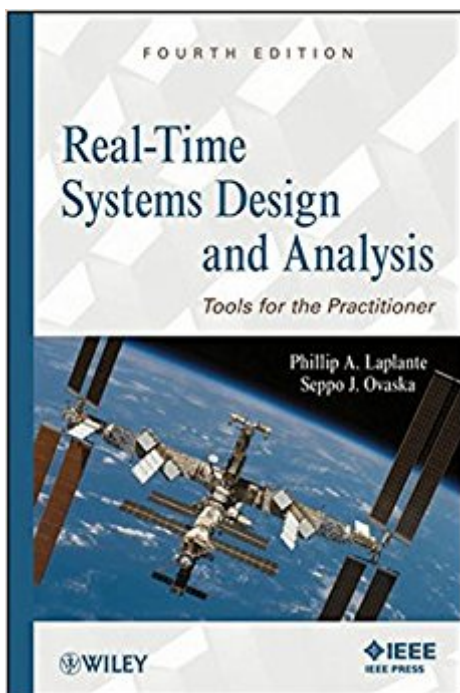The book was found

Real-Time Systems Design And Analysis: Tools For The Practitioner

Synopsis

The leading text in the field explains step by step how to write software that responds in real time From power plants to medicine to avionics, the world increasingly depends on computer systems that can compute and respond to various excitations in real time. The Fourth Edition of Real-Time Systems Design and Analysis gives software designers the knowledge and the tools needed to create real-time software using a holistic, systems-based approach. The text covers computer architecture and organization, operating systems, software engineering, programming languages, and compiler theory, all from the perspective of real-time systems design. The Fourth Edition of this renowned text brings it thoroughly up to date with the latest technological advances and applications. This fully updated edition includes coverage of the following concepts: Multidisciplinary design challenges   Time-triggered architectures   Architectural advancements Automatic code generation   Peripheral interfacing   Life-cycle processes   The final chapter of the text offers an expert perspective on the future of real-time systems and their applications. The text is self-contained, enabling instructors and readers to focus on the material that is most important to their needs and interests. Suggestions for additional readings guide readers to more in-depth discussions on each individual topic. In addition, each chapter features exercises ranging from simple to challenging to help readers progressively build and fine-tune their ability to design their own real-time software programs. Now fully up to date with the latest technological advances and applications in the field, Real-Time Systems Design and Analysis remains the top choice for students and software engineers who want to design better and faster real-time systems at minimum cost.

Book Information

Hardcover: 584 pages

Publisher: Wiley-IEEE Press; 4 edition (November 22, 2011)

Language: English

ISBN-10: 0470768649

ISBN-13: 978-0470768648

Product Dimensions: 6.4 x 1.5 x 9.1 inches

Shipping Weight: 2 pounds (View shipping rates and policies)

Average Customer Review: 4.6 out of 5 stars   16 customer reviews

Best Sellers Rank: #680,032 in Books (See Top 100 in Books)   #140 inÂ Books > Science & Math > Physics > Waves & Wave Mechanics   #619 inÂ Books > Computers & Technology > Databases

Customer Reviews

PHILLIP A. LAPLANTE, PhD, PE, is Professor of Software Engineering at Penn State, where he specializes in software and systems engineering, project management, and software testing and security. Dr. Laplante spent several years as a software engineer and project manager working on avionics, computer-aided design, and software test systems. He has authored or edited twenty-seven books and has published more than 200 scholarly articles. SEPPO J. OVASKA, DSc, is Professor of Industrial Electronics at Aalto University, Finland. He has served as a visiting scholar at Utah State University, Virginia Tech, and the University of Passau, Germany, and has published more than 100 articles in peer-reviewed journals. Prior to his academic career, Dr. Ovaska developed control systems for high-rise elevators; those contributions led to nine international patents.

GREAT TEXT... RARELY DO I RATE BOOKS SO HIGHLY... THIS IS A GOLD MINE FOR THOSE WHO WANT TO LEARN MORE.

This book presents the reader with a carefully selected set of topics regarding real time systems analysis, design and even implementation. A very resourceful overview on several semi-independent subjects, it may not be appropriate if you are looking for in-depth treating of any specific aspect.

Real-time systems are the unsung, unheralded absolute workhorses of the technological age. The airliner you fly wouldn't exist or be flyable without real-time systems. The American B-2 bomber flies only because real-time systems are constantly altering the shape of its wing surfaces. Medical suites ranging from your dentists' and doctor's offices to surgical suites are loaded - and dependent - on real-time systems. Ever your recent year car is dependent on real-time systems. In fact, the majority of microprocessors ever made don't reside in the computer systems you have in your home or workplace, but in devices utilizing real-time systems. How do they work? This excellent introductory text explains how - and, if you want to, teaches you how to design, program and implement such systems. Now in its fourth edition, this text is essentially the "bible" as the introduction to the subject. You don't need to know much about programming when using this book, though some knowledge of a high-level language like C or Java would be very helpful. The authors,

whose words flow smoothly, understand their audience is unsophisticated in the areas they address and introduce and define technical terms as they are required. They start at the very beginning by defining what real-time systems are. Next comes An exposition of the hardware required and the necessary architecture. Things start getting tougher with the introduction of real-time operating systems and programming languages used in such applications. (These chapters are, by the way, absolute gems of technical clarity.) The overview of programming languages is particularly illuminating. From this point on, it is a pretty steep climb, though the authors make it as scalable as they can. You're getting into actual application development territory here and a lot of folks may get lost. Dedicated students will find the instruction comprehensible - if they already have or are willing to developed a sophisticated understanding of real-time systems which is, after all, what the text is intended to achieve. There is no question of why this book is considered the classic introductory text in the field and why it is popular enough to warrant frequent updates: it is a great way to learn real-time systems.Jerry

This book has come a long way since I used the first two editions as class texts in the 90s. The new edition is considerably more comprehensive and also covers such latter-day phenomena as multi-core processors and real-time Java.I was pleased to find a practical, realistic example of a state machine. I would have liked to see more: In my opinion, state diagramming is not appreciated enough as a practical problem analysis tool, which also lends itself beautifully to collaborative efforts. A couple of Statecharts examples show up in case studies later on. More would have been welcome but may be too much to ask for given the breadth of the book's coverage.Tasks/threads are discussed throughout but little is said about the design of the thread architecture of a real-time system. In my view it is a self-contained artifact that can often be modeled directly on concurrency found in the problem domain - in simple cases a state diagram - quite independently of the rest of the design.Â Design of Multithreaded Software: The Entity-Life Modeling ApproachI am pleased that the book sides with the practitioner. Refreshingly, it points out that what a theoretician may call "provable correctness" is only one aspect of the practical correctness of an implemented system.

Good books on real time embedded systems are rare and I think that if you were looking for a book that explains the why behind things, then you may stop here.This book targets beginners and advanced professionals of embedded systems.If you are a beginner, this book is the right one for you. The book starts from scratch and explains multiple sides of embedded systems like hardware and the different software aspects.If you're an experienced embedded systems engineer, but you

just know some of the aspects of real time systems you've always worked with, because you have to follow a certain development process, and you would like to learn about other aspects, programming languages or processes, than you can find a lot of new things to learn in this book. As the author mentioned in the introduction "depth is occasionally sacrificed for breadth" in this book, which is a good decision made by the authors because it makes the reader understand the principles first but if the reader needs or would like to deepen his knowledge on a specific topic, every chapter mentions further literature at the end.I also liked the examples from real life applications and the interesting and funny anecdotes/ stories from what engineers experienced during their project work,Also challenging exercises and case studies help you test your knowledge and read again the paragraphs you did not pay enough attention to.As the book is meant to handle all embedded systems, it doesn't use a specific hardware board. Those who would like to learn embedded systems on hardware should buy one of the boards of known companies like Arduino, microchip, Franzis (for users in Germany) etc and either use internet tutorials or buy a book which is specific to that board. The knowledge you have from "Real Time Systems Design and analysis" would anyway be a benefit and would go beyond the knowledge you need to make those boards work.Here are some of the questions that you may ask yourself and that this book answers:What is the difference between a microcontroller and a microprocessor?What's the difference between Interrupt driven I/O and DMA?What are the different scheduling techniques?How can I avoid dead-locks and starvation?What are swapping, overlying and paging? How can I achieve an efficient memory management?Should I buy or build and embedded systems, and if I decide to buy, what are the criteria I should use to evaluate the different products?Which programming language should I use for my real time application? What are the cardelli's metrics?What is the difference between C, C++ and Ada?Is assembler still used in embedded applications?Which coding standards should I follow?What are the different compiler optimizations?What is code generation, is it suitable for my application?How do I write the requirements of my system? How can I validate them?What are formal methods, and where can they be used in the development cycle?Does software engineering have any principles like electrical or mechanical engineering?How should I design my software if I use a procedural or an object-oriented approach?Have you ever heard of the waterfall model? Or of the V model? Are there any other software development models?And many other questions...

Real-Time Systems Design and Analysis: Tools for the Practitioner Real-Time Systems: Design Principles for Distributed Embedded Applications (Real-Time Systems Series) The Real Book of

Real Estate: Real Experts. Real Stories. Real Life. Hawaii Real Estate Wholesaling Residential Real Estate Investor & Commercial Real Estate Investing: Learn to Buy Real Estate Finance Hawaii Homes & Find Wholesale Real Estate Houses in Hawaii Graphic Design Success: Over 100 Tips for Beginners in Graphic Design: Graphic Design Basics for Beginners, Save Time and Jump Start Your Success (graphic ... graphic design beginner, design skills) A PRACTITIONER'S GUIDE TO BUSINESS ANALYTICS: Using Data Analysis Tools to Improve Your Organizationâ ™s Decision Making and Strategy Nurse Practitioner's Business Practice and Legal Guide, Second Edition (Buppert, Nurse Practitioner's Business Practice and Legal Guide) Nurse Practitioner's Business Practice And Legal Guide (Buppert, Nurse Practitioner's Business Practice and Legal Gu) Nurse Practitioner's Business Practice And Legal Guide (Buppert, Nurse Practitioner's Business Practice and Legal Guide) Immigrant Students and Literacy: Reading, Writing, and Remembering (Practitioner Inquiry Series) (Practitioner Inquiry (Paperback)) Sound Systems: Design and Optimization: Modern Techniques and Tools for Sound System Design and Alignment The Smart Real Estate Investor: Real Estate Book Bundle 3 Manuscripts Expert Strategies on Real Estate Investing, Finding and Generating Leads, Funding, Proven Methods for Investing in Real Estate Real Estate: 25 Best Strategies for Real Estate Investing, Home Buying and Flipping Houses (Real Estate, Real Estate Investing, home buying, flipping houses, ... income, investing, entrepreneurship) The Smart Real Estate Investor: Real Estate Book Bundle 2 Manuscripts Expert Strategies on Real Estate Investing, Starting with Little or No Money, Proven Methods for Investing in Real Estate Real Estate: 30 Best Strategies to Prosper in Real Estate - Real Estate Investing, Financing & Cash Flow (Real Estate Investing, Flipping Houses, Brokers, Foreclosure) Humanitarian Intelligence: A Practitioner's Guide to Crisis Analysis and Project Design (Security and Professional Intelligence Education Series) Tactics Time 2: 1001 Real Chess Tactics From Real Chess Games (Tactics Time Chess Tactics Books) Power Tools for Synthesizer Programming: The Ultimate Reference for Sound Design: Second Edition (Power Tools Series) The Real Analysis Lifesaver: All the Tools You Need to Understand Proofs (Princeton Lifesaver Study Guides) Fundamentals Of Information Systems Security (Information Systems Security & Assurance) - Standalone book (Jones & Bartlett Learning Information Systems Security & Assurance)

FAQ & Help