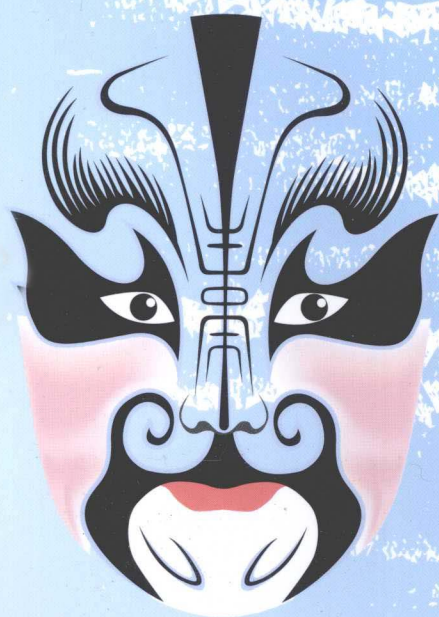


工程师经验手记

构建嵌入式Linux

核心软件系统实战

杨铸 李奎 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



工程师经验手记

构建嵌入式 Linux 核心 软件系统实战

杨 铸 李 奎 编著

北京航空航天大学出版社

内 容 简 介

本书以实战的方式,讲解了构建嵌入式 Linux 核心软件系统的五大组件:交叉编译工具链、BootLoader、Linux Kernel、根文件系统、图形界面系统。其中包含了大量解决实际工作中常遇到的典型问题的方法、技巧和经验。

本书适合大学本、专科学生,培训机构学生,自学人员以及研究生学习嵌入式 Linux 及图形界面软件系统的移植和开发;同时,从事该方向的软件开发工程师也可将本书作为案头的技术手册来进行查阅和参考。

图书在版编目(CIP)数据

构建嵌入式 Linux 核心软件系统实战 / 杨铸,李奎编著. —
北京:北京航空航天大学出版社,2013.4
ISBN 978-7-5124-1084-8

I. ①构… II. ①杨…②李… III. ①Linux 操作系统—程序设计 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2013)第 042236 号

版权所有,侵权必究。

构建嵌入式 Linux 核心软件系统实战

杨 铸 李 奎 编著
责任编辑 苗长江 王 彤

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:23.25 字数:496 千字

2013 年 4 月第 1 版 2013 年 4 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-1084-8 定价:49.00 元(含光盘 1 张)

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

创作动机

很多初学者常问我：

- (1) 学习嵌入式软件系统移植、开发，难吗？
- (2) 学习嵌入式 Linux 系统软件移植、开发，有前途吗？
- (3) 能让我用较短的时间一窥嵌入式 Linux 核心软件系统的全貌吗？

不少刚从事嵌入式 Linux 软件系统移植和开发工作的朋友常问我：

- (1) 能推荐一本全面介绍构建嵌入式 Linux 及图形界面软件系统的书吗？
- (2) 在嵌入式 Linux 及图形界面软件系统的移植和开发实际工作中，常遇到一些棘手的问题，不知如何下手解决。能告诉我解决这些常见问题的方法、技巧和经验吗？

本书将站在一个草根工程师的角度，以自身的实践经历，试着回答这些问题。期望能为嵌入式软件开发在中国的发展勉尽薄力。

本书内容及组织方式

第 1 章

“知其然，知其所以然”。很多学习嵌入式的读者都是为了学习而学习，有些时候对所学的知识缺乏使用率和实用性的认知。

嵌入式是一门涉及软硬件知识的整体学科，如何学好嵌入式是很多嵌入式爱好者所关心的问题。学习嵌入式学科有三个问题是我们所关注的：一是学习的成本；二是学习的效率；三是它的发展前景。本章在硬件平台上选择了 S3C2440，软件选择了 Linux 系统。读者会发现在互联网时代的今天，透过网络搜索这些资源是多么容易和有趣的事情，因此跨入嵌入式大门不再是难事。互联网为我们学习嵌入式降低了成本并提高了效率（很多资源都可以通过网络获取）。

如果你还在担心学习的知识是否有前景，那么建议你浏览第 1 章节的内容，相信这一次面对面的“交谈”会使得你对基于 ARM 的 Linux 开发更感兴趣。

第 2 章

“工欲善其事，必先利其器”。本章节为读者在接下来的学习提供了搭建嵌入式环境的方法。其中包括虚拟机下安装 Linux 系统的方法，串口调试开发板工具 mini-com 和 SecureCRT 的使用，NFS 远程文件系统及 FTP 文件传输服务。

本书中使用 Linux 系统桌面版 ubuntu 作为宿主机，相中它的原因是它的简单和

专一。需要使用到任何软件,只需要一个 apt-get 命令便可以得到。如果读者想要看到华丽的界面效果,相信直接在电脑上安装一个 ubuntu 桌面版是最好的选择。

学习硬件需要有调试终端,嵌入式硬件没有友好的终端屏幕可供调试,因此常用的调试终端为串口,并需要借助宿主机中的屏幕来显示开发板中的内容。读者可以选择 Linux 中的串口通信软件 minicom 或是 Windows 下的串口通信软件 SecureCRT 来“倾听”开发板的“吐槽”,作为一个嵌入式学习者,往往都是由此而成长起来的。

NFS 远程文件系统非常“亲民”,因为我们的开发分为两步,一是在 PC 机上做好开发板上需要的资源,二是部署资源到开发板。NFS 可以让我们不用做第二步,直接通过 NFS 服务把 PC 机上做好的资源挂载到开发板上来进行调试,当确定没有问题的时候,再部署资源到开发板。这样做即省时,也避免频繁烧写对 Nand Flash 等存储设备造成不必要的寿命损耗。

我们为读者制作了一个学习的光盘,非常重要。里面有本书学习所使用的所有资源和源代码。读者可以通过 FTP(文件传输协议)软件,把学习中使用到的资源或源代码传输到你的 Linux 系统中直接使用,简化学习过程,提高学习效率。

第 3 章

“不在乎沿途风景,只在乎最终结果”。读者应该很想知道这本书到底讲了哪些东西,或是想绕开中间繁杂的环节,直接查看谜底。

本章节将是以下所有章节的“知识结晶”。读者按照所提供的步骤,便可以享受到整个嵌入式的构建过程之旅。而在整个构建的“旅途”中,所涉及的“地方”是第 4 章的交叉编译工具链的制作,第 5 章的引导程序的制作,第 6 章的 Linux 内核系统的构建,第 7 章的根文件系统的构建,第 8 章的 Qt 图形界面的构建。作为终点站的第 8 章,如果你已经能够看到开发板上运行的图形界面,那么恭喜你享受了一个完美的旅行过程。

若是读者对某一个部分比较感兴趣可以停下来,到下面所列举的章节中,做一次旅游体验师吧,因为只有自己真正体会了,这些知识才真正属于你。

第 4 章

翻译官——编译器的制作。

作为程序员与计算机来说,我们需要制作一个工具,这个工具能够把我们用 C 语言等写的源代码转化为能让 CPU 识别的语言(机器指令)。本章所需要掌握的重点是在配置制作编译器的过程中了解其中参数的含义,对于以后编译各类嵌入式设备所需要的交叉编译工具链以及编译过程中需要指定的选项能够心知肚明。

本章共讲解了 3 种编译器制作的方法:crosstool、源码、crosstool-ng。第一种制作出的交叉编译器也是我们本书中通篇使用的交叉编译器;源码方式制作的交叉编译器是为好奇的读者想了解交叉编译器制作的每个环节而设计的,是基于源代码的方式自己动手写交叉编译脚本的整个过程;crosstool-ng 和 crosstool 类似,也是通过开源的脚本来制作交叉编译器,但不同的是它支持更新的 CPU 和内核版本,并且在制作之前,可以使用图形界面进行配置选择其中的功能。

第5章

扬帆远航——引导程序的制作。

本书的重点为第5章至第8章。作为本书学习的起点,引导程序的制作以及由此而获得的知识是非常丰富多彩的。由于引导程序涉及对硬件的操作,也有软件设计的部分,所以对嵌入式不熟悉的读者更需要认真地来对待这一“扬帆”的时刻。

本章节分为两个部分:一是和书的构建流程相关的 uboot,在 5.1~5.3 节;一是 5.4 节简化版本的引导程序制作过程。uboot 部分本章节只介绍和构建流程相关的知识内容,而 uboot 作为嵌入式知识的精华,读者有必要去深入认识和理解,通过互联网可以找得到太多它的故事。而 5.4 节简化版本的引导程序为 iBoot(iotaBoot-loader 的简称),它非常小,目的也非常明确,只为引导它所“期待”的内核。因此读者可以在做完 uboot 之后,把它当作一个饭后的“甜点”来消理解,相信读者对引导程序及嵌入式开发会有更加深刻的认识。

第6章

舵手——Linux 内核的构建。

本章是本书的重点和难点。重点在于它的启动完成预示着所有的软硬件资源都由其管理,掌握“生杀大权”。难点在于 Linux 内核代码非常庞大,读者需要掌握学习内核的方法和进行知识储备。

本章节以构建流程为流水线,逐步讲解构建开发板上的资源的过程。本章节并没有对涉及的内核框架和驱动做详细说明,而是以穿针引线的方式针对性地讲解开发板上资源的构建过程。哪些是需要改动的,为什么改动,使得读者根据“舵手”的指引,完成本章对构建一个完整 Linux 系统的理解。

这一部分中的驱动移植的内容,如需彻底理解,可能需要读者具备驱动开发相关的知识。由于本书并不打算讲解 Linux 下的驱动开发,因此如果读者想彻底了解相关内容的话,可以自行学习 Linux 下驱动开发的相关书籍,IFL 嵌入式小组编著的《深入浅出嵌入式底层软件开发》是个不错的入口。

第7章

展翅翱翔的基石——根文件系统构建。

根文件系统总是在等待,等待其上能够出现的有趣应用。或者说强大的应用程序需要建立在一个稳定的基石上。本章节将展示作为基石的根文件系统的整个构建过程是怎样的。7.1 和 7.2 节分别对目前的文件系统做了一个简单介绍。7.3 节作为本章的重点,读者需要仔细地按照步骤来完成根文件系统的构建过程。7.4 节是根文件系统的一点小小的“追求”。我们移植了一些应用在它上面,以展示根文件系统已经移植成功,并且可以安装应用了。

第8章

梦想的实现——Qt 图形界面系统构建。

本章节将完成整个系统构建过程的最后一步,也是本次旅途的终点站。

作为一个美好的回忆,我们将会在此章节完成图形界面菜单在根文件系统上的

构建过程,读者也可以基于这一系统来制作相关的图形界面应用程序了。8.1 节是对 Qt 的简介。8.2 节是本章的重点,介绍整个 Qt 图形界面程序构建的过程和部署。8.3 节之后是 Qt 应用程序的开发过程和调试部分。Qt 已经提供了众多的案例代码供学者学习,读者可以关注 8.4 节 gdbserver 的代码调试,很多嵌入式的调试环境也都是基于它的。

感 谢

本书由 IFL 嵌入式小组的杨铸、李奎负责编写。感谢 IFL 嵌入式小组的唐攀、孙夏玉不辞辛劳地提供了本书所用到的部分素材。正是整个 IFL 嵌入式小组的鼎力相助,才使得本书得以顺利完成。

感谢我的父母,是你们从小对我朴实无华的谆谆教导,在我心灵的深处种下了要勤奋学习、要努力工作、要懂得感恩的火种,你们给了我强大的精神鼓励和支持,使得本书得以顺利完成。对了,还有我可爱的小外甥——张和洋,希望你能健康成长,少玩电脑,少打游戏。

感谢北京航空航天大学出版社胡晓柏主任对本书的支持和关怀,正是他耐心的鼓励和支持,才使得本书在最短的时间内与读者见面。

感谢安博教育的成宝宗、柳斌、何桂忠、张亮、马林、马锋、王柱、刘志强、修宸、杨建光、马伯骊、丁晓峰、关东升、何凯霖、胡德昆、黄曼绮、江军、刘鹏、罗福强、罗佳、杨剑、杨菊英、张云和对本书的写作和出版提供的帮助。

感谢瞿庆松、罗鹏、张稷勇、刘升、党旭、李欢对本书的认真校对,并提出的修改意见。

感谢广州友善之臂科技有限公司以及北京铁匠铺科技有限公司提供的开发板,他们出品的开发板和相关资料质量很高,使得本书的写作有了一个很好的硬件平台。

最后,要感谢伟大的互联网,本书不少的素材都来源于互联网。正是由于互联网的普及,使得各种技术资料的获得和技术知识的交流变得非常容易,从而使得本书的创作事半功倍。

另外,由于版权问题,书中涉及的部分软件和资料不能收录在随书光盘中,请读者上网搜索自行下载,也可访问作者博客:<http://user.qzone.qq.com/308337370>,
<http://blog.csdn.net/ielife>。作者提供相关内容的下载链接。

限于作者水平有限,书中难免有遗漏和不足之处,恳请广大读者批评指正。联系方式是 E-mail,杨铸:scyz@263.net,李奎:eabi010@gmail.com;并开通了 QQ 技术讨论群:47753328。

作者 2013 年于
北京昌平
成都少城公园
重庆西永微电园

目 录

第 1 章 嵌入式 Linux 系统开发综述	1
1.1 嵌入式 Linux 系统的发展现状	1
1.2 基于 ARM 处理器的 Linux 系统开发过程综述	3
1.3 选择 mini2440 与 tq2440 开发平台的理由	5
第 2 章 嵌入式 Linux 开发环境的搭建	7
2.1 嵌入式系统环境搭建	7
2.1.1 为什么选择 Windows+vm 虚拟机+ubuntu 的开发方式	7
2.1.2 主机环境安装	8
2.2 开发工具使用说明	21
2.2.1 超级终端——minicom 的使用	21
2.2.2 超级终端——SecureCRT 的使用	22
2.2.3 设置虚拟机与主机通信	24
2.2.4 NFS 远程文件系统	25
2.2.5 FTP 服务器软件配置	26
2.3 光盘目录结构说明	27
第 3 章 体验嵌入式 Linux 系统之旅	29
3.1 烧写引导程序到开发板	29
3.1.1 使用 JTAG 接口烧写引导程序到开发板	29
3.1.2 使用 JLINK 接口烧写引导程序到开发板	34
3.1.3 测试引导程序是否正常	38
3.2 安装 USB 驱动与 dnw 软件	38
3.2.1 USB 驱动安装	38
3.2.2 开发板 USB Slave 驱动安装	39
3.3 使用 dnw 软件下载内核镜像到开发板	39
3.4 使用 dnw 软件下载根文件系统到开发板	40
3.5 运行测试	40
3.5.1 系统执行流程分析	41
3.5.2 测试图形界面程序	43
3.6 小 结	43

第 4 章 制作交叉编译工具链	44
4.1 交叉编译器的组成结构	44
4.1.1 arm-linux-gcc	44
4.1.2 glibc	45
4.1.3 binutils	45
4.2 基于 crosstool 制作交叉编译工具链	46
4.2.1 获取 crosstool 源码和补丁文件	46
4.2.2 获取 crosstool 脚本需要的源代码文件	47
4.2.3 crosstool 脚本代码修改	47
4.2.4 编译安装及测试	48
4.3 源代码编译的制作方式	49
4.3.1 获取源码及工具	49
4.3.2 配置工作环境	50
4.3.3 配置编译安装 binutils	51
4.3.4 配置编译安装无 abi 支持的 GCC	51
4.3.5 编译 glibc 所需要的内核头文件	55
4.3.6 配置编译安装 glibc	55
4.3.7 配置编译安装完整的 GCC	57
4.3.8 测试编译成功的交叉编译工具链	59
4.3.9 使用新选项 sysroot 对交叉编译工具链进行完美化	60
4.4 基于 crosstool-ng 工具制作交叉编译工具链	61
4.4.1 获取 crosstool-ng 的源代码	62
4.4.2 获取 crosstool-ng 脚本需要的源代码文件	62
4.4.3 准备 crosstool-ng 的安装环境	62
4.4.4 安装 crosstool-ng	63
4.4.5 配置编译的交叉编译工具链参数	63
4.4.6 编译交叉工具链	70
4.4.7 测试编译成功的交叉编译工具链	71
4.5 小 结	72
第 5 章 构建 BootLoader	73
5.1 BootLoader 介绍	73
5.1.1 BootLoader 概述	73
5.1.2 BootLoader 的分类	77
5.2 多平台引导程序——U-Boot	78
5.2.1 U-Boot 简述	78
5.2.2 U-Boot 的功能特性	79

5.2.3 U-Boot 目录结构	79
5.3 U-Boot 的移植过程	80
5.3.1 安装和使用源代码阅读工具 Source Insight	80
5.3.2 U-Boot 的编译初步	82
5.3.3 分析 U-Boot 的第一阶段代码(cpu/arm920t/start.S)	83
5.3.4 分析 U-Boot 的第二阶段代码	88
5.3.5 继续移植、编译 U-Boot	90
5.3.6 U-Boot 常用命令使用简介	94
5.3.7 U-Boot 命令实现框架的分析	98
5.3.8 U-Boot 引导 Linux 操作系统的过程分析	104
5.3.9 让 U-Boot 支持从 USB slave 接口获得数据	110
5.3.10 让 U-Boot 支持读写 Yaffs 文件系统	110
5.3.11 增加 mtd 设备层支持	112
5.3.12 光盘中的补丁使用	113
5.4 实战:制作小型的能够快速引导内核的 iBoot	113
5.4.1 iBoot 简介	113
5.4.2 iBoot 源码目录结构及说明	114
5.4.3 iBoot 代码解释	114
5.5 小 结	138
第 6 章 构建嵌入式 Linux 内核	139
6.1 Linux 内核简介	139
6.2 Linux 内核版本历史	140
6.3 Linux 内核源代码目录结构	141
6.4 Linux 编译运行体验	142
6.5 Kbuild——Linux 内核构造框架	144
6.5.1 内核 make 流程	144
6.5.2 Kbuild 简介	145
6.5.3 make %config 的实现过程	145
6.5.4 make menuconfig 配置解析	147
6.5.5 Kbuild 机制实现原理	148
6.5.6 Kconfig 语法	149
6.5.7 实战:添加 DM9000 网卡驱动	153
6.5.8 zImage 文件组成结构	154
6.5.9 uImage 和 zImage 的关系	156
6.5.10 zImage 在内存中的布局	157
6.5.11 内核的真实执行过程	161

6.5.12	在移植中需要为开发板做哪些改动	161
6.6	创建目标平台——my2440	162
6.6.1	基于三星 SMDK2440 创建目标平台	162
6.6.2	时钟源频率的更改	165
6.6.3	机器码的修改	165
6.6.4	运行测试	166
6.7	Nand Flash 驱动的移植与分区更改	169
6.7.1	内核如何管理 Nand Flash	169
6.7.2	更改 Nand Flash 的分区结构	171
6.7.3	配置内核支持 Nand Flash	172
6.7.4	测试 Nand Flash 分区信息	172
6.8	yaffs2 文件系统移植	173
6.8.1	yaffs2 文件系统说明	173
6.8.2	获得 yaffs2 文件系统的内核补丁	177
6.8.3	配置内核支持 yaffs2 文件系统	177
6.8.4	测试 yaffs2 文件系统	180
6.9	网卡设备的移植——DM9000	181
6.9.1	内核中网卡的移植方法	181
6.9.2	DM9000 网卡芯片特性	182
6.9.3	DM9000 网卡移植过程	182
6.9.4	配置内核支持 DM9000 网卡驱动	190
6.9.5	测试 DM9000 网卡设备工作状态	191
6.10	显示设备 LCD 的移植	192
6.10.1	显示屏简介	192
6.10.2	2440LCD 控制器	192
6.10.3	内核中的 frame buffer 显示框架	197
6.10.4	设置 LCD 在内核中的硬件资源	200
6.10.5	增加各种 LCD 设备类型的支持	205
6.10.6	配置内核支持 LCD 平台设备驱动	208
6.10.7	测试 LCD 在开发板上的运行情况	209
6.11	修改 Linux 内核的 Logo 信息	209
6.11.1	Linux Logo 显示流程	209
6.11.2	使用工具制作 Logo	210
6.11.3	去除屏幕上显示的光标	211
6.12	UDA1341 音频设备移植	212
6.12.1	数字音频处理简介	212

6.12.2	Linux 音频驱动框架	212
6.12.3	UDA1341 与 S3C2440 硬件接口说明	213
6.12.4	S3C2410-UDA1341 驱动主要结构	215
6.12.5	移植 UDA1341 驱动	217
6.12.6	UDA1341 音频测试	219
6.13	SD 卡设备移植	220
6.13.1	SD 卡简介	220
6.13.2	MMC/SD 卡 SDIO 接口与 S3C2440 硬件接口	220
6.13.3	Linux 内核 MMC/SD 驱动程序框架	222
6.13.4	移植 MMC/SD 卡驱动	222
6.13.5	测试 SD 卡	224
6.14	触摸屏设备驱动移植	225
6.14.1	触摸屏硬件接口说明	225
6.14.2	内核 input 子系统	226
6.14.3	配置内核支持触摸屏设备	227
6.14.4	测试触摸屏	232
6.15	LED 设备移植	233
6.15.1	LED 硬件接口	233
6.15.2	Linux 的杂项(misc)设备	233
6.15.3	移植 LED 设备驱动	238
6.15.4	配置内核支持 LED 设备	242
6.15.5	测试 LED 设备	242
6.16	用户按键设备移植	242
6.16.1	按键的硬件接口	242
6.16.2	移植按键设备驱动	244
6.16.3	配置内核支持 my2440 按键	248
6.16.4	测试按键	249
6.17	看门狗设备移植	249
6.17.1	看门狗工作原理	249
6.17.2	配置内核支持看门狗设备	250
6.17.3	测试看门狗	251
6.18	内核中其余部分的移植步骤	252
6.18.1	PWM 蜂鸣器移植	252
6.18.2	RTC 实时时钟移植	253
6.18.3	USB 设备移植	254
6.18.4	其他必选项	256

6.19	利用光盘补丁制作内核镜像	257
6.20	小 结	257
第 7 章	构建嵌入式 Linux 文件系统	259
7.1	嵌入式 Linux 文件系统简介	259
7.1.1	嵌入式文件系统概述	259
7.1.2	MTD 设备与 Flash 文件系统简介	259
7.2	嵌入式 Linux 常用的文件系统	261
7.2.1	ramfs 文件系统	261
7.2.2	tmpfs 文件系统	261
7.2.3	romfs 文件系统	263
7.2.4	cramfs 文件系统	264
7.2.5	jffs2 文件系统	264
7.2.6	yaffs 文件系统	265
7.2.7	ubi 文件系统	265
7.3	详解制作根文件系统	266
7.3.1	FHS 标准介绍	266
7.3.2	编译安装 busybox,生成/bin、/sbin、/usr/bin、/usr/sbin 目录	267
7.3.3	利用交叉编译工具链构建/lib 目录	268
7.3.4	手工构建/etc 目录	271
7.3.5	手工构建最简化的/dev 目录	272
7.3.6	使用启动脚本完成/proc、/sys、/dev、/tmp、/var 等目录的完整构建	274
7.3.7	制作根文件系统的 jffs2 镜像文件	279
7.3.8	制作根文件系统的 yaffs2 镜像文件	279
7.3.9	使用 U-Boot 的 nfs 命令挂载远程文件系统	290
7.3.10	使用 dnw 工具烧写到开发板测试	290
7.4	构建嵌入式 Linux 应用程序系统	291
7.4.1	辅助处理工具的移植	291
7.4.2	mp3 播放器 madplay 的移植	295
7.4.3	主要网络服务器的移植与使用	298
7.4.4	数据库程序的移植与使用	304
7.5	小 结	307
第 8 章	构建 Qt 图形系统	309
8.1	Qt 系统简介	309
8.1.1	Qt 的分类和发展	310
8.1.2	Qt 的应用领域	311

8.1.3 Qt 的资源获取	312
8.1.4 Qt 的环境搭建	312
8.1.5 Qt Creator IDE 开发环境	313
8.1.6 Qt Designer 工具开发 GUI 图形应用	315
8.2 配置目标机环境	316
8.2.1 编译 tslib 库	316
8.2.2 编译 Qt 源码	317
8.2.3 配置目标机的 Qt 运行环境	320
8.2.4 制作 Qt 根文件系统	322
8.2.5 测试运行触摸屏和 Qt 程序	326
8.2.6 Qt 运行时的段错误问题解决	326
8.2.7 解决黑屏问题	332
8.3 Qt 应用程序开发指南	333
8.3.1 建立工程	333
8.3.2 如何使用信号与槽	337
8.3.3 移植 Qt 程序到开发板中运行	338
8.4 gdb/gdbserver 远程代码调试	340
8.4.1 gdb/gdbserver 远程调试介绍	340
8.4.2 gdb 源代码的下载和编译	340
8.4.3 gdb 远程调试命令	341
8.4.4 gdb/gdbserver 远程调试应用程序实例	342
8.5 Qt 的快速有效开发	345
8.5.1 Qt 最新特性说明	345
8.5.2 Qt Quick(QML)使用	348
8.5.3 Qt 移动开发	353
8.5.4 让 Android 系统支持 Qt 应用程序	353
8.6 小 结	354
参考文献	355
后 记	356

第 1 章

嵌入式 Linux 系统开发综述

1.1 嵌入式 Linux 系统的发展现状

Linux 诞生至今已有 20 年,从最早的 0.01 版本上升到目前 3.0 版本。Linux 项目的创始人 Linus Torvalds 也在 2012 年获得了全球科技最高荣誉奖——2012 年千禧年科技奖。该奖用于奖励在科学领域和 IT 界有重大贡献的人物,被誉为“工程界的诺贝尔奖”。

Linux 受到高度的肯定和重视,与它秉承的开源特性和市场实用性强的特点是不分开的。在如今,Linux 在手机、电脑、机器人上的卓越表现足以证明它在 IT 领域所发挥的领袖气质。以下从发展火热的领域对 Linux 系统的发展现状做出说明。

1. 移动消费领域发展现状

互联网高速的发展,特别是移动互联网时代的发展带动了嵌入式系统的火热竞争,表 1-1 列出欧洲国家 2012 年操作系统的市场占有率。

表 1-1 欧洲国家 2012 年操作系统的市场占有率

国 家	操作系统	2011-05-15(%)	2012-05-15(%)	改 变(%)
英 国	Symbian	10.0	1.7	-8.3
	RIM	20.6	12.8	-7.8
	iOS	16.6	29.0	12.4
	WP7	1.1	3.1	2.0
	WinMobile	1.5	0.4	-1.1
	Android	48.3	52.5	4.2
	Bada	1.6	0.1	-1.5
	Other	0.2	0.4	0.2

续表 1-1

国 家	操作系统	2011-05-15(%)	2012-05-15(%)	改 变(%)
德 国	Symbian	27.4	5.7	-21.7
	RIM	4.8	1.7	-3.1
	iOS	17.1	18.2	1.1
	WP7	5.1	3.7	-1.4
	WinMobile	0.9	0.5	-0.4
	Android	37.3	68.6	31.3
	Bada	6.4	0.8	-5.6
	Other	1	0.8	-0.2
法 国	Symbian	18.2	5.3	-12.9
	RIM	12.2	8.8	-3.4
	iOS	21.3	16.2	-5.1
	WP7	1.6	3.2	1.6
	WinMobile	1.2	0.2	-1.0
	Android	36.2	56.0	19.8
	Bada	8.5	9.2	0.7
	Other	0.8	1.2	0.4

从以上移动手机领域不难发现,底层使用 Linux 操作系统的 Android 系统已经占据到了 50% 以上的市场份额。

而对于国内来说更是如此,特别是嵌入式应用在国内发展的火热,使得 Linux 被应用到了更多领域,不再仅仅是作为服务器而被人熟知。

2. 操作系统的演化和发展

目前在 PC 领域受关注的系统包括微软的 Windows 系列、苹果公司的 Mac OS 以及 Linux 的桌面版(Red Hat、ubuntu、Fedora)等等。从普及度来看,很明显是微软占据主导地位,然而这只是对于用户而言,在服务器上,还是 Linux 系统使用居多。

然而,据最新数据统计,2012 年的移动终端的销售量已经超过了 PC 产业,而且发展势头迅猛。不难猜测多元化的嵌入式市场对操作系统的稳定性、高效性、安全性、适用性以及功耗、成本考虑会比 PC 更加严格,因此 Linux 的开源与成熟度逐渐被人们认可,广泛地应用在了嵌入式市场,并且占据着重要的地位。

在智能终端中,Android(Linux)、iOS、Symbian、Windows Phone7、Windows Phone8、Blackberry、WinCE、WebOS、Meego、Bada、Tizen 等系统是目前使用率较多的,而其中主导系统为 Android、iOS,其中 Android 系统使用了开源的 Linux。对于学习来说,开源的 Linux 系统无疑是最合适的选择。

3. ARM 处理器的广泛应用

可以毫不夸张地说,ARM 处理器已经统领了嵌入式市场,几乎占据了智能产品

市场的90%，其中通信设备、消费类电子、智能家电、工业控制、汽车导航、医疗设备占据了ARM总使用率的60%以上。

伴随着3G互联网的发展，嵌入式技术的进步，智能产品的发展在以更快的速度融入我们的生活。而中国政府也在大力推广物联网。那么在智能领域、电子传感领域、无线通信领域等都将更多地依赖基于ARM的嵌入式系统的技术，而ARM公司也针对这一特点，开发了Cortex-A、Cortex-R和Cortex-M系列分别适用于这些领域。有了成熟的ARM硬件和开源稳定的Linux操作系统，基于ARM处理器的Linux系统开发也逐渐成为了一门学科，在各大高校成立了专门的专业，嵌入式这个名词也被更多的人所熟知。

1.2 基于ARM处理器的Linux系统开发过程综述

嵌入式系统是以应用为中心，以计算机技术为基础，且软硬件可裁减，适合于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。它一般由以下几部分组成：

- 嵌入式微处理器；
- 外围硬件设备；
- 嵌入式操作系统；
- 特定的应用程序。

基于ARM的Linux系统属于嵌入式系统。本书使用的处理器为ARM9核心的S3C2440处理器，整个硬件系统的框架结构如图1-1所示。

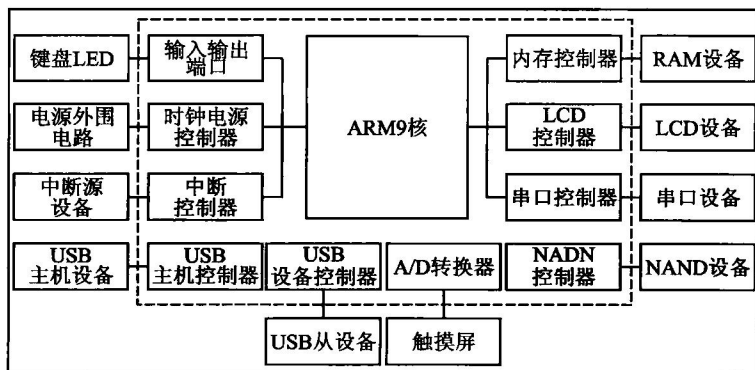


图 1-1 嵌入式硬件框架基本结构

其中ARM9核与片上的外设控制器组成了S3C2440 CPU芯片，并通过控制器连接外围接口设备。软件开发者通过向ARM9核发送指令（能被ARM9识别的机器代码），操作设备控制器的命令、数据、状态寄存器，从而达到控制硬件工作的目的。其中的控制方法：一是要去查看S3C2440提供的寄存器功能；二是对于复杂的硬件