

Wrox Programmer to Programmer™

嵌入式系统经典丛书

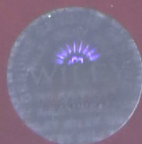
嵌入式 ARM 开发实战

Professional Embedded ARM Development

[美] James A. Langbridge 著

陈青华 张龙杰 司维超 译

 **wrox**
A Wiley Brand



清华大学出版社

嵌入式 ARM 开发实战

[美] James A. Langbridge 著

陈青华 张龙杰 司维超 译

清华大学出版社

James A. Langbridge
Professional Embedded ARM Development
EISBN: 978-1-118-78894-3
Copyright © 2014 by John Wiley & Sons, Inc.
All Rights Reserved. This translation published under license.

Trademarks: Wiley, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2014-4852

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

* 图书在版编目(CIP)数据

嵌入式 ARM 开发实战 / (美) 兰布里奇(Langbridge, J.A.) 著; 陈青华, 张龙杰, 司维超 译. —北京: 清华大学出版社, 2015
(嵌入式系统经典丛书)

书名原文: Professional Embedded ARM Development

ISBN 978-7-302-39619-2

I. ①嵌… II. ①兰…②陈…③张…④司… III. ①微处理器—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2015)第 049228 号

责任编辑: 王 军 于 平

装帧设计: 牛静敏

责任校对: 曹 阳

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 148mm×210mm 印 张: 12.25 字 数: 566 千字

版 次: 2015 年 5 月第 1 版 印 次: 2015 年 5 月第 1 次印刷

印 数: 1~3500

定 价: 49.80 元

译者序

由清华大学出版社引进并出版的《嵌入式 ARM 开发实战》(Wrox 出版社, 作者 James A. Langbridge)是一本关于嵌入式 ARM 系统的入门教材。

从教材内容来看, 本书涵盖 ARM 嵌入式项目的所有方面, 包括处理器和扩展处理器、汇编程序、调试和优化。与传统的嵌入式 ARM 教材相比, 本书有以下特点:

- 介绍了各种常见的 ARM 处理器, 涵盖了嵌入式 ARM 系统启动、运行及开发的过程, 使读者能轻松地进入 ARM 开发。
- 教材内容秉承“由程序员为程序员而著”的理念, 通过程序示例的讲解使读者快速开始系统编程和调试。
- 注重代码的调试和优化, 通过列举真实案例描述了何为调试以及如何优化。
- 书中所有源代码可以从 www.wrox.com 和 <http://www.tupwk.com.cn/downpage> 下载, 加入 Wrox P2P 论坛可以与 Wrox 作者、编辑或其他行业专家进行交流。

作者 James A. Langbridge 长期从事 ARM 系统开发, 拥有 15 年以上的嵌入式系统开发经验。本书既可以作为嵌入式领域开发人员的入门教材, 也可作为开发人员深入理解 ARM 处理器及内核的参考教材。

本书由海军航空工程学院的陈青华、张龙杰和司维超老师翻译。参与本次翻译工作的还有洪贝、乔福超、初丰文。由于本书内容新、翻译难度较大, 加之译者水平有限, 书中难免有疏漏或错误, 欢迎广大读者批评指正。

陈青华

作者简介

James A. Langbridge 不喜欢以第三人称谈论自己，但不管怎样还是会尝试一下。James 出生于新加坡，随父母居住多国，后定居法国南特，与父母及女儿同住。

作为嵌入式系统顾问，他在工业、军事、移动电话和航空安全系统领域拥有 15 年以上的从业经验。他主要从事初级开发，用汇编语言创建引导区加载或优化程序，以及充分利用大多数小型处理器。工作之余，James 培训嵌入式系统工程师或者热衷于创新小发明，虽然同伴们对此不太感兴趣。

James 在年仅 6 岁时开始他的计算机编程人生，并一直从未停止修修补补。在加入英国广播公司电脑部前，当最终别无选择只能使用个人电脑时，他开始使用 Apple IIs、ZX80s 和 ZX81s。

技术编辑简介

Chris Shore，在总部位于英国剑桥的 ARM 有限公司任职于培训和教育主管部门。拥有 13 年的培训客户经验。每年，Chris 开设从芯片设计到软件优化的 200 门课程。Chris 所教授的学员分布在除南极洲之外的所有大洲。这只是因为南极洲机会有限，但也只是时间问题。他经常在行业会议上发表演讲。

在剑桥大学计算机专业毕业获得学位后，在 1999 年加入 ARM 公司前，Chris 作为软件咨询师工作了 15 年，主要致力于嵌入式实时系统研究。他是特许工程师和工程与技术学会会员，同时也是伦敦玛丽皇后学院的董事会成员。业余时间，Chris 喜欢饲养蜜蜂、弹吉他，并热衷于周游列国。

Jean-Michel Hautbois，居住于法国，从事专业性软件开发已经超过 15 年了。现从事嵌入式 Linux 咨询顾问工作，主要负责基于 ARM 的 SoC 芯片的视频产品的开发。他参与新硬件产品开发和重要的决策过程。闲暇时间，Jean-Michel 喜欢和妻子、儿子外出旅游。

致 谢

对于那些在此项目上给予我帮助的人们，我表示无上感激。首先，所有初入这一领域的初级开发员们，希望我已回答了你们的各种问题。感谢 LinkedIn ARM 项目组提供的优秀资源，你们所贡献的所有问题和答案对于我有莫大的帮助。

感谢 ARM 公司的每一位成员对于我的帮助。在此，我要特别感谢两位技术编辑 Chris Shore 和 Jean-Michel Hautbois 先生。没有你们的帮助，这本书不可能完成。我还要感谢 Philippe Lancon 先生的支持，Atmel 和 Silicon Labs 的大力帮助，以及所有在这个项目上给予我帮助的朋友和家庭成员。当然，还要感谢 Wiley 团队，每一次我有疑问时，你们总是帮助我解决。最后，还要感谢 Christina Haviland、Mary James、San Dee Phillips，以及每一位帮助我完成这本书的人，感谢你们。

前 言

在嵌入式系统的领域内,不借助 ARM CPU 难以长期工作。ARM CPU 因其较低的耗电量成为理想的移动嵌入式系统。自 2012 年以来,几乎所有 PDA 和智能手机都使用了 ARM CPU,其占据了 32 位嵌入式系统 75%和嵌入式 RISC 系统的 90%的份额。2005 年,售出的 10 亿多部手机中 98%使用至少一个 ARM 处理器。ARM 处理器使用在手机、平板计算机、MP3 播放器、手持游戏机、计算器,甚至计算机外围设备,如蓝牙芯片和硬盘驱动器中。

每两个月都有超过 10 亿的 ARM 处理器被送到使用者手中。令人惊讶的是,ARM 并非作为处理器而是作为设计内核被使用。ARM 的合作伙伴们使用这些设计作为基础,添加自己的外部设备、外围设备,或修改 ARM 的原有速度和功耗功率,从而制成适合自己的处理器。通过与制造商合作,ARM 创造了一个巨大的生态系统。结果是诞生了一系列非凡的处理器,可用于所有类型的设备。使用共同的架构,开发人员能够轻松地切换各个处理器。

ARM 处理器不是只拥有有限的微型处理器性能。从用于最小系统的微控制器设备到运用 64 位处理器的服务器都可运用 ARM 处理器。

本书介绍嵌入式 ARM 系统,如何启动并运行该系统,如何在该平台上进行开发以及在此巨大的生态系统中可运用的设备。

本书适用对象

本书主要面向初入嵌入式领域的开发人员。对大多数例子来讲，需要基本了解 C 语言，但不需要掌握汇编语言的知识。

该书也便于开发人员更好地掌握处理器的内部构件和深入理解内核工作。

本书涵盖内容

本书涵盖了 ARM 处理器技术的高级内容，着重关注运用于 Cortex-A、Cortex-R 和 Cortex-M 设备的 ARMv7 架构。如果使用了 Cortex 系列处理器，你会感觉运用自如。但如果使用 ARM 经典内核，你还可以掌握架构和平台之间的区别。

本书结构安排

本书为不具备 ARM 处理器使用经验的读者提供尽可能丰富的信息。要理解 ARM 哲学，有必要了解 ARM 的起源和产生过程。本书涵盖了嵌入式项目的所有方面：理解处理器和扩展处理器，理解汇编程序，用更常见的 C 语言生成第一个程序，并且继续进行调试和优化。

第 1 章“ARM 发展史”，概述了 ARM 的传奇历史和起源，以及发展现状。

第 2 章“ARM 嵌入式系统”，介绍了何为嵌入式系统，讲述了嵌入式系统的优势。

第 3 章“ARM 架构”，列举了 ARM 处理器的组成元件及其有效的使用方法。

第 4 章“ARM 汇编语言”，介绍了 ARM 汇编语言，并解释了理解汇编语言的重要性。

第 5 章“ARM 入门”，展示一些模拟器和真实芯片来编写程序，不仅有 ARM 处理器启动而且包括更复杂程序应用基础的编程。本章还列举了一些真实世界的场景项目。

第 6 章“Thumb 指令集”，介绍了 Thumb 模式和 Thumb-2 扩展。Cortex-M 处理器仅用于 Thumb 模式，但在代码密度要求高的地方 Thumb 可用于每一个现代处理器。

第 7 章“汇编指令”，列举了 ARM 统一汇编语言中的一系列汇编指令。用浅显易懂的程序介绍了这一系列汇编指令的用法。

第 8 章“NEON”，介绍了 NEON，ARM 先进的单指令多数据处理器，并展示了如何运用它来加速数学密集的例程。

第 9 章“调试”，通过列举真实案例描述了何为调试、调试程序所需要的以及调试的成效。

第 10 章“编写优化 C 程序”，描述了应用程序的最终部分——优化。通过简单案例，可以学习如何编写优化代码，并了解处理器内部实施进一步的优化的过程。

附录 A“术语”，解释了使用嵌入式系统(具体指 ARM 嵌入式系统)工作时遇到的词汇和术语。

附录 B“ARM 架构版本”，列举了不同的 ARM 架构并解释了每一个版本架构的技术优点，以及处理器所属的架构。

附录 C“ARM 内核版本”，详细介绍 ARM 内核，介绍每个处理器的变化。重点讨论了始于 ARM6 的每个处理器。引入更多细节介绍现代的 Cortex-级处理器。

附录 D“NEON 内部函数和指令”，列举出可用于 ARM NEON 引擎的不同指令，同时使用 C 语言介绍了用于优化 NEON 计算的内部函数执行方式。

附录 E“汇编语言指令”，列举并详细描述了运用于 UAL 的不

同汇编指令，同时列举了运用于不同 Cortex-M 处理器的一系列 Thumb 指令。

使用本书的必要条件

大多数人猜想嵌入式系统被昂贵的电子设备和高价的软件授权所包围，但事实是仅用很少的投入就可以开始嵌入式开发。首先，需要一台开发计算机。本书给出了 Linux 系统下的例子，也可以在 Windows 和 MacOS 系统下使用。本书提供了免费版权的编译器，可以使用免费的 ARM 模拟器来编写第一个程序，但是稍后使用一个小型的 ARM 系统将是明智的：不论是来自于一个制造商(指定两个)的评估版，还是使用廉价的小型计算机，如 Raspberry Pi(树莓派)或者 Arduino Due。

要运行书中的示例，需要具备以下条件：

- Linux 系统开发计算机
- Mentor 图形编译器套件
- Atmel SAM D20 Xplained Pro evaluation board
- Silicon Lab's STK3200 and STK3800 evaluation boards
- Raspberry Pi

示例的源代码可以从 Wrox 网站上下载，网址如下：

www.wiley.com/go/profembeddedarmdev

约定

为了有助于你从书中获得最大收获并能跟踪正在发生什么，全书使用了一些约定。



注释：注释表明提示、小窍门、技巧和当前的旁白。

源代码

使用本书中的示例时，你既可以选择手动输入所有的代码也可以使用随书的源代码文件。本书使用的所有源代码可以在 www.wrox.com 上下载。针对本书，代码下载位于如下页面 **Download Code** 选项卡：

www.wrox.com/go/profembeddedarmdev

也可以通过 ISBN(本书的 ISBN 为 978-1-118-78894-3)在 www.wrox.com 和 <http://www.tupwk.com.cn/downpage> 上搜索本书来找到代码。并且当前所有 Wrox 图书完整的代码列表可以在 www.wrox.com/dynamic/books/download.aspx。在第 5 章的开头，你会发现该章的主要代码文件的列表。在这一章，也会发现引用的可供下载的代码文件的名称。

www.wrox.com 网页上的大多数代码被压缩成 ZIP、RAR 格式，或者适合开发平台类似的格式。

下载代码后，用你最喜欢的压缩工具解压缩。另外，你可以去 Wrox 代码主下载页面 www.wrox.com/dynamic/books/download.aspx 上去看本书和其他所有 Wrox 书适用的代码。

勘误表

我们竭尽全力确保本书在文字或代码中没有错误。然而，人无

完人，而错误时有发生。如果你在我们任一本书中发现错误，如拼写错误或者代码片段的错误，我们将会感激你的反馈。通过发送勘误表，你可能会节省其他读者数小时的困惑，与此同时，你也可以帮助我们提供更高质量的信息。

请到网址 www.wrox.com/go/profembeddedarmdev 找到本书的勘误表页面，然后单击勘误表链接。在这个页面你能看到由 Wrox 编辑为本书提出的所有勘误。如果在本书勘误表页面不能找到“你发现的”错误，登录 www.wrox.com/contact/techsupport.shtml，在该页面完成表单发送你所找到的错误给我们。我们会核对信息，如果合适，留言到勘误表页面并在随后的图书编辑中解决。

p2p.wrox.com

作者和同行可以登录 <http://p2p.wrox.com> 加入 P2P 论坛讨论。论坛是一个基于 Web 的系统，便于你发布关于 Wrox 图书和相关技术的消息并且与其他读者和技术用户交流。论坛提供了电子邮件订阅功能。Wrox 作者、编辑或其他行业专家还有你的读者也在这些论坛上。

在 <http://p2p.wrox.com> 上，不仅当你阅读本书时，而且在你开发自己的应用程序时，会发现许多不同的对你有帮助的论坛。要想加入论坛，只需按照如下步骤：

- (1) 访问 <http://p2p.wrox.com> 并单击 Register 链接。
- (2) 阅读使用条款并单击 Agree。
- (3) 完成所需要的信息填写，以及任何你要提供的可选信息并单击 Submit。
- (4) 你会收到一封电子邮件，邮件描述如何验证你的账户和完成加入过程。



注释：不用加入 P2P 论坛就可以在论坛中阅读帖子，但是要想发布自己的帖子必须要加入论坛。

加入后，可以发布新消息和回复其他用户发布的消息。你可以阅读网页上的任何时间的消息。如果你想从一个特定的论坛获取新的消息并发送电子邮件给你，在论坛名单中单击订阅中的本论坛图标。

关于如何使用 Wrox P2P 论坛获取更多信息，一定要读关于 P2P 论坛软件是如何工作的常见问题的解答，以及许多针对 P2P 和 Wrox 图书的常见问题。要想阅读 FAQ，请单击任何 P2P 页面的 FAQ 链接。

目 录

第 I 部分 ARM 系统和开发

第 1 章	ARM 发展史	3
1.1	ARM 的起源	4
1.1.1	Acorn 决定创建新处理器的理由	6
1.1.2	Acorn 变成 ARM 的原因	7
1.1.3	ARM 不实际生产微处理器的原因	8
1.2	ARM 的命名约定	10
1.2.1	如何辨别正在使用的处理器	11
1.2.2	ARM7TDMI 和 ARM926EJ-S 处理器的差异	13
1.2.3	ARM7 和 ARMv7 的差异	15
1.2.4	Cortex-M 和 Cortex-A 的差异	15
1.3	制造商文献资料	16
1.4	ARM 在今天都做些什么	16
1.5	本章小结	17
第 2 章	ARM 嵌入式系统	19
2.1	ARM 嵌入式系统的定义	22
2.1.1	芯片上的系统	23
2.1.2	嵌入式系统和系统程序设计之间的区别	24
2.1.3	优化的重要性	25
2.1.4	RISC 架构的优势	29
2.2	选择合适的处理器	32
2.3	如何着手	34
2.3.1	可用的电路板	35

2.3.2	现有操作系统	38
2.3.3	最适合我的目的的编译器	39
2.3.4	准备进行调试	40
2.3.5	是否有完整的开发环境	41
2.3.6	还需要知道什么	42
2.4	本章小结	43
第 3 章	ARM 架构	45
3.1	理解基础知识	46
3.1.1	寄存器	47
3.1.2	堆栈	48
3.1.3	内部 RAM	48
3.1.4	缓存	49
3.2	开始了解不同的 ARM 子系统	51
3.2.1	处理器寄存器介绍	51
3.2.2	CPSR 介绍	55
3.2.3	计算单元	57
3.2.4	流水线	58
3.2.5	紧耦合存储器	60
3.2.6	协处理器	61
3.3	理解不同的概念	62
3.3.1	异常的概念	62
3.3.2	处理不同的异常	65
3.3.3	操作模式	66
3.3.4	向量表	68
3.3.5	存储器管理	70
3.4	不同的技术	73
3.4.1	JTAG 调试(D)	73
3.4.2	增强的 DSP(E)	73
3.4.3	向量浮点(F)	74

3.4.4	EmbeddedICE(I)	75
3.4.5	Jazelle(J)	75
3.4.6	长乘法(M)	75
3.4.7	Thumb(T)	76
3.4.8	合成器(S)	76
3.4.9	TrustZone	77
3.4.10	NEON	78
3.4.11	big.LITTLE	78
3.5	本章小结	80
第4章	ARM 汇编语言	81
4.1	汇编语言介绍	82
4.2	与计算机对话	82
4.3	学习汇编语言的理由	84
4.3.1	速度	85
4.3.2	空间	87
4.3.3	趣味性	88
4.3.4	编译器并不完美	88
4.3.5	通过汇编语言理解计算机科学	89
4.3.6	使用汇编语言编写	89
4.4	使用汇编语言	90
4.4.1	编写启动加载程序	90
4.4.2	逆向工程	91
4.4.3	优化	92
4.5	ARM 汇编语言	93
4.5.1	指令格式	93
4.5.2	布局	94
4.5.3	条件代码	94
4.5.4	更新条件标志	98
4.5.5	寻址模式	101