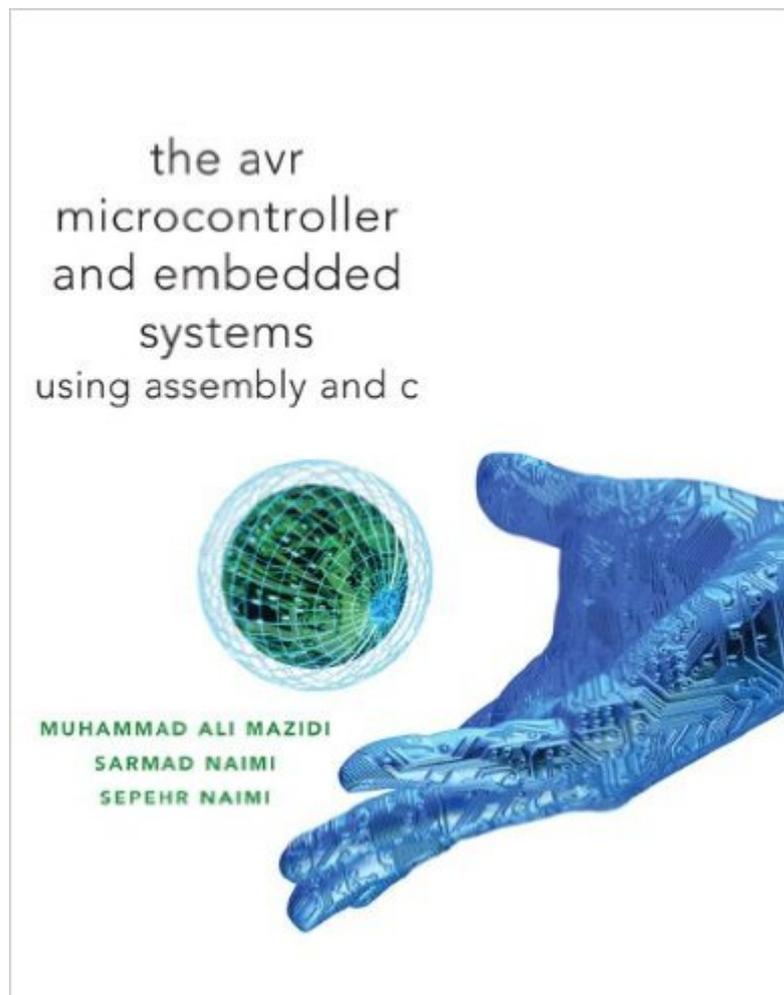


The book was found

AVR Microcontroller And Embedded Systems: Using Assembly And C (Pearson Custom Electronics Technology)



Synopsis

The AVR Microcontroller and Embedded Systems: Using Assembly and C features a step-by-step approach in covering both Assembly and C language programming of the AVR family of Microcontrollers. It offers a systematic approach in programming and interfacing of the AVR with LCD, keyboard, ADC, DAC, Sensors, Serial Ports, Timers, DC and Stepper Motors, Opto-isolators, and RTC. Both Assembly and C languages are used in all the peripherals programming. In the first 6 chapters, Assembly language is used to cover the AVR architecture and starting with chapter 7, both Assembly and C languages are used to show the peripherals programming and interfacing.Â Â

Book Information

Series: Pearson Custom Electronics Technology

Paperback: 792 pages

Publisher: Pearson; 1 edition (January 31, 2010)

Language: English

ISBN-10: 0138003319

ISBN-13: 978-0138003319

Product Dimensions: 8.2 x 1.7 x 10.9 inches

Shipping Weight: 3.9 pounds (View shipping rates and policies)

Average Customer Review: 4.3 out of 5 starsÂ Â See all reviewsÂ (15 customer reviews)

Best Sellers Rank: #170,656 in Books (See Top 100 in Books) #7 inÂ Books > Engineering & Transportation > Engineering > Energy Production & Extraction > Power Systems #14 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Embedded Systems #110 inÂ Books > Computers & Technology > Hardware & DIY > Personal Computers

Customer Reviews

The textbook only covers basics. It is a book for beginners and for low level college introductory microcontroller courses. The book not very useful for real projects. The author give me an impression that he just started to learn about AVR right before starting to write this book. In this book, it is not much explanation on the hardware features; being able to connect to external memory for ATmega128. Author use for examples Atmega32. Interfaces examples no cover CAN and USB connections to AVR. No cover connection to graphics matrix displays.

I will start of by saying that out of all the text books I have come across this is by far one of the best.I

need this book for part of an engineering degree, focused on the AVR micro-controller. Coming from a programming degree I found this book very good, clear and it is filled with useful examples. One thing I loved about the book is that it was written in a way that it can be used by people with little experience with programming, or anyone who is new to Assembly and embedded systems programming. I loved the introductory chapters as they gave you a feel and solid background in AVR and computing. One more note that I found very important was the fact that as the examples got more complicated they often had both C and assembly side by side which helps give a better perspective of what you are doing. 5 stars. :-PAVR Microcontroller and Embedded Systems: Using Assembly and C (Pearson Custom Electronics Technology)

OK let me start by saying--there are errors in the book. And the author won't build a smartphone with an AVR for you. But if you are somewhat of a beginner with firmware programming of AVR's, this is the book for you! The first 6 chapters deal with coding using Assembler. Now, offering advice on coding with C vs. assembler is like discussing religion--everyone has an opinion. Here's mine: you should always have a basic knowledge of how to code in assembler for a device--it gives you a broad understanding of what is happening when the electrons start flying, and allows you to use assembler blocks later in your C files. Of course C is almost always better once you start a real involved project, which is why most of the book mainly involves coding in C. In my case, my device is the ATXMEGA128B1. There are large differences in my include file from the one the author uses (m32def.inc) but I'm able to note the changes by referencing the Atmel libraries of app notes online. You should have a basic knowledge of C prior to purchasing the book. The author starts from the beginning with assembly language but assumes you have a basic knowledge of C. What you are really paying for is the EXCELLENT explanation and uses of the various commands and syntax each language offers, with LOTS of examples (which is why the book is 700+ pages). And you will spot the errors very quickly. I haven't come across an error yet that caused me to be confused--they are mostly just typos. If you are looking for a book to do your project for you, look elsewhere. But if you want to gain a broad knowledge of firmware coding of AVR's, THIS BOOK IS IT!

This is the book assigned for an Assembly class I am taking. The scope of this book is not stated on , but this book is intended for college students learning Assembly with some C knowledge, (obviously using Atmel AVR's to learn with.) This book was clearly written and published very quickly, without a lot of effort and without much proofing. Many sections in this book are unclear and awkwardly worded. There are several blatant errors that are all results of the book not being proofed

before publishing. The authors/publishers also offer classroom PowerPoint slides to colleges who use this book. These slides are riddled with errors, typos, and mistakes. This could have been a very good book, even a five star book. All of the information that you need to know/learn is there. It is unfortunate that the authors didn't spend a little more time in writing and proofing the book before publishing it.

very good book, highly recommended to anyone looking to get into mcu programming. lots of examples, very easy to read. if you have no experience, then i will recommend this book. if you have some experience, then i would look elsewhere.

When I checked this book out from the library for the first time, the only thing I knew about microcontrollers was that I wanted to learn how to program and use them in projects. I knew I could get an Arduino and get up and running in a hurry, but I really wanted to understand what was going on from a more fundamental level. This book was just the ticket. It starts out by explaining (very concisely) exactly what a computer is and how it actually works. Then you learn the details of the AVR architecture, and start writing programs using the AVR assembly instruction set. I can't express in words how cool it was to write my first computer programs in assembly, even though they just flashed LED's and the like. Assembly code describes EXACTLY what is happening inside a computer. I used to think only geniuses could understand the lowest workings of a computer, but I'm not a smart man and I figured it out. So if you want to understand to, buy this book (or find it in a library it is pricey!!). More details on how I got started from scratch on AVR:-Watch Patrick Hood Daniel's youtube series on AVR micros. Some of the hardest parts about getting started are installing atmel studio or WinAVR on your pc, and he walks you through much of the confusion.-The AVR micro course from Bruce Land (Cornell) via youtube & the course website is also a great resource for learning-I used the "tinyISP" from Adufruit as a programmer, and the first micro I worked with was the ATTiny85. I used an unregulated 4xAA battery pack as a power supply, a breadboard I bought on ebay, and various leds & wires I had from an old radio shack electronics kit from when I was a kid. That's really all you need to do some basic stuff.-The "AVR freaks" forum is your new best friend.-Don't expect to understand/ make everything work right away. This stuff takes patience, but is well worth it in the end.

[Download to continue reading...](#)

AVR Microcontroller and Embedded Systems: Using Assembly and C (Pearson Custom Electronics Technology) Some Assembly Required: Assembly Language Programming with the AVR

Microcontroller The Atmel AVR Microcontroller: MEGA and XMEGA in Assembly and C (Explore Our New Electronic Tech 1st Editions) The Atmel AVR Microcontroller: MEGA and XMEGA in Assembly and C (with Student CD-ROM) (Explore Our New Electronic Tech 1st Editions) PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18 How to Use Custom Dashboards in Google Analytics: Using Custom Dashboards to Grow Your Blog DSP Software Development Techniques for Embedded and Real-Time Systems (Embedded Technology) Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers (Embedded Technology) 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) Programming the PIC Microcontroller with MBASIC (Embedded Technology) Beginner's Guide To Embedded C Programming: Using The Pic Microcontroller And The Hitech Picc-Lite C Compiler The 8051 Microcontroller and Embedded Systems (2nd Edition) PIC Microcontroller And Embedded Systems Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C Mosfet Modeling for VLSI Simulation: Theory And Practice (International Series on Advances in Solid State Electronics) (International Series on Advances in Solid State Electronics and Technology) The Physics And Modeling of Mosfets (International Series on Advances in Solid State Electronics) (International Series on Advances in Solid State Electronics and Technology (Unnumbered)) Introduction to Electronic Commerce (3rd Edition) (Pearson Custom Business Resources)

[Dmca](#)