Unboxing Android USB

A Hands-On Approach with Real World Examples

Rajaram Regupathy

Apress®

Unboxing Android USB

ISBN-13 (pbk): 978-1-4302-6208-4

ISBN-13 (electronic): 978-1-4302-6209-1

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning
Lead Editor: Saswata Mishra
Technical Reviewers: Jan Axelson and Prathap Rajmohan

Coordinating Editor: Mark Powers
Copy Editor: Kezia Endsley
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com/9781430262084. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

To my father

Memories of the past
Remains in my heart
Those days were pleasant
But, still they cannot return

Now, I see my destination,
I ain’t any incarnation
Need to work hard
To be on the right path
With god on my side
Hope to reach it with some pride.

—Rajaram

Contents at a Glance

Contents

About the Author

Rajaram Regupathy has more than 15 years of professional experience in developing firmware and system software-embedded products. He enjoys designing and developing new technology products from scratch. He has patents in embedded domain and is also a senior ACM member. A Linux and open source enthusiast, he has published books on Linux USB stack programming and written numerous open source articles.

About the Technical Reviewers

Jan Axelson is the author of *USB Complete, USB Embedded Hosts*, and other books about hardware programming and design. Jan enjoys experimenting with computer interfaces, especially USB.

Prathap Rajmohan has 15 years of experience in embedded software development and has architected and developed embedded TCP/IPv4/IPv6 and USB stacks for VoIP phones and mobile devices. He is a Principal Software Engineer at Logitech, where he develops embedded firmware for Linux-based products. He holds a B.E. in Electronics and Communication from Anna University and an M.S. in Software Systems from BITS, Pilani.

About the Contributor

Sakethram Bommisetti works as a Senior Software Engineer with Ericsson India and received his bachelor’s degree in Engineering from NIT Nagpur. He has experience in porting different Android USB frameworks and Android USB kernel, and contributed to the development of examples in this book.

Foreword

Like millions of others around the world, I use Android every day. That's because Android is the OS that powers my phone.

Of course, Android isn't limited to phones; it's also popular in embedded systems of all kinds. And every phone or other Android device with a USB port needs programming to manage the USB communications.

When Rajaram Regupathy mentioned to me that he was thinking about writing a book on Android USB programming, I was delighted. I first encountered Rajaram in the USB Experts group he manages on LinkedIn. We discovered we were traveling parallel paths, exploring and writing about the USB universe from different perspectives. Although I'd been writing about USB hardware and programming for 15 years, Android USB programming was something I hadn't yet explored.

Now, writing a programming book is no easy task. You need the fortitude to pore over reams of documentation, the expertise to test and debug what the documents promise (sometimes the documents are wrong!), careful attention to include everything your readers need and nothing more, and an ability to present the information in a clear and logical way.

Rajaram was exactly the person for the job. If you need to program USB communications for Android, this book will put you on the road to success. I'm happy to add *Unboxing Android USB* to the short list of books that I recommend on USB technology.

Jan Axelson

Author of *USB Complete:*
The Developer's Guide and USB Embedded Hosts

Acknowledgments

This is my second book and I look at it as a product that I have taken through various stages, from conceptualizing it, developing it, and finally realizing it. This book would not have been possible without collaboration and support by many people at various stages. I take this opportunity to thank them all.

First, I would like to thank my Acquisition Editor, Saswata Mishra, who played a key role from conceptualization to the production stages of this book, providing effective input and suggestions. I would also like to thank Mark Powers, my Coordinating Editor, who helped make the process completely easy.

Sincere thanks to Jan for her kindness in accepting my request to review the book, for taking the time to review it, and for sharing valuable comments from her experiences. I also thank Prathap for his critical review of the text.

Special thanks to Sakethram for helping me out by preparing the Android examples and also by providing critical inside information.

I also thank my colleagues at Cypress for helping to include the FX3-based examples. Last but not least, I thank my wife, my daughter, and others for making this journey yet again a smooth one.

Introduction

The Android open platform, which was introduced in 2007, is now in more than 50 million devices. The application store statistics show billions of downloads. It has literally conquered the mobile handset market, overtaking many established players. It is also expanding beyond mobile platforms into unique products such as the Android Stick, which converts a normal TV to a smart one.

If you are a developer who works on embedded systems, there is no escape from this ever-growing platform. This inevitability creates a need for good reference books for engineers who are interested in getting started with Android. There are many books in the market covering Android application programming and its development environment. If you are looking for something like that in this book, you are in the wrong place. This book is much more than that. The book explains the complete Android framework, from the API to the internals of Android, along with the kernel below them.

This book exclusively covers the internals of the Android USB framework. Why USB? Similar to the Android platform, USB is also inevitable in the embedded world. On the Android platform, USB is the primary connectivity solution, as an interface used to debug and also as an interface used to charge the batteries of the Android device.

Does this mean this book is only for USB engineers? In fact, it will be useful to any developer working on the Android platform. Why?

If you are a multimedia developer on the Android platform, you need USB for media transfer or to play back audio. This book explores MTP and USB audio in both USB device and USB host modes.

If you are a core developer who works on charging, you need to understand the USB charging specifications, which are explained in the book.

If you are a networking developer interested in tethering, USB plays a role using the RNDIS specification, which is explained in the book.

If you are an application developer interested in managing USB devices from an Android platform, this book explores the Android USB Service framework, which manages USB functionalities.

Last but not least, Android Debug Bridge (ADB), the debugging tool of Android, is over USB and knowledge of its internals is a definite value-add for any application or platform developer. This book details the internals of ABD to the kernel level.

This book covers everything about USB on Android, from the different USB classes supported in device mode to the USB host framework that manages the USB devices connected to the Android platform. Each chapter explains USB class specification before exploring how the functionality (class) is implemented on the Android platform. This gives readers a clean perspective as to what the USB specification demands and how it is implemented in Android.

The Android framework has migrated to different versions by now. As a platform or application developer, it's important you know about the major changes each version introduced. The book covers the major changes in the USB framework between the versions, including interesting bug fixes that were undocumented in the Android specifications.

Intended Audience

The primary audience for this book are application developers and engineers who work hands-on with Android. This book is for an application developer who has an idea for a USB app and wonders how to implement it. This book will be a definite guide for the developer to manage USB on Android.

Because the book covers APIs to the Linux kernel, core platform developers will find it easy to put data point to debug. Thus, core Android platform developers working on USB, audio, media, and others are the next primary audience for the book.

Technical managers, architects, and senior managers who look for the eagle-eye view of a system are a secondary audience for the book. The book will enable them to understand the different blocks of the Android USB subsystem and help estimate the complexity involved.

Student and engineers can use this book as a do-it-yourself reference, as it explains the different blocks of the Android USB framework, from the application level to the kernel.

What You'll Learn

Understand the Android USB framework, from the APIs to the kernel layer, and enable advanced USB application development.

Learn all the major USB functionalities by exploring the USB class specifications not covered in any of the USB books.

Learn the newly introduced Android Open Accessory (AOA) protocol and explore the developing NFC reader using the AOA protocol.

Learn about critical changes in the Android USB framework among different Android versions.

Learn how USB charging works, with an explanation of the USB battery specification.

Learn how to switch between MTP and mass storage and vice versa, in order to share storage with a host PC.

Salient Features

Real-world useful applications enhance your Android experience, including reverse tethering, AOA audio, AOA NFC reader, switching between MTP and UMS, and more. Complete project source is available, which will help you try it on your own.

Covers advanced technical topics (Android and USB) that aren't covered in other texts.

All design diagrams (Microsoft Visio) are on the CD for reuse by developers and architects.

Covers the major differences in the Android USB framework between Android versions.

Covers all major USB functions, such as MTP, audio, charging, and mass storage, along with Google-defined USB functions like ADB and AOA, all by exploring their specifications.

Chapter Introduction

Though there are different types of Android-powered devices, this book details the Android USB framework with a mobile hand-held device in mind. The following section provides a brief description of each chapter in this book.

Getting Started: The Android USB Framework

Android defines its requirement through the Compatibility Definition Document (CDD) and mandates that Android devices comply with this specification. This chapter provides a brief overview of the USB requirements defined in the Android CDD. The chapter subsequently explains various USB-related Android APIs that the Android framework exports for application developers in order to manage USB functionalities or devices.

Discovering and Managing USB Within Android

Discovering and managing a device is the first step and a crucial part any programming activity. This chapter describes how USB function discovery is made inside the Android framework when an Android device is connected in USB device mode. The chapter also details how a USB device is detected inside the Android framework when an Android device is connected in host mode.

USB Storage

Media is one of the key features of mobile devices and is predominantly managed using USB. Media over USB is managed using two USB specifications: Media Transfer Protocol (MTP) and Mass Storage Class (UMS). This chapter briefly details these two specifications and provides an overview of the USB specification's requirements. The chapter also details how media files are transferred to a host PC when the Android device is in USB device mode (both UMS and MTP).

This chapter also explains how a USB-based external media device (say, a USB flash drive or an MTP device) is managed by the Android framework in USB host mode.

USB Tethering

Tethering is a method by which mobile devices shares their Internet connectivity with other devices, such as personal computers or laptops. An Android device uses the RNDIS protocol over USB to tether and share Internet connectivity with other devices. The RNDIS protocol is Microsoft-specific and is very similar to the USB ECM class specification. This chapter provides a brief overview of the RNDIS specification and explains the USB part of the Android framework that facilitates tethering.

USB Accessory

Android Open Accessory (AOA), an Android-specific class defined by Google, was introduced in the Ice Cream Sandwich version of Android to facilitate Android devices in managing external devices. The chapter details the AOA protocol and its operations with an example application. With the Jelly Bean version of Android, the AOA protocol was improved to support the USB Human Interface Device (HID) class. The chapter provides a brief overview of the USB HID class and its implementation inside the Android framework.

USB Audio

The USB audio specification defines transport that provides an efficient way to propagate and control digital audio. With the Jelly Bean version of Android, an Android system in USB device mode supports the USB audio class. This support of digital audio over USB is packed with the AOA protocol. This chapter provides a brief overview of USB audio specification and subsequently explains the Android framework that implements the device audio class. The chapter explains the device and host audio implementations within the Android framework.

Android Debug Bridge

Android Debug Bridge (ADB) is a command-line client/server debug tool that allows you to communicate with an Android-powered device using USB as a transport. This chapter details the ADB protocol defined by Google and subsequently explains how the Android USB framework implements the ADB protocol.

Appendix A: Battery Charging Using USB

Most battery-powered hand-held devices use a USB port to generate power for charging the battery. Android-powered hand-held devices also use USB as the primary power source to charge the battery. This USB class is covered as part of this appendix since there is no real Android USB framework for battery management. This is because USB charging specification focuses on the charging current and other low-level details; there is no USB-level protocol. This chapter provides a brief overview of the USB charging specification and subsequently explains the USB part of the Android battery manager framework.

Appendix B: Using libusb in Android

Protocols like USB allow developers to write driver at user space to manage its functionality. The USB user space driver called libusb is available in almost all popular desktop operating systems. Since libusb is a generic driver, it can be used with any USB device. This chapter explores how to write a simple application over libusb on the Android platform.