

## Freertos Documentation

Nowadays, embedded systems - the computer systems that are embedded in various kinds of devices and play an important role of specific control functions, have permitted various aspects of industry. Therefore, we can hardly discuss our life and society from now onwards without referring to embedded systems. For wide-ranging embedded systems to continue their growth, a number of high-quality fundamental and applied researches are indispensable. This book contains 19 excellent chapters and addresses a wide spectrum of research topics on embedded systems, including basic researches, theoretical studies, and practical work. Embedded systems can be made only after fusing miscellaneous technologies together. Various technologies condensed in this book will be helpful to researchers and engineers around the world.

This user's guide does far more than simply outline the ARM Cortex-M3 CPU features; it explains step-by-step how to program and implement the processor in real-world designs. It teaches readers how to utilize the complete and thumb instruction sets in order to obtain the best functionality, efficiency, and reuseability. The author, an ARM engineer who helped develop the core, provides many examples and diagrams that aid understanding. Quick reference appendices make locating specific details a snap! Whole chapters are dedicated to: Debugging using the new CoreSight technology Migrating effectively from the ARM7 The Memory Protection Unit Interfaces, Exceptions, Interrupts ...and much more! The only available guide to programming and using the groundbreaking ARM Cortex-M3 processor Easy-to-understand examples, diagrams, quick reference appendices, full instruction and Thumb-2 instruction sets are included T teaches end users how to start from the ground up with the M3, and how to migrate from the ARM7

Este livro presume que você tenha um conhecimento básico pelo menos na linguagem C e desejável, mas não indispensável, do Arduino e do microcontrolador STM32 da ST Microelectronics. O livro apresenta um estudo sobre sistemas operacionais de tempo real e e plataformas de desenvolvimento e programação multitarefa com RTOS, apresentando um enfoque da definição e implementação visando reaproveitamento de código e portabilidade entre plataformas de hardware. Atenção! Se estiver interessado em simplesmente aprender o FreeRTOS, recomendo o meu livro Programando Multitarefa na prática: Utilizando a linguagem C/C++, freeRTOS e Arduino (Segunda Edição) Agora se estiver interessado em como usar este sistema como base para um código reutilizável e portátil entre várias plataformas, a presente obra vai mostrar um exemplo prático, que irá ilustrar como separar o código dependente de plataforma do independente e como o RTOS (ou algum outro tipo de Sistema operacional) auxilia nesta tarefa ao disponibilizar funções multiplataforma para gerencia a execução das tarefas, temporizações, filas, etc. Ao final são anexados todos os códigos fontes nas duas plataformas, para permitir uma inspeção do resultado da estratégia proposta. A base teórica para sistemas embarcados e sistemas operacionais, assim como a escolha do freeRTOS é apresentada, contribuindo muito para aqueles que não tem familiaridade com estes conceitos. Explicações sobre as duas plataformas (Arduíno e STM32) introduzem a escolha do kit de desenvolvimento STM32F411E-Discovery da ST Microelectronics e a montagem de hardware baseado no Arduino Mega, que teria os mesmos recursos de hardware. Utiliza-se também, em ambas as plataformas, o módulo WiFi ESP8266 a Espressif, amplamente utilizado em projetos makers e profissionais. A integração com esta é feita através de comandos AT, implementado nas duas plataformas. Com este módulo, leituras de giroscópio lidos dos sensores presentes nos dois hardwares (chamados aqui de plataformas), são enviadas via comandos GET HTTP para um servidor com PHP, que salva os dados em arquivos em disco (mas poderia ser um banco de dados relacional, NOSQL, etc). Um script simples é adicionada neste livro, como exemplo de um cenário inicial de exemplo que pode ser trocado para outros protocolos para IoT, por exemplo. O valor deste livro é que ele mostra o uso de multitarefas e de APIS do RTOS como base para possibilitar a implementação de código mais genérico, separando a parte específica de plataforma daquela que independente do hardware, por definir as ações que são desejadas e não a forma como este hardware ou aquele irá implementá-lo. O código ilustrativo é apresentado, compondo aproximadamente 4/5 da quantidade de página e tem o valor de não deixar o leitor sem verificar como se poderia implementar estes sistemas multiplataforma em suas aplicações.

This book presents an overview of smart camera systems, considering practical applications but also reviewing fundamental aspects of the underlying technology. It introduces in a tutorial style the principles of sensing and signal processing, and also describes topics such as wireless connection to the Internet of Things (IoT) which is expected to be the biggest market for smart cameras. It is an excellent guide to the fundamental of smart camera technology, and the chapters complement each other well as the authors have worked as a team under the auspice of GFP (Global Frontier Project), the largest-scale funded research in Korea. This is the third of three books based on the Integrated Smart Sensors research project, which describe the development of innovative devices, circuits, and system-level enabling technologies. The aim of the project was to develop common platforms on which various devices and sensors can be loaded, and to create systems offering significant improvements in information processing speed, energy usage, and size. This book contains extensive reference lists, introduces the reader to the subject in a tutorial style and also reviews state-of-the-art results, which allows it to be used as a guide for starting researchers.

This volume contains the papers presented at SBMF 2009: the Brazilian Symposium on Formal Methods, held during August 19–21, 2009 in Gramado, Rio Grande do Sul, Brazil. The SBMF programme included three invited talks given by Leonardo de Moura (Microsoft Research), Sebastian Uchitel (University of Buenos Aires and Imperial College London), and Daniel Kroening (University of Oxford). The symposium was accompanied by two short courses: – Introduction to Software Testing, given by Marci ´o Eduardo Delamaro (University of Sao ´ Paulo) – Formal Models for Automatic Test Case Generation, given by Patr ´ıcia Machado and Wilkerson Andrade (Federal University of Campina Grande) This year, the SBMF symposium had a special section on the Grand Challenge in Verified Software, inspired by recent advances in theory and tool support. Work on the grand challenge started with the creation of a Verified Software Repository with two principal aims: – To collect a set of verified software components – To conduct a series of industrial-scale verification experiments with theoretical significance and impact on tool support This special session on the grand challenge was dedicated to two pilot projects currently underway: – The Flash File Store. The challenge is to verify the correctness of a fault-tolerant, POSIX-compliant file store implemented on flash memory. Verification issues include dependability guarantees as well as software correctness. Levels of abstraction include requirements specification, software design, executable code, device drivers, and flash translation layers. The challenge was inspired by the requirements for forthcoming NASA space missions. – FreeRTOS.

Embedded Software Development: The Open-Source Approach delivers a practical introduction to embedded software development, with a focus on open-source components. This programmer-centric book is written in a way that enables even novice practitioners to grasp the development process as a whole. Incorporating real code fragments and explicit, real-world open-source operating system references (in particular, FreeRTOS) throughout, the text: Defines the role and purpose of embedded systems, describing their internal structure and interfacing with software development tools Examines the inner workings of the GNU compiler collection (GCC)-based software development system or, in other words, toolchain Presents software execution models that can be adopted profitably to model and express concurrency Addresses the basic nomenclature, models, and concepts related to task-based scheduling algorithms Shows how an open-source protocol stack can be integrated in an embedded system and interfaced with other software components Analyzes the main components of the FreeRTOS Application Programming Interface (API), detailing the implementation of key operating system concepts Discusses advanced topics such as formal verification, model checking, runtime checks, memory corruption, security, and dependability Embedded Software Development: The Open-Source Approach capitalizes on the authors' extensive research on real-time operating systems and communications used in embedded applications, often carried out in strict cooperation with industry. Thus, the book serves as a springboard for further research.

This book describes the field of State-of-Charge (SoC) indication for rechargeable batteries. An overview of the state-of-the-art of SoC indication methods including available market solutions from leading semiconductor companies is provided. All disciplines are covered, from electrical, chemical, mathematical and measurement engineering to understanding battery behavior. This book will therefore is for persons in engineering and involved in battery management.

Product development is the magic that turns circuitry, software, and materials into a product, but moving efficiently from concept to manufactured product is a complex process with many potential pitfalls. This practical guide pulls back the curtain to reveal what happens—or should happen—when you take a product from prototype to production. For makers looking to go pro or product development team members keen to understand the process, author Alan Cohen tracks the development of an intelligent electronic device to explain the strategies and tactics necessary to transform an abstract idea into a successful product that people want to use. Learn 11 deadly sins that kill product development projects Get an overview of how electronic products are manufactured Determine whether your idea has a good chance of being profitable Narrow down the product's functionality and associated costs Generate requirements that describe the final product's details Select your processor, operating system, and power sources Learn how to comply with safety regulations and standards Dive into development—from rapid prototyping to manufacturing Alan Cohen, a veteran systems and software engineering manager and lifelong technophile, specializes in leading the development of medical devices and other high-reliability products. His passion is to work with engineers and other stakeholders to forge innovative technologies into successful products.

Build a strong foundation in designing and implementing real-time systems with the help of practical examples Key Features Get up and running with the fundamentals of RTOS and apply them on STM32 Enhance your programming skills to design and build real-world embedded systems Get to grips with advanced techniques for implementing embedded systems Book Description A real-time operating system (RTOS) is used to develop systems that respond to events within strict timelines. Real-time embedded systems have applications in various industries, from automotive and aerospace through to laboratory test equipment and consumer electronics. These systems provide consistent and reliable timing and are designed to run without intervention for years. This microcontrollers book starts by introducing you to the concept of RTOS and compares some other alternative methods for achieving real-time performance. Once you've understood the fundamentals, such as tasks, queues, mutexes, and semaphores, you'll learn what to look for when selecting a microcontroller and development environment. By working through examples that use an STM32F7 Nucleo board, the STM32CubeIDE, and SEGGER debug tools, including SEGGER J-Link, Ozone, and SystemView, you'll gain an understanding of preemptive scheduling policies and task communication. The book will then help you develop highly efficient low-level drivers and analyze their real-time performance and CPU utilization. Finally, you'll cover tips for troubleshooting and be able to take your new-found skills to the next level. By the end of this book, you'll have built on your embedded system skills and will be able to create real-time systems using microcontrollers and FreeRTOS. What you will learn Understand when to use an RTOS for a project Explore RTOS concepts such as tasks, mutexes, semaphores, and queues Discover different microcontroller units (MCUs) and choose the best one for your project Evaluate and select the best IDE and middleware stack for your project Use professional-grade tools for analyzing and debugging your application Get FreeRTOS-based applications up and running on an STM32 board Who this book is for This book is for embedded engineers, students, or anyone interested in learning the complete RTOS feature set with embedded devices. A basic understanding of the C programming language and embedded systems or microcontrollers will be helpful.

Communication and Power Engineering are the proceedings of the joint International conferences organized by IDES in the year 2016. The aim of these conference proceedings is to bringing together the researchers, scientists, engineers, and scholar students in all areas of Computer Science, Power Engineering, Electrical & Electronics and provides an international forum for the dissemination of original research results, new ideas and practical development experiences, focused on both theory and practices. The conference deals with the frontier topics in the Computer Science, Electrical and Electronics Engineering subjects. The Institute of Doctors Engineers and Scientists - IDES is formed to promote, and organize technical research Meetings, Conference, Discussions, Seminars, Workshops, Study tours, Industry visits; and to publish professional Journals, Magazines and Newsletters; and to carry on research and development on the above fields; and to research, design, and develop products or materials and projects. There are total 35 research papers included in this book covering all the frontier topics in Computer Science, Electrical and Electronics Engineering subjects. The authors of each chapter are researchers from various universities. Contents: Foreword Handwritten Script Identification from Text Lines A Rule based Approach for Noun Phrase Extraction from English Text Document Recommending Investors using Association Rule Mining for Crowd Funding Projects Colour Texture Classification Using Anisotropic Diffusion and Wavelet

Transform Competitive Advantage of using Differential Evolution Algorithm for Software Effort Estimation Comparative Analysis of Cepstral analysis and Autocorrelation Method for Gender Classification A Simulative Study on Effects of Sensing Parameters on Cognitive Radio's Performance Analysis of Cyclotomic Fast Fourier Transform by Gate level Delay Method Dynamic Resource Allocation in Next Generation Networks using FARIMA Time Series Model Classification of Mimetic Spectral Signatures using Orthogonal Subspace Projection with Complex Wavelet Filter Bank based Dimensionality Reduction An Illumination Invariant Face Recognition Approach based on Fourier Spectrum Optimal Load Frequency Controller for a Deregulated Reheat Thermal Power System Design and Implementation of a Heuristic Approximation Algorithm for Multicast Routing in Optical Networks Infrastructure Management Services Toolkit A Novel Approach for Residential Society Maintenance Problem for Better Human Life Smart Suspect Vehicle Surveillance System Formal Performance Analysis of Web Servers using an SMT Solver and a Web Framework Modified GCC Compiler Pass for Thread-Level Speculation by Modifying the Window Size using Openmp Overview and Evaluation of an IoT Product for Application Development A TCP in CR-MANET with Unstable Bandwidth Impact of Digital Ecosystem on Business Environment A Two-Factor Single Use Password Scheme Design & Implementation of Wireless System for Cochlear Devices Software Code Clone Detection and Removal using Program Dependence Graphs Social Sentimental Analytics using Big Data Tools Predicting Flight Delay using ANN with Multi-core Map Reduce Framework New Network Overlay Solution for Complete Networking Virtualization Review upon Distributed Facts Hard Drive Schemes throughout Wireless Sensor Communities Detection of Rapid Eye Movement Behaviour Sleep Disorder using Time and Frequency Analysis of EEG Signal Applied on C4-A1 Channel Analysis of PV/ WIND/ FUEL CELL Hybrid System Interconnected With Electrical Utility Grid Analysis of Wind Speed Prediction Technique by hybrid Weibull-ANN Model An efficient FPGA Implementation of DES and Triple-DES Encryption Systems A Novelty Comparison of Power with Assorted Parameters of a Horizontal Wind Axis Turbine for NACA 5512 Retaliation based Enhanced Weighted Clustering Algorithm for Mobile Ad-hoc Network (R-EWCA) Chest CT Scans Screening of COPD based Fuzzy Rule Classifier Approach Author Index

This volume contains the proceedings of TESTCOM/FATES 2008, a joint conference of two communities: TESTCOM was the 20th edition of the IFIP TC6/ WG6.1 International Conference on Testing of Communicating Systems and FATES was the 8th edition of the International Workshop on Formal Approaches to Testing of Software. TESTCOM/FATES 2008 was held at the Campus Innovation Center in Tokyo, Japan during June 10-13, 2008. Testing is one of the most important techniques for validating and checking the correctness of communication and software systems. Testing, however, is also a laborious and very cost-intensive task during the development process of such systems. TESTCOM is a series of international conferences addressing the problems of testing communicating systems, including communication protocols, services, distributed platforms, and middleware. FATES is a series of international workshops discussing the challenges of using rigorous and formal methods for testing software systems in general. TESTCOM/FATES aims at being a forum for researchers, developers, and testers to review, discuss, and learn about new approaches, concepts, theories, methodologies, tools, and experiences in the field of testing of communicating systems and software. TESTCOM has a long history. Previously it was called the International Workshop on Protocol Test Systems (IWPTS) and changed its name to the International Workshop on Testing of Communicating System (IWTC) later. The previous conferences were held in Vancouver, Canada (1988); Berlin, Germany (1989); McLean, USA (1990); Leidschendam, The Netherlands (1991); Montr?eal, Canada (1992); Pau, France (1993); Tokyo, Japan (1994); Evry, France (1995); Darmstadt, Germany (1996); Cheju Island, Korea (1997); Tomsk, Russia (1998); Budapest, Hungary (1999); Ottawa, Canada (2000); Berlin, Germany (2001); Sophia Antipolis, France (2002); Oxford, UK (2004); Montr?eal, Canada (2005); New York, USA (2006) and Tallinn, Estonia (2007).

Deep learning networks are getting smaller. Much smaller. The Google Assistant team can detect words with a model just 14 kilobytes in size—small enough to run on a microcontroller. With this practical book you'll enter the field of TinyML, where deep learning and embedded systems combine to make astounding things possible with tiny devices. Pete Warden and Daniel Situnayake explain how you can train models small enough to fit into any environment. Ideal for software and hardware developers who want to build embedded systems using machine learning, this guide walks you through creating a series of TinyML projects, step-by-step. No machine learning or microcontroller experience is necessary. Build a speech recognizer, a camera that detects people, and a magic wand that responds to gestures Work with Arduino and ultra-low-power microcontrollers Learn the essentials of ML and how to train your own models Train models to understand audio, image, and accelerometer data Explore TensorFlow Lite for Microcontrollers, Google's toolkit for TinyML Debug applications and provide safeguards for privacy and security Optimize latency, energy usage, and model and binary size The Second International Conference on Networks and Communications (NeCoM 2010), the Second International Conference on Wireless and Mobile Networks (WiMoN 2010), and the Second International Conference on Web and Semantic Technology (WeST 2010) were held in Chennai, India, during July 23–25, 2010. They attracted many local and international delegates, presenting a balanced mixture of intellects from the East and from the West. The goal of these conferences is to bring together researchers and practitioners from academia and industry to focus on understanding computer networks, wireless networks, mobile networks and the Web, semantic technologies and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in the areas of all computer networks and Semantic Web technologies. The NeCoM 2010, WiMoN 2010 and WeST 2010 committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer-review

process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity.

This book is about the Zynq-7000 All Programmable System on Chip, the family of devices from Xilinx that combines an application-grade ARM Cortex-A9 processor with traditional FPGA logic fabric. Catering for both new and experienced readers, it covers fundamental issues in an accessible way, starting with a clear overview of the device architecture, and an introduction to the design tools and processes for developing a Zynq SoC. Later chapters progress to more advanced topics such as embedded systems development, IP block design and operating systems. Maintaining a 'real-world' perspective, the book also compares Zynq with other device alternatives, and considers end-user applications. The Zynq Book is accompanied by a set of practical tutorials hosted on a companion website. These tutorials will guide the reader through first steps with Zynq, following on to a complete, audio-based embedded systems design.

Using FreeRTOS and libopencm3 instead of the Arduino software environment, this book will help you develop multi-tasking applications that go beyond Arduino norms. In addition to the usual peripherals found in the typical Arduino device, the STM32 device includes a USB controller, RTC (Real Time Clock), DMA (Direct Memory Access controller), CAN bus and more. Each chapter contains clear explanations of the STM32 hardware capabilities to help get you started with the device, including GPIO and several other ST Microelectronics peripherals like USB and CAN bus controller. You'll learn how to download and set up the libopencm3 + FreeRTOS development environment, using GCC. With everything set up, you'll leverage FreeRTOS to create tasks, queues, and mutexes. You'll also learn to work with the I2C bus to add GPIO using the PCF8574 chip. And how to create PWM output for RC control using hardware timers. You'll be introduced to new concepts that are necessary to master the STM32, such as how to extend code with GCC overlays using an external Winbond W25Q32 flash chip. Your knowledge is tested at the end of each chapter with exercises. Upon completing this book, you'll be ready to work with any of the devices in the STM32 family. Beginning STM32 provides the professional, student, or hobbyist a way to learn about ARM without costing an arm! What You'll Learn Initialize and use the libopencm3 drivers and handle interrupts Use DMA to drive a SPI based OLED displaying an analog meter Read PWM from an RC control using hardware timers Who This Book Is For Experienced embedded engineers, students, hobbyists and makers wishing to explore the ARM architecture, going beyond Arduino limits.

Most microcontroller-based applications nowadays are large, complex, and may require several tasks to share the MCU in multitasking applications. Most modern high-speed microcontrollers support multitasking kernels with sophisticated scheduling algorithms so that many complex tasks can be executed on a priority basis. ARM-based Microcontroller Multitasking Projects: Using the FreeRTOS Multitasking Kernel explains how to multitask ARM Cortex microcontrollers using the FreeRTOS multitasking kernel. The book describes in detail the features of multitasking operating systems such as scheduling, priorities, mailboxes, event flags, semaphores etc. before going onto present the highly popular FreeRTOS multitasking kernel. Practical working real-time projects using the highly popular Clicker 2 for STM32 development board (which can easily be transferred to other boards) together with FreeRTOS are an essential feature of this book. Projects include: LEDs flashing at different rates; Refreshing of 7-segment LEDs; Mobile robot where different sensors are controlled by different tasks; Multiple servo motors being controlled independently; Multitasking IoT project; Temperature controller with independent keyboard entry; Random number generator with 3 tasks: live, generator, display; home alarm system; car park management system, and many more. Explains the basic concepts of multitasking Demonstrates how to create small multitasking programs Explains how to install and use the FreeRTOS on an ARM Cortex processor Presents structured real-world projects that enables the reader to create their own

This book constitutes the refereed proceedings of the 17th International Conference on Formal Engineering Methods, ICFEM 2015, held in Paris, France, in November 2015. The 27 revised full papers presented were carefully reviewed and selected from 82 submissions. The papers cover a wide range of topics in the area of formal methods and software engineering and are devoted to advancing the state of the art of applying formal methods in practice. They focus in particular on combinations of conceptual and methodological aspects with their formal foundation and tool support.

Since the release of V0.01 in 2006, to the present V4.0 version, RT-Thread has developed a reputation among developers for its open source strategy. RT-Thread has gained a large following among members of the embedded open source community in China with hundreds of thousands of enthusiasts. RT-Thread is widely used in energy, automotive, medical, consumer electronics, among other applications, making it a mature and stable open source embedded operating system. The purpose of RT-Thread RTOS Design and Implementation is to create an easy learning curve for mastering RT-Thread, so that more developers can participate in the development of RT-Thread and work together to create an open source, tiny, and beautiful Internet of Things operating system. The book's first part introduces the RT-Thread kernel and starts with an overview of RT-Thread before covering thread management, clock management, inter-thread synchronization, inter-thread communication, memory management, and interrupt management. The second part begins with RT-Thread kernel porting and explains how to port RT-Thread to a hardware board to run it. The second part also introduces RT-Thread components and discusses the Env development environment, FinSH console, device management, and network framework. Additional topics covered include: The I/O device framework Virtual file systems Peripheral interfaces Devices including the PIN device, UART device, and ADC device, among others. Each chapter features code samples, as well as helpful tables and graphs, so you can practice as you learn as well as perform your own experiments.

This book constitutes the refereed proceedings of the 32nd IFIP TC 11 International Conference on ICT Systems Security and Privacy Protection, SEC 2017, held in Rome, Italy, in May 2017. The 38 revised full papers presented were carefully reviewed and selected from 199 submissions. The papers are organized in the following topical sections: network security and cyber

attacks; security and privacy in social applications and cyber attacks defense; private queries and aggregations; operating systems and firmware security; user authentication and policies; applied cryptography and voting schemes; software security and privacy; privacy; and digital signature, risk management, and code reuse attacks.

This book describes state-of-the-art approaches to Fog Computing, including the background of innovations achieved in recent years. Coverage includes various aspects of fog computing architectures for Internet of Things, driving reasons, variations and case studies. The authors discuss in detail key topics, such as meeting low latency and real-time requirements of applications, interoperability, federation and heterogeneous computing, energy efficiency and mobility, fog and cloud interplay, geo-distribution and location awareness, and case studies in healthcare and smart space applications.

This book is intended to provide a senior undergraduate or graduate student in electrical engineering or computer science with a balance of fundamental theory, review of industry practice, and hands-on experience to prepare for a career in the real-time embedded system industries. It is also intended to provide the practicing engineer with the necessary background to apply real-time theory to the design of embedded components and systems. Typical industries include aerospace, medical diagnostic and therapeutic systems, telecommunications, automotive, robotics, industrial process control, media systems, computer gaming, and electronic entertainment, as well as multimedia applications for general-purpose computing. This updated edition adds three new chapters focused on key technology advancements in embedded systems and with wider coverage of real-time architectures. The overall focus remains the RTOS (Real-Time Operating System), but use of Linux for soft real-time, hybrid FPGA (Field Programmable Gate Array) architectures and advancements in multi-core system-on-chip (SoC), as well as software strategies for asymmetric and symmetric multiprocessing (AMP and SMP) relevant to real-time embedded systems, have been added. Companion files are provided with numerous project videos, resources, applications, and figures from the book. Instructors' resources are available upon adoption. FEATURES: • Provides a comprehensive, up to date, and accessible presentation of embedded systems without sacrificing theoretical foundations • Features the RTOS (Real-Time Operating System), but use of Linux for soft real-time, hybrid FPGA architectures and advancements in multi-core system-on-chip is included • Discusses an overview of RTOS advancements, including AMP and SMP configurations, with a discussion of future directions for RTOS use in multi-core architectures, such as SoC • Detailed applications coverage including robotics, computer vision, and continuous media • Includes a companion disc (4GB) with numerous videos, resources, projects, examples, and figures from the book • Provides several instructors' resources, including lecture notes, Microsoft PP slides, etc.

This second edition of Real-Time Embedded Multithreading contains the fundamentals of developing real-time operating systems and multithreading with all the new functionality of ThreadX Version 5. ThreadX has been deployed in approximately 500 million devices worldwide. General concepts and terminology are detailed along with problem solving of com  
Extend the capabilities and power of your applications using Real-Time Operating System features. This book combines two powerful tools: Arduino and freeRTOS. Resources addressed:  
Interrupts: Addresses communication between hardware interrupts and tasks. Tasks: Allow parallel programming to better organize execution and code. Semaphores: Allows you to control concurrent access to resources and communication between tasks. Queues: It allows to communicate multiple items between tasks and is explored through several examples, in association with interruptions and tasks. Task notification: Sending values to task directly through task notification, without using queues or semaphores. Software Timer: Without having to control for interruptions, call a function of your own in time or after a timeout only once. We will approach the concepts, through brief explanations and listings of sample source codes, which will often be expanded in stages. In this way we will present and explain the mechanisms of programming in multiple tasks and their mechanisms of support, control of access to resources, communication between tasks. Understanding concepts will be given by their incremental introduction, tracking changes and improvements in the code, which you can go testing on your Arduino (if you prefer), or just go through the accompanying explanation. Some companion or book listings are posted on the internet as a supplement. The Arduino platform, which further popularized digital electronics (even for those with no specific training) and at the same time facilitated the creation of product prototypes, for startups, makers, and even for engineers and programmers of experienced embedded systems. freeRTOS, the Real-Time Operating System, which supports a large amount of microcontrollers and development environment, and has become a de facto standard. The union of these two platforms, facilitated by the development of a freeRTOS package that can be easily added to the Arduino IDE (and in this book you'll see how to do this), will allow you to learn how to develop powerful and easy-to-maintain applications. Each has its own style of studying programming. I prefer to read over, examining areas of greater interest, and then "lay hands on the mass." You may prefer to follow step by step what is presented and then venture into making your modifications and creating your solutions. Think of this book as a complement to your Arduino programming knowledge or programming for embedded systems in general. The focus is to get you started (or increase your knowledge) in multitasking for MCUs, using freeRTOS in your projects, whatever platform you prefer among the many supported platforms.

The Designer's Guide to the Cortex-M Family is a tutorial-based book giving the key concepts required to develop programs in C with a Cortex M- based processor. The book begins with an overview of the Cortex- M family, giving architectural descriptions supported with practical examples, enabling the engineer to easily develop basic C programs to run on the Cortex-M0/M0+/M3 and M4. It then examines the more advanced features of the Cortex architecture such as memory protection, operating modes and dual stack operation. Once a firm grounding in the Cortex M processor has been established the book introduces the use of a small footprint RTOS and the CMSIS DSP library. With this book you will learn: The key differences between the Cortex M0/M0+/M3 and M4 How to write C programs to run on Cortex-M based processors How to make best use of the Coresight debug system How to do RTOS development The Cortex-M operating modes and memory protection Advanced software techniques that can be used on Cortex-M microcontrollers How to optimise DSP code for the cortex M4 and how to build real time DSP systems An Introduction to the Cortex microcontroller software interface standard (CMSIS), a common framework for all Cortex M- based microcontrollers Coverage of the CMSIS DSP library for Cortex M3 and M4 An evaluation tool chain IDE and debugger which allows the accompanying example projects to be run in simulation on the PC or on low cost hardware

The purpose of this book is to help new Harmony users climb the steep learning curve so they can use the MPLAB Harmony development environment to develop reliable and reproducible applications using Microchip devices. The "learn-by-doing" method used in the book provides a deep understanding of the underlying structures and architecture of Harmony which can be applied to the PIC32MZ (EF) starter kit and/or custom boards. Microchip provides a wide variety of hardware products supported by a comprehensive suite of software development tools. For

newbies, however, choosing which tools and products can be overwhelming. The Microchip starter kits can be used to demonstrate the hardware and software capabilities, but the learning curve can be daunting. Even though Microchip provides wikidot tutorials and thousands of pages of documentation, there is little or no guidance to explain the "how and why" when developing Harmony applications. My first book, "Harmony for PIC32MX Applications," is based on the Ethernet Starter Kit II (MX chip) and targets the new-to-Harmony user. This book is also for a new Harmony user, but it addresses functionality available with the more powerful MZ chip and includes features that are of interest to the intermediate-level developer. I have added new sections to this book that: (a) use the Pin Manager to connect the board peripherals to the application code, and (b) added an entire chapter that addresses how Harmony can be used in conjunction with the real-time operating system, FreeRTOS. For more detailed information, including the Table of Contents and Introduction chapters, go to [www.capritytechnologybooks.com](http://www.capritytechnologybooks.com).

Gain the knowledge and skills necessary to improve your embedded software and benefit from author Jacob Beningo's more than 15 years developing reusable and portable software for resource-constrained microcontroller-based systems. You will explore APIs, HALs, and driver development among other topics to acquire a solid foundation for improving your own software. Reusable Firmware Development: A Practical Approach to APIs, HALs and Drivers not only explains critical concepts, but also provides a plethora of examples, exercises, and case studies on how to use and implement the concepts. What You'll Learn Develop portable firmware using the C programming language Discover APIs and HALs, explore their differences, and see why they are important to developers of resource-constrained software Master microcontroller driver development concepts, strategies, and examples Write drivers that are reusable across multiple MCU families and vendors Improve the way software documented Design APIs and HALs for microcontroller-based systems Who This Book Is For Those with some prior experience with embedded programming.

MicroC/OS II Second Edition describes the design and implementation of the MicroC/OS-II real-time operating system (RTOS). In addition to its value as a reference to the kernel, it is an extremely detailed and highly readable design study particularly useful to the embedded systems student. While documenting the design and implementation of the ker

Enhance your cross-platform programming abilities with the powerful features and capabilities of Qt 6 Key Features Leverage Qt and C++ capabilities to create modern, cross-platform applications that can run on a wide variety of software applications Explore what's new in Qt 6 and understand core concepts in depth Build professional customized GUI applications with the help of Qt Creator Book Description Qt is a cross-platform application development framework widely used for developing applications that can run on a wide range of hardware platforms with little to no change in the underlying codebase. If you have basic knowledge of C++ and want to build desktop or mobile applications with a modern graphical user interface (GUI), Qt is the right choice for you. Cross-Platform Development with Qt 6 and Modern C++ helps you understand why Qt is one of the favorite GUI frameworks adopted by industries worldwide, covering the essentials of programming GUI apps across a multitude of platforms using the standard C++17 and Qt 6 features. Starting with the fundamentals of the Qt framework, including the features offered by Qt Creator, this practical guide will show you how to create classic user interfaces using Qt Widgets and touch-friendly user interfaces using Qt Quick. As you advance, you'll explore the Qt Creator IDE for developing applications for multiple desktops as well as for embedded and mobile platforms. You will also learn advanced concepts about signals and slots. Finally, the book takes you through debugging and testing your app with Qt Creator IDE. By the end of this book, you'll be able to build cross-platform applications with a modern GUI along with the speed and power of native apps. What you will learn Write cross-platform code using the Qt framework to create interactive applications Build a desktop application using Qt Widgets Create a touch-friendly user interface with Qt Quick Develop a mobile application using Qt and deploy it on different platforms Get to grips with Model/View programming with Qt Widgets and Qt Quick Discover Qt's graphics framework and add animations to your user interface Write test cases using the Qt Test framework and debug code Build a translation-aware application Follow best practices in Qt to write high-performance code Who this book is for This book is for application developers who want to use C++ and Qt to create modern, responsive applications that can be deployed to multiple operating systems such as Microsoft Windows, Apple macOS, and Linux desktop platforms. Although no prior knowledge of Qt is expected, beginner-level knowledge of the C++ programming language and object-oriented programming system (OOPs) concepts will be helpful.

Many sensors are currently available at prices lower than USD 100 and cover a wide range of biological signals: motion, muscle activity, heart rate, etc. Such low-cost sensors have metrological features allowing them to be used in everyday life and clinical applications, where gold-standard material is both too expensive and time-consuming to be used. The selected papers present current applications of low-cost sensors in domains such as physiotherapy, rehabilitation, and affective technologies. The results cover various aspects of low-cost sensor technology from hardware design to software optimization.

Artificial intelligence (AI) stands out as a transformational technology of the digital age. Its practical applications are growing very rapidly. One of the chief reasons AI applications are attaining prominence, is in its design to learn continuously, from real-world use and experience, and its capability to improve its performance. It is no wonder that the applications of AI span from complex high-technology equipment manufacturing to personalized exclusive recommendations to end-users. Many deployments of AI software, given its continuous learning need, require computation platforms that are resource intense, and have sustained connectivity and perpetual power through central electrical grid. In order to harvest the benefits of AI revolution to all of humanity, traditional AI software development paradigms must be upgraded to function effectively in environments that have resource constraints, small form factor computational devices with limited power, devices with intermittent or no connectivity and/or powered by non-perpetual source or battery power. The aim this book is to prepare current and future software engineering teams with the skills and tools to fully utilize AI capabilities in resource-constrained devices. The book introduces essential AI concepts from the perspectives of full-scale software development with emphasis on creating niche Blue Ocean small form factored computational environment products.

This book constitutes the refereed proceedings of the 7th International Conference on Wired/Wireless Internet Communications, WWIC 2009, held in Enschede, The Netherlands in May 2008. The 13 revised full papers were carefully reviewed and selected from 39 submissions. The papers are organized in topical sessions on energy efficient WSN design, routing & transport protocols for WSNs, security and protocol design and mobility & handover management.

This volume of Advances in Intelligent Systems and Computing highlights key scientific achievements and innovations in all areas of automation, informatization, computer science, and

artificial intelligence. It gathers papers presented at the IITI 2017, the Second International Conference on Intelligent Information Technologies for Industry, which was held in Varna, Bulgaria on September 14–16, 2017. The conference was jointly co-organized by Technical University of Varna (Bulgaria), Technical University of Sofia (Bulgaria), VSB Technical University of Ostrava (Czech Republic) and Rostov State Transport University (Russia). The IITI 2017 brought together international researchers and industrial practitioners interested in the development and implementation of modern technologies for automation, informatization, computer science, artificial intelligence, transport and power electrical engineering. In addition to advancing both fundamental research and innovative applications, the conference is intended to establish a new dissemination platform and an international network of researchers in these fields.

IoT is empowered by various technologies used to detect, gather, store, act, process, transmit, oversee, and examine information. The combination of emergent technologies for information processing and distributed security, such as Cloud computing, Artificial intelligence, and Blockchain, brings new challenges in addressing distributed security methods that form the foundation of improved and eventually entirely new products and services. As systems interact with each other, it is essential to have an agreed interoperability standard, which is safe and valid. This book aims at providing an introduction by illustrating state-of-the-art security challenges and threats in IoT and the latest developments in IoT with Cloud, AI, and Blockchain security challenges. Various application case studies from domains such as science, engineering, and healthcare are introduced, along with their architecture and how they leverage various technologies Cloud, AI, and Blockchain. This book provides a comprehensive guide to researchers and students to design IoT integrated AI, Cloud, and Blockchain projects and to have an overview of the next generation challenges that may arise in the coming years.

Formal Methods: Foundations and Applications 12th Brazilian Symposium on Formal Methods, SBMF 2009 Gramado, Brazil, August 19-21, 2009 Revised Selected Papers Springer Science & Business Media This book discusses topics in mission-oriented sensor networks and systems research and practice, enabling readers to understand the major technical and application challenges of these networks, with respect to their architectures, protocols, algorithms, and application design. It also presents novel theoretical and practical ideas, which have led to the development of solid foundations for the design, analysis, and implementation of energy-efficient, reliable, and secure mission-oriented sensor network applications. Covering various topics, including sensor node architecture, sensor deployment, mobile coverage, mission assignment, detection, localization, tracking, data dissemination, data fusion, topology control, geometric routing, location privacy, secure communication, and cryptograph, it is a valuable resource for computer scientists, researchers, and practitioners in academia and industry.

An introduction to the engineering principles of embedded systems, with a focus on modeling, design, and analysis of cyber-physical systems. The most visible use of computers and software is processing information for human consumption. The vast majority of computers in use, however, are much less visible. They run the engine, brakes, seatbelts, airbag, and audio system in your car. They digitally encode your voice and construct a radio signal to send it from your cell phone to a base station. They command robots on a factory floor, power generation in a power plant, processes in a chemical plant, and traffic lights in a city. These less visible computers are called embedded systems, and the software they run is called embedded software. The principal challenges in designing and analyzing embedded systems stem from their interaction with physical processes. This book takes a cyber-physical approach to embedded systems, introducing the engineering concepts underlying embedded systems as a technology and as a subject of study. The focus is on modeling, design, and analysis of cyber-physical systems, which integrate computation, networking, and physical processes. The second edition offers two new chapters, several new exercises, and other improvements. The book can be used as a textbook at the advanced undergraduate or introductory graduate level and as a professional reference for practicing engineers and computer scientists. Readers should have some familiarity with machine structures, computer programming, basic discrete mathematics and algorithms, and signals and systems.

Inhaltsangabe: Einleitung: Zur Bereitstellung von Verkehrsinformationen wurde im Bereich Verkehrstelematik am Institut für Automation und Kommunikation e.V. Magdeburg (ifak) ein mobiler Stausensor entwickelt. Dieser wird stationär über der Fahrbahn befestigt und erfasst mithilfe von Verkehrssensoren die Anzahl und Geschwindigkeit von Fahrzeugen auf einer oder mehreren Fahrspuren. Zusätzlich findet eine Klassifikation der Fahrzeuge in Gruppen wie Pkw oder Lkw statt. Die gewonnenen Messdaten werden per Funk an einen Server übermittelt, der durch die Auswertung dieser Daten gezielt Verkehrslenkungsmaßnahmen durchführen kann. Dies ist beispielsweise die Information der Verkehrsteilnehmer durch Wechselverkehrszeichen oder über Internetplattformen. Damit das System autark arbeitet, wird es aus einem Bleiakkumulator mit Energie versorgt, der über ein Solarmodul geladen wird. Mehrere Prototypen des mobilen Stausensors wurden seit 2006 an verschiedenen Standorten in Düsseldorf, Halle und Magdeburg getestet. Während dieser Testphase bestätigten sich die Vorteile der zeitnahen Bereitstellung von Verkehrsinformationen für eine gezielte Beeinflussung des Verkehrs. Andererseits zeigte der Langzeiteinsatz der Prototypen auch Mängel auf. Während der sonnenarmen Wintermonate reichte die durch das Solarmodul aufgenommene Energie in einigen Fällen nicht aus, um die Leistungsaufnahme des Systems zu decken. Die Abbildung 1.2 zeigt den Verlauf der Akkumulatorspannung eines Testgeräts vom Juni 2008 bis Juni 2009. Auf dieser ist ein deutliches Absinken der Spannung in den Wintermonaten zu erkennen. Durch das Erreichen einer Mindestspannung führte dies im Dezember zu mehreren Systemausfällen. Darüber hinaus zeigten sich Mängel durch Verbindungsabbrüche und Systemabstürze bei einigen Prototypen. Hierfür wird die Ursache in der komplexen Betriebssoftware des Systems vermutet. Das Ziel der Studienarbeit ist die Weiterentwicklung des mobilen Stausensors, ausgehend von der vorhandenen Mikrocontrollerplattform. Um Ausfälle des Systems bei geringer Sonneneinstrahlung zu vermeiden, soll die Leistungsaufnahme des Systems analysiert und gegebenenfalls optimiert werden. Besonderes Augenmerk wird dabei auf die energieeffiziente Erzeugung der Versorgungsspannungen und die geringe Verlustleistung der Bauelemente gelegt. Die Komplexität der Betriebssoftware soll durch den Einsatz eines Betriebssystems verringert werden. Dazu werden im zweiten Schritt verfügbare Betriebssysteme [...]

Master the technique of using ESP32 as an edge device in any IoT application where wireless communication can make life easier Key Features Gain practical experience in working with ESP32 Learn to interface various electronic devices such as sensors, integrated circuits (ICs), and displays Apply your knowledge to build real-world automation projects Book Description Developing IoT Projects with ESP32 provides end-to-end coverage of secure data communication techniques from sensors to cloud platforms that will help you to develop production-grade IoT solutions by using the ESP32 SoC. You'll learn how to employ ESP32 in your IoT projects by interfacing with different sensors and actuators using different types of serial protocols. This book will show you how some projects require immediate output for end-users, and cover different display technologies as well as examples of driving different types of displays. The book features a dedicated chapter on cybersecurity packed with hands-on examples. As you progress, you'll get to grips with BLE technologies and BLE mesh networking and work on a complete smart home project where all nodes communicate over a BLE mesh. Later chapters will show you how IoT requires cloud connectivity most of the time and remote access to smart devices. You'll also see how cloud platforms and third-party integrations enable endless possibilities for your end-users, such as insights with big data analytics and predictive maintenance to minimize costs. By the end of this book, you'll have developed the skills you need to start using ESP32 in your next wireless IoT project and meet the project's requirements by building effective, efficient, and secure solutions. What you will learn Explore advanced use cases like UART communication, sound and camera features, low-energy scenarios, and scheduling with an RTOS Add different types of displays in your projects where immediate output to users is required Connect to Wi-Fi and Bluetooth for local network communication Connect cloud platforms through different IoT messaging protocols Integrate ESP32 with third-party services such as voice assistants and IFTTT Discover best practices for implementing IoT security features in a production-grade solution Who this book is for If you are an embedded software developer, an IoT software architect or developer, a technologist, or anyone who wants to learn how to use ESP32 and its

applications, this book is for you. A basic understanding of embedded systems, programming, networking, and cloud computing concepts is necessary to get started with the book.

Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications, Second Edition provides the techniques and technologies in software engineering to optimally design and implement an embedded system. Written by experts with a solution focus, this encyclopedic reference gives an indispensable aid on how to tackle the day-to-day problems encountered when using software engineering methods to develop embedded systems. New sections cover peripheral programming, Internet of things, security and cryptography, networking and packet processing, and hands on labs. Users will learn about the principles of good architecture for an embedded system, design practices, details on principles, and much more. Provides a roadmap of key problems/issues and references to their solution in the text Reviews core methods and how to apply them Contains examples that demonstrate timeless implementation details Users case studies to show how key ideas can be implemented, the rationale for choices made, and design guidelines and trade-offs

[Copyright: bdbc93d709ecd82c3d9da503d7e4789b](#)