

Embedded Programming With Android

Create the perfectly customized system by unleashing the power of Android OS on your embedded device About This Book Understand the system architecture and how the source code is organized Explore the power of Android and customize the build system Build a fully customized Android version as per your requirements Who This Book Is For If you are a Java programmer who wants to customize, build, and deploy your own Android version using embedded programming, then this book is for you. What You Will Learn Master Android architecture and system design Obtain source code and understand the modular organization Customize and build your first system image for the Android emulator Level up and build your own Android system for a real-world device Use Android as a home automation and entertainment system Tailor your system with optimizations and add-ons Reach for the stars: look at the Internet of Things, entertainment, and domotics In Detail Take a deep dive into the Android build system and its customization with Learning Embedded Android Programming, written to help you master the steep learning curve of working with embedded Android. Start by exploring the basics of Android OS, discover Google’s “repo” system, and discover how to retrieve AOSP source code. You’ll then find out to set up the build environment and the first AOSP system. Next, learn how to customize the boot sequence with a new animation, and use an Android “kitchen” to “cook” your custom ROM. By the end of the book, you’ll be able to build customized Android open source projects by developing your own set of features. Style and approach This step-by-step guide is packed with various real-world examples to help you create a fully customized Android system with the most useful features available.

Master the Android mobile development platform Build compelling Java-based mobile applications using the Android SDK and the Eclipse open-source software development platform. Android: A Programmer’s Guide shows you, step-by-step, how to download and set up all of the necessary tools, build and tune dynamic Android programs, and debug your results. Discover how to provide web and chat functions, interact with the phone dialer and GPS devices, and access the latest Google services. You’ll also learn how to create custom Content Providers and database-enable your applications using SQLite. Install and configure Java, Eclipse, and Android plugin Create Android projects from the Eclipse UI or command line Integrate web content, images, galleries, and sounds Deploy menus, progress bars, and auto-complete functions Trigger actions using Android Intents, Filters, and

Receivers Implement GPS, Google Maps, Google Earth, and GTalk Build interactive SQLite databases, calendars, and notepads Test applications using the Android Emulator and Debug Bridge

Barr Group’s Embedded C Coding Standard was developed to help firmware engineers minimize defects in embedded systems. Unlike the majority of coding standards, this standard focuses on practical rules that keep bugs out - including techniques designed to improve the maintainability and portability of embedded software. The rules in this coding standard include a set of guiding principles, as well as specific naming conventions and other rules for the use of data types, functions, preprocessor macros, variables, and other C language constructs. Individual rules that have been demonstrated to reduce or eliminate certain types of defects are highlighted. The BARR-C standard is distinct from, yet compatible with, the MISRA C Guidelines for Use of the C Language in Critical Systems. Programmers can easily combine rules from the two standards as needed.

Build, customize, and debug your own Android system About This Book Master Android system-level programming by integrating, customizing, and extending popular open source projects Use Android emulators to explore the true potential of your hardware Master key debugging techniques to create a hassle-free development environment Who This Book Is For This book is for Android system programmers and developers who want to use Android and create indigenous projects with it. You should know the important points about the operating system and the C/C++ programming language. What You Will Learn Set up the Android development environment and organize source code repositories Get acquainted with the Android system architecture Build the Android emulator from the AOSP source tree Find out how to enable WiFi in the Android emulator Debug the boot up process using a customized Ramdisk Port your Android system to a new platform using VirtualBox Find out what recovery is and see how to enable it in the AOSP build Prepare and test OTA packages In Detail Android system programming involves both hardware and software knowledge to work on system level programming. The developers need to use various techniques to debug the different components in the target devices. With all the challenges, you usually have a deep learning curve to master relevant knowledge in this area. This book will not only give you the key knowledge you need to understand Android system programming, but will also prepare you as you get hands-on with projects and gain debugging skills that you can use in your future projects. You will start by exploring the basic setup of AOSP, and building and testing an emulator image. In the first project, you will learn how to customize and extend the Android emulator. Then you’ll move on to the real challenge—building your own Android system on VirtualBox. You’ll see how to debug the init process, resolve the bootloader issue, and enable various hardware interfaces. When you have a complete system, you will learn how to patch and upgrade it through recovery. Throughout the book, you will get to know useful tips on how to integrate and reuse existing open source projects such as LineageOS (CyanogenMod), Android-x86, Xposed, and GApps in your own system. Style and approach This is an easy-to-follow guide full of hands-on examples and system-level programming tips.

Programming Embedded Systems

Embedded C Coding Standard

Hands-On Mobile and Embedded Development with Qt 5

Develop high performance applications for embedded systems with C++ and Qt 5

Embedded Linux System Design and Development

Tools and Techniques for Low-Power Networking

A comprehensive guide that will get you up and running with embedded software development using Qt5 Key Features Learn to create fluid, cross-platform applications for embedded devices Achieve optimum performance in your applications with QT Lite project Explore the implementation of Qt with IoT using QtMqtt, QtKNX, and QtWebSockets Book Description Qt is an open-source toolkit suitable for cross-platform and embedded application development. This book uses inductive teaching to help you learn how to create applications for embedded and Internet of Things (IoT) devices with Qt 5. You’ll start by learning to develop your very first application with Qt. Next, you’ll build on the first application by understanding new concepts through hands-on projects and written text. Each project will introduce new features that will help you transform your basic first project into a connected IoT application running on embedded hardware. In addition to practical experience in developing an embedded Qt project, you will also gain valuable insights into best practices for Qt development, along with exploring advanced techniques for testing, debugging, and monitoring the performance of Qt applications. Through the course of the book, the examples and projects are demonstrated in a way so that they can be run both locally and on an embedded platform. By the end of this book, you will have the skills you need to use Qt 5 to confidently develop modern embedded applications. What you will learn Understand how to develop Qt applications using Qt Creator under Linux Explore various Qt GUI technologies to build resourceful and interactive applications Understand Qt’s threading model to maintain a responsive UI Get to grips with remote target load and debug under Qt Creator Become adept at writing IoT code using Qt Learn a variety of software best practices to ensure that your code is efficient Who this book is for software and hardware professionals with experience in different domains who are seeking new career opportunities in embedded systems and IoT. Working knowledge of the C++ Linux command line will be useful to get the most out of this book.

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
• Real-time programming tips and techniques
• 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.

An introduction to embedding systems for C and C++++ programmers encompasses such topics as testing memory devices, writing and erasing Flash memory, verifying nonvolatile memory contents, and much more. Original. (Intermediate).

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.

Linux: Embedded Development

Node.js for Embedded Systems

Embedded System Design

Mobile Applications Development with Android

Embedded Android

Introduction to Embedded Systems, Second Edition

As the embedded world expands, developers must have a strong grasp of many complex topics in order to make faster, more efficient and more powerful microprocessors to meet the public’s growing demand. Embedded Software: The Works covers all the key subjects embedded engineers need to understand in order to succeed, including Design and Development, Programming, Languages including C/C++, and UML, Real Time Operating Systems Considerations, Networking, and much more. New material on Linux, Android, and multi-core gives engineers the up-to-date practical know-how they need in order to succeed. Colin Walls draws upon his experience and insights from working in the industry, and covers the complete cycle of embedded software development: its design, development, management, debugging procedures, licensing, and reuse. For those new to the field, or for experienced engineers looking to expand their skills, Walls provides the reader with detailed tips and techniques, and rigorous explanations of technologies. Key features include: New chapters on Linux, Android, and multi-core - the cutting edge of embedded software development! Introductory roadmap guides readers through the book, providing a route through the separate chapters and showing how they are linked About the Author Colin Walls has over twenty-five years experience in the electronics industry, largely dedicated to embedded software. A frequent presenter at conferences and seminars and author of numerous technical articles and two books on embedded software, he is a member of the marketing team of the Mentor Graphics Embedded Software Division. He writes a regular blog on the Mentor website (blogs.mentor.com/colinwalls). New chapters on Linux, Android, and multi-core - the cutting edge of embedded software development! Introductory roadmap guides readers through the book, providing a route through the separate chapters and showing how they are linked

Develop the next killer Android App using Java programming! Android is everywhere! It runs more than half the smartphones in the U.S.—and Java makes it go. If you want to cash in on its popularity by learning to build Android apps with Java, all the easy-to-follow guidance you need to get started is at your fingertips. Inside, you’ll learn the basics of Java and grasp how it works with Android; then, you’ll go on to create your first real, working application. How cool is that? The demand for Android apps isn’t showing any signs of slowing, but if you’re a mobile developer who wants to get in on the action, it’s vital that you get the necessary Java background to be a success. With the help of Java Programming for Android Developers For Dummies, you’ll quickly and painlessly discover the ins and outs of using Java to create groundbreaking Android apps—no prior knowledge or experience required! Get the know-how to create an Android program from the ground up Make sense of basic Java development concepts and techniques Develop the skills to handle programming challenges Find out how to debug your app Don’t sit back and watch other developers release apps that bring in the bucks! Everything you need to create that next killer Android app is just a page away!

Embedded Programming with AndroidBringing Up an Android System from ScratchAddison-Wesley Professional

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.

Embedded Programming in Ada

Hands-On Embedded Programming with Qt

Building Embedded Linux Systems

A Unified Hardware/Software Introduction

A Practical Real-World Approach

Create versatile and robust embedded solutions for MCUs and RTOSes with modern C++

Interested in developing embedded systems? Since they don’t tolerate inefficiency, these systems require a disciplined approach to programming. This easy-to-read guide helps you cultivate a host of good development practices, based on classic software design patterns and new patterns unique to embedded programming. Learn how to build system architecture for processors, not operating systems, and discover specific techniques for dealing with hardware difficulties and manufacturing requirements. Written by an expert who’s created embedded systems ranging from urban surveillance and DNA scanners to children’s toys, this book is ideal for intermediate and experienced programmers, no matter what platform you use. Optimize your system to reduce cost and increase performance Develop an architecture that makes your software robust in resource-constrained environments Explore sensors, motors, and other I/O devices Do more with less: reduce RAM consumption, code space, processor cycles, and power consumption Learn how to update embedded code directly in the processor Discover how to implement complex mathematics on small processors Understand what interviewers look for when you apply for an embedded systems job “Making Embedded Systems is the book for a C programmer who wants to enter the fun (and lucrative) world of embedded systems. It’s very well written—entertaining, even—and filled with clear illustrations.” —Jack Ganssle, author and embedded system expert.

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.

This book introduces a modern approach to embedded system design, presenting software design and hardware design in a unified manner. It covers trends and challenges, introduces the design and use of single-purpose processors (“hardware”) and general-purpose processors (“software”), describes memories and buses, illustrates hardware/software tradeoffs using a digital camera example, and discusses advanced computation models, controls systems, chip technologies, and modern design tools. For courses found in EE, CS and other engineering departments.

In this DIY guide, you will learn how to use Arduino – the open-source hardware board for makers, hobbyists, and inventors. You will learn how to develop your own projects, create prototypes, and produce professional-quality embedded systems. A simple step-by-step demonstration system accompanies you from vision to reality – and just like riding a bike, you’ll get better at it, the more you do it. Featuring a wealth of detailed diagrams and more than 50 fully functional examples, this book will help you get the most out of this versatile tool and bring your electronic inventions to life.

Build apps for Android, iOS, and Raspberry Pi with C++ and Qt

Create fast and reliable embedded solutions with Linux 5.4 and the Yocto Project 3.1 (Dunfell)

Designing Embedded Systems with Arduino

Android Programming

The Works

Building, Customizing, Managing and Operating Android System Services

Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Fully updated for Android Studio 2, the goal of this book is to teach the skills necessary to develop Android based applications using the Android Studio Integrated Development Environment (IDE) and the Android 6 Software Development Kit (SDK). Beginning with the basics, this book provides an outline of the steps necessary to set up an Android development and testing environment. An overview of Android Studio is included covering areas such as tool windows, the code editor and the Designer tool. An introduction to the architecture of Android is followed by an in-depth look at the design of Android applications and user interfaces using the Android Studio environment. More advanced topics such as database management, content providers and intents are also covered, as are touch screen handling, gesture recognition, camera access and the playback and recording of both video and audio. This edition of the book also covers printing, transitions and cloud-based file storage. The concepts of material design are also covered in detail, including the use of floating action buttons, Snackbars, tabbed interfaces, card views, navigation drawers and collapsing toolbars. In addition to covering general Android development techniques, the book also includes Google Play specific topics such as implementing maps using the Google Maps Android API, in-app billing and submitting apps to the Google Play Developer Console. The key new features of Android Studio 2, Instant Run and the new AVD emulator environment, are also covered in detail. Chapters also cover advanced features of Android Studio such as Gradle build configuration and the implementation of build variants to target multiple Android device types from a single project code base.

Assuming you already have some Java programming experience, are ready to download Android Studio and the Android SDK, have access to a Windows, Mac or Linux system and ideas for some apps to develop, you are ready to get started. The number of Android devices running on Intel processors has increased since Intel and Google announced, in late 2011, that they would be working together to optimize future versions of Android for Intel Atom processors. Today, Intel processors can be found in Android smartphones and tablets made by some of the top manufacturers of Android devices, such as Samsung, Lenovo, and Asus. The increase in Android devices featuring Intel processors has created a demand for Android applications optimized for Intel Architecture: Android Application Development for the Intel® Platform is the perfect introduction for software engineers and mobile app developers. Through well-designed app samples, code samples and case studies, the book teaches Android application development based on the Intel platform—including for smartphones, tablets, and embedded devices—covering performance tuning, debugging and optimization. This book is jointly developed for individual learning by Intel Software College and China Shanghai JiaoTong University.

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.

Technologies and Algorithms

The Android Developer's Cookbook

Design Patterns for Great Software

Near Field Communication with Arduino, Android, and PhoneGap

Using Web Technologies to Build Connected Devices

Android Application Development for the Intel Platform

Build safety-critical and memory-safe stand-alone and networked embedded systems Key FeaturesKnow how C++ works and compares to other languages used for embedded developmentCreate advanced GUIs for embedded devices to design an attractive and functional UIIntegrate proven strategies into your design for optimum hardware performanceBook Description C++ is a great choice for embedded development, most notably, because it does not add any bloat, extends maintainability, and offers many advantages over different programming languages. Hands-On Embedded Programming with C++17 will show you how C++ can be used to build robust and concurrent systems that leverage the available hardware resources. Starting with a primer on embedded programming and the latest features of C++17, the book takes you through various facets of good programming. You'll learn how to use the concurrency, memory management, and functional programming features of C++ to build embedded systems. You will understand how to integrate your systems with external peripherals and efficient ways of working with drivers. This book will also guide you in testing and optimizing code for better performance and implementing useful design patterns. As an additional benefit, you will see how to work with Qt, the popular GUI library used for building embedded systems. By the end of the book, you will have gained the confidence to use C++ for embedded programming. What you will learnChoose the correct type of embedded platform to use for a projectDevelop drivers for OS-based embedded systemsUse concurrency and memory management with various microcontroller units (MCUs)Debug and test cross-platform code with LinuxImplement an infotainment system using a Linux-based single board computerExtend an existing embedded system with a Qt-based GUICommunicate with the FPGA side of a hybrid FPGA/SOC systemWho this book is for If you want to start developing effective embedded programs in C++, then this book is for you. Good knowledge of C++ language constructs is required to understand the topics covered in the book. No knowledge of embedded systems is assumed.

Want to get started building applications for Android, the world's hottest, fast-growing mobile platform? Already building Android applications and want to get better at it? This book brings together all the expert guidance—and code—you'll need! Completely up-to-date to reflect the newest and most widely used Android SDKs, The Android Developer's Cookbook is the essential resource for developers building apps for any Android device, from phones to tablets. Proven, modular recipes take you from the absolute basics to advanced location-based services, security techniques, and performance optimization. You'll learn how to write apps from scratch, ensure interoperability, choose the best solutions for common problems, and avoid development pitfalls. Coverage includes: Implementing threads, services, receivers, and other background tasks Providing user alerts Organizing user interface layouts and views Managing user-initiated events such as touches and gestures Recording and playing audio and video Using hardware APIs available on Android devices Interacting with other devices via SMS, web browsing, and social networking Storing data efficiently with SQLite and its alternatives Accessing location data via GPS Using location-related services such as the Google Maps API Building faster applications with native code Providing backup and restore with the Android Backup Manager Testing and debugging apps throughout the development cycle Turn to The Android Developer's Cookbook for proven, expert answers—and the code you need to implement them. It's all you need to jumpstart any Android project, and create high-value, feature-rich apps that sell!

Jump into the world of Near Field Communications (NFC), the fast-growing technology that lets devices in close proximity exchange data, using radio signals. With lots of examples, sample code, exercises, and step-by-step projects, this hands-on guide shows you how to build NFC applications for Android, the Arduino microcontroller, and embedded Linux devices. You'll learn how to write apps using the NFC Data Exchange Format (NDEF) in PhoneGap, Arduino, and node.js that help devices read messages from passive NFC tags and exchange data with other NFC-enabled devices. If you know HTML and JavaScript, you're ready to start with NFC. Dig into NFC's architecture, and learn how it's related to RFID Write sample apps for Android with PhoneGap and its NFC plugin Dive into NDEF: examine existing tag-writer apps and build your own Listen for and filter NDEF messages, using PhoneGap event listeners Build a full Android app to control lights and music in your home Create a hotel registration app with Arduino, from check-in to door lock Write peer-to-peer NFC messages between two Android devices Explore embedded Linux applications, using examples on Raspberry Pi and BeagleBone

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

Making Embedded Systems

Learning Embedded Android N Programming

Linux for Embedded and Real-time Applications

Android System Programming

Programming Android

Embedded Systems

Mobile Applications Development with Android: Technologies and Algorithms presents advanced techniques for mobile app development, and addresses recent developments in mobile technologies and wireless networks. The book covers advanced algorithms, embedded systems, novel mobile app architecture, and mobile cloud computing paradigms. Divided into three sections, the book explores three major dimensions in the current mobile app development domain. The first section describes mobile app design and development skills, including a quick start on using Java to run an Android application on a real phone. It also introduces 2D graphics and UI design, as well as multimedia in Android mobile apps. The second part of the book delves into advanced mobile app optimization, including an overview of mobile embedded systems and architecture. Data storage in Android, mobile optimization by dynamic programming, and mobile optimization by loop scheduling are also covered. The last section of the book looks at emerging technologies, including mobile cloud computing, advanced techniques using Big Data, and mobile Big Data storage. About the Authors Meikang Qiu is an Associate Professor of Computer Science at Pace University, and an adjunct professor at Columbia University. He is an IEEE/ACM Senior Member, as well as Chair of the IEEE STC (Special Technical Community) on Smart Computing. He is an Associate Editor of a dozen of journals including IEEE Transactions on Computers and IEEE Transactions on Cloud Computing. He has published 320+ peer-reviewed journal/conference papers and won 10+ Best Paper Awards. Wenyun Dai is pursuing his PhD at Pace University. His research interests include high performance computing, mobile data privacy, resource management optimization, cloud computing, and mobile networking. His paper about mobile app privacy has been published in IEEE Transactions on Computers. Keke Gai is pursuing his PhD at Pace University. He has published over 60 peer-reviewed journal or conference papers, and has received three IEEE Best Paper Awards. His research interests include cloud computing, cyber security, combinatorial optimization, business process modeling, enterprise architecture, and Internet computing. .

"Get the Java skills you will need to start developing Android apps apps!"--Cover.

How can we build bridges from the digital world of the Internet to the analog world that surrounds us? By bringing accessibility to embedded components such as sensors and microcontrollers, JavaScript and Node.js might shape the world of physical computing as they did for web browsers. This practical guide shows hardware and software engineers, makers, and web developers how to talk in JavaScript with a variety of hardware platforms. Authors Patrick Mulder and Kelsey Breseman also delve into the basics of microcontrollers, single-board computers, and other hardware components. Use JavaScript to program microcontrollers with Arduino and Espruino Prototype IoT devices with the Tessel 2 development platform Learn about electronic input and output components, including sensors Connect microcontrollers to the Internet with the Particle Photon toolchain Run Node.js on single-board computers such as Raspberry Pi and Intel Edison Talk to embedded devices with Node.js libraries such as Johnny-Five, and remotely control the devices with Bluetooth Use MQTT as a message broker to connect devices across networks Explore ways to use robots as building blocks for shared experiences

Current multimedia and telecom applications require complex, heterogeneous multiprocessor system on chip (MPSoC) architectures with specific communication infrastructure in order to achieve the required performance. Heterogeneous MPSoC includes different types of processing units (DSP, microcontroller, ASIP) and different communication schemes (fast links, non standard memory organization and access). Programming an MPSoC requires the generation of efficient software running on MPSoC from a high level environment, by using the characteristics of the architecture. This task is known to be tedious and error prone, because it requires a combination of high level programming environments with low level software design. This book gives an overview of concepts related to embedded software design for MPSoC. It details a full software design approach, allowing systematic, high-level mapping of software applications on heterogeneous MPSoC. This approach is based on gradual refinement of hardware/software interfaces and simulation models allowing to validate the software at different abstraction levels. This book combines Simulink for high level programming and SystemC for the low level software development. This approach is illustrated with multiple examples of application software and MPSoC architectures that can be used for deep understanding of software design for MPSoC.

Beginning NFC

Embedded Software Design and Programming of Multiprocessor System-on-Chip

Embedded Linux Primer

Porting, Extending, and Customizing

A Cyber-Physical Systems Approach

Building Applications with the Android SDK

Android U-Boot Linux NAND U-Boot Goldfish Android ROM

The First Practical, Hands-On Guide to Embedded System Programming for Android Today, embedded systems programming is a more valuable discipline than ever, driven by fast-growing, new fields such as wearable technology and the Internet of Things. In this concise guide, Roger Ye teaches all the skills you'll need to write the efficient embedded code necessary to make tomorrow's Android devices work. The first title in Addison-Wesley's new Android™ Deep Dive series for intermediate and expert Android developers, Embedded Programming with Android™ draws on Roger Ye's extensive experience with advanced projects in telecommunications and mobile devices. Step by step, he guides you through building a system with all the key components Android hardware developers must deliver to manufacturing. By the time you're done, you'll have the key programming, compiler, and debugging skills you'll need for real-world projects. First, Ye introduces the essentials of bare-metal programming: creating assembly language code that runs directly on hardware. Then, building on this knowledge, he shows how to use C to create hardware interfaces for booting a Linux kernel with the popular U-Boot bootloader. Finally, he walks you through using filesystem images to boot Android and learning to build customized ROMs to support any new Android device. Throughout, Ye provides extensive downloadable code you can run, explore, and adapt. You will Build a complete virtualized environment for embedded development Understand the workflow of a modern embedded systems project Develop assembly programs, create binary images, and load and run them in the Android emulator Learn what it takes to bring up a bootloader and operating system Move from assembler to C, and explore Android's goldfish hardware interfaces Program serial ports, interrupt controllers, real time clocks, and NAND flash controllers Integrate C runtime libraries Support exception handling and timing Use U-Boot to boot the kernel via NOR or NAND flash processes Gain in-depth knowledge for porting U-Boot to new environments Integrate U-Boot and a Linux kernel into an AOSP and CyanogenMod source tree Create your own Android ROM on a virtual Android device

Explore Qt framework and APIs for building cross-platform applications for mobile devices, embedded systems, and IoT Key Features Build cross-platform applications and deploy them across mobile and connected devices Design 2D and 3D UIs for embedded systems using Yocto and Qt Creator Build machine to machine automation solution using QtSensors, QtMQTT, and QtWebSockets Book Description Qt is a world-class framework, helping you to develop rich graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. The framework helps you connect the dots across platforms and between online and physical experience. This book will help you leverage the fully-featured Qt framework and its modular cross-platform library classes and intuitive APIs to develop applications for mobile, IoT, and industrial embedded systems. Considerations such as screen size, device orientation changes, and small memory will be discussed. We will focus on various core aspects of embedded and mobile systems, such as connectivity, networking, and sensors; there is no IoT without sensors. You will learn how to quickly design a flexible, fast, and responsive UI that looks great. Going further, you will implement different elements in a matter of minutes and synchronize the UI elements with the 3D assets with high precision. You will learn how to create high-performance embedded systems with 3D/2D user interfaces, and deploy and test on your target hardware. The book will explore several new features, including Qt for WebAssembly. At the end of this book, you will learn about creating a full software stack for embedded Linux systems using Yocto and Boot to Qt for Device Creation. What you will learn Explore the latest features of Qt, such as preview for Qt for Python and Qt for WebAssembly Create fluid UIs with a dynamic layout for different sized screens Deploy embedded applications on Linux systems using Yocto Design Qt APIs for building applications for embedded and mobile devices Utilize connectivity for networked and machine automated applications Discover effective techniques to apply graphical effects using Qt Quick apps Who this book is for The book is ideal for mobile developers, embedded systems engineers and enthusiasts who are interested in building cross-platform applications with Qt. Prior knowledge of C++ is required.

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.

Embedded Software

Simulink and System C Case Studies

A Fundamental Technology for Makers

Mastering Embedded Linux Programming

The Big Nerd Ranch Guide

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 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 in 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 system 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 exper

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.

This book is intended for the practicing programmer or the advanced engineering or computer science student. The presentation of the material presumes a certain level of sophistication on the part of the reader, and concepts found in introductory programming texts are not covered. Some familiarity with operating systems, basic knowledge of block-structured language is recommended. The material is intended for system programming, but the basic concepts apply in any context.

Famed author Jack Ganssle has selected the very best embedded systems design material from the Newnes portfolio and compiled into this volume. The result is a book covering the gamut of embedded design—from hardware to software to integrated embedded systems—with a strong pragmatic emphasis. In addition to specific design techniques and practices, this book also discusses various design patterns and how to successfully apply theory to actual design tasks. The material has been selected for its timelessness as well as for its relevance to contemporary embedded design issues. This book will be an essential working reference for anyone involved in embedded system design! Table of Contents: Chapter 1. Motors - Stuart Ball Chapter 2. Testing - Arnold S. Berger Chapter 3. System-Level Design - Stuart Ball Chapter 4. Actuator and Control Applications and Circuits (Hard Tasks) - Lewin ARW Edwards Chapter 5. Installing and Using a Version Control System - Chris Keydel and Olaf Meding Chapter 6. Embedded State Machine Implementation - Martin Gomez Chapter 7. Firmware Musings - Jack Ganssle Chapter 8. Hardware Musings - Jack Ganssle Chapter 9. Closed Loop Controls, Rabbits, and Hounds - John M. H. Reagle Chapter 10. Examples David J. Katz and Rick Gentile Chapter 11. Analog I/Os - Jean LaBrosse Chapter 12. Optimizing DSP Software - Robert Oshana Chapter 13. Embedded Processors - Peter Wilson *Hand-picked content selected by embedded systems luminary Jack Ganssle *Real-world best design practices including chapters on FPGAs, DSPs, and microcontrollers *Covers both hardware and software aspects of embedded systems *Linux® is being adopted by 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 available. This 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 a target 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 code. 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 basics of 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, tftpd, strace, and gdb are among the packages discussed.

Android Studio 2 Development Essentials

With C and GNU Development Tools

Bringing Up an Android System from Scratch

Programming Embedded Systems in C and C++

ANDROID A PROGRAMMERS GUIDE

Getting Started with Bluetooth Low Energy

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 perf, 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.

Android Programming: The Big Nerd Ranch Guide is an introductory Android book for programmers with Java experience. Based on Big Nerd Ranch's popular Android Bootcamp course, this guide will lead you through the wilderness using hands-on example apps combined with clear explanations of key concepts and APIs. This book focuses on practical techniques for developing apps compatible with Android 4.1 (Jelly Bean) and up, including coverage of Lollipop and material design. Write and run code every step of the way, creating apps that integrate with other Android apps, download and display pictures from the web, play sounds, and more. Each chapter and app has been designed and tested to provide the knowledge and experience you need to get started in Android development. Big Nerd Ranch specializes in developing and designing innovative applications for clients around the world. Our experts teach others through our books, bootcamps, and onsite training. Whether it's Android, iOS, Ruby and Ruby on Rails, Cocoa, Mac OS X, JavaScript, HTML5 or UX/UI, we've got you covered. The Android team is constantly improving and updating Android Studio and other tools. As a result, some of the instructions we provide in the book are no longer correct. You can find an addendum addressing breaking changes at: <https://github.com/bignerdranch/AndroidCourseResources/raw/master/2ndEdition/Errata/2eAddendum.pdf>.

With Bluetooth Low Energy (BLE), smart devices are about to become even smarter. This practical guide demonstrates how this exciting wireless technology helps developers build mobile apps that share data with external hardware, and how hardware engineers can gain easy and reliable access to mobile operating systems. This book provides a solid, high-level overview of how devices use BLE to communicate with each other. You'll learn useful low-cost tools for developing and testing BLE-enabled mobile apps and embedded firmware and get examples using various development platforms—including iOS and Android for app developers and embedded platforms for product designers and hardware engineers. Understand how data is organized and transferred by BLE devices Explore BLE's concepts, key limitations, and network topology Dig into the protocol stack to grasp how and why BLE operates Learn how BLE devices discover each other and establish secure connections Set up the tools and infrastructure for BLE application development Get examples for connecting BLE to iPhones, iPads, Android devices, and sensors Develop code for a simple device that transmits heart rate data to a mobile device

Learn Java for Android Development

Java Programming for Android Developers For Dummies

Inside the Android OS

Hands-On Embedded Programming with C++17

Embedded Programming with Android

Android qian ru shi bian cheng