Embedded Linux Development Using Eclipse Author Doug Abbott Nov 2008

A practical tutorial guide which introduces you to the basics of Yocto Project, and also helps you with its real hardware use to boost your Embedded Linux-based project. If you are an embedded systems enthusiast and willing to learn about compelling features offered by the Yocto Project, then this book is for you. With prior experience in the embedded Linux domain, you can make the most of this book to efficiently create custom Linux-based systems.

Another day without Test-Driven Development means more time wasted chasing bugs and watching your code deteriorate. You thought TDD was for someone else, but it's not! It's for you, the embedded C programmer. TDD helps you prevent defects and build software with a long useful life. This is the first book to teach the hows and whys of TDD for C programmers. TDD is a modern programming practice C developers need to know. It's a different way to program---unit tests are written in a tight feedback loop with the production code, assuring your code does what you think. You get valuable feedback every few minutes. You find mistakes before they become bugs. You get immediate notification of side effect defects. You get to spend more time adding valuable features to your product. James is one of the few experts in applying TDD to embedded C. With his 1.5 decades of training,coaching, and practicing TDD in C, C++, Java, and C# he will lead you from being a novice in TDD to using the techniques that few have mastered. This book is full of code written for embedded C programmers. You don't just see the end product, you see code and tests evolve. James leads you through the thought process and decisions made each step of the way. You'll learn techniques for test-driving code right nextto the hardware, and you'll learn design principles and how to apply them to C to keep your code clean and flexible. To run the examples in this book, you will need a C/C++ development environment on your machine, and the GNU GCC tool chain or Microsoft Visual Studio for C++ (some project conversion may be needed).

Over 79 hands-on recipes for professional embedded Linux developers to optimize and boost their Yocto Project know-how Key Features Optimize your Yocto setup to speed up development and debug build issues Use what is quickly becoming the standard embedded Linux product builder framework—the Yocto Project Recipe-based implementation of best practices to optimize your Linux system Book Description The Yocto Project has become the de facto distribution build framework for reliable and robust embedded systems with a reduced time to market.You'll get started by working on a build system where you set up Yocto, create a build directory, and learn how to debug it. Then, you'll explore everything about the BSP layer, from creating a custom layer to debugging device tree issues. In addition to this, you'll learn how to add a new hardware layer, packages, data, scripts, and configuration files to your system. You will then cover topics based on application development, such as using the Software Development Kit and how to use the Yocto project in various development environments. Toward the end, you will learn how to debug, trace, and profile a running system. This second edition has been updated to include new content based on the latest Yocto release. What you will learn Optimize your Yocto Project setup to speed up development and debug build issues Use Docker containers to build Yocto Project-based systems Take advantage of the user-friendly Toaster web interface to the Yocto Project build system Build and debug the Linux kernel and its device trees Customize your root filesystem with already-supported and new Yocto packages Optimize your production systems by reducing the size of both the Linux kernel and root filesystems Explore the mechanisms to increase the root filesystem security Understand the open source licensing requirements and how to comply with them when cohabiting with proprietary programs Create recipes, and build and run applications in C, C++, Python, Node.js, and Java Who this book is for If you are an embedded Linux developer with the basic knowledge of Yocto Project, this book is an ideal way to broaden your knowledge with recipes for embedded development.

We are very pleased to introduce Open Source Development, Communities and Quality. The International Conference on Open Source Systems has come to its fourth edition – OSS 2008. Now, Free, Libre, and Open Source software is by all means now one of the most relevant subjects of study in several disciplines, ranging from information technology to social sciences and including also law, business, and political sciences. There are several conference tracks devoted to open source software with several publications appearing in high quality journals and magazines. OSS 2008 has been organized with the purpose of being the reference venue for those working in this area, being the most prominent conference in this area. For this reason OSS 2008 has been located within the frameworks of the 20 World Computer Congress, WCC 2008, in Milan, the largest event of IFIP in 2008. We believe that this conference series, and the IFIP working group it represents, can play an important role in meeting these challenges, and hope that this book will become a valuable contribution to the open source body of research.

Technological, Economic, and Social Perspectives
Embedded Linux System Design and Development
Software Engineering for Embedded Systems
Exploring Raspberry Pi
Exploring BeagleBone
Constraint Logic Programming using Eclipse

This handbook of research is one of the few texts to combine Open Source Software (OSS) in public and private sector activities into a single reference source. It examines how the use of OSS affects practices in society, business, government, education, and law.

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.

*The Eclipse environment solves the problem of having to maintain your own Integrated Development Environment (IDE), which is time consuming and costly. Embedded tools can also be easily integrated into Eclipse. The C/C++CDT is ideal for the embedded community with more than 70% of embedded developers using this language to write embedded code. Eclipse simplifies embedded system development and then eases its integration into larger platforms and frameworks. In this book, Doug Abbott examines Eclipse, an IDE, which can be vital in saving money and time in the design and development of an embedded system. Eclipse was created by IBM in 2001 and then became an open-source project in 2004. Since then it has become the de-facto IDE for embedded developers. Virtually all of the major Linux vendors have adopted this platform, including MontVista, LynuxWorks, and Wind River. *Overview of the latest C/C++ Developer's Toolkit (CDT) *Includes case studies of Eclipse use including Monta Vista, LynuxWorks, and Wind River (IDE) essential to streamlining your embedded development process *Overview of the latest C/C++ Developer's Toolkit (CDT) *Includes case studies of Eclipse use including Monta Vista, LynuxWorks, and Wind River*

This new edition of Linux for Embedded and Real-Time Applications provides a practical introduction to the basics and the latest developments in this rapidly evolving technology. Ideal for those new to using Linux in an embedded environment, it takes a hands-on approach and covers key concepts plus specific applications. Key features include: Substantially updated to focus on a specific ARM-based single board computer (SBC) as a target for embedded application programming Includes an introduction to Android programming With this book you will learn: The basics of Open Source, Linux and the embedded space How to set up a simple system and tool chain How to use simulation for initial application testing Network, graphics and Android programming How to use some of the many Linux components and tools How to configure and build the Linux kernel, BusyBox and U-Boot bootloader Provides a hands-on introduction for engineers and software developers who need to get up to speed quickly on embedded Linux, its operation and its capabilities – including Android Updated and changed accompanying tools, with a focus on the author's specially-developed Embedded Linux Learning Kit

Embedded Linux Primer
Embedded Linux Systems with the Yocto Project
Embedded Linux Development Using Eclipse
Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives
International Conference, EUC 2006, Seoul, Korea, August 1-4, 2006, Proceedings
Open Source Development, Communities and Quality

Embedded Linux Development Using EclipseNewnes
In-depth instruction and practical techniques for building with the BeagleBone embedded Linux platform Exploring BeagleBone is a hands-on guide to bringing gadgets, gizmos, and robots to life using the popular BeagleBone embedded Linux platform. Comprehensive content and deep detail provide more than just a BeagleBone instruction manual-you'll also learn the underlying engineering techniques that will allow you to create your own projects. The book begins with a foundational primer on essential skills, and then gradually moves into communication, control, and advanced applications using C/C++, allowing you to learn at your own pace. In addition, the book's companion website features instructional videos, source code, discussion forums, and more, to ensure that you have everything you need. The BeagleBone's small size, high performance, low cost, and extreme adaptability have made it a favorite development platform, and the Linux software base allows for complex yet flexible functionality. The BeagleBone has applications in smart buildings, robot control, environmental sensing, to name a few; and, expansion boards and peripherals dramatically increase the possibilities. Exploring BeagleBone provides a reader-friendly guide to the device, including a crash course in computer engineering. While following step by step, you can: Get up to speed on embedded Linux, electronics, and programming Master interfacing electronic circuits, buses and modules, with practical examples Explore the Internet-connected BeagleBone and the BeagleBone with a display Apply the BeagleBone to sensing applications, including video and sound Explore the BeagleBone's Programmable Real-Time Controllers Updated to cover the latest Beagle boards, Linux kernel versions, and Linux software releases. Includes new content on Linux kernel development, the Linux Remote Processor Framework, CAN bus, IoT frameworks, and much more! Hands-on learning helps ensure that your new skills stay with you, allowing you to design with electronics, modules, or peripherals even beyond the BeagleBone. Insightful guidance and online peer support help you transition from beginner to expert as you master the techniques presented in Exploring BeagleBone, the practical handbook for the popular computing platform.
The following list describes what you can get from this book: Information that lets you get set up to develop using the Yocto Project. Information to help developers who are new to the open source environment and to the distributed revision control system Git, which the Yocto Project uses. An understanding of common end-to-end development models and tasks. Information about common development tasks generally used during image development for embedded devices. Information on using the Yocto Project integration of the QuickEMUlator (QEMU), which lets you simulate running on hardware an image you have built using the OpenEmbedded build system. Many references to other sources of related information.
Optimize and boost your Linux-based system with Yocto Project and increase its reliability and robustness efficiently and cost-effectively.About This Book* Optimize your Yocto Project tools to develop efficient Linux-based projects* Practical approach to learning Linux development using Yocto Project* Demonstrates concepts in a practical and easy-to-understand wayWho This Book Is ForIf you are an embedded Linux developer with a basic knowledge of Yocto Project and want to broaden your knowledge with examples of embedded development, then this book is for you. This book is also for professionals who want to find new insights into working methodologies for Linux development.What You Will Learn* Understand the basic concepts involved in Poky workflows along with configuring and preparing the Poky build environment.* Configure a build server and customize images using Toaster.* Generate images and fit packages into created images using BitBake.* Support the development process by setting up and using Package feeds.* Debug Yocto Project by configuring Poky.* Build an image for the BeagleBone Black, RaspberryPi 3, and Wandboard, and boot it from an SD card.In DetailYocto Project is turning out to be the best integration framework for creating reliable embedded Linux projects. It has the edge over other frameworks because of its features such as less development time and improved reliability and robustness.Embedded Linux Development using Yocto Project starts with an in-depth explanation of all Yocto Project tools, to help you perform different Linux-based tasks. The book then moves on to in-depth explanations of Poky and BitBake. It also includes some practical use cases for building a Linux subsystem project using Yocto Project tools available for embedded Linux. The book also covers topics such as SDK, recipetool, and others.By the end of the book, you will have learned how to generate and run an image for real hardware boards and will have gained hands-on experience at building efficient Linux systems using Yocto Project.Style and approachA clear, concise, and straightforward book that will enable you to use and implement the latest features of Yocto Project.

Linux for Embedded and Real-time Applications
Linux Kernel Programming
Intelligent Computing
Embedded Linux Development with Yocto Project
IFIP 20th World Computer Congress, Working Group 2.3 on Open Source Software, September 7-10, 2008, Milano, Italy
Building Embedded Linux Systems

Leverage the power of Linux to develop captivating and powerful embedded Linux projects About This Book Explore the best practices for all embedded product development stages Learn about the compelling features offered by the Yocto Project, such as customization, virtualization, and many more Minimize project costs by using open source tools and programs Who This Book Is For If you are a developer who wants to build embedded systems using Linux, this book is for you. It is the ideal guide for you if you want to become proficient and broaden your knowledge. A basic understanding of C programming and experience with systems programming is needed. Experienced embedded Yocto developers will find new insight into working methodologies and ARM specific development competence. What You Will Learn Use the Yocto Project in the embedded Linux development process Get familiar with and customize the bootloader for a board Discover more about real-time layer, security, virtualization, CGL, and LSB See development workflows for the U-Boot and the Linux kernel, including debugging and optimization Understand the open source licensing requirements and how to comply with them when cohabiting with proprietary programs Optimize your production systems by reducing the size of both the Linux kernel and root filesystems Understand device trees and make changes to accommodate new hardware on your device Design and write multi-threaded applications using POSIX threads Measure real-time latencies and tune the Linux kernel to minimize them In Detail Embedded Linux is a complete Linux distribution employed to operate embedded devices such as smartphones, tablets, PDAs, set-top boxes, and many more. An example of an embedded Linux distribution is Android, developed by Google. This learning path starts with the module Learning Embedded Linux Using the Yocto Project. It introduces embedded Linux software and hardware architecture and presents information about the bootloader. You will go through Linux kernel features and source code and get an overview of the Yocto Project components available. The next module Embedded Linux Projects Using Yocto Project Cookbook takes you through the installation of a professional embedded Yocto setup, then advises you on best practices. Finally, it explains how to quickly get hands-on with the Freescale ARM ecosystem and community layer using the affordable and open source Wandboard embedded board. Moving ahead, the final module Mastering Embedded Linux Programming takes you through the product cycle and gives you an in-depth description of the components and options that are available at each stage. You will see how functions are split between processes and the usage of POSIX threads. By the end of this learning path, your capabilities will be enhanced to create robust and versatile embedded projects. This Learning Path combines some of the best that Packt has to offer in one complete, curated package. It includes content from the following Packt products: Learning Embedded Linux Using the Yocto Project by Alexandru Vaduva Embedded Linux Projects Using Yocto Project Cookbook by Alex Gonzalez Mastering Embedded Linux Programming by Chris Simmonds Style and approach This comprehensive, step-by-step, pragmatic guide enables you to build custom versions of Linux for new embedded systems with examples that are immediately applicable to your embedded developments. Practical examples provide an easy-to-follow way to learn Yocto project development using the best practices and working methodologies. Coupled with hints and best practices, this will help you understand embedded Linux better.

Up-to-the-Minute, Complete Guidance for Developing Embedded Solutions with Linux Linux has emerged as today's #1 operating system for embedded products. Christopher Hallinan's Embedded Linux Primer has proven itself as the definitive real-world guide to building efficient, high-value, embedded systems with Linux. Now, Hallinan has thoroughly updated this highly praised book for the newest Linux kernels, capabilities, tools, and hardware support, including advanced multicore processors. Drawing on more than a decade of embedded Linux experience, Hallinan helps you rapidly climb the learning curve, whether you're moving from legacy environments or you're new to embedded programming. Hallinan addresses today's most important development challenges and demonstrates how to solve the problems you're most likely to encounter. You'll learn how to build a modern, efficient embedded Linux development environment, and then utilize it as productively as possible. Hallinan offers up-to-date guidance on everything from kernel configuration and initialization to bootloaders, device drivers to file systems, and BusyBox utilities to real-time configuration and system analysis. This edition adds entirely new chapters on UDEV, USB, and open source build systems. Tour the typical embedded system and development environment and understand its concepts and components. Understand the Linux kernel and userspace initialization processes. Preview bootloaders, with specific emphasis on U-Boot. Configure the Memory Technology Devices (MTD) subsystem to interface with flash (and other) memory devices. Make the most of BusyBox and latest open source development tools. Learn from expanded and updated coverage of kernel debugging. Build and analyze real-time systems with Linux. Learn to configure device files and driver loading with UDEV. Walk through detailed coverage of the USB subsystem. Introduces the latest open source embedded Linux build systems. Reference appendices include U-Boot and BusyBox commands.

Learn how to write high-quality kernel module code, solve common Linux kernel programming issues, and understand the fundamentals of Linux kernel internals Key FeaturesDiscover how to write kernel code using the Loadable Kernel Module frameworkExplore industry-grade techniques to perform efficient memory allocation and data synchronization within the kernelUnderstand the essentials of key internals topics such as kernel architecture, memory management, CPU scheduling, and kernel synchronizationBook Description Linux Kernel Programming is a comprehensive introduction for those new to Linux kernel and module development. This easy-to-follow guide will have you up and running with writing kernel code in next-to-no time. This book uses the latest 5.4 Long-Term Support (LTS) Linux kernel, which will be maintained from November 2019 through to December 2025. By working with the 5.4 LTS kernel throughout the book, you can be confident that your knowledge will continue to be valid for years to come. You'll start the journey by learning how to build the kernel from the source. Next, you'll write your first kernel module using the powerful Loadable Kernel Module (LKM) framework. The following chapters will cover key kernel internals topics including Linux kernel architecture, memory management, and CPU scheduling. During the course of this book, you'll delve into the fairly complex topic of concurrency within the kernel, understand the issues it can cause, and learn how they can be addressed with various locking technologies (mutexes, spinlocks, atomic, and refcount operators). You'll also benefit from more advanced material on cache effects, a primer on lock-free techniques within the kernel, deadlock avoidance (with lockdep), and kernel lock debugging techniques. By the end of this kernel book, you'll have a detailed understanding of the fundamentals of writing Linux kernel module code for real-world projects and products. What you will learnWrite high-quality modular kernel code (LKM framework) for 5.x kernelsConfigure and build a kernel from sourceConfigure and build a kernel from source to grips with key programming details regarding memory management within the kernelUnderstand and work with various dynamic kernel memory alloc/ dealloc APIsDiscover key internals aspects regarding CPU scheduling within the kernelGain an understanding of kernel concurrency issuesFind out how to work with key kernel synchronization primitivesWho this book is for This book is for Linux programmers beginning to find their way with Linux development. If you're a Linux kernel and driver developer looking to overcome frequent and common kernel development issues, or understand kernel internals, you'll find plenty of useful information. You'll need a solid foundation of Linux CLI and C programming before you can jump in.

Linux® is being adopted at an increasing number of embedded systems developers, who have been won over by its sophisticated scheduling and networking, its cost-free license, its open development model, and the support offered by rich and powerful programming tools. While there is a great deal of hype surrounding the use of Linux in embedded systems, there is not a lot of practical information. Building Embedded Linux Systems is the first in-depth, hard-core guide to putting together an embedded system based on the Linux kernel. This indispensable book features arcane and previously undocumented procedures for: Building your own GNU development toolchain Using an efficient embedded development framework Selecting, configuring, building, and installing a target-specific kernel

Creating a complete target root filesystem Setting up, manipulating, and using solid-state storage devices Installing and configuring a bootloader for the target Cross-compiling a slew of utilities and packages Debugging your embedded system using a plethora of tools and techniques Details are provided for various target architectures and hardware configurations, including a thorough review of Linux's support for embedded hardware. All explanations rely on the use of open source and free software packages. By presenting how to build the operating system components from pristine sources and how to find more documentation or help, this book greatly simplifies the task of keeping complete control over one's embedded operating system, whether it be for technical or sound financial reasons Author Karim Yaghmour, a well-known designer and speaker who is responsible for the Linux Trace Toolkit, starts by discussing the strengths and weaknesses of Linux as an embedded operating system. Licensing issues are included, followed by a discussion of the basics of building embedded Linux systems. The configuration, setup, and use of over forty different open source and free software packages commonly used in embedded Linux systems are also covered. uClibc, BusyBox, U-Boot, OpenSSH, thttpd, tftp, strace, and gdb are among the packages discussed.

and its Application to Wireless Networking

Interfacing to the Real World with Embedded Linux

Linux Bible

Hands-On RTOS with Microcontrollers

Mobile Phone Programming

Proceedings of the 2020 Computing Conference, Volume 1

This book constitutes the refereed proceedings of the International Conference on Embedded and Ubiquitous Computing, EUC 2006, held in Seoul, Korea, August 2006. The book presents 113 revised full papers together with 3 keynote articles, organized in topical sections on power aware computing, security and fault tolerance, agent and distributed computing, wireless communications, real-time systems, embedded systems, multimedia and data management, mobile computing, network protocols, middleware and P2P, and more.

Today, Linux is included with nearly every embedded platform. Embedded developers can take a more modern route and spend more time tuning Linux and taking advantage of open source code to build more robust, feature-rich applications. While Gene Sally does not neglect porting Linux to new hardware, modern embedded hardware is more sophisticated than ever: most systems include the capabilities found on desktop systems. This book is written from the perspective of a user employing technologies and techniques typically reserved for desktop systems. Modern guide for developing embedded Linux systems Shows you how to work with existing Linux embedded system, while still teaching how to port Linux Explains best practices from somebody who has done it before

Modern embedded systems are used for connected, media-rich, and highly integrated handheld devices such as mobile phones, digital cameras, and MP3 players. All of these embedded systems require networking, graphic user interfaces, and integration with PCs, as opposed to traditional embedded processors that can perform only limited functions for industrial applications. While most books focus on these controllers, Modern Embedded Computing provides a thorough understanding of the platform architecture of modern embedded computing systems that drive mobile devices. The book offers a comprehensive view of developing a framework for embedded systems-on-chips. Examples feature the Intel Atom processor, which is used in high-end mobile devices such as e-readers, Internet-enabled TVs, tablets, and net books. Beginning with a discussion of embedded platform architecture and Intel Atom-specific architecture, modular chapters cover system boot-up, operating systems, power optimization, graphics and multi-media, connectivity, and platform tuning. Companion lab materials compliment the chapters, offering hands-on embedded design experience. Learn embedded systems design with the Intel Atom Processor, based on the dominant PC chip architecture. Examples use Atom and offer comparisons to other platforms Design embedded processors for systems that support gaming, in-vehicle infotainment, medical records retrieval, point-of-sale purchasing, networking, digital storage, and many more retail, consumer and industrial applications Explore companion lab materials online that offer hands-on embedded design experience

Master the techniques needed to build great, efficient embedded devices on Linux About This Book Discover how to build and configure reliable embedded Linux devices This book has been updated to include Linux 4.9 and Yocto Project 2.2 (Morty) This comprehensive guide covers the remote update of devices in the field and power management Who This Book Is For If you are an engineer who wishes to understand and use Linux in embedded devices, this book is for you. It is also for Linux developers and system programmers who are familiar with embedded systems and want to learn and program the best in class devices. It is appropriate for students studying embedded techniques, for developers implementing embedded Linux devices, and engineers supporting existing Linux devices. What You Will Learn Evaluate the Board Support Packages offered by most manufacturers of a system on chip or embedded module Use Buildroot and the Yocto Project to create embedded Linux systems quickly and efficiently Update IoT devices in the field without compromising security Reduce the power budget of devices to make batteries last longer Interact with the hardware without having to write kernel device drivers Debug devices remotely using GDB, and see how to measure the performance of the systems using powerful tools such as perk, ftrace, and valgrind Find out how to configure Linux as a real-time operating system In Detail Embedded Linux runs many of the devices we use every day, from smart TVs to WiFi routers, test equipment to industrial controllers - all of them have Linux at their heart. Linux is a core technology in the implementation of the inter-connected world of the Internet of Things. The comprehensive guide shows you the technologies and techniques required to build Linux into embedded systems. You will begin by learning about the fundamental elements that underpin all embedded Linux projects: the toolchain, the bootloader, the kernel, and the root filesystem. You'll see how to create each of these elements from scratch, and how to automate the process using Buildroot and the Yocto Project. Moving on, you'll find out how to implement an effective storage strategy for flash memory chips, and how to install updates to the device remotely once it is deployed. You'll also get to know the key aspects of writing code for embedded Linux, such as how to access hardware from applications, the implications of writing multi-threaded code, and techniques to manage memory in an efficient way. The final chapters show you how to debug your code, both in applications and in the Linux kernel, and how to profile the system so that you can look out for performance bottlenecks. By the end of the book, you will have a complete overview of the steps required to create a successful embedded Linux system. Style and approach This book is an easy-to-follow and pragmatic guide with in-depth analysis of the implementation of embedded devices. It follows the life cycle of a project from inception through to completion, at each stage giving both the theory that underlies the topic and practical step-by-step walkthroughs of an example implementation.

Linux: Embedded Development

Test Driven Development for Embedded C

Learn to leverage the power of Yocto Project to build efficient Linux-based products

Learning Embedded Linux Using the Yocto Project

Professional Linux Programming

Mastering Embedded Linux Programming

Expand Raspberry Pi capabilities with fundamental engineering principles Exploring Raspberry Pi is the innovators guide to bringing Raspberry Pi to life. This book favors engineering principles over a 'recipe' approach to give you the skills you need to design and build your own projects. You'll understand the fundamental principles in a way that transfers to any type of electronics, electronic modules, or external peripherals, using a "learning by doing" approach that caters to both beginners and experts. The book begins with basic Linux and programming skills, and helps you stock your inventory with common parts and supplies. Next, you'll learn how to make parts work together to achieve the goals of your project, no matter what type of components you use. The companion website provides a full repository that structures all of the code and scripts, along with links to video tutorials and supplementary content that takes you deeper into your project. The Raspberry Pi's most famous feature is its adaptability. It can be used for thousands of electronic applications, and using the Linux OS expands the functionality even more. This book helps you get the most from your Raspberry Pi, but it also gives you the fundamental engineering skills you need to build basic applications Build your inventory of parts so you can always "make it work" Understand interfacing, controlling, and communicating with almost any component Explore advanced applications with video, audio, real-world interactions, and more Be free to adapt and create with Exploring Raspberry Pi.

This book provides a solid overview of mobile phone programming for readers in both academia and industry. Coverage includes all commercial realizations of the Symbian, Windows Mobile and Linux platforms. The text introduces each programming language (JAVA, Python, C/C++) and offers a set of development environments "step by step," to help familiarize developers with limitations, pitfalls, and challenges.

Constraint logic programming lies at the intersection of logic programming, optimisation and artificial intelligence. It has proved a successful tool in many areas including production planning, transportation scheduling, numerical analysis and bioinformatics. Eclipse is one of the leading software systems that realise its underlying methodology. Eclipse is exploited commercially by Cisco, and is freely available and used for teaching and research in over 500 universities. This book has a two-fold purpose. It's an introduction to constraint programming, appropriate for one-semester courses for upper undergraduate or graduate students in computer science or for programmers wishing to master the practical aspects of constraint programming. By the end of the book, the reader will be able to understand and write constraint programs that solve complex problems. Second, it provides a systematic introduction to the Eclipse system through carefully-chosen examples that guide the reader through the language and illustrate its power, versatility and utility.

The peripheral component interconnect (PCI) bus is the dominant bus system used to connect the different elements making up today's high-performance computer systems. Different PCI implementations have also been developed for such applications as telecommunications and embedded computing. If an application calls for high speed, high reliability, flexible configuration, and bus mastering, then PCI is the only logical bus choice. This book is an applications-oriented introduction to the PCI bus, with an emphasis on implementing PCI in a variety of computer architectures. Special attention is given to industrial and mission-critical applications of PCI bus. -Fully describes PCI electrical specifications, mechanical requirements, and signal types -Covers advanced topics through numerous design examples to increase the readers understanding of the subject -Includes updated coverage of PCI-X 2.0

Embedded Linux Development Using Yocto Project Cookbook

A Practical Real-World Approach

Porting, Extending, and Customizing

A First Course

Practical recipes to help you leverage the power of Yocto to build exciting Linux-based systems, 2nd Edition

Modern Embedded Computing

Embedded Android is for Developers wanting to create embedded systems based on Android and for those wanting to port Android to new hardware, or creating a custom development environment. Hackers and moders will also find this an indispensible guide to how Android works.

Presents instructions for creating Android applications for mobile devices using Java.

This book focuses on the core areas of computing and their applications in the real world. Presenting papers from the Computing Conference 2020 covers a diverse range of research areas, describing various detailed techniques that have been developed and implemented. The Computing Conference 2020, which provided a venue for academic and industry practitioners to share new ideas and development experiences, attracted a total of 514 submissions from pioneering academic researchers, scientists, industrial engineers and students from around the globe. Following a double-blind, peer-review process, 160 papers (including 15 poster papers) were selected to be included in these proceedings. Featuring state-of-the-art intelligent methods and techniques for solving real-world problems, the book is a valuable resource and will inspire further research and technological improvements in this important area.

This Expert Guide gives you the techniques and technologies in software engineering to optimally design and implement your embedded system. Written by experts with a solutions focus, this encyclopedic reference gives you an indispensable aid to tackling the day-to-day problems when using software engineering methods to develop your embedded systems. With this book you will learn: The principles of good architecture for an embedded system Design practices to help make your embedded project successful Details on principles that are often a part of embedded systems, including digital signal processing, safety-critical principles, and development processes Techniques for setting up a performance engineering strategy for your embedded system software How to develop user interfaces for embedded systems Strategies for testing and deploying your embedded system, and ensuring quality development processes Practical techniques for optimizing embedded software for performance, memory, and power Advanced guidelines for developing multicore software for embedded systems How to manage the embedded development process Includes contributions from: Frank Schirrmeister, Shelly Gretlein, Bruce Douglass, Erich Styger, Gary Stringham, Jean Labrosse, Jim Trudeau, Mike Brogioli, Mark Pitchford, Catalin Dan Udma, Markus Levy, Pete Wilson, Whit Waldo, Inga Harris, Xinxin Yang, Srinivasa Addepalli, Andrew McKay, Mark Kraeling and Robert Oshana. Road map of key problems/issues and references to their solution in the text Review of core methods in the context of how to apply them Examples demonstrating timeless implementation details Short and to- the- point case studies show how key ideas can be implemented, the rationale for choices made, and design guidelines and trade-offs

Methods, Practical Techniques, and Applications

Concepts, Techniques, Tricks, and Traps

Computational Technologies

Embedded Linux Development Using Yocto Projects - Second Edition

Embedded Android

A comprehensive guide to kernel internals, writing kernel modules, and kernel synchronization

There's a great deal of excitement surrounding the use of Linux in embedded systems -- for everything from cell phones to car ABS systems and water-filtration plants -- but not a lot of practical information. Building Embedded Linux Systems offers an in-depth, hard-core guide to putting together embedded systems based on Linux. Updated for the latest version of the Linux kernel, this new edition gives you the basics of building embedded Linux systems, along with the configuration, setup, and use of more than 40 different open source and free software packages in common use. The book also looks at the strengths and weaknesses of using Linux in an embedded system, plus a discussion of licensing issues, and an introduction to real-time, with a discussion of real-time options for Linux. This indispensable book features arcane and previously undocumented procedures for: Building your own GNU development toolchain Using an efficient embedded development framework Selecting, configuring, building, and installing a target-specific kernel Creating a complete target root filesystem Setting up, manipulating, and using solid-state storage devices Installing and configuring a bootloader for the target Cross-compiling a slew of utilities and packages Debugging your embedded system using a plethora of tools and techniques Using the uClibc, BusyBox, U-Boot, OpenSSH, thttpd, tftp, strace, and gdb packages By presenting how to build the operating system components from pristine sources and how to find more documentation or help, Building Embedded Linux Systems greatly simplifies the task of keeping complete control over your embedded operating system.

* Detailed installation instructions and step-by-step descriptions of key desktop and server components help new users get up and running immediately * Descriptions of the various distributions from people in the Linux community help users zero in on the best Linux for their needs * The perfect migration guide for Windows and Macintosh desktop users who want to switch to Linux, as well as for systems administrators who want to set up secure, fully functioning server systems * Covers Linux embedded systems, firewalls, and routers plus desktops and servers * Includes Fedora Core 3, Debian Linux, SUSE Linux, Knoppix, Gentoo Linux, Slackware Linux, Mandrake Linux, Damn Small Linux, and a Linux firewall and router on DVD

This book is broken into four primary sections addressing key topics that Linux programmers need to master: Linux nuts and bolts, the Linux kernel, the Linux desktop, and Linux for the Web Effective examples help get readers up to speed with building software on a Linux-based system while using the tools and utilities that contribute to streamlining the software development process Discusses using emulation and virtualization technologies for kernel development and application testing Includes useful insights aimed at helping readers understand how their applications code fits in with the rest of the software stack Examines cross-compilation, dynamic device insertion and removal, key Linux projects (such as Project Utopia), and the internationalization capabilities present in the GNOME desktop

Based upon the authors' experience in designing and deploying an embedded Linux system with a variety of applications, Embedded Linux System Design and Development contains a full embedded Linux system development roadmap for systems architects and software programmers. Explaining the issues that arise out of the use

of Linux in embedded systems, the book facilitates movement to embedded Linux from traditional real-time operating systems, and describes the system design model containing embedded Linux. This book delivers practical solutions for writing, debugging, and profiling applications and drivers in embedded Linux, and for understanding Linux BSP architecture. It enables you to understand: various drivers such as serial, I2C and USB gadgets; uClinux architecture and its programming model; and the embedded Linux graphics subsystem. The text also promotes learning of methods to reduce system boot time, optimize memory and storage, and find memory leaks and corruption in applications. This volume benefits IT managers in planning to choose an embedded Linux distribution and in creating a roadmap for OS transition. It also describes the application of the Linux licensing model in commercial products.
Programming Android

Yocto Project Development Manual
Embedded Linux Development using Yocto Projects
Pro Linux Embedded Systems
Embedded Software for the IoT

Harness the power of Linux to create versatile and robust embedded solutions Key FeaturesLearn how to develop and configure robust embedded Linux devicesExplore the new features of Linux 5.4 and the Yocto Project 3.1 (Dunfell)Discover different ways to debug and profile your code in both user space and the Linux kernelBook Description If you're looking for a book that will demystify embedded Linux, then you've come to the right place. Mastering Embedded Linux Programming is a fully comprehensive guide that can serve both as means to learn new things or as a handy reference. The first few chapters of this book will break down the fundamental elements that underpin all embedded Linux projects: the toolchain, the bootloader, the kernel, and the root filesystem. After that, you will learn how to create each of these elements from scratch and automate the process using Buildroot and the Yocto Project. As you progress, the book will show you how to implement an effective storage strategy for flash memory chips and install updates to a device remotely once it's deployed. You'll also learn about the key aspects of writing code for embedded Linux, such as how to access hardware from apps, the implications of writing multi-threaded code, and techniques to manage memory in an efficient way. The final chapters demonstrate how to debug your code, whether it resides in apps or in the Linux kernel itself. You'll also cover the different tracers and profilers that are available for Linux so that you can quickly pinpoint any performance bottlenecks in your system. By the end of this Linux book, you'll be able to create efficient and secure embedded devices using Linux. What you will learnUse Buildroot and the Yocto Project to create embedded Linux systemsTroubleshoot BitBake build failures and streamline your Yocto development workflowUpdate IoT devices securely in the field using Mender or balenaPrototype peripheral additions by reading schematics, modifying device trees, soldering breakout boards, and probing pins with a logic analyzerInteract with hardware without having to write kernel device driversDivide your system up into services supervised by BusyBox runitDebug devices remotely using GDB and measure the performance of systems using tools such as perf, ftrace, eBPF, and CallgrindWho this book is for If you're a systems software engineer or system administrator who wants to learn how to implement Linux on embedded devices, then this book is for you. It's also aimed at embedded systems engineers accustomed to programming for low-power microcontrollers, who can use this book to help make the leap to high-speed systems on chips that can run Linux. Anyone who develops hardware that needs to run Linux will find something useful in this book – but before you get started, you'll need a solid grasp on POSIX standard, C programming, and shell scripting.

Build Complete Embedded Linux Systems Quickly and Reliably Developers are increasingly integrating Linux into their embedded systems: It supports virtually all hardware architectures and many peripherals, scales well, offers full source code, and requires no royalties. The Yocto Project makes it much easier to customize Linux for embedded systems. If you're a developer with working knowledge of Linux, Embedded Linux Systems with the Yocto Project™ will help you make the most of it. An indispensable companion to the official documentation, this guide starts by offering a solid grounding in the embedded Linux landscape and the challenges of creating custom distributions for embedded systems. You'll master the Yocto Project's toolbox hands-on, by working through the entire development lifecycle with a variety of real-life examples that you can incorporate into your own projects. Author Rudolf Streif offers deep insight into Yocto Project's build system and engine, and addresses advanced topics ranging from board support to compliance management. You'll learn how to Overcome key challenges of creating custom embedded distributions Jumpstart and iterate OS stack builds with the OpenEmbedded Build System Master build workflow, architecture, and the BitBake Build Engine Quickly troubleshoot build problems Customize new distros with built-in blueprints or from scratch Use BitBake recipes to create new software packages Build kernels, set configurations, and apply patches Support diverse CPU architectures and systems Create Board Support Packages (BSP) for hardware-specific adaptations Provide Application Development Toolkits (ADT) for round-trip development Remotely run and debug applications on actual hardware targets Ensure open-source license compliance Scale team-based projects with Toaster, Build History, Source Mirrors, and Autobuilder

The open source nature of Linux has always intrigued embedded engineers, and the latest kernel releases have provided new features enabling more robust functionality for embedded applications. Enhanced real-time performance, easier porting to new architectures, support for microcontrollers and an improved I/O system give embedded engineers even more reasons to love Linux! However, the rapid evolution of the Linux world can result in an eternal search for new information sources that will help embedded programmers to keep up! This completely updated second edition of noted author Doug Abbott's respected introduction to embedded Linux brings readers up-to-speed on all the latest developments. This practical, hands-on guide covers the many issues of special concern to Linux users in the embedded space, taking into account their specific needs and constraints. You'll find updated information on: • The GNU toolchain • Configuring and building the kernel • BlueCat Linux • Debugging on the target • Kernel Modules • Devices Drivers • Embedded Networking • The RTAI environment • And much more The accompanying CD-ROM contains all the source code from the book's examples, helpful software and other resources to help you get up to speed quickly. This is still the reference you'll reach for again and again! * 100+ pages of new material adds depth and breadth to the 2003 embedded bestseller. * Covers new Linux kernel 2.6 and the recent major OS release, Fedora. * Gives the engineer a guide to working with popular and cost-efficient open-source code.

This book offers readers an idea of what embedded Linux software and hardware architecture looks like, cross-compiling, and also presents information about the bootloader and how it can be built for a specific board. This book will go through Linux kernel features and source code, present information on how to build a kernel source, modules, and the Linux root filesystem. You'll be given an overview of the available Yocto Project components, how to set up Yocto Project Eclipse IDE, and how to use tools such as Wic and Swabber that are still under development. It will present the meta-realtime layer and the newly created meta-cgl layer, its purpose, and how it can add value to poky.
PCI Bus Demystified
Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools
Embedded and Ubiquitous Computing
Designing Connected, Pervasive, Media-rich Systems
Create fast and reliable embedded solutions with Linux 5.4 and the Yocto Project 3.1 (Dunfell)
Tools and Techniques for Building with Embedded Linux

Optimize and boost your Linux-based system with Yocto Project and increase its reliability and robustness efficiently and cost-effectively. About This Book Optimize your Yocto Project tools to develop efficient Linux-based projects Practical approach to learning Linux development using Yocto Project Demonstrates concepts in a practical and easy-to-understand way Who This Book Is For If you are an embedded Linux developer with a basic knowledge of Yocto Project and want to broaden your knowledge with examples of embedded development, then this book is for you. This book is also for professionals who want to find new insights into working methodologies for Linux development. What You Will Learn Understand the basic concepts involved in Poky workflows along with configuring and preparing the Poky build environment. Configure a build server and customize images using Toaster. Generate images and fit packages into created images using BitBake. Support the development process by setting up and using Package feeds. Debug Yocto Project by configuring Poky. Build an image for the BeagleBone Black, RaspberryPi 3, and Wandboard, and boot it from an SD card. In Detail Yocto Project is turning out to be the best integration framework for creating reliable embedded Linux projects. It has the edge over other frameworks because of its features such as less development time and improved reliability and robustness. Embedded Linux Development using Yocto Project starts with an in-depth explanation of all Yocto Project tools, to help you perform different Linux-based tasks. The book then moves on to in-depth explanations of Poky and BitBake. It also includes some practical use cases for building a Linux subsystem project using Yocto Project tools available for embedded Linux. The book also covers topics such as SDK, recipetool, and others. By the end of the book, you will have learned how to generate and run an image for real hardware boards and will have gained hands-on experience at building efficient Linux systems using Yocto Project.

In this book wedescribe the basic elements of present computational technologies that use the algorithmic languages C/C++. The emphasis is on GNU compilers and libraries, FOSS for the solution of computational mathematics problems and visualization of the obtained data. At the beginning, a brief introduction to C is given with emphasis on its easy use in scientific and engineering computations.We describe the basic elements of the language, such as variables, data types, executable statements, functions, arrays, pointers, dynamic memory and file management. After that, we present some observations on the C++ programming language.We discuss the issues of program compiling, linking, and debugging. A quick guide to Eclipse is also presented in the book. The main features for editing, compiling, debugging and application assembling are considered.As examples, wesolve the standard problems of computational mathematics: operations with vectors and matrices, linear algebra problems, solution of nonlinear equations, numerical differentiation and integration, interpolation, initial value problems for ODEs and so on. Finally, basic features ofcomputational technologies are illustrated with model problems. All programs are implemented in C/C++ with using the GSL library. Gnuplot is employed to visualize the results of computations.

In-depth instruction and practical techniques for buildingwith the BeagleBone embedded Linux platform Exploring BeagleBone is a hands-on guide to bringinggadgets, gizmos, and robots to life using the popular BeagleBoneembedded Linux platform. Comprehensive content and deep detailprovide more than just a BeagleBone instructionmanual—you'll also learn the underlying engineeringtechniques that will allow you to create your own projects. Thebook begins with a foundational primer on essential skills, andthen gradually moves into communication, control, and advancedapplications using C/C++, allowing you to learn at your own pace.In addition, the book's companion website featuresinstructional videos, source code, discussion forums, and more, toensure that you have everything you need. The BeagleBone's small size, high performance, low cost,and extreme adaptability have made it a favorite developmentplatform, and the Linux software base allows for complex yetflexible functionality. The BeagleBone has applications in smartbuildings, robot control, environmental sensing, to name a few;and, expansion boards and peripherals dramatically increase thepossibilities. Exploring BeagleBone provides areader-friendly guide to the device, including a crash coursein computer engineering. While following step by step, you can: Get up to speed on embedded Linux, electronics, andprogramming Master interfacing electronic circuits, buses and modules, withpractical examples Explore the Internet-connected BeagleBone and the BeagleBonewith a display Apply the BeagleBone to sensing applications, including videoand sound Explore the BeagleBone's Programmable Real-TimeControllers Hands-on learning helps ensure that your new skills stay withyou, allowing you to design with electronics, modules, orperipherals even beyond the BeagleBone. Insightful guidance andonline peer support help you transition from beginner to expert asyou master the techniques presented in Exploring BeagleBone,the practical handbook for the popular computing platform.