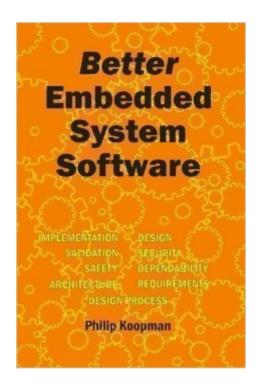# Better Embedded System Software

Synopsis

This book distills the experience of more than 90 design reviews on real embedded systems into a set of bite-size lessons learned in the areas of software development process, requirements, architecture, design, implementation, verification & validation, and critical system properties.Each chapter describes an area that tends to be a problem in embedded system design, symptoms that tend to indicate you need to make changes, the risks of not fixing problems in this area, and concrete ways to make your embedded system software better. Each of the 29 chapters is self-sufficient, permitting developers with a busy schedule to cherry-pick the best ideas to make their systems better right away.If you are relatively new to the area but have already learned the basics, this book will be an invaluable asset for taking your game to the next level. If you are experienced, this book provides a way to fill in any gaps. Once you have mastered this material, the book will serve as a source of reminders to make sure you haven't forgotten anything as you plan your next project. The product image of the back of the book lists the chapters.

Book Information

Hardcover: 397 pages

Publisher: Drumnadrochit Education (2010)

Language: English

ISBN-10: 0984449000

ISBN-13: 978-0984449002

Product Dimensions: 9.1 x 6.1 x 1.2 inches

Shipping Weight: 1.6 pounds

Average Customer Review: 4.9 out of 5 stars See all reviews (21 customer reviews)

Best Sellers Rank: #553,547 in Books (See Top 100 in Books)   #60 in Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Embedded Systems

Customer Reviews

Dr. Koopmann possesses both academic knowledge and real world experience in developing well-structured safety-assured systems containing embedded software and hardware. His experience earned in the trenches of over 90 design reviews of embedded computer-based projects is clearly described and exquisitely organized to guide the novice systems designer or experienced professional in correct, safe and efficiently designed software for embedded systems. He employs formal tools such as state charts and real-time scheduling calculations in just the situations and areas where they are most fruitful in successful systems development. With the practiced eye of

both an academic and profession embedded systems team leader he advocates just the most useful pieces of documentation and formal technique recognizing that just the right and productive system specification, documentation and testing is necessary to ensure the correct and efficient operation of an embedded systems project; that the right documentation (that he clearly describes) is necessary to ensure a safe reliable product, but that unnecessary is expensive to create and will not be consulted. The top-down and carefully structured organization of this handbook allows gleaning just the heuristics that you as a developer need at a given stage of systems development and to extract valuable hints relevant to the current stage of the project.

This book is an excellent reference for embedded product development on both large and small scale projects. I've worked for various tech companies from small to huge for 20 years before forming my own embedded consulting business 5+ years ago, and this book contains many great tips and best practices that I've experienced or advocated in all these work situations. Any development team could truly benefit from reading this book and incorporating some or all of its tenets into their own development process.The first 9 chapters focus on process and product requirements - although these topics aren't generally regarded as "sexy", these are often the areas where the most teams seem to need alot of help. The chapters on software architecture, implementation and critical areas such as task scheduling, data integrity, ISRs, and watchdog/reset implementation are "spot on" and serve as great teaching points to novices as well as useful reminders to the gray-haired. The chapters on validation, safety, security, and reliability are also quite good and touch on many important aspects, even though these topics are themselves often complex enough to warrant their own scope-specific books. Mr Koopman's book is very well organized, clearly and concisely written, and the chapters stand well on their own for quick referencing. It is very obvious that the author is highly experienced in these topics and wastes no time in hitting all the key points. There are more specifically-scoped books out which go into greater detail within a narrower band of interest, but I have yet to find another book which provides the quantity of useful/practical points and principles across all aspects of embedded software development as this one.

Better Embedded System Software DSP Software Development Techniques for Embedded and Real-Time Systems (Embedded Technology) Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit Sleep Smarter: The Ultimate Guide To Sleep Better, Feel Better By Having Healthy Sleeping Habits (sleep smarter, sleep better, healthy sleep habits, sleep

... healthy sleep, sleep apnea, feel better) Embedded System Design: A Unified Hardware/Software Introduction Applied Control Theory for Embedded Systems (Embedded Technology) Analog Interfacing to Embedded Microprocessor Systems, Second Edition (Embedded Technology Series) Real-Time UML Workshop for Embedded Systems, Second Edition (Embedded Technology) Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers (Embedded Technology) TCP/IP Embedded Internet Applications (Embedded Technology) Linux for Embedded and Real-time Applications, Third Edition (Embedded Technology) Linux for Embedded and Real-time Applications (Embedded Technology) Linux for Embedded and Real-time Applications, Second Edition (Embedded Technology) ARM System Developer's Guide: Designing and Optimizing System Software (The Morgan Kaufmann Series in Computer Architecture and Design) Making Embedded Systems: Design Patterns for Great Software An Embedded Software Primer Embedded Linux Primer: A Practical Real-World Approach (Prentice Hall Open Source Software Development Series) Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development Embedded Linux Systems with the Yocto Project (Prentice Hall Open Source Software Development) Make: Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software (Learning by Discovery)