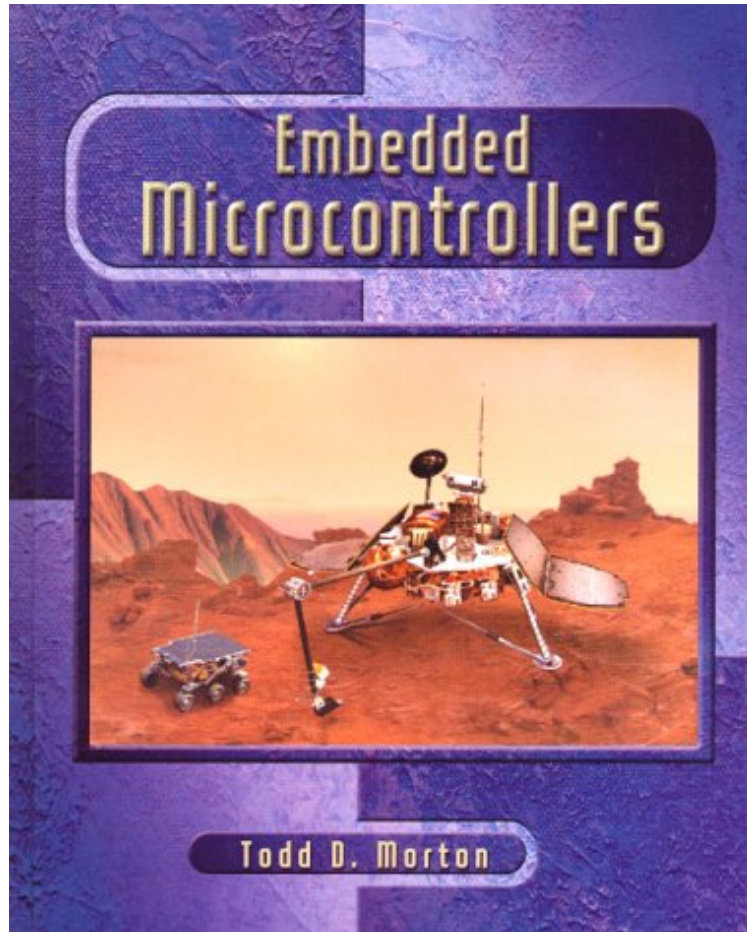


# Embedded Microcontrollers

*Todd D. Morton*

*ePub | \*DOC | audiobook | ebooks | Download PDF*



[Download](#)

[Read Online](#)

#2382072 in Books 2000-09-17Ingredients: Example IngredientsOriginal language:EnglishPDF # 1 9.10 x 1.60 x 7.50l, 2.60 #File Name: 0139075771694 pages | File size: 29.Mb

**Todd D. Morton : Embedded Microcontrollers** before purchasing it in order to gage whether or not it would be worth my time, and all praised Embedded Microcontrollers:

0 of 0 people found the following review helpful. GoodBy SpencerArrived in better condition than I expected it to be in. Pretty easy to follow and definitely something someone should pick up if they are interested in MCU's2 of 2 people found the following review helpful. Difficult for a beginnerBy aje3892I used this book for a beginner 68HC12 class at my college and I thought it was extremely hard to understand. Most of the examples involve C programming and at the time I hadn't taken a course on C so I didn't know what it was talking about. I think motorola has a lot of useful books if you want to program using assembly language. They have a small black book that goes over all the instructions and then also another book that goes over all the registers. I also beleive this book is wickedly expensive.5 of 8 people found the following review helpful. Outstanding book on embedded systemsBy Jean J. Labrosse"Embedded Microcontrollers" is an outstanding book for anyone who wishes to understand the inner workings of embedded systems. This book covers both software and hardware aspects and Mr. Morton does an

excellent job in presenting the material with ample illustrations, tables, timing diagrams, schematics and lots of practical code examples. This book will certainly become a favorite in Colleges and Universities. If you're into embedded system design, you need this book on your bookshelf!

This practical book on designing real-time embedded systems using 8-and 16-bit microcontrollers covers both assembly and C programming and real-time kernels. Using a large number of specific examples, it focuses on the concepts, processes, conventions, and techniques used in design and debugging. Chapter topics include programming basics; simple assembly code construction; CPU12 programming model; basic assembly programming techniques; assembly program design and structure; assembly applications; real-time I/O and multitasking; microcontroller I/O resources; modular and C code construction; creating and accessing data in C; real-time multitasking in C; and using the MICROC/OS-II preemptive kernel. For anyone who wants to design small- to medium-sized embedded systems.

From the Back Cover This text provides a comprehensive look at embedded microcontrollers specialized devices used for a variety of applications and which include a memory, a microprocessor, and an I/O interface. Embedded Microcontrollers is one of the first books based on the Motorola MC68HC12. It is for use in any microcontroller course covering embedded systems, or for anyone who wants to design small to medium-sized embedded systems. From a wealth of practical experience, the author has prepared a work that fills a significant void, both for learning the new MC68HC12 and for learning the fundamentals of embedded real-time applications. For the novice or the experienced, this text provides essential, up-to-date information as well as practical tips all in a format and style that are engaging and easily understood. Excerpt. Reprinted by permission. All rights reserved. Preface This book is intended for anyone who wants to design small- to medium-sized embedded systems. It was written primarily for electronics engineering technology students, but should also be appropriate for most engineering students and practicing engineers. It is currently being used for two one-quarter courses in electronics engineering technology. The first is a required junior-level course on microprocessor-based applications, and the second is a senior elective course on embedded systems. The first course covers assembly language only, and the second course uses C. Prerequisites for this book include a background in electronic circuits, basic digital logic, and general C programming. The C programming course is required only for Parts 4 and 5. A background in C and/or C++ is required because this book does not cover C language programming basics. It only addresses the concepts and techniques required for using C in a microcontroller-based embedded system. SOFTWARE AND HARDWARE USED IN THE BOOK When writing a book on embedded systems, it is difficult to maintain the balance between being too specific to one hardware/software set and too general to be practical. I have tried to keep this balance by focusing on the concepts, processes, conventions, and techniques used in design and debugging. The book focuses on the M68HC12 microcontroller from Motorola, but has also been successfully used in courses that use the M68HC 11 family. It is intended to supplement, not replace, vendor documentation. I expect every student to have the complete vendor documentation for both the MCU and the development board he or she is using. For example, for the M68HC12 MCU the student should have the CPU12 Reference Manual and the Technical Specifications for the specific part. If the M68HC 11 is used, the famous "pink books" are a requirement. The development hardware used throughout the text is the Motorola 68HC912B32 EVB. In the first part of the text, only a single board is required, and all code is loaded into RAM. In the second half of the book, the background debug system is used, and the code is loaded into the target Flash ROM. This requires either two EVBs or one EVB as a target and a 68HC12 BDM pod such as the Noral 68HC 12 BDM debugger. The concepts regarding the debug process and testing should apply to most modern development systems. The development software used throughout is the Introl-CODE development system. Except for the sections that specifically address development using the Introl-CODE system, the C code is all ANSI-C, and therefore it would be reasonable to use another compiler. The real-time kernel covered in this text is MicroC/OS-II. It is available in source form and is widely used. Many of the concepts apply to other kernels, especially the applications for typical kernel services. CHAPTER DESCRIPTIONS The book is divided into five parts. The first half of the book emphasizes assembly code and the second half focuses on C. Hardware is covered throughout the text, especially in Part 3. Because of the use of pseudo-C from Chapter 6 on, it is reasonable to cover that material with an emphasis on C instead of assembly. Part 1 Introduction. This part introduces the reader to the background and perspective required in learning about embedded systems. Part 2 Assembly Language Programming. This part introduces programming in assembly language and covers the CPU 12 programming model and program design. Students should be able to build a prewritten program after completing Chapter 3, and write complete programs that are executed by the D-Bug 12 monitor in RAM after completing Chapter 6. Chapter 7 covers some basic applications that are appropriate for assembly code. Part 3 Microcontroller Hardware and I/O. Real-time concepts and I/O hardware are introduced, including interrupts and basic multitasking. All of the 68HC912B32 I/O resources are covered with the exception of the BDLC. MCU configuration for stand-alone systems along with bus expansion are covered in Chapters 10 and 11. Part 4 Programming Microcontrollers in C. Concepts in C for programming real-time embedded systems are covered. Emphasis is placed on memory usage and program efficiency appropriate for small MCUs. Part 5 Real-Time

Multitasking Kernels. In this part basic multitasking design is covered along with using MicroC/OS-II, an off-the-shelf kernel. ACKNOWLEDGMENTS Without the help and patience of the following people, this text would not have been possible: Rich Pennington at Intral Corp, Jim Sibigtroth at Motorola University and Austin Community College, Jean Labrosse at Micrium, Tony Plutino and Dave Hyder at Motorola, Marsh Faber and Mel Downs at Hewlett-Packard, Phil Meek and Harry Erickson at Noral Micrologics, Dave Garza and staff at Prentice-Hall, Kathleen Kitto and Andrew Pace at WWU, George Sweiss at ITT Technical Institute, Malvern Phillips at British Columbia Institute of Technology, the contributors to the Motorola 68HC 11 and 68HC 12 listserv, and, of course, all of my students. You will finally get those labs and homework assignments graded.