异步图书
www.epubit.com

Pearson

[美]罗杰·叶（Roger Ye） 著
师蓉 译

Android 嵌入式编程

Embedded Programming with Android

实用的 Android 嵌入式编程指南

从零开始构建 Android 系统

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

Pearson

[美]罗杰·叶（Roger Ye）著
师蓉 译

Android 嵌入式编程

Embedded Programming with Android

人民邮电出版社
北京

图书在版编目（CIP）数据

Android嵌入式编程 / （美）罗杰·叶（Roger Ye）著 ； 师蓉译. — 北京 ： 人民邮电出版社，2019.1
ISBN 978-7-115-49380-4

Ⅰ. ①A… Ⅱ. ①罗… ②师… Ⅲ. ①移动终端－应用程序－程序设计－指南 Ⅳ. ①TN929.53-62

中国版本图书馆CIP数据核字(2018)第212259号

版权声明

Authorized translation from the English language edition, entitled EMBEDDED PROGRAMMING WITH ANDROID: BRINGING UP AN ANDROID SYSTEMFROM SCRATCH,1E,byYE, ROGER, published by Pearson Education,Inc.,Copyright ©2016.
All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.
CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright©2018.

本书中文简体版由 **Pearson Education,Inc** 授权人民邮电出版社出版。未经出版者书面许可，不得以任何方式或任何手段复制和抄袭本书内容。
本书封面贴有 **Pearson Education**（培生教育出版集团）激光防伪标签，无标签者不得销售。
版权所有，侵权必究。

◆ 著 [美] 罗杰·叶（Roger Ye）
译 师 蓉
责任编辑 吴晋瑜
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
固安县铭成印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：18.25
字数：402 千字 2019 年 1 月第 1 版
印数：1－2 000 册 2019 年 1 月河北第 1 次印刷

著作权合同登记号 图字：01-2016-2068 号

定价：69.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

安徽大学图书馆 藏书 www.lib.ahu.edu.cn

内容提要

本书主要介绍 Android 嵌入式系统编程的相关内容，通过讲解裸机编程、启动加载程序、构建 Android 系统等知识点，旨在帮助读者夯实编程基础，掌握 Android 嵌入式系统的开发、编译及调试技巧。

本书包括三部分内容。第一部分重点介绍裸机编程，主要介绍底层开发和 Android 系统编程的基本原理，涵盖使用 U-Boot 启动 Linux 内核所必需的硬件接口，裸机编程环境中串口的硬件接口编程、实时时钟、NAND 闪存控制器等内容。第二部分介绍将 U-Boot 移植到 Goldfish 平台的方法。第三部分则完成使用虚拟设备为 Android 设备构建定制的 ROM 的实现。本书适合有一定经验的从事 Android 系统开发的开发人员参考，也适合想要探索 Android 底层开发知识的计算机专业学生阅读。

前 言

计算机正变得越来越普及。计算设备逐渐从传统的台式计算机发展到平板电脑和移动设备上。与传统的大型计算机和台式计算机相比，嵌入式计算机在新的平台上发挥着越来越重要的作用。嵌入式系统编程在不同的使用场景中显得非常不同。在某些情况下，它由使用汇编语言和 C 语言直接在硬件上编程的应用程序组成。在其他情况下，它发生在实时操作系统（RTOS）上。在复杂情况下，它可以使用现代操作系统（例如 Linux 或 Windows）的桌面系统。

由于可能存在多种不同的使用场景和硬件架构，因此在学校或者大学中用标准的教学方式讲授嵌入式编程是非常困难的——存在太多基于大量不同架构的硬件平台。处理器或者微处理器可以是简单的 8 位模式，也可以是复杂的 32 位甚至是 64 位设备。但在大多数情况下，学生会根据主板使用某个公司的编译器和调试器学习在专用硬件上嵌入式编程。显然，这种开发环境是独特的，是很难复制的。为了克服这些挑战，本书使用虚拟技术和开源工具来提供任何程序员都可以轻易从互联网上获取的开发环境。

读者对象

如果读者想学习嵌入式系统编程，尤其是 Android 嵌入式系统编程，一定要阅读本书。首先，本书包含许多可供读者尝试的示例，可供初学者从中获得一些实际经验。其次，读者不需要担心硬件平台或开发工具——书中所有示例都是在从互联网下载的开源工具的基础上构建的，并且这些示例都可以在 Android 模拟器上进行测试。本书的源代码托管在 GitHub 上。我们在附录 A 中介绍了配置编译环境的方法，并解释了使用 GitHub 上源代码的方法。

> **注意**
>
> Git 是很多开源项目使用的版本控制工具。如果你还没有听说过它，可以在互联网上搜索“git”或“GitHub”查找相应的使用教程。
>
> GitHub 是互联网上的免费 git 仓库，可以用于托管开源项目。

如果你刚开始作为嵌入式系统软件工程师的职业生涯，所做的第一个项目有可能就是将 U-Boot 移植到新硬件系统中。本书将详细介绍将 U-Boot 移植到 Android 模拟器上的方法。

如果你是有经验的软件开发人员，你可能知道很难在自己的项目中调试复杂的设备驱动程序。我们在本书中将探索一种将硬件接口调试与设备驱动程序开发分开的方式，阐释在裸机环境中调试串口、中断控制器、定时器、实时时钟和 NAND 闪存等的方法。然后，我们将阐释如何将这些示例与 U-Boot 驱动程序整合起来。相同的方法还可以用于 Linux 或 Windows 驱动程

此为试读，需要完整PDF请访问：www.ertongbook.com

序开发。

要想让本书物尽其用，读者必须熟悉 C 语言、基本的操作系统概念和 ARM 汇编语言。本书适合想要探索底层开发知识的计算机科学专业毕业生或者经验丰富的开发人员阅读，也可供想要从事 Android 系统开发的专业人士参考。

本书内容结构

我们在本书中会讨论嵌入式系统编程的全部内容——从基本的裸机编程到启动加载程序，再到 Android 系统的启动。本书的重点是向读者灌输普通的编程知识以及开发编译器和调试的技巧，旨在提供嵌入式系统编程基础知识，为读者提供一条进入嵌入式系统编程领域更好的路径。

本书是以一种面向过程的方式组织的。读者可以决定如何根据自身情况阅读本书，也就是说，以什么顺序阅读本书中的各个章节和探讨内容。

本书由三部分组成。第一部分重点介绍裸机编程。其中包含底层开发和 Android 系统编程的基本原理。第 1 章～第 4 章介绍与裸机编程相关的基础知识，包括使用汇编语言代码直接在硬件上运行程序的方法。第 5 章的重点将转移到 C 编程语言。第一部分的剩余章节将探讨使用 U-Boot 启动 Linux 内核所必需的最小硬件接口集。第 5～8 章的重点是在裸机编程环境中串口的硬件接口编程、中断控制器、实时时钟和 NAND 闪存控制器。

第二部分从第 9 章开始，介绍了将 U-Boot 移植到 Goldfish 平台的方法。正如我们将在第 10 章所介绍的，使用 U-Boot 可以启动 Linux 内核和 Android 系统。第 5～8 章完成的工作有助于 U-Boot 移植——将硬件复杂度从 U-Boot 中的驱动程序框架中分离出来。同样的技术也可以用于 Linux 驱动程序开发。在这一部分，我们还会使用 Android SDK 中的文件系统映像启动 Android 系统。要支持两种不同的启动过程（NOR 和 NAND 闪存），我们必须从 Android SDK 中定制文件系统。由于这项工作发生在二进制级别，因此我们仅限于在文件级别执行定制，也就是说，我们不能修改任何文件的内容。定制文件系统的策略将在本书第三部分中介绍。

第三部分会从启动加载程序转移到内核，再到文件系统。我们使用虚拟设备演示如何为 Android 设备构建定制的 ROM。我们会探讨支持新设备以及将启动加载程序和 Linux 内核整合到 Android 源代码树中的方式。在第 11 章中，我们将深入探讨 Android 模拟器的环境设置过程和标准构建过程。在第 12 章中，我们会为包含 U-Boot 和 Linux 内核整合的虚拟设备创建定制的 ROM。在学完所有内容后，读者可以对 Android 系统开发人员在移动设备制造级别所做的工作有全面的了解。

各章节的主要内容如下。第一部分由第 1～8 章组成，主要介绍裸机编程。

- 第 1 章给出了嵌入式系统编程的简介，并说明了本书的知识范围。
- 第 2 章介绍了 Android 模拟器，并简要介绍了本书使用的硬件接口。
- 第 3 章详细介绍了我们在项目中使用的开发环境和工具，并给出一个可以测试开发环境的示例。

- 第 4 章介绍了开发汇编语言的基础知识。我们用两个示例来分析汇编和链接一个程序的方法。有了二进制的概念后，我们就会分析它如何被加载到 Android 模拟器上。
- 第 5 章介绍了 C 的启动代码，并说明了如何从汇编语言转移到 C 语言环境。我们还将探讨 Goldfish 平台的 Goldfish 硬件接口和串口。
- 第 6 章详细介绍了将 C 运行时库整合到裸机编程环境中的方法。我们介绍了不同风格的 C 运行时库，并以 Newlib 作为说明如何整合 C 运行时库的示例。
- 第 7 章探讨了 Goldfish 平台的中断控制器、定时器和实时时钟。我们会通过各种示例演示处理这些硬件接口的方式。本章开发的所有示例代码都可以用于第 9 章的 U-Boot 移植。
- 第 8 章探讨了 Goldfish 平台的 NAND 闪存接口。它同样是 U-Boot 移植的重要部分。在第 10 章中，我们会探讨从 NAND 闪存引导 Android 系统的方法。

第二部分由第 9 章和第 10 章组成，介绍了 U-Boot 移植和调试的过程。有了可工作的 U-Boot 映像后，我们就可以用它启动自己的 Goldfish 内核和 Android 映像。

- 第 9 章给出了 U-Boot 移植的详细过程。
- 第 10 章讨论了如何构建我们自己的 Goldfish Linux 内核。然后，内核映像被用于演示使用 U-Boot 启动 Goldfish Linux 内核的不同场景。我们会讨论 NOR flash 和 NAND 闪存启动过程。

第三部分探讨将 U-Boot 和 Linux 内核集成到 Android 开源项目（AOSP）和 CyanogenMod 源树中的方法。

- 第 11 章介绍了在 AOSP 和 CyanogenMod 上构建 Android 模拟器的详细方法。
- 第 12 章介绍如何在虚拟 Android 设备上创建自己的 Android ROM。这个 Android ROM 可由我们在第 9 章创建的 U-Boot 激活。

示例代码

本书给出了很多可用于测试各章内容的示例。我们建议读者在阅读本书时输入并运行这些示例代码。这样做可以为读者提供良好的实践经验和有价值的见解，以便读者能更好地理解各章所涵盖的主题。

对于第 3～8 章来说，目录结构是按章节来组织代码的。有些文件夹在所有示例中都很常见，例如包含 include 和驱动程序文件的那些文件夹。所有其他文件夹都是与章节相关的，例如 c03、c04 和 c05，这些文件夹包含对应章节的示例代码。

常见的生成文件是位于顶层目录中的 `makedefs.arm`。本书给出了每个示例的生成文件。示例代码的一个生成文件模板如下所示。`PROJECTNAME` 被定义为一个示例代码的文件夹。这个生成文件模板可用于第 3～8 章的项目。

```
#
# The base directory relative to this folder
```

```
#
ROOT=../..
PROJECTNAME=
#
# Include the common make definitions.
#
include ${ROOT}/makedefs.arm
#
# The default rule, which causes the ${PROJECTNAME} example to be built.
#
all: ${COMPILER}
all: ${COMPILER}/${PROJECTNAME}.axf
#
# The rule to debug the target using Android emulator.
#
debug:
        @ddd --debugger arm-none-eabi-gdb ${COMPILER}/${PROJECTNAME}.axf &
        @emulator -verbose -show-kernel -netfast -avd hd2 -shell -qemu -monitor
telnet::6666,server -s -S -kernel ${COMPILER}/${PROJECTNAME}.axf
#
# The rule to clean out all the build products.
#
clean:
        @rm -rf ${COMPILER} ${wildcard *~}
#
# The rule to create the target directory.
#
${COMPILER}:
        @mkdir -p ${COMPILER}

#
# Rules for building the ${PROJECTNAME} example.
#

${COMPILER}/${PROJECTNAME}.axf: ${COMPILER}/${PROJECTNAME}.o
${COMPILER}/${PROJECTNAME}.axf: ${PROJECTNAME}.ld
SCATTERgcc_${PROJECTNAME}=${PROJECTNAME}.ld
ENTRY_${PROJECTNAME}=ResetISR
#
# Include the automatically generated dependency files.
#
ifneq (${MAKECMDGOALS},clean)
-include ${wildcard ${COMPILER}/*.d} __dummy__
endif
```

本书中的源代码可在 GitHub 上找到，要了解更多细节，请参阅附录 A。

致 谢

感谢培生技术出版集团的主编 Laura Lewin 和 Bernard Goodwin，感谢他们给我在 Addison-Wesley 出版这本书的机会。感谢 Addison-Wesley 出版社团队的全体成员。感谢策划编辑 Michael Thurston，他审查了所有章节并提出了关于内容介绍的宝贵意见；感谢 Olivia Basegio 和 Michelle Housley 帮助我与 Addison-Wesley 的团队协作；感谢执行编辑 Elizabeth Ryan 确保这个项目按照计划进行；感谢文字编辑 Jill Hobbs 为提高本书可读性所做的努力。

如果没有经过技术审查，本书就不可能出版。在此感谢所有指出本书错误并提供宝贵意见的审稿人。尤其要感谢 Android 专家、《Enterprise Android》和《Programming Android》的合著者：Zigurd Mednieks 和 G. Blake Meike。

还要感谢我所有的朋友以及摩托罗拉公司和埃莫森公司的所有同事。我们一起开发了很多促成过去十年来技术繁荣的伟大产品。我们还一起见证了改变了我们今天生活的高科技产品的推出。

最后，我要感谢我的爱人和女儿，谢谢她们在我写这本书时给我的支持和鼓励。

作者简介

Roger Ye 是一名对嵌入式系统及其相关技术有着极大兴趣的嵌入式系统程序员。他曾在摩托罗拉公司、埃莫森公司和英特尔公司担任技术经理。在摩托罗拉公司和埃莫森公司工作时，他参与了移动设备和电信基础设施的嵌入式系统项目。目前他是英特尔公司安全支持部门的技术经理，领导着开发 Android 应用程序的团队。

目 录

第一部分 裸机编程

此为试读，需要完整PDF请访问：www.ertongbook.com

第三部分 Android 系统集成

第一部分
裸机编程

第1章
嵌入式系统编程简介

在阅读第一本教科书时，我以为它肯定是在告诉我这个世界的真相。很多年后的今天，回首过去，我才明白每本书只是从作者的角度讲述这个世界的真相。

同样的想法也适用于嵌入式系统编程。市面上有很多与嵌入式系统编程相关的书籍，每位作者都不可避免地从自己的角度分享经验。在本书中，我也从这个角度来介绍嵌入式系统编程。

1.1 什么是嵌入式系统

嵌入式系统是为最终用户提供专用功能的计算设备或组件。它可以是一个大系统的一部分，也可以是一个独立的设备。我们生活中的很多应用程序和其他设备都被称为嵌入式系统。有些是我们每天都会直接使用的，例如 DVD 播放器、扫描仪、打印机、开关和路由器等。其他则是隐藏在大系统中的，例如基站、卫星、电梯控制装置、汽车发动机控制装置、医院设备控制装置和成像系统等。

嵌入式系统既可以是一个简单的设备，也可以是一个复杂的系统。它们包含低成本的设备和复杂的高成本系统。这些设备和组件可以使用任何满足设计目标的硬件架构。

显然，要为这个宽泛的主题提供完整的介绍是很难的。我们在本书中将通过一个典型的示例探索嵌入式系统的世界以及嵌入式系统编程的发展概况。

1.2 裸机编程

裸机编程意味着直接将代码写在硬件上，也就是说，在程序下面没有其他软件层。这种做法在对微处理器（MCU）编程时很常见。

很多书籍都侧重于嵌入式编程，却很少详细讲解裸机编程。然而，如果你在互联网上搜索裸机编程，就会发现很多关于这个主题的文章和讨论。很多书籍通常不讲解裸机编程的原因是这种编程严重依赖于硬件，因此很难做到让关于这个主题的书籍适用于所有读者，让他们都能

从中受益。我们谈论裸机编程时，必然会涉及设计特定的硬件参考板。当然，并不是所有读者手里都有参考板。本书将用虚拟环境解决这个硬件依赖问题，具体来说，是用 Android 模拟器作为硬件参考板。

想要进行裸机编程的原因有很多，但最简单的系统中的硬件限制是主要动机之一。在最简单的嵌入式系统中，可能会使用微处理器。这种系统中的硬件资源可能非常有限，以至于它不能运行自己的操作系统。在这种情况下，直接在硬件上运行的小程序是唯一的选择。

我们有时也会在高级或者复杂的系统中进行裸机编程。在带有微处理器的复杂系统中，最终用户环境中可能会有操作系统。然而，对于研究实验室中的芯片级硬件验证来说，可能很难使用整个软件堆栈在操作系统中验证初始芯片。最简单的方法是直接在硬件上创建一个简单的环境，以便芯片设计师和验证团队可以专注于硬件验证本身。如果你在硬件参考板开发团队工作过，就会知道很多硬件模块的初始代码是由验证团队或者硬件设计师提供的。他们提供直接在硬件上运行的代码片段，而不是依赖于一个操作系统。实际上，操作系统的设备驱动程序开发人员在使用硬件规范作为参考的同时，可能会根据测试代码开发硬件驱动程序。

裸机编程主要是使用 C 语言和汇编语言编写的，因为这两种语言都可以在没有（或者只有最少的）运行时库支持的情况下使用。这意味着可以在内存的任何位置加载程序。我们可以通过执行复位向量运行程序，让硬件首先取指令。随后，添加 C 运行时库做一些简单但非常有用的事情，例如用 printf 函数提供错误消息。

为了让读者能跟紧我们的步伐，我们将使用汇编语言、C 语言和你可以在计算机上下载并运行的虚拟硬件——即 Android 模拟器或 Goldfish 模拟器。我们将从头开始编写程序。最初，程序是以汇编语言编写的，但我们会尝试尽快将其转换成 C 语言。在这个过程中，我们将探讨硬件接口并直接在虚拟硬件上创建实验代码。稍后我们会在 U-Boot 移植中使用这个实验代码。

注意

Goldfish 虚拟硬件板是在 Android 模拟器中定义的虚拟硬件，我们将在第 2 章详细讨论它。

在完成系统构建后，我们就会尽可能地整合现有的技术，以减少用于构建最终系统的时间。我们会整合 C 运行时库的代码和重用硬件外设的代码，并逐步构建启动加载程序（U-Boot）、Goldfish 内核和文件系统。在本书结束时，读者应该能对嵌入式系统是如何构建的以及开发环境是什么样的有一个清楚的了解。

本书侧重于动手实践。此外，读者可能需要自己探索与 ARM 架构、汇编语言或 C 语言以及 Android 系统相关的细节。一些有用的资源如下。

- 《ARM System Developer's Guide》（作者 Andrew N. Sloss、Dominic Symes 和 Chris Wright）。
- ARM 体系结构参考手册 ARMv7-A 和 ARMv7-R 版。

此为试读，需要完整PDF请访问：www.ertongbook.com