    - 7.1.4 Interrupt Identification Methods . . . . . 304
    - 7.1.5 Priority Handling . . . . . 306

- 7.2 Interrupt Handling in the MSP430 . . . . . 310
  - 7.2.1 System Resets . . . . . 311
  - 7.2.2 (Non)-Maskable Interrupts . . . . . 311
  - 7.2.3 Maskable Interrupts. . . . . 312
- 7.3 Interrupt Software Design . . . . . 315
  - 7.3.1 Fundamental Interrupt Programming Requirements . . . . . 315
  - 7.3.2 Examples of Interrupt-Based Programs . . . . . 316
  - 7.3.3 Multi-Source Interrupt Handlers . . . . . 318
  - 7.3.4 Dealing with False Triggers . . . . . 322
  - 7.3.5 Interrupt Latency and Nesting . . . . . 323
  - 7.3.6 Interrupts and Low-Power Modes . . . . . 324
  - 7.3.7 Working with MSP430 Low-Power Modes . . . . . 327
- 7.4 Timers and Event Counters . . . . . 330
  - 7.4.1 Base Timer Structure . . . . . 330
  - 7.4.2 Fundamental Applications: Interval Timer and Event Counter . . . . . 332
  - 7.4.3 Signature Timer Applications. . . . . 336
  - 7.4.4 MSP430 Timer Support. . . . . 340
- 7.5 Embedded Memory Technologies . . . . . 356
  - 7.5.1 A Classification of Memory Technologies . . . . . 356
  - 7.5.2 Flash Memory: General Principles . . . . . 359
  - 7.5.3 MSP430 Flash Memory and Controller . . . . . 361
- 7.6 Bus Arbitration and DMA Transfers . . . . . 364
  - 7.6.1 Fundamental Concepts in Bus Arbitration . . . . . 364
  - 7.6.2 Direct Memory Access Controllers . . . . . 367
  - 7.6.3 MSP430 DMA Support . . . . . 374
- 7.7 Chapter Summary . . . . . 379
- 7.8 Problems . . . . . 380
  
- 8 External World Interfaces . . . . . 383**
  - 8.1 General Purpose Input-Output Ports . . . . . 383
    - 8.1.1 Structure of an Input-Output Pin. . . . . 384
    - 8.1.2 Configuring GPIO Ports . . . . . 386
    - 8.1.3 GPIO Ports in MSP430 Devices . . . . . 388
  - 8.2 Interfacing External Devices to GPIO Ports . . . . . 390
    - 8.2.1 Electrical Characteristics in I/O Pins . . . . . 391
    - 8.2.2 Schmitt-Trigger Inputs. . . . . 395
    - 8.2.3 Output Limits in I/O Pins . . . . . 397
    - 8.2.4 Maximum Power and Thermal Dissipation. . . . . 399
    - 8.2.5 Switching Characteristics of I/O Pins . . . . . 402
  - 8.3 Interfacing Switches and Switch Arrays. . . . . 402
    - 8.3.1 Switch Types . . . . . 403
    - 8.3.2 Working with Switches and Push-Buttons . . . . . 403

- 8.3.3 Dealing with Real Switches . . . . . 405
- 8.3.4 Characteristics of Switch Bounce . . . . . 406
- 8.3.5 Problems Associated to Contact Bounce . . . . . 406
- 8.3.6 Switch Debouncing Techniques . . . . . 407
- 8.3.7 Hardware Debouncing Techniques . . . . . 407
- 8.3.8 Software Debouncing Techniques . . . . . 410
- 8.3.9 Switch Arrays. . . . . 414
- 8.3.10 Linear Switch Arrays . . . . . 414
- 8.3.11 Interfacing Keypads . . . . . 415
- 8.4 Interfacing Display Devices to Microcontrollers . . . . . 419
- 8.5 Interfacing LEDs and LED-Based Displays to MCUs . . . . . 420
  - 8.5.1 Interfacing Discrete LEDs . . . . . 421
  - 8.5.2 Interfacing LED Arrays . . . . . 422
- 8.6 Interfacing Liquid Crystal Displays . . . . . 426
  - 8.6.1 Principles of Operation of LCDs. . . . . 426
  - 8.6.2 Control Signals for LCDs . . . . . 428
  - 8.6.3 Direct LCD Drivers . . . . . 430
  - 8.6.4 Multiplexed LCD Drivers . . . . . 431
  - 8.6.5 Active Matrix LCDs . . . . . 434
  - 8.6.6 Commercial LCD Modules . . . . . 436
- 8.7 LCD Support in MSP430 Devices. . . . . 439
  - 8.7.1 MSP430x4xx LCD Controllers . . . . . 440
  - 8.7.2 MSP430x6xx LCD Controllers . . . . . 442
- 8.8 Managing Large DC Loads and Voltages. . . . . 444
  - 8.8.1 Using Transistor Drivers . . . . . 444
  - 8.8.2 Using Driver ICs . . . . . 448
- 8.9 Managing AC Loads . . . . . 449
  - 8.9.1 Mechanical Relays . . . . . 449
  - 8.9.2 Solid-State Relays. . . . . 451
  - 8.9.3 Opto-Detectors and Opto-Isolators . . . . . 452
- 8.10 MCU Motor Interfacing. . . . . 453
  - 8.10.1 Working with DC Motors . . . . . 453
  - 8.10.2 Servo Motor Interfacing. . . . . 455
  - 8.10.3 Stepper Motor Interfaces . . . . . 456
  - 8.10.4 Variable Reluctance Stepper Motors . . . . . 456
  - 8.10.5 Permanent Magnet Stepper Motors . . . . . 460
  - 8.10.6 Hybrid Stepper Motors . . . . . 466
  - 8.10.7 Microstepping Control. . . . . 467
  - 8.10.8 Software Considerations in Stepper  
Motor Interfaces . . . . . 468
  - 8.10.9 Choosing a Motor: DC Versus Servo  
Versus Stepper . . . . . 471
- 8.11 Summary . . . . . 473
- 8.12 Problems . . . . . 473

- 9 Principles of Serial Communication . . . . . 475**
  - 9.1 Data Communications Fundamental . . . . . 475
  - 9.2 Types of Serial Channels . . . . . 477
  - 9.3 Clock Synchronization in Serial Channels . . . . . 479
    - 9.3.1 Asynchronous Serial Channels . . . . . 479
    - 9.3.2 Asynchronous Packet Format . . . . . 480
    - 9.3.3 Error Detection Mechanism . . . . . 481
    - 9.3.4 Asynchronous Communication Interface Modules. . . . . 482
    - 9.3.5 UART Structure and Functionality . . . . . 482
    - 9.3.6 UART Interface . . . . . 484
    - 9.3.7 UART Configuration and Operation . . . . . 487
    - 9.3.8 Physical Standards for Serial Channels . . . . . 489
    - 9.3.9 The Universal Serial Bus (USB). . . . . 494
  - 9.4 Synchronous Serial Communication . . . . . 498
    - 9.4.1 The Serial Peripheral Interface Bus: SPI . . . . . 499
    - 9.4.2 The Inter-Integrated Circuit Bus: I<sup>2</sup>C . . . . . 501
  - 9.5 Serial Interfaces in MSP430 MCUs. . . . . 507
    - 9.5.1 The MSP430 USART . . . . . 507
    - 9.5.2 The MSP430 USI . . . . . 516
    - 9.5.3 The MSP430 USCI. . . . . 520
    - 9.5.4 The MSP430 Enhanced USCI . . . . . 533
  - 9.6 Chapter Summary . . . . . 535
  - 9.7 Problems . . . . . 535
  
- 10 The Analog Signal Chain . . . . . 537**
  - 10.1 Introduction to Analog Processing . . . . . 537
  - 10.2 Analog Signal Chain in MSP430 . . . . . 538
  - 10.3 Sensors and Signal Conditioning. . . . . 539
    - 10.3.1 Active and Passive Sensors . . . . . 539
    - 10.3.2 Figures of Merit for Sensors . . . . . 540
  - 10.4 Operational Amplifiers . . . . . 544
    - 10.4.1 Basic OA Linear Configurations. . . . . 545
    - 10.4.2 Antialiasing Filter . . . . . 551
    - 10.4.3 Operational Amplifiers in MSP430 . . . . . 551
  - 10.5 Comparators. . . . . 552
    - 10.5.1 Internal Comparators in MSP430 (A and A+) . . . . . 556
  - 10.6 Analog-to-Digital and Digital-to-Analog: An Overview. . . . . 558
  - 10.7 Sampling and Quantization Principles . . . . . 561
    - 10.7.1 Sampling and Nyquist Principle . . . . . 561
    - 10.7.2 Sampling and Aliasing . . . . . 562
    - 10.7.3 Quantization. . . . . 563
    - 10.7.4 Quantization and Noise . . . . . 567
    - 10.7.5 Quantization and Hardware Imperfections . . . . . 568
    - 10.7.6 Sample & Hold Circuits . . . . . 570

|  |  |            |
|--|--|------------|
| 10.8   | Digital-to-Analog Converters . . . . .       | 572        |
| 10.8.1   | R-2R Ladder . . . . .                        | 572        |
| 10.8.2   | Smoothing Filters . . . . .                  | 575        |
| 10.8.3   | Using MSP430 DACs . . . . .                  | 575        |
| 10.9   | Analog-to-Digital Converters . . . . .       | 578        |
| 10.9.1   | Slope ADC . . . . .                          | 578        |
| 10.9.2   | Successive Approximation ADC . . . . .       | 579        |
| 10.9.3   | Oversampled Converter: Sigma-Delta . . . . . | 582        |
| 10.9.4   | Closing Remarks . . . . .                    | 584        |
| 10.9.5   | Using the MSP430 ADCs . . . . .              | 585        |
| 10.10  | Chapter Summary . . . . .                    | 593        |
| 10.11  | Problems . . . . .                           | 594        |
| <b>Appendix A: Software Planning Using Flowcharts . . . . .</b>                |  | <b>597</b> |
| <b>Appendix B: MSP430 Instruction Set . . . . .</b>                            |  | <b>605</b> |
| <b>Appendix C: Code Composer Introduction Tutorial . . . . .</b>               |  | <b>615</b> |
| <b>Appendix D: CPUX: An Overview of the Extended<br/>MSP430X CPU . . . . .</b> |  | <b>623</b> |
| <b>Appendix E: Copyright Notices . . . . .</b>                                 |  | <b>629</b> |
| <b>References . . . . .</b>  |  | <b>631</b> |
| <b>Author Biography . . . . .</b>  |  | <b>635</b> |
| <b>Index . . . . .</b>   |  | <b>637</b> |