

1. Record Nr.	UNINA9910457346103321
Autore	Walls Colin
Titolo	Embedded software [[electronic resource]] : the works // Colin Walls
Pubbl/distr/stampa	Amsterdam ; ; Boston, : Elsevier/Newnes, c2006
ISBN	1-280-62952-5 9786610629527 0-08-046109-3
Descrizione fisica	1 online resource (417 p.)
Disciplina	005.1
Soggetti	Embedded computer systems - Programming Computer systems - Programming Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	front cover; copyright; Dedication; table of contents; front matter; Foreword: What Do You Expect - Perfection?; Preface; How This Book Came About; Where This Book Came From; What You Will Find Here; Who This Book Is For; How to Use This Book; Acknowledgments; Contributors; A Good Cause; Contact Me; What's on the CD-ROM?; Code Fragments; body; 1 Embedded Software; 1.1 What Makes an Embedded Application Tick?; 1.2 Memory in Embedded Systems; 1.3 Memory Architectures; 1.4 How Software Influences Hardware Design; 1.5 Migrating Your Software to a New Processor Architecture 1.6 Testing Computers on Wheels1.7 Embedded Software for Transportation Applications; 1.8 How to Choose a CPU for Your System on Chip Design; 1.9 An Introduction to USB Software; 1.10 USB On-the-Go; 2 Design and Development; 2.1 Emerging Technology for Embedded Systems Software Development; 2.2 Making Development Tool Choices; 2.3 Eclipse - Bringing Embedded Tools Together; 2.4 A Development System That Crosses RTOS Boundaries; 2.5 Embedded Software and UML; 2.6 Model-Based Systems Development with xtUML; 3 Programming; 3.1 Programming for Exotic Memories; 3.2 Self-Testing in Embedded Systems 3.3 A Command-Line Interpreter3.4 Traffic Lights: An Embedded

Software Application; 3.5 PowerPC Assembler; 4 C Language; 4.1 C Common; 4.2 Using C Function Prototypes; 4.3 Interrupt Functions and ANSI Keywords; 4.4 Optimization for RISC Architectures; 4.5 Bit by Bit; 4.6 Programming Floating-Point Applications; 4.7 Looking at C - A Different Perspective; 4.8 Reducing Function Call Overhead; 4.9 Structure Layout - Become an Expert; 4.10 Memory and Programming in C; 4.11 Pointers and Arrays in C and C++; 5 C++; 5.1 C++ in Embedded Systems - A Management Perspective
5.2 Why Convert from C to C++? 5.3 Clearing the Path to C++; 5.4 C++ Templates - Benefits and Pitfalls; 5.5 Exception Handling in C++; 5.6 Looking at Code Size and Performance with C++; 5.7 Write-Only Ports in C++; 5.8 Using Nonvolatile RAM with C++; 6 Real Time; 6.1 Real-Time Systems; 6.2 Visualizing Program Models of Embedded Systems; 6.3 Event Handling in Embedded Systems; 6.4 Programming for Interrupts; 7 Real-Time Operating Systems; 7.1 Debugging Techniques with an RTOS; 7.2 A Debugging Solution for a Custom Real-Time Operating System; 7.3 Debugging - Stack Overflows
7.4 Bring in the Pros - When to Consider a Commercial RTOS 7.5 On the Move; 7.6 Introduction to RTOS Driver Development; 7.7 Scheduling Algorithms and Priority Inversion; 7.8 Time versus Priority Scheduling; 7.9 An Embedded File System; 7.10 OSEK - An RTOS Standard; 8 Networking; 8.1 What's Wi-Fi?; 8.2 Who Needs a Web Server?; 8.3 Introduction to SNMP; 8.4 IPv6 - The Next Generation Internet Protocol; 8.5 The Basics of DHCP; 8.6 NAT Explained; 8.7 PPP - Point-to-Point Protocol; 8.8 Introduction to SSL; 8.9 DHCP Debugging Tips; 8.10 IP Multicasting; 9 Embedded Systems and Programmable Logic
9.1 FPGAs and Processor Cores: The Future of Embedded Systems?

Sommario/riassunto

Embedded systems are everywhere. In the average household, there are around 40 different microprocessors, not counting PCs (which contribute another 5-10 each), or cars, which typically contain a few dozen. And these numbers are set to rise by a couple of orders of magnitude over the next decade. This comprehensive new reference is a must-have for the tens of thousands of embedded designers working to make all those microprocessors faster, more efficient, and more powerful. A good embedded software developer must have a strong grasp of many complex topics, and this practical guide cover
