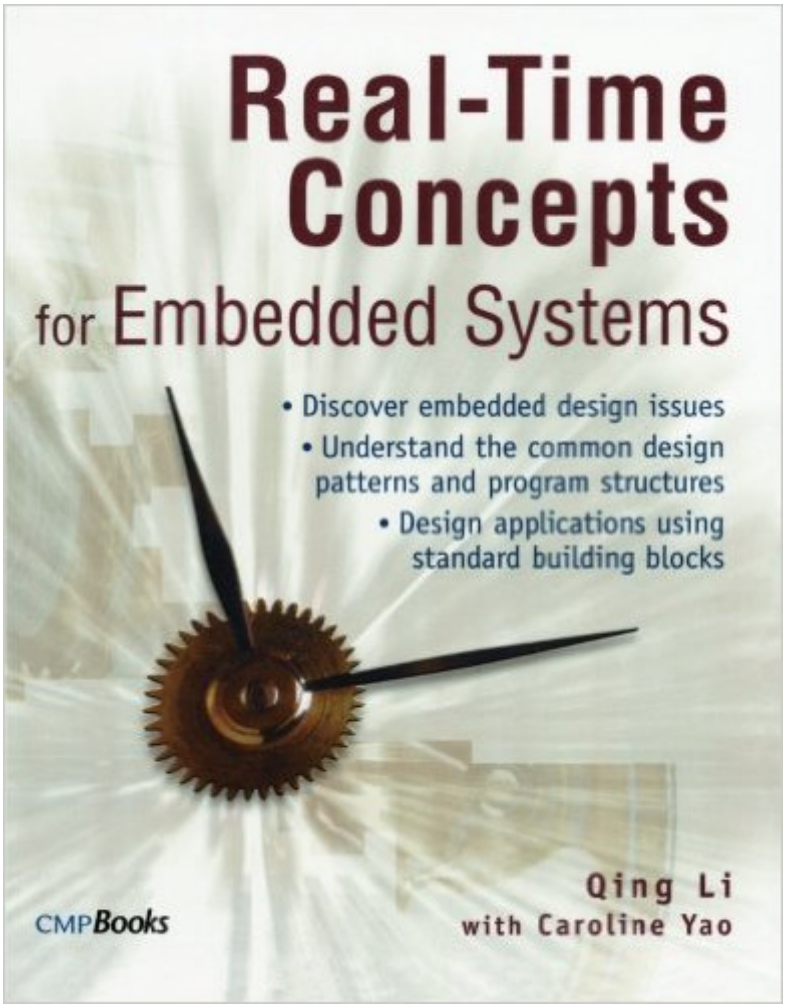
Real-Time Concepts For Embedded Systems



Synopsis

'... a very good balance between the theory and practice of real-time embedded system designs.' â ”Jun-ichiro itojun Hagino, Ph.D., Research Laboratory, Internet Initiative Japan Inc., IETF IPv6 Operations Working Group (v6ops) co-chair 'A cl

Book Information

Paperback: 306 pages

Publisher: CRC Press; 1st edition (July 2003)

Language: English

ISBN-10: 1578201241

ISBN-13: 978-1578201242

Product Dimensions: 7 x 0.7 x 9 inches

Shipping Weight: 1.3 pounds (View shipping rates and policies)

Average Customer Review: 3.8 out of 5 starsÂ Â See all reviewsÂ (25 customer reviews)

Best Sellers Rank: #200,272 in Books (See Top 100 in Books) #18 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Embedded Systems #132 inÂ Books > Computers & Technology > Databases & Big Data > Data Processing #222 inÂ Books > Textbooks > Computer Science > Operating Systems

Customer Reviews

When I first started reading this work, it seemed decidedly to be for beginners only. I was pleasantly surprised however, how many new concepts and ideas I learned by the time I completed it. I found the "points to remember" and Chapters 11, 13-16 to be particularly useful because they highlight the pitfalls that I've seen even the most seasoned RTOSers fall into. I ended up buying the book for all my firmware developers and made the above mentioned sections mandatory reading for them. Although I have read several books on real time this is definitely the most complete because it references the real time concepts in the context of a real live OS - VxWorx. The information is presented in such a way that it can easily be applied to any RTOS such as QNX, InTime, MsCE, and MuOS. All my firmware new hires are given this book on the first day of work, and they can't start coding until they've finished it.

This book is a good introduction to readers who are beginners in the world of embedded programming and real-time operating systems. A real-time OS facilitates the creation of real-time systems, but does not guarantee that they are real-time. Nor does an RTOS necessarily have high

throughput. Instead, they enable, through specialized scheduling algorithms and deterministic behavior, the guarantee that system deadlines can be met. Thus a real-time OS is valued more for how quickly it can respond to an event than for the total amount of work it can do.This book, written by a senior architect at Wind River Systems, is very VxWorks centric. This is OK, since VxWorks is one of the better commercial real-time operating systems out there, having been on the scene in various forms since 1983. This book will introduce you to all of the terminology you will need to be familiar with before you write code that uses VxWorks. I highly recommend it for anybody who is or is planning to get involved in embedded systems. Since so many robot designers end up programming with VxWorks, I highly recommend it especially to them. The concepts of scheduling, tasks, semaphores, queues, exceptions, and timers are all covered very well, even for those readers who have no previous exposure to operating systems in general. There is plenty of pseudocode and instructive diagrams. However, when it comes to actually writing an embedded system with VxWorks, no book that I know of actually gives out that kind of knowledge. There is no substitute to attaching yourself to a person experienced in this sort of work and learning by example. But first, read this book so that you understand what it is that they are teaching you. For real newbies, read "Designing Embedded Hardware" to get an idea of what devices make up an embedded system, and then "Programming Embedded Systems with C and C++" to get an idea of how these devices are controlled. At that point, you will be ready for this book.

"Real-Time Concepts" is a book that tackles one of the most difficult subject areas of embedded systems programming. Bugs that are introduced because of problems with the behavior of a real-time system can be extremely subtle and difficult to correct. These bugs can also have disasterous effects. Even Computer Science students who take an Operating Systems class probably never deal with a Real Time Operating System.In this book, the authors attempt to explain the concepts of real time programming. In my opinion, they generally succeed at their task. They keep the code examples to a bare minimum, which I think is a good strategy, because code examples can often obscure a concept, rather than explain it.If the book has a weakness, I think that it is the introductory sections. The first 50 pages or so cover no new ground and have been covered much more extensively by other authors in the CMP Books series. However, the book really hits its stride at Chapter 4, "Introduction to Real Time Operating Systems". From then on, the book teaches you what you will need to know to be able to design a real-time system.Of particular note, I was impressed with the authors' explanation of Rate Monotonic Analysis (RMA). This is one of the more difficult to understand concepts and the authors explained it in a way that the novice

reader could understand it.In summary, "Real-Time Concepts" should be required reading for anyone entering the field of embedded systems design. For the CS professional who may be new to embedded systems, investing the time to read this book before you start to code is time well-spent. Thid book should be part of every Embedded Systems professional’s bookshelf.

Well-written book which offers a "one-book-says-it-all" look at both embedded concepts and real-time design. The books is definitely recommended reading if undertaking real-time embedded design for the first time. The first few chapters offers pratical hands-on explanations with references to an actual embedded OS. The book then goes on to do an excellent job of explaining typical RTOS concepts such scheduling, concurrency, preventing deadlocks and handling priority inversion. It discusses how to incorporate realtime concepts into the embedded project and it identifies considerations that may affect your design. Good reading.

This book is a good introduction to the world of designing and building embedded systems. Chapter 3’s discussion on context switching and scheduling in RTOSes is quite useful. A suggestion - the discussion could also have included some of the newer scheduling techniques like rate monotonic scheduling, etc. I also found the detailed explanation of mutexes and semaphores in Chapter 6 to be of value. One feeling I had is that more examples using popular RTOSes like VxWorksÃ Â™ could have been included.I liked the authors’ summarization of the key points at the end of each chapter. Overall, a good book that I would recommend for an embedded designer’s bookshelf.

for Embedded and Real-Time Systems Real-Time Software Design for Embedded Systems Real-Time Embedded Components and Systems with Linux and RTOS (Engineering) Real-Time Embedded Components And Systems: With Linux and RTOS Applied Control Theory for Embedded Systems (Embedded Technology) Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit Analog Interfacing to Embedded Microprocessor Systems, Second Edition (Embedded Technology Series) Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers (Embedded Technology)