



普通高等教育“十二五”计算机类规划教材

ARM嵌入式Linux

驱动程序开发

© 廉文娟 郭华 范延滨 编

ARM Qianrushi Linux
Qudong Chengxu Kaifa

85



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”计算机类规划教材

ARM 嵌入式 Linux 驱动程序开发

廉文娟 郭华 范延滨 编

附录 (CIP) 目录张全付图

昆明理工大学图书馆
呈贡校区
中文藏书章



03002206313



机械工业出版社

机械工业出版社
地址: 北京市西城区百万庄大街24号
邮编: 100037
电话: (010) 88379696
网址: <http://www.cmpbook.com>
E-mail: custserv@cmpbook.com

本书基于Linux 2.6内核,以S3C2410A为例详细介绍了Linux在ARM上的移植过程和驱动程序设计的理论知识,并在此基础上由浅入深地讲解了Linux驱动程序的设计步骤及调试方法。全书共分11章,主要内容包括嵌入式系统开发概述,创建嵌入式系统开发环境,嵌入式文件系统分析与设计,BootLoader分析与设计,Linux系统的移植,Linux设备驱动程序,串口驱动程序,基于ZLG7290的键盘和数码管驱动设计,LCD驱动程序分析与设计,USB设备驱动程序分析与设计,网络设备驱动程序分析与设计等。

为配合教学,本书提供配套资料:实验所用到的工具软件;实验所用到的开源软件源代码包;全书所使用到的大部分实例代码;本书配套PPT课件。读者可登录机械工业出版社教育服务网www.cmpedu.com注册下载,或发E-mail到xufan666@163.com索取。

本书既可作为高等院校相关专业的教材,也可作为嵌入式系统开发与应用技术人员和Linux设备驱动开发人员的参考书。

图书在版编目(CIP)数据

ARM 嵌入式 Linux 驱动程序开发/廉文娟等编. —北京:机械工业出版社, 2014. 3

普通高等教育“十二五”计算机类规划教材
ISBN 978-7-111-45500-4

I. ①A… II. ①廉… III. ①Linux 操作系统—高等学校—教材
IV. ①TP316.89

中国版本图书馆CIP数据核字(2014)第008503号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:徐凡 责任编辑:徐凡

版式设计:霍永明 责任校对:李锦莉 刘秀丽

封面设计:张静 责任印制:张楠

北京京丰印刷厂印刷

2014年3月第1版·第1次印刷

184mm×260mm·21.75印张·537千字

标准书号:ISBN 978-7-111-45500-4

定价:43.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网: <http://www.cmpedu.com>

销售一部:(010) 68326294

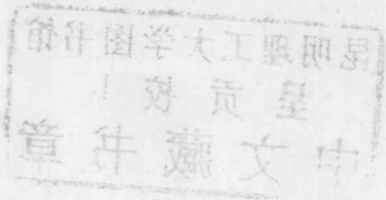
机工官网: <http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博: <http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版



前 言

驱动程序是连接上层应用层和底层硬件层之间的桥梁,是嵌入式系统开发中不可或缺的重要组成部分。当前在市面上能够见到的有关 ARM 嵌入式 Linux 驱动程序开发方面的书籍至少有几十种,它们各有特色,也各有侧重。一方面,从体系结构上看,它们大都比较综合,涉及嵌入式系统设计原理、嵌入式处理器原理、嵌入式 Linux 移植以及 Linux 驱动程序框架和简单驱动的实现。此类图书偏重于嵌入式 Linux 开发设计的流程设计,驱动设计也是以原理为主,比较适合初学者作为参考。另一方面,从内容上看,绝大多数此类图书,以某个公司的现有开发板为原型,进行驱动的讲解和设计,过分依赖于开发板。

本书详细介绍 Linux 在 ARM 上的移植过程和驱动程序设计的理论知识,并在此基础上由浅入深,详细讲解 Linux 的驱动程序设计步骤以及 Linux 驱动程序的调试方法。本书基于 Linux 2.6 内核,以 S3C2410A 为主进行介绍。

全书共分 11 章,主要内容包括嵌入式系统介绍、嵌入式系统开发环境、嵌入式文件系统分析与设计、BootLoader 分析与设计、Linux 系统的移植、Linux 设备驱动程序、串口驱动程序、基于 ZLG7290 的键盘和数码管驱动设计、LCD 驱动程序分析与设计、USB 设备驱动程序分析与设计、网络设备驱动程序分析与设计。

本书是在作者教学和课程实验基础上编写的。为了培养学生嵌入式 Linux 驱动程序开发的能力,注重理论和实践相结合。在讲叙理论的基础上,本书紧紧围绕“经验”和“实践”,在相关驱动程序开发的安排上,着重突出“应用”和“实用”两个基本原则,安排具有代表性、应用广泛的典型实例,易于读者学习借鉴。特别在程序设计思路,尽可能地用简洁的语言来清晰阐述开发的步骤。同时,程序代码部分做了很详细的中文注释,有利于读者举一反三,快速应用和提高。

本书基于一线教师多年的教学经验,从章节编排到案例选择均适合教学,具有针对性。本书内容丰富,专业性强,语言通俗易懂,具有较好的操作性。

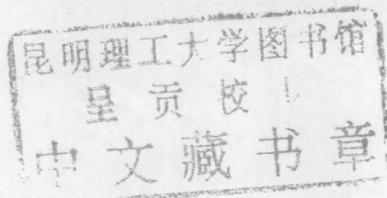
本书适合嵌入式系统设计人员以及高等院校相关专业师生学习和使用,也可以作为 ARM 嵌入式系统接口设计人员和嵌入式 Linux 系统的内核、驱动及应用程序开发人员的参考资料。

本书由廉文娟、郭华、范延滨编写,廉文娟统稿。

为便于读者学习,本书配有教学课件、源代码和相关文档,读者可以到机械工业出版社教育服务网 www.cmpedu.com 下载或发邮件到 xufan666@163.com 索取。

由于作者水平有限,书中难免存在一些错误和不足。对于本书的相关意见或建议,请发 E-mail 到 wenjuan.lian@126.com 与作者联系。

编 者



目 录

前言

第1章 嵌入式系统概述 1

1.1 嵌入式系统的概念 1

1.1.1 嵌入式系统的定义 1

1.1.2 嵌入式系统的特点 2

1.1.3 嵌入式系统的发展趋势 3

1.2 嵌入式系统的分类 4

1.3 嵌入式系统的组成结构 6

1.4 嵌入式处理器简介 9

1.5 ARM 处理器 11

1.5.1 ARM 系列处理器概述 11

1.5.2 ARM 处理器的寄存器文件 18

1.5.3 ARM 处理器的异常与中断 22

1.5.4 ARM 处理器的指令系统 25

1.5.5 S3C2410A 的性能 35

1.6 嵌入式操作系统 39

1.6.1 嵌入式操作系统的发展阶段 39

1.6.2 嵌入式实时操作系统 39

1.6.3 嵌入式操作系统选型 40

1.6.4 典型嵌入式操作系统 41

本章小结 44

第2章 创建嵌入式系统开发环境 45

2.1 Linux 常用命令 45

2.1.1 命令基础 45

2.1.2 文件操作命令 47

2.1.3 目录操作命令 50

2.1.4 文件与目录权限操作 51

2.1.5 Linux 用户和组操作的常用命令 51

2.1.6 文件压缩与打包命令 54

2.1.7 进程操作的相关命令 56

2.1.8 网络命令 57

2.2 交叉编译环境的预备知识 59

2.2.1 GCC 编译器 59

2.2.2 Glibc 库和 GDB 调试工具 62

2.2.3 make 命令和 makefile 文件 68

2.2.4 Binutils 工具包 72

2.3 搭建交叉编译环境 74

2.3.1 交叉编译简介 74

2.3.2 搭建 S3C2410A 交叉编译环境 75

2.3.3 程序执行方式 76

本章小结 79

第3章 嵌入式文件系统分析与设计 80

3.1 Linux 文件系统类型 80

3.2 Linux 根文件系统 83

3.3 Linux 系统的引导过程 85

3.4 嵌入式文件系统 88

3.4.1 嵌入式文件系统的特点和种类 88

3.4.2 常见的嵌入式文件系统 90

3.4.3 内存技术设备 92

3.4.4 RAMDisk 技术 93

3.4.5 网络文件系统 97

3.5 Busybox 102

本章小结 106

第4章 BootLoader 分析与设计 107

4.1 嵌入式系统的引导代码 107

4.1.1 初识 BootLoader 107

4.1.2 BootLoader 的启动流程 109

4.2 常用的嵌入式 Linux BootLoader 113

4.2.1 U-Boot 113

4.2.2 VIVI 114

4.2.3 Blob 115

4.2.4 RedBoot 115

4.2.5 其他 BootLoader 116

4.3 U-Boot 分析与移植 116

4.3.1 U-Boot 源代码结构 116

4.3.2 U-Boot 源代码编译 118

4.3.3 U-Boot 移植步骤 123

4.3.4 U-Boot 启动流程分析 124

4.4 U-Boot 使用 130

4.4.1 U-Boot 的常用命令 130

4.4.2 U-Boot 的环境变量 137

本章小结	139	7.4.2 串口通信应用	228
第5章 Linux 系统的移植	140	本章小结	232
5.1 Linux 内核配置	140	第8章 基于 ZLG7290 的键盘和数码管	
5.1.1 Linux 内核源代码结构	140	驱动设计	233
5.1.2 Linux 内核配置系统	141	8.1 ZLG7290 介绍	233
5.2 Linux 内核编译	148	8.1.1 ZLG7290 功能描述	233
5.2.1 Kbuild 编译工具	148	8.1.2 ZLG7290 工作原理	234
5.2.2 Linux 内核编译命令及编译过程	156	8.1.3 ZLG7290 内部寄存器	235
5.2.3 Linux 内核编译结果	161	8.1.4 控制命令详解	236
5.3 Linux 内核移植	162	8.2 IIC 总线	238
5.3.1 Linux 内核移植方法	162	8.2.1 IIC 总线概述	238
5.3.2 Linux 内核移植过程	163	8.2.2 IIC 总线标准	239
5.4 Linux 内核启动过程分析	173	8.3 ZLG7290 硬件电路设计	243
5.4.1 Linux 内核启动流程	173	8.4 S3C2410A 中与 IIC 相关的 I/O 端口和寄存器	244
5.4.2 Linux 内核启动源代码分析	173	8.4.1 S3C2410A 的 IIC 总线接口	244
本章小结	186	8.4.2 各种模式下的操作流程	246
第6章 Linux 设备驱动程序	187	8.4.3 IIC 总线特殊寄存器	246
6.1 设备驱动基础	187	8.5 Linux 的 IIC 驱动程序分析与设计	251
6.1.1 设备驱动的概念	187	8.5.1 IIC 总线驱动的关键数据结构	251
6.1.2 设备类型与数据交换方式	189	8.5.2 S3C2410A 总线 IIC 设备驱动实现	253
6.1.3 内核空间和用户空间	194	8.5.3 ZLG7290 的 IIC 驱动安装与测试	255
6.2 设备驱动开发	195	本章小结	259
6.2.1 Linux 设备驱动程序框架	195	第9章 LCD 驱动程序分析与设计	260
6.2.2 设备驱动中的关键数据结构	198	9.1 LCD	260
6.2.3 设备驱动中的基本函数	202	9.2 LCD 的显示接口	262
6.2.4 字符设备驱动开发举例	205	9.3 嵌入式处理器的 LCD 控制器	270
本章小结	207	9.3.1 LCD 控制器	270
第7章 串口驱动程序	208	9.3.2 LCD 控制器的设置	271
7.1 串口概述	208	9.3.3 LCD 的字符显示缓存	276
7.2 S3C2410A 串口控制器	210	9.4 基于 framebuffer 的 LCD 驱动程序	277
7.2.1 S3C2410A 串口控制器原理	211	9.4.1 framebuffer 概述	277
7.2.2 S3C2410A 串口控制器相关寄存器	213	9.4.2 framebuffer 设备驱动程序的结构	277
7.3 嵌入式 Linux 串口驱动程序设计	216	9.4.3 framebuffer 设备驱动程序的设计	281
7.3.1 串口驱动程序框架	216	9.5 LCD 驱动程序在 S3C2410A 上的移植	289
7.3.2 串口驱动程序的注册	218	9.5.1 移植步骤	289
7.3.3 ops 中的串口操作函数	220		
7.3.4 串口驱动中与控制台有关的函数	224		
7.4 串口应用程序设计	226		
7.4.1 串口终端函数	226		

9.5.2 LCD 驱动程序分析	292
本章小结	301
第 10 章 USB 设备驱动程序分析与设计	302
10.1 USB 接口概述	302
10.1.1 USB 系统组成	302
10.1.2 USB 的电气特性	302
10.1.3 USB 总线的拓扑结构	303
10.1.4 USB 的通信协议	304
10.2 USB 设备驱动程序设计	305
10.2.1 USB 设备驱动程序分类	305
10.2.2 USB 驱动的框架程序	305
10.2.3 设备端 USB 驱动程序	314
10.3 USB 驱动程序在 S3C2410A 上的移植	315
本章小结	317
第 9 章 LCD 驱动程序分析与设计	260
9.1 LCD	260
9.2 LCD 的显示接口	263
9.3 嵌入式系统驱动的 LCD 控制	270
9.3.1 LCD 控制	270
9.3.2 LCD 控制器的配置	271
9.3.3 LCD 的寄存器	276
9.4 基于 framebuffer 的 LCD 驱动程序	277
9.4.1 framebuffer 概述	277
9.4.2 framebuffer 设备驱动程序	277
9.4.3 framebuffer 设备驱动程序	277
9.5 LCD 驱动程序在 S3C2410A 上的移植	281
9.5.1 移植	281
9.5.2 移植	281

第 11 章 网络设备驱动程序分析与设计	318
11.1 网络协议	318
11.2 以太网	321
11.2.1 以太网的拓扑结构	322
11.2.2 以太网协议	322
11.2.3 以太网帧结构	323
11.3 嵌入式网络接口	324
11.4 嵌入式 Linux 网络驱动	329
11.4.1 Linux 网络设备驱动程序的体系结构	330
11.4.2 网络驱动程序的基本方法	330
11.4.3 网络驱动中的主要数据结构	333
11.5 网络驱动程序的移植及分析	335
本章小结	339
参考文献	340

第1章 嵌入式系统概述

1.1 嵌入式系统的概念

1.1.1 嵌入式系统的定义

嵌入式系统的外延广泛。一台 PC104 微型工业控制计算机是典型的工控类嵌入式系统；一部手机是典型的消费类嵌入式系统；一张 IC 卡是典型的智能卡类嵌入式系统。然而，它们几乎又是完全不同的。因此，嵌入式系统定义的标准化是困难的，一般有如下几种定义方式。

(1) 电气电子工程师协会 (Institute of Electrical and Electronics Engineers, IEEE) 的定义

嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置” (Devices used to control, monitor, or assist the operation of equipment, machinery or plants)。

(2) 国内的定义

嵌入式系统是“以应用为中心、以计算机技术为基础、软件/硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统”。

国内从事嵌入式系统的工作者一般认为 IEEE 的定义并没有充分体现出嵌入式系统的精髓，而认为国内的专家学者所给出的定义更为深刻、确切，更具有普遍意义。

①嵌入式系统是面向用户、面向产品、面向应用的。它必须与具体应用相结合才会具有生命力，才更具有优势。因此，可以理解该定义的含义如下：嵌入式系统是一个计算机系统，它与应用紧密结合且具有很强的专用性，它必须结合实际系统需求而进行合理的设计。

②嵌入式系统是计算机技术、微电子技术等多种高新技术的融合体，它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。因此，进入嵌入式系统的行业者，必须要有一个正确的定位。例如，风河的 Vxworks 之所以在火星车上得以应用，是因为其具有高实时性和高可靠性。

③嵌入式系统必须能够根据应用需求对软硬件进行裁剪，以满足应用系统的功能、可靠性、成本、体积等要求。所以，如果能建立相对通用的软硬件基础，然后在其上开发出适应各种需要的系统，是一个比较好的发展模式。目前的嵌入式系统的核心往往是一个只有几 KB (千字节) 到几十 KB 的微内核，需要根据实际的使用进行功能扩展或者裁减，由于微内核的存在，使得这种扩展能够非常顺利的进行。

(3) 行业的定义

AMD 认为：“除桌面微型电脑与笔记微型电脑之外，其他的都叫做嵌入式产品”。显然，这个定义极为宽泛。在这种定义下，x86 系列的嵌入式产品市场变得巨大，因此对生产 x86 体系结构 CPU 的厂商更加实用。

(4) 本书的注解

实际上,嵌入式系统是一个外延极广的词汇。本书认为,凡是结合产品功能的、含有嵌入式处理器的、运行嵌入式操作系统的、设计专门化的计算系统都叫嵌入式系统。一般而言,嵌入式系统的构架可以分成嵌入式处理器、IO 控制器、嵌入式操作系统和嵌入式应用软件四部分。该定义更便于读者明确对嵌入式系统学习的目标、方向和内容等。一个嵌入式系统的开发者,一般需要具有如下知识结构:

①以嵌入式处理器为核心的知识体系,包括计算机组成与结构、具体嵌入式微处理器(如 ARM、MIPS 等)、模拟/数字电子技术、汇编语言程序设计、硬件描述语言等。

②以嵌入式操作系统为核心的知识体系,包括操作系统原理、具体嵌入式操作系统(如 Linux、 μ C/OS-II/III 等)、C/C++ 语言程序设计、数据结构等。

③以产品应用为核心的知识体系,包括模拟电子技术、传感器技术、与应用相关的专业知识(如机械原理、医疗技术)等。

④以设计技术为核心的知识体系,包括硬件设计技术、软件设计技术、软硬件协同设计方法学、编译技术等。

本书的注解不在于对嵌入式系统定义的标准化,而是注重定义的通俗性、功能性、层次性和知识性,便于初学者依照自己的知识体系来理解和学习。

1.1.2 嵌入式系统的特点

从整体上看,嵌入式系统包括嵌入式处理器、定时器、控制器、存储器、传感器等一系列微电子芯片和嵌入在存储器中的嵌入式操作系统、控制应用软件等,实现诸如实时控制、通信、监视、管理、移动计算、数据处理等各种智能化处理任务。嵌入式系统以应用为中心,以微电子技术、控制技术、计算机技术和通信技术为基础,强调硬件与软件的协同性与整合性,软件与硬件可剪裁,以满足系统对功能、成本、体积和功耗等要求。

嵌入式系统具有以下特点:

①嵌入性。嵌入式系统是嵌入到“特定对象体系”中的实体。因此,嵌入式系统通常不是应用对象的主体而只是辅助性的装置,要求其体积小、功耗低、抗干扰能力强、成本低等。

②系统性。嵌入式系统是一个计算机系统。因此,嵌入式系统会具有软硬件两方面,软件包括嵌入式操作系统和应用软件;硬件包括嵌入式处理器子系统、外部设备控制器等多种接口电路。

③智能性。嵌入式系统的硬件核心是嵌入式处理器,软件核心是嵌入式操作系统。一个嵌入式系统至少含有一个嵌入式处理器,加载一个嵌入式操作系统,运行相关应用程序,智能地实现预定的功能。

④专用性。嵌入式系统是为“特定对象体系”专门设计的。其主要体现在嵌入式处理器上,称为“专用处理器”。例如,语音处理系统、视频处理系统、电机控制系统等其嵌入式处理器的功能是不同的,因此要熟悉嵌入式处理器的类型、性能等。

⑤实时性。嵌入式系统是“特定对象体系”的一个模块。它常被用作控制、监视或辅助操作机器和设备的装置,因此要具有很好的实时性,以满足对象系统的要求。

⑥恰适性。嵌入式系统要恰好满足“特定对象体系”的需要。因此,对于一个成功的

嵌入式系统设计, 对其量体定制、软硬件都无冗余是最好的。

⑦系统设备电子化。PC 中的资源有缓存和内存 (电子的)、外存 (磁性的、光介质)、打印机 (机电一体的)、显示器 (如 LCD、CRT) 等, 种类繁多。嵌入式系统的内存、外存都是电子的 (通常使用 Flash Memory 等), 其特点是可以被嵌入式处理器直接操作。

⑧编程语言低级化。嵌入式系统的编程语言更趋于低级语言, 主要为汇编语言、C 语言、C++ 语言等。

⑨操作系统实时化。嵌入式系统的重要特性是实时性, 因此要求嵌入式操作系统应该为实时嵌入式操作系统 (Real-Time embedded Operating System)。

⑩系统性能可测化。嵌入式系统要求具有软硬件可测试, 以保证系统无错误。

⑪开发系统专用化。嵌入式产品分为集成开发环境 (Integrated Development Environment, IDE)、在线仿真器 (In-Circuit Emulator, ICE)、实时操作系统 (Real Time Operating System, RTOS)。ARM 芯片上都包含专用的在线调试电路, 如 ARM 的 Embedded ICE, ARM 微处理器基于 JTAG 接口等。

1.1.3 嵌入式系统的发展趋势

嵌入式技术是继网络技术之后的又一主流技术。嵌入式技术从 20 世纪 90 年代全面展开, 随着信息化、智能化、网络化的发展, 嵌入式系统也已获得了广阔的发展空间。目前, 嵌入式技术已成为智能领域的核心技术。

(1) 嵌入式开发是一项系统工程

嵌入式开发是一项系统工程, 因此要求嵌入式系统厂商不仅要提供嵌入式软硬件系统本身, 同时还需要提供强大的硬件开发工具和软件包支持。

(2) 嵌入式处理器的发展趋势

嵌入式处理器的位宽从 32 位走向 64 位, 进一步增强了 DSP 功能, 集成更多的功能逐步实施片上系统 (SOC)。例如, 嵌入式设备为了适应网络发展的要求, 必然要求硬件上提供各种网络通信接口。除了支持 TCP/IP、UART 协议, 还有的支持 IEEE 1394、USB、CAN、Bluetooth 或 IrDA、3G 通信接口中的一种或者几种, 同时也需要提供相应的通信组网协议软件和物理层驱动软件。

(3) 嵌入式操作系统的发展趋势

嵌入式操作系统伴随着嵌入式系统的发展经历了四个比较明显的发展阶段: 无操作系统的嵌入算法阶段; 以嵌入式处理器为基础、简单操作系统为核心的嵌入式系统; 通用的嵌入式实时操作系统阶段; 以基于 Internet 为标志的嵌入式系统。具有代表性的嵌入式操作系统产品主要有 Linux、Windows CE、VxWorks、 μ C/OS-II、QNX、Palm OS、TinyOS、Contiki、RT-Thread、Delta OS、Hopen OS、CASSPDA 以及 HBOS 等。

(4) 嵌入式系统推动了物联网的崛起

物联网 (Internet of Thing, IOT) 的崛起, 也显示出了嵌入式系统的巨大发展潜力。物联网是物与物、物与人、人与人之间的信息采集、传输与处理。物联网的核心设备是智能终端, 物联的实现就是智能终端的网络化运行。嵌入式系统又是智能终端的“大脑”和“中枢神经”, 因此嵌入式系统是物联网产业发展的核心推动力。

物联网为人类生活引入了简单而神圣的概念——“智慧”或“智能”, 形成了一系列的

“智慧”与“智能”领域。例如，智慧中国、智慧城市、智慧社区、智能建筑、智能家居、智能工业、智慧农业、智能电网、智能医疗、智能交通、智能物流、智能旅游、智能安防、智能汽车、智能手机、智能电脑、智能家电、智能机器人、环境与安全检测、金融与服务、国防军事等。

在这一系列的“智慧”与“智能”领域中，嵌入式技术构成了“智慧”与“智能”的支撑技术，而嵌入式操作系统又是这些“智慧”与“智能”平台稳定运行的保障。

1.2 嵌入式系统的分类

根据不同的分类标准，嵌入式系统可有许多不同的分类方法。例如，可以按照嵌入式系统的复杂度分类，按照嵌入式处理器的位数分类，按照嵌入式系统的实时性分类，按照嵌入式系统的组成分类，按照嵌入式系统的领用领域分类，等等。

1. 按照嵌入式系统的复杂度分类

嵌入式系统的应用领域广泛，简单的有用 4 位单片机实现的电动玩具，复杂的有基于 32/64 位处理器的飞机飞行控制系统。按照其复杂程度，嵌入式系统可以分为简单嵌入式系统和复杂嵌入式系统。在实际应用中，简单嵌入式系统在数量上占有主要的市场，而复杂嵌入式系统是学习的主流。

(1) 简单嵌入式系统

简单嵌入式系统的软硬件复杂度都比较低。例如，常用的有 8 位或 16 位单片机系统，不使用操作系统，只有一个监控程序和应用程序。

(2) 复杂嵌入式系统

复杂嵌入式系统的软硬件复杂度都比较高，通常，在硬件上需要使用 32 位微处理器，在软件上含有嵌入式操作系统。这类嵌入式系统集成度高、接口丰富、功能强大。

2. 按照嵌入式系统的实时性分类

嵌入式系统的重要特性就是实时性。按照其实时性要求，嵌入式系统可分为硬实时嵌入式系统、软实时嵌入式系统和非实时嵌入式系统。

(1) 硬实时系统

硬实时系统是指系统要确保在最坏情况下的服务时间，即对于事件响应时间的截止期限必须得到满足，而且其响应时间是可时刻计算的。例如，汽车、舰船、飞机、导弹、卫星的控制等就是这样的系统，工业控制系统通常也是实时系统。

(2) 软实时系统

明确地说，软实时系统就是那些从统计的角度来说，一个任务能够得到确保的处理时间，到达系统的事件也能够在此期限前得到处理，但违反截止期限并不会带来致命的错误。例如，手机就是一种软实时系统，MP3 等媒体播放器也是软实时系统。

(3) 非实时系统

非实时系统对实时性没有明确的要求，也就是说没有事件响应时间的截止期限，对事件的响应完全由系统自身的工作状态所决定，如 PDA、PC 等。

系统的实时性需要由硬件和软件共同起作用来实现，硬件上通过中断来保证事件能够被实时感知；软件上通过实时操作系统来保证事件能够被实时响应。

3. 按照嵌入式处理器的位数分类

嵌入式处理器是嵌入式系统的硬件核心。嵌入式处理器一般可分为 4/8 位、16 位、32 位嵌入式处理器。那么,嵌入式系统一般也可根据所采用的嵌入式处理器分为 8 位、16 位、32 位嵌入式系统。

8 位和 16 位的嵌入式系统通常是以 8 位和 16 位微控制器(单片机)为核心构成的简单嵌入式系统,广泛地应用于控制系统、家用电器、汽车电子等。

32 位嵌入式系统通常是以 32 位微处理器或微控制器为核心构成的复杂嵌入式系统,广泛地应用于需要进行手机、消费电子、医疗电子、网络设备等。

4. 按嵌入式系统的组成分类

按照其组成,嵌入式系统可分为嵌入式系统硬件和嵌入式系统软件两大部分。

(1) 嵌入式系统硬件

嵌入式系统硬件是以嵌入式处理器为核心,以丰富的功能接口部件为外围扩展层构建的。也正是基于这些丰富的外设接口,才带来了嵌入式系统越来越丰富的应用。外设接口部件一般是以其接口控制器芯片(如 USB 控制器)为核心搭建的,通过标准的扩展总线与嵌入式处理器实现连接。在嵌入式系统硬件设计中,通常只要把处理器和控制器进行物理连接就可以实现外设接口扩展了。

嵌入式处理器又以处理器核(如 ARM 核)为核心,片内集成适当的接口控制器(称为片内外设)构成的。随着嵌入式处理器高度集成化技术的发展,使得片内可以实现的控制器的越来越多,功能也越来越强。例如,基于 ARM 核处理器 S3C2410A 的内部就封装了 Cache、I2C、UART、USB、LCD 控制器等十几种片内外设。在嵌入式系统设计中,所需要的外围接口电路越来越少。

(2) 嵌入式系统软件

一般来说,嵌入式系统软件由嵌入式操作系统和应用软件两部分组成。本书中的嵌入式系统软件可以分成启动代码(BootLoader)、操作系统内核与驱动、文件系统与应用程序等几部分。

① BootLoader 是嵌入式系统的启动代码,主要用来初始化处理器和必须使用的控制器,传递内核启动参数给嵌入式操作系统内核,使得内核可以按照所设定的参数要求启动。另外,BootLoader 通常都具有从 Flash 搬运内核代码到 RAM 并跳转到内核代码地址运行的功能。

② 嵌入式操作系统内核主要有四个任务:进程管理、进程间通信与同步、内存管理及 I/O 资源管理。驱动程序主要提供给上层应用程序,是处理器、外设接口控制器和外部设备进行通信的一个媒介。

③ 文件系统则可以让嵌入式软件工程师灵活方便地管理系统。

④ 应用程序才是真正针对需求的、有可能是嵌入式软件工程师完全自主开发的。

总的来说,嵌入式系统的硬件可以说是系统的基石;嵌入式系统的软件是系统的灵魂,也是在系统的基石上面镌刻出来的工艺品。对于任何一个需求明确的嵌入式系统来说,两者缺一不可。在对系统做了相对完整而细致的需求分析之后,通常采用软件和硬件基本同步进行的方式来开发,前期硬件系统的设计要比软件系统设计稍微提前,到了后期软件系统开发的工作量会比硬件系统开发的工作量大一些。

1.3 嵌入式系统的组成结构

嵌入式系统一般由硬件平台和软件平台两部分组成。其中，硬件平台由嵌入式处理器、外设控制器芯片和硬件设备组成；而软件平台由 BootLoader、嵌入式操作系统、驱动程序和应用软件组成。

随着芯片技术的不断发展，嵌入式处理器的主频也越来越高，多处理器、多核处理器平台也逐渐应用在嵌入式领域。嵌入式系统的组成也将由单核、低频设计进入多核、高频设计。

1. 典型嵌入式系统的组成结构

图 1-1 完整地描述了典型嵌入式系统的组成结构。对于一个具体的嵌入式系统，图 1-1 中的各种资源并不一定全部使用，是可裁剪的，但是这一切都需要嵌入式系统开发者根据需求自己去取舍。这就要求嵌入式系统开发者必须同时具有较好的软硬件知识和专业知识。

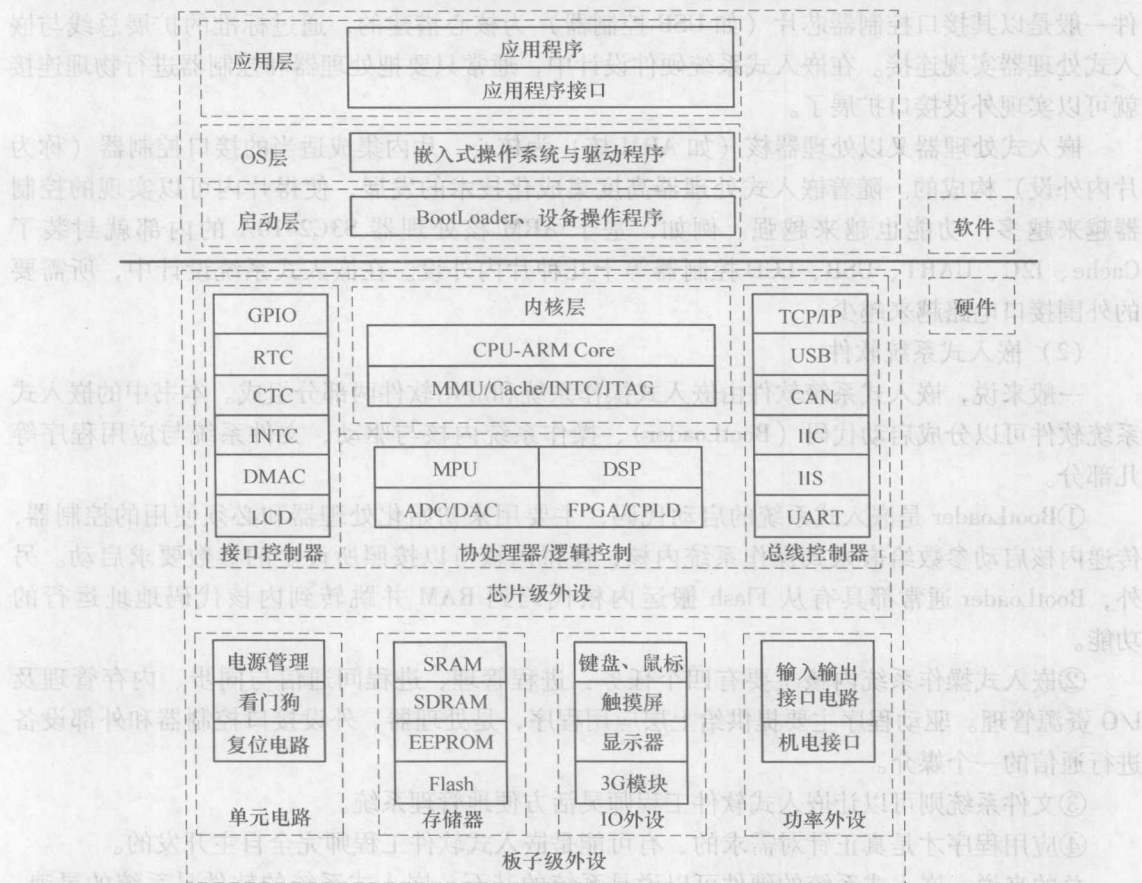


图 1-1 典型嵌入式系统的组成结构

2. 典型嵌入式系统的硬件结构

典型嵌入式系统的硬件层大致可分为三层：内核层、芯片级外设层（片内外设/片内接

口层)、板子级外设层(板内外设/板内接口层)。

(1) 内核层

内核层主要包括嵌入式处理器核、存储器管理、缓存、调试单元等部件,内核层由处理器核开发商提供。目前,处理器体系结构有 50 多个系列,基于 CISC 结构的嵌入式处理器核有 Intel 公司的 Atom 系列等;基于 RISC 结构的嵌入式处理器核有 IBM 公司的 Power PC 系列、MIPS 公司的 MIPS 系列和 ARM 公司的 ARM 系列等。ARM 公司是 ARM 核的生产商,比较流行的 ARM 核有 ARM7TDMI、ARM9、ARM10、ARM11、Cortex-A/R/M 等。

(2) 芯片级外设层

芯片级外设位于嵌入式处理器芯片内部,也称为片内外设。芯片级外设层主要包括与内核层集成在同一个芯片内部的各种接口控制器、总线控制器、协处理器、逻辑控制单元等,决定了嵌入式处理器本身的扩展能力。片内外设具有容易扩展和强抗干扰性能。

在 ARM 嵌入式处理器中,ARM 公司仅仅提供了基于 ARM 核的系统芯片解决方案和技术授权。而芯片级外设是由各个半导体厂商根据自己产品的市场定位和应用领域来设计开发的,其功能是预设计的、可编程的。处理器生产商把所选择的芯片级外设与内核层集成在同一芯片之中,形成一个片上系统(System on a Chip, SoC),称之为嵌入式处理器。嵌入式系统开发者需要学习芯片级外设的设计技术和编程方法,然后才能对其进行设计、编程和使用。

基于 SoC 技术,将内核层和芯片级外设层集成在一个芯片之中,可以形成一个基于应用的解决方案。以 SoC 为核心应用最少的外围部件和连线就能完成一个应用系统,极大地简化了系统设计的复杂度。

(3) 板子级外设层

板子级外设也称为板内外设。板内外设是终端用户依据系统功能设计于线路板上的外围接口电路、控制器等,实现对外设的控制、匹配、驱动等,以达到服务于嵌入对象的目标。板子级外设层一般包括存储设备、通信接口设备、扩展设备和机电设备的扩展与驱动等。这些接口和设备是需要用户根据具体工程要求来具体设计、开发的。

例如,用户选取 S3C2410A 嵌入式处理器开发一款数据终端设备,要求系统具有串行口、以太网口、USB 口、LCD 口等。因为 S3C2410A 具有串行口、USB 口、LCD 口等片内外设,因此用户只需要设计电平转换电路就可以简单地完成串行口、USB 口、LCD 口的开发;但是 S3C2410A 并不具有以太网口的片内外设,用户必须选择适当的以太网口控制器,设计板内外设以太网口。

3. 典型嵌入式系统的软件结构

软件层大致可分为三层:启动层、操作系统层、应用层。

(1) 启动层

嵌入式系统硬件需要进行初始化和操作,这部分工作是由设备启动层来完成的。启动层直接对硬件进行操作和控制,为上层驱动程序提供所需的操作支持,类似 PC 系统中的 BIOS。

参考 Windows CE 中板级支持包(Board Support Package, BSP)中的定义,启动层也称为 BSP,一般包括 BootLoader、硬件抽象层(Hardware Abstraction Layer, HAL)、OEM 适配层(OEM Adapter Layer, OAL)和设备操作程序。实际上,BootLoader、HAL、BSP 等概念

没有统一的定义,但都定义了从系统上电启动到操作系统内核加载之前的功能和操作,其区别是实现的功能多少不同。

① BootLoader

BootLoader 就是在操作系统内核运行之前所运行的一段小程序。通过这段小程序,可以初始化硬件设备,建立内存空间的映射图,从而将系统的软硬件环境带到一个合适的状态,以便为最终调用操作系统内核准备好正确的环境。通常,BootLoader 是严重地依赖于硬件而实现的。因此,在嵌入式系统中建立一个通用的 BootLoader 几乎是不可能的。其主要功能如下:

- 片级初始化。片级初始化是纯硬件初始化,把嵌入式处理器从上电的默认状态设置成系统要求的工作状态。
- 板级初始化。板级初始化包括硬件和软件初始化,设置板内的各种控制器的寄存器、设置软件的数据结构和参数。
- 加载内核。加载内核即将嵌入式操作系统和应用程序的映像从 Flash 存储器复制到系统内存当中,然后跳转到嵌入式操作系统内核的第一条指令处继续执行。

② 设备操作程序

设备操作程序主要为上层驱动程序和软件提供设备的操作接口。在一个嵌入式系统中,嵌入式操作系统可能有,也可能没有,但是设备操作程序是必不可少的。设备操作程序就是一组库函数,用来对硬件进行初始化和操作,并向上层软件提供良好的访问接口。大多数设备操作程序都具备打开、关闭、读操作、写操作等基本功能。这些功能一般用函数的形式来实现,函数之间的组织结构主要有分层结构和混合结构两种。

- 分层结构。分层结构包括硬件接口和调用接口。硬件接口直接操作和控制硬件,把所有与硬件相关的细节都封装在硬件接口函数中;调用接口对调用接口进行封装,为上层软件提供服务 and 函数接口。
- 混合结构。在设备驱动程序当中,没有明确的层次关系,上层调用接口和硬件接口混在一起,相互调用。

与混合结构相比,分层的优点是:在硬件需要升级,需要更新设备驱动程序时,只需要改动硬件接口中的函数即可,而上层调用接口中的函数不用做任何修改。

(2) 操作系统层

对于使用嵌入式操作系统的嵌入式系统而言,嵌入式操作系统一般是以内核映像的方式下载到目标系统中。以 Linux 为例,在系统开发完成后,会将使用到的 Linux 部分做成内核映像文件,与文件系统一起传送到目标系统中;然后通过 BootLoader 指定的地址运行 Linux 内核,启动已经下载好的 Linux;再通过操作系统解开文件系统,运行应用程序。

- 必需部件。内核中的必需部件包括进程管理、进程间通信、内存管理、设备管理等。
- 嵌入式操作系统。常用的嵌入式操作系统有 Linux、Windows CE、VxWorks、μC/OS-II/III、RT-Thread 等。
- 嵌入式中间件。它是在操作系统内核、设备驱动程序和应用软件之外的所有系统软件,其基本思路是:把原本属于应用软件层的一些通用的功能模块抽取出来,形成独立的一层软件,从而为运行在它上面的那些应用软件提供一个灵活、安全、移植性好、相互通信、协同工作的平台。

器(3)应用层

应用层又可以分为应用程序接口 (Application Programming Interface, API) 层和应用程序层。API 层是一系列复杂的函数、消息和结构的集合体; 应用程序层是建立在系统主任务 (Main Task) 基础上的, 应用程序可以调用 API 函数, 用户的应用程序也可以创建自己的任务, 任务间的协调主要依赖于系统的消息队列。

1.4 嵌入式处理器简介

PC 的核心是其中央处理器 (Central Processing Unit, CPU), 嵌入式系统的核心是嵌入式处理器 (Embedded Processing Unit, EPU)。PC 的 CPU 主要由 Intel、AMD 等少数公司垄断生产, 而嵌入式微处理器却是多种多样的。目前, 嵌入式微处理器已超过 1500 余种, 几乎每个半导体生产商都有自己的产品。但是, 对于 32 位嵌入式处理器, ARM 系列占据 80% 以上的市场。

嵌入式系统应用广泛, 嵌入式处理器种类繁多。嵌入式处理器基本上可以分为嵌入式微处理器 (Embedded Micro Processing Unit, EMPU)、嵌入式微控制器 (Micro Controlling Unit, EMCU)、嵌入式数字信号处理器 (Embedded Digital Signal Processing, EDSP)、嵌入式片上系统 (Embedded System on Chip, ESoC), 如图 1-2 所示。这种分类仅仅是概念上的, 是分块式的、并不严格的。

EMPU、EMCU、EDSP、ESoC 又都可以按照如下方法分类。

按数据位数分: 可以分为 8 位、16 位、32 位、64 位等。

按存储结构分: 可以分为冯·诺依曼结构、哈佛结构等。

按指令系统结构分: 可以分为复杂指令集 (Complex Instruction Set Computer, CISC) 结构、精简指令集 (Reduced Instruction Set Computer, RISC) 结构。

按应用领域分: 可以分为应用处理器、通信类、多媒体类、控制类等。

按处理器体系结构分: 可以分为 Intel IA x86、PowerPC、ARM、MIPS、68K/ColdFire 等系列。

1. 嵌入式微处理器

EMPU 是从通用处理器演变而来的, 也可谓是通用计算机中 CPU 的微缩版。相对于通用处理器而言, EMPU 不仅仍然具有较高的运算性能、可以运行嵌入式操作系统等特点, 而且在功耗、工作温度、抗电磁干扰、可靠性等方面都进行了各种增强设计, 其性能具有显著地提高。典型的 EMPU 核有 x86 系列、ARM 系列、PowerPC 系列、MIPS 系列等。

在以 EMPU 为核心的嵌入式系统中, 系统板上除了 EMPU, 还必须设计有 ROM、RAM、总线接口、各种外设控制器等器件, 通常称为单板机。单板机不仅降低了系统的可靠性、技术的保密性等, 而且存在其体积大、功耗大等缺点。

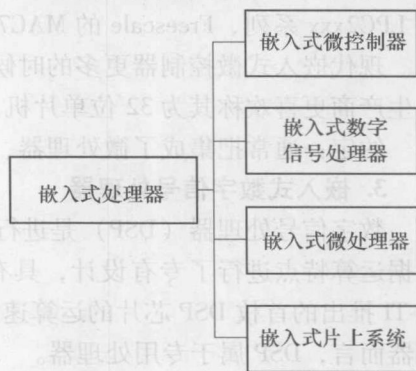


图 1-2 嵌入式处理器分类

约定：通常把集成了处理器核、存储器管理器、高速缓存的芯片称为嵌入式微处理器。

2. 嵌入式微控制器

微电子学和集成电路技术的高速发展为解决单板机的缺陷提供了有效的方法，即把单板机集成到一块集成电路芯片上，实现单板机的单片化，这通常称为单片机。单片机的显著特点是集成了存储器以及丰富的片内外设资源，适合于控制、高可靠性、无人值守等系统。因此与微处理器相对应也称为微控制器，在以后的研究中将不再区分单片机和微控制器。

微控制器的基本特点是单板机的单片化。因此，微控制器必须是以某一种微处理器内核为核心，再在芯片内部集成存储器、总线、各种必要外设控制器等。

在传统的嵌入式微控制器中，通常是以 8 位或 16 位微处理器为主，内部一般都集成了程序存储器（ROM/EPROM/Flash）、数据存储器（RAM）、总线、总线逻辑、定时器/计数器、看门狗、通用 I/O、串行口、脉宽调制输出（PWM）、数模/模数转换器（A-D/D-A）等外设。例如，MCS-51 系列是典型的 8 位微控制器。传统的嵌入式微控制器更多的时候被称为单片机。

在现代嵌入式微控制器中，通常是以 32 位 EMPU 为核心，将众多的外围设备控制器（如存储器控制器 MMU、通用串行口总线控制器 USB、以太网控制器、数模/模数转换器、LCD 控制器、通用 IO 口等）集成到一块芯片中，实现一个定制的“计算”系统。例如，在 ARM 系列中，常见的微控制器有 SAMSUNG 的 S3Cxxx、Atmel 的 AT91SAMxxx 系列、NXP 的 LPC2xxx 系列、Freescale 的 MAC71xx 系列、TI 的 OMAP 系列、Luminary 的 LM3Sxxx 系列等。现代嵌入式微控制器更多的时候被称为应用微处理器、通信微处理器等，也有越来越多的生产商更喜欢称其为 32 位单片机。

约定：通常把集成了微处理器、存储器和片内外设的芯片称为嵌入式微控制器。

3. 嵌入式数字信号处理器

数字信号处理器（DSP）是进行信号处理运算的专用处理器。其体系结构和指令算法都根据运算特点进行了专有设计，具有较高的编译效率和较快的指令执行速度。例如，1982 年 TI 推出的首枚 DSP 芯片的运算速度就比同类微处理器快了几十倍。相对微处理器、微控制器而言，DSP 属于专用处理器。

推动 EDSP 发展的主要因素是嵌入式系统的智能化。例如，各种带有智能逻辑的消费类产品、生物信息识别终端、ADSL 接入、实时语音解压系统、虚拟现实显示等。这类智能化算法一般都是运算量较大，特别是向量运算、指针线性寻址等较多，而这些正是 DSP 的长处所在。

DSP 越来越广泛地应用于各个领域。DSP 应用系统的传统设计可采取主从式结构，在一块电路板上，DSP 作为从机负责数字信号处理运算；EMPU 作为主机来完成输入、控制、显示等其他功能。比如 TI 专门推出的一款双核处理器 OMAP，包含有一个 ARM 核和一个 DSP 核，OMAP 处理器把主从式设计在芯片级上合二为一，其典型的应用实例是诺基亚手机。

4. 嵌入式片上系统

电子系统设计经历了从“板级”到“片级”的过程。对于板级设计，通常需要根据设计要求选一个微处理器或微控制器和多个外设控制器（如 LCD 接口、USB 接口、TCP/IP 接口、3G 接口、蓝牙接口等控制器），然后把所有的集成电路都设计到一块印制电路板上。对于片级设计，就是把微处理器或微控制器和外设控制器都集成到一个集成电路芯片中，在