



单片机与嵌入式系统丛书



<http://www.phei.com.cn>

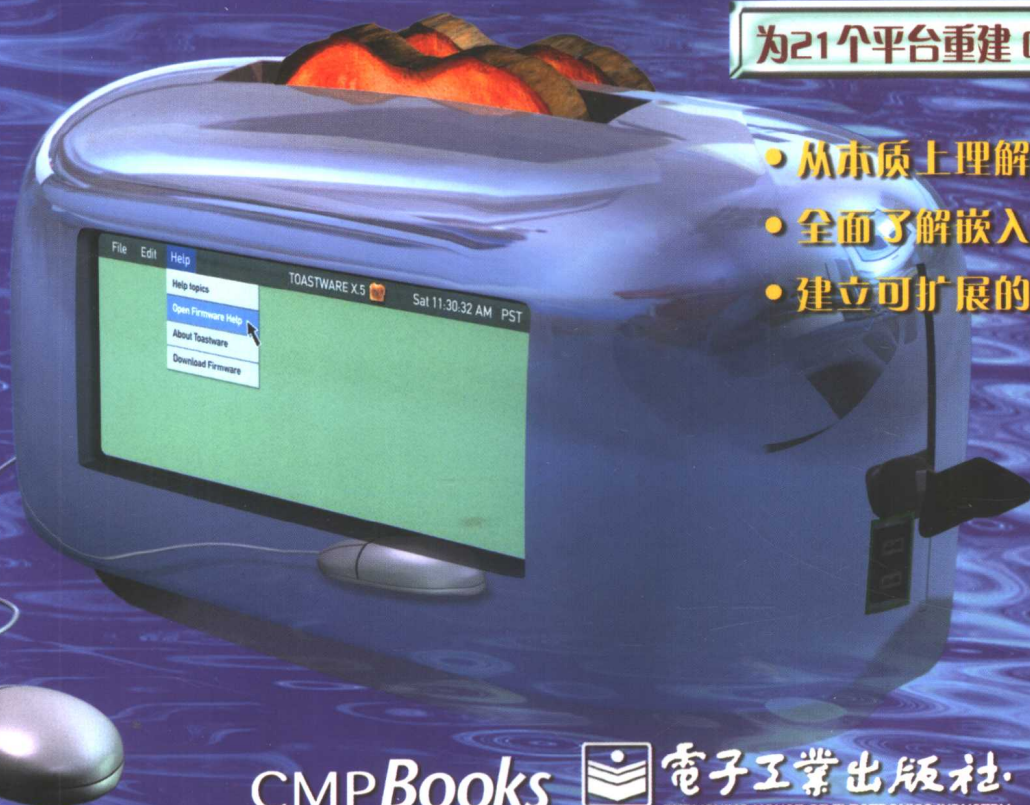
嵌入式系统固件揭秘

Embedded Systems Firmware Demystified

• [美] Ed Sutter 著 • 张晓林 等译

为21个平台重建 GNU X-Tools

- 从本质上理解硬件知识
- 全面了解嵌入式系统
- 建立可扩展的开发平台



CMPBooks



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



含光盘

单片机与嵌入式系统丛书

嵌入式系统固件揭秘

Embedded Systems Firmware Demystified

[美] Ed Sutter 著

张晓林 等译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书通过一个完整的嵌入式系统设计的全过程,向读者展示了嵌入式系统的基本框架及设计、编程、调试等技术细节。书中详细讲述了嵌入式系统中的存储器、微处理器与微控制器、数据总线与地址总线等基本概念,以及在设计中要考虑的要素。另外还给出了关键程序的源代码,使读者通过本书可学会如何看懂嵌入式系统的原理图,了解系统是如何工作的,掌握嵌入式系统开发平台的主要固件。

本书适于从事嵌入式系统研发的技术人员及高校相关专业的师生阅读。

Copyright©2002 by Lucent Technologies, except where noted otherwise. Published by CMP Books, CMP Media LLC. 1601 West 23rd Street, Suite 200, Lawrence, Kansas 66046, USA. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体专有翻译出版权由美国 CMP Books 授予电子工业出版社。该专有出版权受法律保护。

版权贸易合同登记号:图字:01-2002-2109

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式系统固件揭秘/(美)休特(Sutter,E.)著;张晓林等译. —北京:电子工业出版社,2003.6
(单片机与嵌入式系统丛书)

书名原文:Embedded Systems Firmware Demystified

ISBN 7-5053-8668-9

I. 嵌… II. ①休…②张… III. 微型计算机—系统设计 IV. TP36

中国版本图书馆 CIP 数据核字(2003)第 029515 号

责任编辑:富 军 赵丽松

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:21.75 字数:447.2 千字 附光盘 1 张

版 次:2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

印 数:5 000 册 定价:39.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077

译 者 序

本书是一本讲述嵌入式系统固件的畅销书，通过介绍一整套嵌入式系统的设计过程，使读者了解嵌入式系统的设计和开发方法。本书共分 13 章，第 1、2、3 章主要介绍了嵌入式系统硬件和软件的基础知识和微控制器开发平台；第 4、5 章讲述了所需的外围硬件和软件接口；第 6、7 章介绍了闪存文件系统；第 8 章介绍了命令的运行；第 9~13 章介绍了网络的连接和文件传输及实际应用中系统平台的调试过程。

本书以介绍嵌入式系统固件开发为主，列举并讲解了大量的程序实例，对设计中的有关技术问题做了相应说明。本书简明扼要，通俗易懂，可供高等学校有关专业学生及研究生、教师和从事嵌入式系统设计开发工作的技术人员阅读参考。

在本书的翻译、整理和校对过程中，路程、李蕾、谭征、徐彬、魏春风、陈飞虎、林琳、刘佳、张志轩等人做了大量的工作。由于译者水平有限，加之工作繁忙，译文中难免有不妥之处，恳切希望广大读者批评指正。

张晓林 等译

前言

嵌入式系统是嵌入到其他产品内部的计算机。奇怪的是，虽然人们知道许多关于编程和计算机的知识，但却在神秘的嵌入式系统世界中感到迷惑。在嵌入式系统编程世界中，大量的细节（包括硬件和软件）使开发过程像是在探险，甚至感到“难于”生存下来。软件、硬件、固件到底有何区别？怎样将 10 万条程序嵌入到比指甲还小的器件中？什么是闪存？为什么需要高速缓存（cache）？任务和进程之间有什么区别？要不要考虑可重入性？在阅读这本《嵌入式系统固件揭秘》的过程中，读者将逐步了解这些问题，不再像起初那样感到迷惑。

嵌入式系统编程覆盖了从最具体的底层编程到最抽象的高层 UNIX 编程的各个方面。它引起了近 20 年来工业界爆发的革命性变化。在 20 世纪 70 年代后期，汇编程序已被认为是够丰富多采了。一般的嵌入式系统可以使用小于 64Kb（位，不是字节）的系统内存。没有什么硬件要留给固件开发者去处理。一般由同一个人完成画图、焊接样品、写固件，把所有的事拉拢到一起。当 Intel 公司引入 8085 芯片时，很明显那些复杂的微处理器还停留在这个水平上。在 20 世纪 80 年代，Motorola 与 Intel 展开了 CPU 大战，C 语言成为了少数敢于用高级语言编程并烧制 EPROM 的人普遍采用的编程语言。今天，微处理器随处可见，范围从在工业界占主导地位的 4 位和 8 位微处理器家族到 1GHz 的几乎需要冷却设备（当然也是由微处理器控制的）冷却的 64 位微处理器。

多年来，这些系统的复杂度像滚雪球式的增大。工业界已经从在后台用二进制代码对 DEC PDP 机器编程发展到给烤面包机的微控制器提供面向对象的设计。系统的发展速度非常迅猛，已有的微处理器、微控制器、RAM、DRAM、SDRAM、管道、超标量体系结构、EPROM、闪存、RISC 和 CISC、RAS、CAS 及高速缓存等，也仅仅是一个开始。

现在，从牙刷（不是开玩笑，是事实）到喷气式战斗机都由某种微处理器控制着。这种趋势自然产生许多工具和技术。这些可供选择的硬件（固件必须使用的集成电路）和软件（用于建立固件应用程序的工具）使人眼花缭乱。

本书的目标是通过一个完整的嵌入式系统设计过程将你带入真正的嵌入式系统工程。而且，这个工程的源代码包括一些固件（一个嵌入式启动平台），可以简化你今后的工程。本书使用一个包括 CPU、内存及一些外部设备的小硬件系统设计实例，给出了基本原理结构图并讲解如何将指令从内存中取出，同时也将讲到设备

的概念。本书将讲述闪存与 EPROM、SRAM 与 DRAM、微处理器与微控制器、数据总线与地址总线，还将讲述如何将 C 语言和汇编语言源代码转换成二进制映像，并装入设备存储器中以便在 CPU 引导（引导闪存）时使用。

本书用几章的内容讲述启动嵌入式系统并执行一个应用程序（包括在汇编程序中的基本引导程序）的基本概念，先不涉及句柄、闪存驱动程序、闪存文件系统、串行和以太网连接等。这样可以使读者理解嵌入式系统工程如何启动、如何建立起支持嵌入式系统的平台。

这些听起来让人兴奋吗？美妙吗？恐慌吗？事实并非如此。本书的目的并不是要讨论最新出现的超量体系结构和在印刷电路板上的铜线所产生的天线辐射效果，也不想涉及高级的抽象设计过程（高级的体系结构和传输线的效果固然重要，但这些不是本书的主题）。本书是面向那些不想被一大堆工业行话和特殊技术细节所烦恼的、又想尽快了解系统的读者撰写的。读完本书，读者将学会如何看原理图、了解引导闪存器件是如何工作的、掌握完整的嵌入式系统开发平台的主要组件。

本书的读者对象

本书的读者应有一些 C 语言编程经历和基本的汇编语言概念。本书并不要求读者有电子技术或硬件知识背景。因此，只要读者有一些编程知识背景将会发现本书大有益处。

计算机科学或电子工程专业方面的学生不需要有固件开发的背景，只要有兴趣，就可以从本书中获益。

初级固件开发者将发现本书中的例子很有帮助，因为本书中的例子包括文档和代码注释，可以将其扩展到固件开发平台。书中将详细讲解引导新硬件和 CPU 与外部设备的交互方式。涉及到从小文件传输协议（Trivial File Transfer Protocol, TFTP）到以太网的底层引导过程。读者可以将本书中的代码或一部分代码引入到自己的固件平台中。

硬件开发者将发现本书讲述的平台有助于在复杂的 CPU 上分析并调试硬件，也会出于好奇，将注意力从硬件上转到学习固件过程上。本书提供了不脱离硬件而进入固件学习的起点（硬件设计者会自然地过渡到固件/软件的编程世界中）。

项目主管也会发现本书非常有用，因为这里呈现的固件包是一个成熟的平台。这个平台可广泛地应用到实时操作系统（RTOS）和目标体系结构中，而且很容易放到新系统中。这个平台是面向目标的，且独立于 RTOS，这使得它很容易转到各种目标或 RTOS 系统中。

本书的内容

第1章 艰难的开始 本章讲述 CPU 的基本结构及其支持的外设，本书讨论的固件将在这些外设上实现。本章将涉及核心处理器、RAM、闪存、串行 I/O 及 CPU 监控等与设计相关的内容，对每项设计要素都将详细讨论（如 CPU 监控、动态与非动态存储器等）。本章还将讲述微处理器如何从存储器中取得指令及高速缓存如何提高工作效率。本章包含了当今许多微处理器常用的外设，没有特指某个专用外设。

第2章 开始动手 本章介绍建立和编写嵌入式设备程序的方法，讲述本机编译与交叉编译环境之间的差异，将讨论文件格式和制作启动闪存的具体步骤，还将解释连接编译文件的重要性，讲述它如何将应用程序的代码部分恰当地装入目标存储器。本章还讨论一些调试方法，为今后进行固件编程打下基础。

第3章 微型监控器 本章介绍嵌入式系统中的启动平台和启动监控器。这里的监控器叫微型监控器。它有许多特性，是对学习固件开发过程极有帮助的工具。

第4章 所需的汇编语言 本章描述微型监控器的复位操作和启动代码，然后建立一些串行驱动程序并讨论如何建立异常处理程序。

第5章 命令行接口 本章描述如何建立在第4章中介绍过的核心功能的命令解释程序。本章从命令行接口（CLI）开始，解释如何分析输入的命令，然后调用相应的函数处理命令。命令行接口界面包括命令解释变量和符号、命令行编辑和历史记录、用户分级及密码保护。

第6章 闪存的接口 本书中共有3章讨论有关闪存的内容。本章是其中的第一章，将描述一个带有多个闪存的平台，讨论涉及到闪存编程的底层细节，并提供一个对带有不同闪存配置的目标系统非常有益的方法。

第7章 闪存文件系统 本章讨论建立驻留在闪存中的小文件系统（Tiny File System, TFS）的方法。这个简单的文件系统可以说明一些针对多闪存文件系统设计的更复杂的方法。本章还将讨论一些 TFS 设计方法的优缺点。

第8章 执行脚本 本章讨论用于命令解释的 CLI 和用于文件存储的 TFS 的特点，使文件像脚本或小程序一样执行。本章还解释将 CLI 和 TFS 相结合以创建一个简单的解释器的令人惊叹的方法细节。本章用 CLI 命令为 CLI/TFS 用户提供了一个简单有效的编程环境。条件分支和子程序只是本系统的一点小功能。

第9章 网络连通性 本章讨论接入网络的必备元素。与闪存正逐渐成为标准的引导存储器一样，以太网也正逐步成为许多嵌入式系统的标准接口。本章讨论几

个用于嵌入式系统的协议，包括 Ethernet、ARP、ICMP、IP、UDP 及 BOOTP/DHCP。最后，本章还利用几小段只知道本身网络包有效负载的程序代码来检查网络各层的封装。

第 10 章 文件/数据传输 本章讨论两个常用的文件传输协议：用于串行接口的 Xmodem 和用于网络的 TFTP，将在启动监视器中以如何使用这些协议为例展开讨论。本书中实现的 Xmodem 和 TFTP 可以在源和目标系统之间完成文件和存储器原始数据的传输。

第 11 章 添加应用程序 本章展示平台如何为应用程序提供基础。本章中的例子是一个非 RTOS 应用程序，但本章中的这个平台却支持各种环境。作为一个开发者，要判断在应用程序取代系统后，平台上还有多少功能可供应用程序使用。本章讨论应用程序如何驻留在 TFS 中，以及装入平台的程序如何自动将对文件从闪存空间传输到 RAM 空间来执行。

第 12 章 基于监视器的调试 本章讨论一些监视器的联机程序调整能力和说明实现这些能力的理由。调试程序对任何一个开发人员来说，都是必不可少的。本章将讲述如何建立基本的符号调试器，以便将内存像一个结构（表）一样显示出来（而不是一堆原始数据）、设置断点、单步执行、预定义程序的执行步骤、清空堆栈等。本章讨论的调试技术不需要片内支持和外部调试器。

第 13 章 将微型监控器接入 ColdFire MCF5272 本章介绍将微型监控器固件包接入 Motorola MCF5272（ColdFire）评估板的过程，详细讲述了本书所附 CD 上的源代码的目录结构，以及在接入的过程中配置监控器的细节。本章完成了以前章节中开始介绍的接入过程。

附录 A 建立基于主机的工具箱 本附录提供一些对与主机文件、串口等连接非常有帮助的代码段。尽管本书是关于固件方面的事实，但也离不开某些最终的本地编程。即使读者可能已经做过许多本地编程，但本附录中提供的代码会加快你的工作速度。

附录 B RTOS 概述 本附录介绍一些实时操作系统（RTOS）的基本概念，使读者可以从单线程的编程开发环境扩展到多线程环境。在这方面有专门的图书，所以本附录不深入讲解，而是提供一些基础知识，使读者理解多任务能做什么，了解优先控制权为什么是既优秀又带有危险性的特点。

源代码

本书包含大量源代码，在叙述过程中，当谈到各种程序时，使用了其中部分或

全部功能。本书中的全部源代码都已转换成 C 源代码，并放在本书所附的 CD 中。这些源代码中有许多部分不是本书讨论的重点，所以在书中没有全部列出来，以使本书叙述更清晰。CD 中提供的源代码是最终工作采用的版本。如果书中所列的程序不完整，请参考 CD 中的代码。

致谢

首先感谢我在朗讯工作的好友和领导 Roger Levy、Paul Wilford，感谢他们的支持与鼓励。同时也感谢我的好友 Patricia Dunkin 和小 Agesino Primatic。感谢他们帮我审定稿件并提出各种好建议。

感谢本书的出版公司 CMP，感谢 Joe Casad、Catherine Janzen 和 Robert Ward 所做的编辑和技术支持（特别感谢 Robert 在本书的全部过程解答我众多的问题，并及时纠正我的偏差）。感谢 Michelle O'Neal、Justin Fulmer、James Hoyt 和 Madeleine Reardon Dimond 为本书排版、图表及索引所做的工作。

最后，感谢我的爱妻 Lucy，是她始终不渝的鼓励才使我写成了本书。还要感谢我的爱子 Tommy，他始终将我的写作放在首位。最要感谢的还是我们的救世主——耶稣基督，是他无私的爱给了我永恒的力量。没有任何力量可以与其媲美！

目 录

第 1 章 艰难的开始	1
1.1 系统要求	1
1.2 中央处理器	2
1.2.1 可编程芯片的选择	3
1.2.2 中断控制器	4
1.2.3 定时-计数单元	4
1.2.4 DMA 控制器	5
1.2.5 串口	6
1.2.6 DRAM 控制单元	6
1.2.7 内存管理单元 (MMU)	6
1.2.8 缓存	7
1.2.9 可编程 I/O 管脚	7
1.2.10 把所有部件集成起来	8
1.3 系统存储器	8
1.3.1 ROM, PROM, EPROM 和 EEPROM	9
1.3.2 RAM	9
1.3.3 闪存 (Flash Memory)	9
1.3.4 其他	10
1.4 CPU 监控	10
1.4.1 复位	11
1.4.2 看门狗定时器 (watchdog timer)	12
1.4.3 带备份电源的 SRAM	12
1.4.4 每日时钟	13
1.5 串口驱动器	13
1.5.1 直接电缆数据传输	13
1.5.2 差分驱动传输	14
1.6 以太网接口	15
1.7 闪存设备的选择	15
1.7.1 闪存锁定工具	15

1.7.2 底部引导 (Bottom-Boot) 和顶部引导 (Top-Boot) 闪存设备	16
1.8 CPU/存储器接口	16
1.8.1 CPU	18
1.8.2 缓存的功能及缺陷	20
1.9 小结	23
第 2 章 开始动手	24
2.1 在 PC 上的实现	24
2.1.1 交互编译过程	26
2.1.2 建立内存映射	30
2.1.3 连接编译文件	30
2.1.4 文本、数据和 BSS	32
2.1.5 Make 文件	34
2.2 建立库	37
2.3 准备活动	38
2.3.1 开始硬件设计	40
2.3.2 认识硬件并善待设计师	41
2.3.3 拥有所有数据的本地备份	41
2.3.4 确信硬件可以工作	42
2.3.5 慢慢开始	42
2.3.6 查看你的生成文件	43
2.4 运行时间	45
2.5 为固件开发进行全面的硬件测试	45
2.5.1 确定电源电压	46
2.5.2 验证时钟的有效性	46
2.5.3 检查启动芯片的片选和读信号	46
2.5.4 取出放大镜	46
2.5.5 小心静电	46
2.5.6 复位时简单的循环	47
2.5.7 一个 LED 的重要作用	47
2.5.8 RAM 和“不需要堆栈的”串行输出	47
2.5.9 开始 C 语言层次	48
2.6 小结	50

第3章 微型监控器	51
3.1 一个嵌入式系统启动平台	51
3.1.1 常驻系统命令集	52
3.1.2 给应用程序提供的 API	54
3.1.3 基于主机的命令集	55
3.2 小结	57
第4章 所需的汇编语言	58
4.1 复位之后	58
4.2 I/O 初始化	64
4.3 建立异常处理	65
4.3.1 ROM 中的异常处理	66
4.3.2 RAM 中的异常处理	72
4.3.3 当心寄存器	74
4.4 小结	74
第5章 命令行接口	75
5.1 命令行接口的特点	75
5.2 命令行接口的数据结构和命令列表	76
5.3 命令行接口处理	77
5.4 命令名下的函数	77
5.5 内部变量和符号处理	82
5.6 命令行重新定向	85
5.7 命令行编辑和记录	91
5.8 用户分级	97
5.9 密码保护	100
5.10 小结	102
第6章 闪存的接口	103
6.1 接口函数	103
6.1.1 闪存库	104
6.1.2 在 RAM 中重新部署闪存操作函数	105
6.1.3 闪存控制结构初始化	107
6.1.4 29F040 系列的闪存操作系统	108

6.1.5 对 16 位与 32 位的扩展 (banks) 操作	116
6.2 闪存驱动的前端 (Front End)	117
6.3 小结	121
第 7 章 闪存文件系统	122
7.1 TFS 在平台上的作用	122
7.2 TFS 的设计标准	123
7.3 文件属性	123
7.3.1 可以自动加载的文件	124
7.3.2 用户级别	125
7.4 高级的详细内容	125
7.5 TFS 所要求的闪存空间	127
7.6 碎片整理	128
7.6.1 简单但是存在潜在危险的方法	129
7.6.2 比较复杂但功能更为强大的方法	129
7.7 TFS 的应用	130
7.7.1 带有安全电源的碎片整理操作	130
7.7.2 没有安全电源的 tfsclean () 函数	135
7.8 增加和删除文件	138
7.8.1 tfsadd () 函数	138
7.8.2 tfsunlink () 函数	147
7.9 加载的应用	148
7.10 文件解压缩	153
7.11 现场执行	154
7.12 小结	155
第 8 章 执行脚本	156
8.1 脚本运行器	156
8.1.1 Exit	159
8.1.2 Goto	160
8.1.3 gosub 和 return	161
8.2 条件转向	163
8.3 一些例子	169
8.3.1 例子#1: ping	169

8.3.2 例子#2: 外壳数组	170
8.3.3 例子#3: 子程序、条件转向、TFS 及其他	172
8.4 小结	174
第 9 章 网络连通性	175
9.1 以太网	175
9.2 ARP	176
9.3 IP	177
9.4 ICMP	177
9.5 UDP 和 TCP	179
9.6 DHCP/BOOTP	180
9.7 嵌入式系统的应用	181
9.7.1 processPACKET () 函数	183
9.7.2 总结	189
9.8 小结	190
第 10 章 文件/数据传输	192
10.1 Xmodem	192
10.1.1 Xdown ()	196
10.1.2 MicroMonitor 中的 Xmodem	199
10.2 TFTP	199
10.3 自升级功能	211
10.3.1 应用程序不知道潜在升级路径	211
10.3.2 应用程序是自升级的一部分	212
10.4 小结	212
第 11 章 添加应用程序	213
11.1 各种存储映像	213
11.2 弱启动	214
11.3 建立应用程序堆栈	214
11.4 连接到监控器的 API	214
11.4.1 Moncom () 函数	215
11.4.2 MonConnect () 函数	216
11.5 应用程序 start () 函数	219

11.6	应用程序 main () 函数	220
11.7	为应用程序创建的驱动程序	221
11.8	基于应用程序的 CLI 使用监控器 CLI	221
11.9	通过应用程序 CLI 运行脚本	223
11.10	小结	224
第 12 章	基于监控器的调试	225
12.1	不同类型的调试方法	226
12.2	断点	226
12.2.1	使用断点进行代码分析	228
12.2.2	一些 CPU 提供了调试吊钩	230
12.3	增加符号能力	230
12.4	显示存储器	231
12.5	将 C 结构覆盖到内存	238
12.5.1	一些示例输出	252
12.6	堆栈跟踪	254
12.7	检测堆栈溢出	260
12.7.1	预填充堆栈内存或者缓冲区	262
12.7.2	利用对每个函数堆栈段的检查	263
12.8	系统评测	265
12.8.1	使用系统节拍 (tick)	265
12.8.2	基本模块	274
12.9	小结	275
第 13 章	将微型监控器接入 ColdFire MCF5272	276
13.1	原始资料代码目录树	277
13.2	编译文件	278
13.3	头文件的结构	287
13.3.1	FORCE_BSS_INIT	287
13.3.2	PLATFORM_XXX	288
13.3.3	闪存结构	288
13.3.4	TFS 结构	289
13.3.5	INCLUDE 列表	290
13.3.6	多样化配置	292

13.4 连接步骤	293
13.4.1 下载第一个镜像	293
13.4.2 启动闪存驱动器	297
13.4.3 启动 TFS	303
13.4.4 启动以太网	305
13.5 小结	306
结束语	307
附录 A 建立基于主机的工具箱	308
A.1 与主机文件连接	308
A.2 与计算机串口的接口	312
A.3 基于 PC 的 UDP 处理: moncmd	315
A.4 小结	318
附录 B RTOS 概述	320
B.1 调度程序	321
B.2 任务、线程和过程	321
B.3 抢占、时间分割和中断	322
B.4 信号机、事件、消息和定时器	323
B.5 重入	326
B.6 好的并行和差的并行	327
B.7 小结	328
本书所附光盘 (CD) 的内容	329

第 1 章 艰难的开始

尽管本书关注的重点主要是固件开发，但是一个好的固件开发者必须对其所依赖的硬件有很好的了解。人们可以将固件编程者与加油站换油的工作人员相比：如果他所知道的只是换油，那他就无法发现各种事故隐患，而一个训练有素的技术工人会马上发现这些迹象。受训不久的技术工人也许可以胜任换油的工作，但却不会注意诸如盖垫密封片泄漏之类的问题。如果这些问题一直不被发现，则某些部件迟早会出问题，并且可能会由于技术工人的知识有限而产生额外的损失。

本章阐述在一个典型系统里，固件工程师需要了解关于硬件方面的知识。这样做的目的并不是把你变成硬件设计者，而是为了使你能做比“换油”更多的事情。为了达到这个目的，本章讨论一些常见的 CPU 支持的外围设备并讨论处理器如何完成工作。从讨论本书涉及的系统开始，你将了解现在基于微处理器系统的一些最常见的特点，同时还将讨论 CPU 获得指令的步骤及使用缓存所引入的好处和缺陷。

至本章的最后（尽管你还并不是一名合格的硬件设计师），你会对固件的硬件平台有更好的理解。

1.1 系统要求

嵌入式系统所需要的硬件因其用途不同而有显著差异。有些微型系统只有不到 1KB 数据空间和 16KB 地址空间，而高性能系统可能运行于 1GHz 64 位处理引擎下，拥有 32MB 快速闪存和 128MB DRAM。本书将关注在这两者之间的系统性能，所讨论的固件的存储空间（32~256KB 的闪存，8~128KB 的 RAM）视所要实现的功能而定，所以这种系统不适合于小的微处理器（8051，68HC05/11/12 等等）。另一方面，这些代码不需要大型机器就能成功运行。

图 1.1 显示了一个完整的系统模型框图。利用这个系统模型，你可以对自己的器件进行编程、与 PC 会话，甚至给网络浏览器提供 HTML 页面服务。如果想节约，则可以去掉复位/看门狗（Watchdog）及有备用电源的 RAM（BBRAM）/每日时间控制器（TOD）。在模型中包括的这些功能是因为它们极其有用，同时它们在大多数的微处理器设计中也成为标准的配备。