

嵌入式Linux系统 开发教程 (第2版)

- ◆ 嵌入式系统基础
- ◆ Linux基础
- ◆ ARM体系结构
- ◆ 嵌入式编程
- ◆ 软硬件开发环境
- ◆ 交叉编译工具
- ◆ Bootloader详解及移植
- ◆ 定制内核移植
- ◆ 嵌入式Linux文件系统
- ◆ 驱动程序开发基础
- ◆ 嵌入式Linux图形设计
- ◆ 视频监控系统设计实例



贺丹丹 编著

清华大学出版社

高等学校计算机应用规划教材

嵌入式 Linux 系统 开发教程 (第 2 版)

贺丹丹 编著

清华大学出版社

北 京

内 容 简 介

本书系统论述了在 Linux 环境下开发嵌入式系统的设计思想、设计方法及开发流程,通过实例与设计项目,帮助读者尽快掌握嵌入式系统的基本概念,提高嵌入式设计技能。本书共 12 章,内容包括嵌入式基础知识、Linux 概述、ARM 体系架构、嵌入式编程、交叉工具链、Bootloader、定制内核、文件系统、驱动程序开发基础、嵌入式图形设计等。本书的最后给出了一个综合实例,帮助读者理解嵌入式 Linux 的开发方法和技巧。

本书可作为高校计算机、通信、电子专业相关课程的教材,也可供广大嵌入式开发人员参考。

本书封面贴有清华大学出版社防伪标签,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 Linux 系统开发教程 / 贺丹丹 编著. —2 版. —北京:清华大学出版社, 2014
(高等学校计算机应用规划教材)

ISBN 978-7-302-36504-4

I. ①嵌… II. ①贺… III. ①Linux 操作系统—程序设计—高等学校—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2014)第 102883 号

责任编辑:刘金喜 蔡 娟

装帧设计:孔祥峰

责任校对:成凤进

责任印制:

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

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 刷 者:

装 订 者:

经 销:全国新华书店

开 本:185mm×260mm

印 张:22.5

字 数:562 千字

版 次:2010 年 5 月第 1 版

2014 年 7 月第 2 版

印 次:2014 年 7 月第 1 次印刷

印 数:1~4000

定 价:36.00 元

产品编号:

前 言

嵌入式 Linux 系统由于具有开源、网络功能强大、内核稳定、高效等特性，在产品开发周期、产品的功能可扩充性、开发时的人力投入等方面都具有显著的优势，因此广泛应用于高、中、低端智能电子设备中。而它与 ARM 的结合，更是一种主流的解决方案。嵌入式 Linux+ARM 已经广泛应用于机顶盒、智能手机、平板电脑、MPC(多媒体个人计算机)、网络设备、工业控制等领域，并且具有良好的市场前景。

嵌入式系统是以应用为中心，以计算机技术为基础，采用可裁剪软硬件，适用于对功能、可靠性、成本、体积、功耗等要求严格的专用计算机系统。

在新兴的嵌入式系统产品中，常见的有 MP3、智能手机、平板电脑、数字播放器、GPS、机顶盒、嵌入式服务器、家庭游戏网关、VoIP、PDA、数字视讯录像机及瘦客户机等。嵌入式系统是未来生活的一个基础平台，将会大大影响人们的生活方式。

本书将系统地讲解嵌入式 Linux 开发流程中的各个步骤，详细解析各个流程中的疑点、难点。本书分 3 个部分，共 12 章。各部分内容如下：

第一部分为基础知识篇，主要讲解嵌入式系统与 Linux 相关的基础知识，其中第 1 章为嵌入式系统基础，主要讲解嵌入式相关的概念、历史、应用及前景；第 2 章为 Linux 概论，主要是与 Linux 基础相关的知识，如 Linux 桌面系统、Linux 常用软件的使用及 Linux 常见命令等；第 3 章为 ARM 体系架构，主要介绍 ARM 架构的相关知识，如 ARM 指令集、ARM 处理器基本原理等；第 4 章为嵌入式编程，将简要介绍嵌入式汇编语言及 C 语言的编程基础。

第二部分为开发入门篇，主要介绍嵌入式开发的基本方法，这部分是本书的重点，也是嵌入式 Linux 学习的难点，读者要认真学习。这部分共 5 章，其中第 5 章介绍了嵌入式开发的软硬件环境，如工具的驱动程序安装、Ubuntu 的安装、DNW 的使用、NFS 的配置和使用，以及 Telnet、ftp 等的配置使用；第 6 章主要讲解交叉编译环境的概念以及工具链的编译、获取；第 7 章介绍了 Bootloader 及典型引导程序的制作，如 Vivi；第 8 章讲解内核的定制；第 9 章介绍了嵌入式 Linux 文件系统，这部分内容较多，希望读者重点掌握。

第三部分是提高篇，主要包括第 10 章驱动程序的开发；第 11 章嵌入式 Linux 的图形设计；第 12 章将给出一个开发实例，使读者能系统地了解嵌入式 Linux 的开发过程。

本课程总学时为 54 学时，各章学时分配见下表(供参考)：

学时分配建议表

课 程 内 容	学 时 数			
	合 计	讲 授	实 验	机 动
第 1 章 嵌入式系统基础	1	1		
第 2 章 Linux 基础	3	2	1	
第 3 章 ARM 体系架构	2	2		
第 4 章 嵌入式编程	4	3	1	
第 5 章 软硬件开发环境	4	3	1	
第 6 章 交叉编译工具	3	2	1	
第 7 章 Bootloader 详解及移植	5	3	2	
第 8 章 定制内核移植	3	2	1	
第 9 章 嵌入式 Linux 文件系统	7	4	3	
第 10 章 嵌入式 Linux 驱动程序开发基础	9	6	3	
第 11 章 嵌入式 Linux 图形设计	8	5	3	
第 12 章 嵌入式视频监控系统开发实例	6	4	2	
合 计	54	37	18	

本书内容丰富，实例典型，有很强的针对性。书中各章不仅详细介绍了实例的具体操作步骤，而且还配有一定数量的练习题供读者学习使用。读者只需按照书中介绍的步骤一步步地实际操作，就能完全掌握本书的内容。

尽管本书只讨论如何在嵌入式系统中使用 Linux，但是对想要在嵌入式系统中使用 BSD(伯克利软件发行中心)的开发者来说也会有一些帮助，但本书所作的许多说明都必须依据 BSD 与 Linux 间的差异重新诠释。

本书可作为高等学校计算机、通信、电子等专业嵌入式设计课程的教材，也可供嵌入式开发技术人员参考。

本书 PPT 教学课件可以通过 <http://www.tupwk.com.cn/downpage> 下载。

本书由贺丹丹编著，此外，马建红、许小荣、张泽、刘荣、张璐、王统、王东、周艳丽、刘波、苏静等也参与了本书的编写，在此，同样致以诚挚的谢意！

由于时间仓促及作者水平所限，本书难免有纰漏和不妥之处，敬请广大读者批评指正。

编 者
2013 年 12 月

目 录

第 1 章 嵌入式系统基础	1	2.1.5 Linux 的种类和特性	30
1.1 嵌入式系统	1	2.2 图形操作界面	33
1.1.1 嵌入式系统的概念	1	2.2.1 Linux 与图形界面	34
1.1.2 嵌入式系统的组成	3	2.2.2 KDE	35
1.1.3 嵌入式系统的发展	5	2.2.3 GNOME	37
1.1.4 嵌入式系统的应用前景	8	2.2.4 GNOME与KDE发展趋势	38
1.2 嵌入式处理器	11	2.3 Linux 的基本命令行操作	39
1.2.1 嵌入式微控制器(EMCU)	11	2.3.1 目录操作	40
1.2.2 嵌入式微处理器(EMPU)	12	2.3.2 文件操作	46
1.2.3 嵌入式数字信号 处理器(EDSP)	13	2.3.3 压缩、解压与打包	50
1.2.4 嵌入式片上系统(ESOC)	13	2.3.4 磁盘管理	51
1.3 嵌入式操作系统	13	2.3.5 用户系统	53
1.3.1 Linux	15	2.3.6 网络管理	55
1.3.2 VxWorks	15	2.4 Linux 内核	58
1.3.3 WinCE	16	思考与练习	59
1.3.4 μ C/OS-II	16	第 3 章 ARM 体系架构	60
1.3.5 eCOS	17	3.1 ARM 微处理器简介	60
1.3.6 Android	17	3.1.1 ARM 微处理器的发展	60
1.3.7 iOS	18	3.1.2 ARM 微处理器的 特点和应用	61
1.3.8 WP 和 Windows RT	18	3.2 ARM 微处理器系列	62
1.4 嵌入式系统设计	18	3.2.1 Classic(传统)系列	62
1.4.1 嵌入式系统开发流程	18	3.2.2 Cortex-M 系列	63
1.4.2 嵌入式系统开发方法	19	3.2.3 Cortex-R 系列	63
思考与练习	20	3.2.4 Cortex-A 系列	64
第 2 章 Linux 基础	22	3.2.5 Cortex-A50 系列	65
2.1 Linux 简介	22	3.3 ARM 编程模型	65
2.1.1 Linux 的历史	23	3.3.1 ARM 硬件架构	65
2.1.2 Linux 特点	23	3.3.2 ARM 微处理器模式	66
2.1.3 Linux 与 Windows	25	3.3.3 ARM 寄存器	67
2.1.4 Linux 的主要组成部分	27	3.3.4 异常处理	68

3.3.5 ARM 的存储器组织.....	72	思考与练习	117
3.4 ARM 指令系统.....	74	第 5 章 软硬件开发环境	119
3.4.1 ARM 指令格式	74	5.1 硬件环境	119
3.4.2 ARM 指令的寻址方式	75	5.1.1 主机硬件环境.....	119
3.4.3 ARM 最常用指令和 条件后缀.....	77	5.1.2 目标板硬件环境.....	120
3.5 ARM 微处理器的应用选型	79	5.2 Windows 软件环境.....	122
思考与练习.....	80	5.2.1 超级终端的设置.....	122
第 4 章 嵌入式编程	82	5.2.2 DNW 的设置.....	123
4.1 ARM 汇编语言程序设计.....	82	5.2.3 设置 GIVEIO 驱动	126
4.1.1 ARM 汇编语言中的 程序结构.....	82	5.3 Linux 软件环境	128
4.1.2 ARM 汇编语言的语句 格式.....	83	5.3.1 Linux 系统的 VMware 安装	129
4.1.3 基于 Linux 下 GCC 的 汇编语言程序结构.....	84	5.3.2 Windows 与 Ubuntu 双 系统安装.....	135
4.1.4 基于 Windows 下 ADS 的 汇编语言程序结构.....	85	5.3.3 Linux 网络服务配置	137
4.1.5 ARM 汇编器所支持的 伪指令.....	86	5.3.4 配置 NFS 服务.....	138
4.2 ARM 汇编与 C 语言编程.....	90	5.3.5 配置 FTP 服务	140
4.2.1 基本的 APCS 规则	91	5.4 刻录镜像文件.....	141
4.2.2 C 语言中内嵌汇编代码.....	93	5.4.1 刻录工具.....	142
4.2.3 从汇编程序中访问 C 程序变量.....	94	5.4.2 使用方法.....	142
4.2.4 在汇编代码中调用 C 函数.....	95	思考与练习	143
4.2.5 在 C 语言代码中调用 汇编函数.....	98	第 6 章 交叉编译工具	145
4.3 基于 Linux 的 C 语言编程.....	99	6.1 工具链软件.....	145
4.3.1 C 语言编程概述.....	99	6.1.1 工具链组成	145
4.3.2 Linux 下的 C 开发工具.....	99	6.1.2 构建工具链	146
4.3.3 vim 编辑器	100	6.2 分步构建交叉编译链.....	147
4.3.4 gedit 编辑器.....	106	6.2.1 准备工具.....	147
4.3.5 编译器 gcc.....	107	6.2.2 基本过程.....	147
4.3.6 调试器 gdb	111	6.2.3 详细步骤.....	148
4.3.7 项目管理器 make	114	6.3 用 Crosstool 工具构建交叉 工具链	155
		6.3.1 准备工具.....	155
		6.3.2 基本过程.....	155
		6.3.3 详细步骤.....	156
		6.4 使用现成的交叉工具.....	159

思考与练习	160	8.4.5 取消不需要的文件系统的 支持	214
第 7 章 Bootloader 详解及移植	162	思考与练习	214
7.1 嵌入式 Bootloader 简介	162	第 9 章 嵌入式 Linux 文件系统	216
7.1.1 Bootloader 功能	162	9.1 嵌入式 Linux 的文件系统	216
7.1.2 基于 Bootloader 软件架构	163	9.1.1 文件系统结构	216
7.1.3 Bootloader 的操作模式	164	9.1.2 文件系统特性	217
7.1.4 Bootloader 的依赖性	164	9.1.3 系统存储设备及其 管理机制	218
7.1.5 Bootloader 的启动方式	164	9.1.4 基于 Flash 闪存的 文件系统	219
7.1.6 Bootloader 启动流程	167	9.1.5 基于 RAM 的文件系统	221
7.1.7 各种 Bootloader	168	9.1.6 网络文件系统	222
7.2 Vivi	169	9.2 根文件系统及其定制	223
7.2.1 Vivi 简介	169	9.2.1 根文件系统架构	223
7.2.2 Vivi 体系架构	169	9.2.2 定制工具 Busybox	225
7.2.3 Vivi 的运行过程分析	170	9.2.3 库文件构建	233
7.2.4 Vivi 的配置与编译	183	9.2.4 设备文件的构建	235
7.2.5 Vivi 命令	185	9.2.5 根文件系统初始化	236
7.3 Bootloader 程序的调试和 刻录	187	9.3 文件系统的制作	239
思考与练习	188	9.3.1 根文件系统的制作	239
第 8 章 定制内核移植	189	9.3.2 NFS 文件系统的制作	245
8.1 Linux 内核源码组织	189	9.3.3 Cramfs 文件系统的制作	247
8.2 内核基本配置	191	9.3.4 Yaffs 文件系统的制作	249
8.2.1 内核配置系统	191	9.3.5 Ramdisk 文件系统的制作	250
8.2.2 Makefile	192	思考与练习	253
8.2.3 具体的配置操作	197	第 10 章 嵌入式 Linux 驱动程序 开发基础	255
8.2.4 添加自己的代码	201	10.1 嵌入式 Linux 驱动程序 概述	255
8.3 内核定制	204	10.1.1 Linux 驱动程序 工作原理	255
8.3.1 获取源码	204	10.1.2 Linux 驱动程序功能	257
8.3.2 移植过程	205	10.2 设备驱动程序的基础知识	257
8.4 内核裁剪	212	10.2.1 Linux 的设备管理机制	257
8.4.1 取消虚拟内存的支持	212	10.2.2 驱动层次结构	261
8.4.2 取消多余的调度器	212		
8.4.3 取消对旧版本二进制 执行文件的支持	213		
8.4.4 取消不必要的设备的支持	213		

10.2.3	设备驱动程序与 外界的接口	262	11.2.4	Qt 的支撑工具及组件	307
10.2.4	设备驱动程序的特点	262	11.2.5	Qt/Embedded 对象 模型	307
10.2.5	驱动程序开发流程	263	11.2.6	信号与插槽机制	309
10.3	模块编程	263	11.2.7	Qt/Embedded 常用的类	312
10.3.1	模块与内核	263	11.3	安装 Qt/Embedded	314
10.3.2	建立模块文件	264	11.3.1	配置	314
10.3.3	编写 makefile	265	11.3.2	编译	315
10.3.4	模块加载	266	11.3.3	测试	315
10.3.5	模块的其他信息	267	11.4	Qt 设计实例——密码 验证程序	315
10.3.6	模块参数	267	11.4.1	快速安装 QDevelop 和 Qt Designer	316
10.4	字符设备驱动程序	268	11.4.2	界面设计	317
10.4.1	相关的数据结构	268	11.4.3	信号与槽	319
10.4.2	字符设备驱动程序 开发流程	274	11.4.4	添加代码	320
10.4.3	字符设备驱动程序 扩展操作	283	11.4.5	编译	323
10.5	网络设备驱动程序	285	11.4.6	程序测试	324
10.5.1	基本概念	285	11.4.7	移植	324
10.5.2	网络数据包处理流程	290	思考与练习		325
	思考与练习	292			
第 11 章	嵌入式 Linux 图形设计	294	第 12 章	嵌入式视频监视系统 开发实例	327
11.1	嵌入式 GUI	294	12.1	系统设计背景	327
11.1.1	嵌入式 GUI 简介	294	12.2	系统总体设计	328
11.1.2	嵌入式 GUI 需求	295	12.2.1	系统总体设计思路	328
11.1.3	嵌入式 GUI 组成	296	12.2.2	系统的设计要求及 特点	328
11.1.4	Qt/Embedded	297	12.2.3	系统总体架构设计	328
11.1.5	MiniGUI	298	12.3	系统详细设计	330
11.1.6	MicroWindows	300	12.3.1	系统的硬件设计与 调试	330
11.1.7	OpenGUI	302	12.3.2	系统的软件设计与 调试	333
11.1.8	Tiny-X	302	12.3.3	USB 数据输入驱动 程序移植	341
11.1.9	各种 GUI 比较	303			
11.2	Qt/Embedded 开发入门	303			
11.2.1	Qt/Embedded 简介	303			
11.2.2	Qt/Embedded 架构	304			
11.2.3	Qt 的开发环境	306			

12.3.4	USB 摄像头数据输入		12.4	系统测试	345
	驱动程序测试	342	12.4.1	准备工作	345
12.3.5	嵌入式网络视频		12.4.2	测试方法	346
	服务器的设计	343	12.4.3	测试结果	346
12.3.6	Video4Linux 程序设计	344			

第1章 嵌入式系统基础

嵌入式系统是近年来随着电子芯片技术的发展而迅速普及的，现已广泛应用到工业、军事、通信、运输、金融、农业、医疗、气象等众多领域。在我们日常生活中，嵌入式系统的应用也是随处可见：汽车里的控制器、电梯、MP3/MP4、智能手机、平板电脑等。当前，嵌入式系统开发领域聚集着无数IT领域的精英，正是他们在嵌入式领域开展的大量艰辛工作，嵌入式系统才会不断向前发展。作为本书的开篇，本章将主要介绍有关嵌入式系统的基础知识。本章从嵌入式系统的基本概念开始，介绍其组成、发展以及应用前景。接着介绍嵌入式处理器，它涵盖了嵌入式微控制器、嵌入式微处理器、嵌入式DSP处理器以及嵌入式片上系统(System On Chip)。之后简要介绍常见的嵌入式操作系统Linux、VxWorks、Windows CE、RT-Linux、Android(安卓)、iOS(iPhone/iPad Operating System)、WP(Windows Phone)、Windows RT等。最后介绍嵌入式系统开发流程、嵌入式系统开发方法等知识。

本章重点：

- 嵌入式系统的概念及发展
- 嵌入式处理器
- 嵌入式操作系统
- 嵌入式系统的应用

1.1 嵌入式系统

嵌入式系统主要融合了计算机软硬件技术、通信技术和微电子技术，它将计算机直接嵌入到应用系统中，利用计算机的高速处理能力以实现某些特定的功能。随着半导体技术和微电子技术的发展，超大规模集成电路制造工艺已经十分成熟，使得系统芯片的集成度大大提高，从而可以实现高性能的系统芯片，最终推动嵌入式系统向更高级的方向发展，进而促使嵌入式系统得到更广泛、更深入的应用。近几年来，嵌入式系统的快速发展，一方面使其成为计算机技术和微电子技术的一个重要研究方向，另一方面也使计算机类别的划分从以前的巨型机、大型机、小型机、微型机变为通用计算机(General Computer)和嵌入式计算机(Embedded Computer)。

1.1.1 嵌入式系统的概念

首先介绍通用计算机，通用计算机就是日常所说的PC机，它由CPU、硬盘、显卡、内存、键盘、鼠标、显示器等组成。它的作用就是为人们提供一台可编程、会计算、能处理数据的机器。人们可以用它作为科学计算的工具，也可以用它作为企业管理的工具。所以，人们把

这样的计算机系统称为“通用”计算机系统。

有些系统并不是这样的。例如，医用的CT扫描仪也是一个系统，里面有计算机，但是这种计算机(或处理器)是某个专用系统中的一个部件。像这样“嵌入”到更大的、专用的系统中的计算机系统，就被称为“嵌入式计算机”、“嵌入式计算机系统”或“嵌入式系统”。在不致引起混淆的情况下，一般把这三者用作同义词，并且一般总是指系统中的核心部分，即嵌入在系统中的计算机。从字面上讲，后者似乎比前者更为广义，因为系统中常常还包括一些机电、光电、热电或者电化的执行部件；有时前者也包含后者，例如一台通用计算机的外部设备中就包含5~10个嵌入式微处理器，键盘、鼠标、软驱、硬盘、显示卡、显示器、Modem、网卡、声卡、打印机、扫描仪、数码相机和USB集线器等均是由嵌入式处理器进行控制的，但是实际上却往往不作严格的区分。

目前嵌入式系统已经渗透到我们生活的每一个角落：工业控制、服务行业、消费电子、教育等，正是由于嵌入式系统的应用范围如此之大，“嵌入式系统”的概念才更加难以定义。举个简单的例子来说：一个消费级的数码相机是否可以叫做嵌入式系统呢？答案是肯定的，它本质上就是一个复杂的嵌入式系统，其实不光是数码相机，我们日常生活中经常用的智能手机、MP3/MP4、PSP、平板电脑也都是嵌入式系统，如图1-1所示。除了这些日常消费品之外，你认为一个PC104的微型工业控制计算机是嵌入式系统吗？当然也是，工业控制是嵌入式系统技术的一个典型应用领域。然而对两者进行比较，你也许会发现两者几乎完全不同，但其中都嵌入有微处理器，由此可以看出所有的嵌入式系统都具有一些共同的特性。

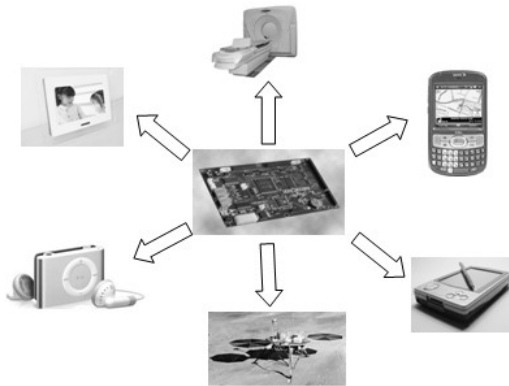


图1-1 嵌入式系统的应用

根据国际电机工程师协会(IEEE)的定义，嵌入式系统是“控制、监视或者辅助装置、机器和运行的装置”，这主要是从应用层次上来定义的，从中可以看出嵌入式系统包括硬件和软件两个载体，另外还可以涵盖机械等附属装置。为了充分体现嵌入式系统的精髓，目前国内普遍认同的一个定义是：以应用为中心，以计算机技术为基础，软硬件可裁剪，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。关于这个定义，可以从以下几个方面来理解。

- 嵌入式系统是面向产品、面向用户、面向应用的。它必须结合实际的应用场合才能发挥其优势。对于三个面向的理解，可以认为嵌入式系统具有很强的专业性，必须结合实际系统需求在软硬件方面进行合理的裁剪。

- 嵌入式系统是一个技术密集、集成度高、需要不断创新的集成系统。嵌入式系统结合了计算机技术、半导体技术、微电子技术以及各个行业的具体专业应用知识，所以，嵌入式系统在设计之前必须有一个正确的定位。例如iOS和Android就是因为其立足于个人电子消费品市场，而在平板电脑领域占有绝对优势的市场份额；而VxWorks之所以在火星探测器上得到应用，则是因为其高实时性和高可靠性。
- 嵌入式系统必须根据应用场合对软硬件进行必要的裁剪来实现需要的功能。对于不同的应用场合，系统的硬件和软件需求一般都是不同的。设计开发需要的软硬件，去除不需要的资源也是使系统满足功能、可靠性、体积、成本所要求的。所以，在相对通用的软硬件基础上，对其开发出适用于不同应用场合的系统，是嵌入式系统的一般发展模式。

1.1.2 嵌入式系统的组成

嵌入式系统一般由嵌入式计算机和执行部件组成，如图1-2所示。其中嵌入式计算机是整个嵌入式系统的核心，主要包括硬件层、中间层、系统软件层以及应用软件层；执行部件则接收嵌入式计算机系统发出的控制指令，执行规定的操作，也被称做被控对象。比较简单的执行部件比如电机、显示屏、扬声器等，比较复杂的如SONY智能机器狗，上面集成多个微型控制电机和传感器，从而可以执行不同的操作和感知各种状态信息。下面主要对嵌入式计算机部分的组成做主要介绍。

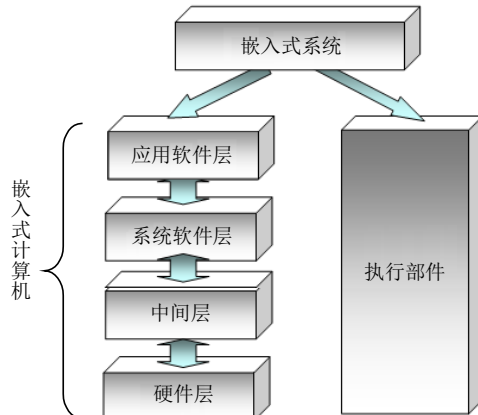


图1-2 嵌入式系统组成

1. 硬件层

硬件层主要包含了嵌入式系统中必要的硬件设备：嵌入式微处理器和协处理器、存储器(SDRAM、ROM等)、设备IO接口等。

- 嵌入式微处理器是嵌入式系统硬件层的核心，主要负责对信息的运算处理，相当于通用计算机中的中央处理器，在后面会有更详细的介绍。
- 存储器则用来存储数据和代码。嵌入式系统的存储器一般包括微处理器、Cache、主存储器和辅助处理器，存储器结构如图1-3所示。

- ◆ Cache是一种容量小、速度快的存储器阵列，它位于主存储器和微处理器内核之间，存放的是最近一段时间微处理器使用最多的程序代码和数据。在需要进行数据读取操作时，微处理器尽可能地从Cache中读取数据，而不是从主存储器中读取，这样就大大改善了系统的性能，提高了微处理器和主存储器之间的数据传输速率。

在嵌入式系统中，Cache全部集成在嵌入式微处理器内，可分为数据Cache、指令Cache和混合Cache，Cache的大小依不同处理器而定。

- ◆ 主存储器是嵌入式微处理器能直接访问的寄存器，用来存放系统和用户的程序及数据。它可以位于微处理器的内部或外部，其容量为256KB~4GB，根据具体的应用而定。一般片内存储器容量小，速度快，片外存储器容量大。

注意：

常用作主存的存储器有如下几类。

- ROM类：NOR Flash、EPROM和PROM等。
- RAM类：SRAM、DRAM和SDRAM等。

其中NOR Flash凭借其可擦写次数多、存储速度快、存储容量大、价格便宜等优点，在嵌入式领域内得到了广泛应用。

- ◆ 辅助处理器用来存放大数据量的程序代码或信息，它的容量大，但读取速度与主存储器相比却慢很多，用来长期保存用户的信息。

嵌入式系统中常用的外存有：硬盘、NAND Flash、CF卡、MMC和SD卡等。

- 设备IO接口提供了系统与内部或者外部的硬件接口，如通过IO实现内部A/D转换，或者通过IO与外部的存储器连接进行存储扩展。

除此之外，有的嵌入式控制模块中还集成有电源电路、时钟电路、定时器等以实现更高级的功能。

2. 中间层

中间层为硬件层与系统软件层之间的部分，有时也称为硬件抽象层(Hardware Abstract Layer, HAL)或者板级支持包(Board Support Package, BSP)。对于上层的软件(比如操作系统)，中间层提供了操作和控制硬件的方法和规则。而对于底层的硬件，中间层主要负责相关硬件设备的驱动等。中间层将系统上层软件与底层硬件分离开来，使系统的底层驱动程序与硬件无关，上层软件开发人员无需关心底层硬件的具体情况，根据中间层提供的接口即可进行开发。

中间层主要包含以下操作：底层硬件初始化、硬件设备配置以及相关的设备驱动。

- 底层硬件初始化操作按照自底而上、从硬件到软件的次序分为三个环节，依次是：

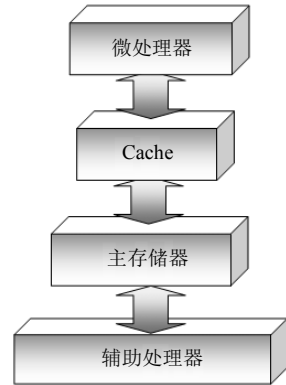


图1-3 嵌入式系统存储器结构

片级初始化、板级初始化和系统级初始化。

- 硬件设备配置对相关系统的硬件参数进行合理的控制以达到正常工作。另一个主要功能是硬件相关的设备驱动。
- 硬件相关的设备驱动程序的初始化通常是一个从高到低的过程。尽管中间层中包含硬件相关的设备驱动程序，但是这些设备驱动程序通常不直接由中间层使用，而是在系统初始化过程中由中间层将它们与操作系统中通用的设备驱动程序关联起来，并在随后的应用中由通用的设备驱动程序调用，实现对硬件设备的操作。

3. 系统软件层

系统软件层由实时多任务操作系统(Real-time Operation System, RTOS)、文件系统、图形用户界面接口(Graphic User Interface, GUI)、网络系统及通用组件模块组成。其中实时多任务操作系统(RTOS)是整个嵌入式系统开发的软件基础和平台。

4. 应用软件层

应用软件层则是开发设计人员在系统软件层的基础之上，根据需要实现的功能，结合系统的硬件环境所开发的软件。它是嵌入式系统开发过程中最重要的环节之一。

1.1.3 嵌入式系统的发展

虽然嵌入式系统是最近几年才流行起来的，但是嵌入式系统这个概念却早在20世纪70年代就开始出现。嵌入式系统的硬件雏形可以归结为单片机。从当时第一个单片机的问世到今天成百上千种嵌入式处理器的出现，嵌入式系统已经有了超过40年的发展历史。一个刚问世的处理系统，一般要在基于硬件与软件双螺旋的支撑下才能逐渐趋于稳定和成熟，嵌入式系统也不例外。

1. 20世纪70年代

1976年，计算机硬件厂商Intel公司发布8048处理器，它是最早的单片机处理芯片。随后Motorola公司推出了68HC05，Zilog公司推出了Z80等一系列的单片机，这些早期单片机系统的出现，使得汽车、家电、工业机器、通信装置以及成千上万种产品可以通过内嵌电子装置来获得更佳的使用性能、更容易使用、处理速度更快、价格更便宜。正是由于电子装置是“内嵌式的”，使得“嵌入式系统”这个初级概念深入人心。今天看来，当时这些装置已经初步具备了嵌入式的应用特点，但是这时的应用只是使用8位的芯片，硬件技术相对落后，比如说只能执行一些单线程的程序，还谈不上“多核”的概念。但是它却标志着“嵌入式系统”出现了硬件雏形。在开创嵌入式系统独立发展的道路上，Intel公司功不可没。在寻求最佳形态嵌入式系统的体系结构中，奠定了单片微型计算机(Sing Chip Microcomputer, SCM)与通用计算机完全不同的发展道路。

不过，人们当时还没有“嵌入式操作系统”这个概念。他们当时做的只是基于8048或者Z80结合实际应用的需要，在DOS或者其他“操作系统”下进行开发设计工作。不像今天硬件系统的多平台开发，当时针对硬件芯片，可能只有一种系统可以开发，而且系统的稳定

性、兼容性也十分差。

2. 20世纪80年代

在20世纪80年代初,通过几年的设计实践,Intel进一步完善了8048,推出了在它的基础上研制成功的8051。这在单片机的历史上是重要的一款,具有划时代的意义。另一方面,硬件技术的逐步发展也推动了整个嵌入式系统的进步。如图1-4和图1-5所示,在各种嵌入式产品中有被广泛使用的Motorola Z80和Intel 51芯片。



图1-4 Z80系列单片机



图1-5 51系列单片机

在此基础上,各大电子厂商不断扩展满足嵌入式应用,如对系统要求的各种外围电路和接口电路,以突显其智能化控制能力。因此,发展微控制器(Micro Control Unit, MCU)的重任不可避免地落在电气、电子技术厂家,比如知名的Philips公司。

在软件方面,从20世纪80年代早期开始,嵌入式系统的程序员开始用商业级的“操作系统”编写嵌入式应用软件,这使得可以获得更短的开发周期、更低的开发资金和更高的开发效率,“嵌入式系统”真正出现了。确切地说,这个时候的操作系统是一个实时核,这个实时核包含了许多传统操作系统的特征,包括任务管理、任务间通信、同步与相互排斥、中断支持、内存管理等功能。其中比较著名的有Integrated System Incorporation(ISI)的PSOS、Ready System公司的VRTX、IMG的VxWorks和QNX公司的QNX等。这些嵌入式操作系统都具有嵌入式的典型特征。

- 它们的系统内核很小,具有可裁剪、可扩充和可移植性,可以移植到各种各样的处理器芯片上。
- 它们均采用占先式的调度,响应的时间很短,任务执行的时间可以确定。
- 较强的实时和可靠性,适合嵌入式应用。
- 这些嵌入式实时多任务操作系统的出现,使得应用开发人员得以从小范围的开发解放出来,同时也促使嵌入式有了更为广阔的应用空间。

3. 20世纪90年代

Philips公司以其在嵌入式应用方面的巨大优势,将51系列单片微型计算机迅速发展到了微控制器。总的来说,单片机是嵌入式系统的独立发展之路,也是向MCU(微控制器)阶段发展的重要因素。嵌入式系统微控制器的设计过程主要是寻求应用系统在芯片上的最大化解决。

因此,专用单片机的发展自然形成了片上系统(System-on-Chip, SoC)化趋势。随着微电子技术、IC设计、EDA工具的发展,基于SoC的单片机应用系统设计会有较大的发展,如图

1-6所示。据不完全统计,全世界嵌入式处理器的品种总量已经达到1000多种,主流的有ARM、MCU、DSP、FPGA等。嵌入式开发设计过程也细分为多个具体的技术:电源技术、传感技术、信号处理与转换、控制技术、显示驱动、无线网络、音视频技术和接口等。



图1-6 32位嵌入式微处理器SoC芯片系列

嵌入式操作系统的发展已进入成熟时期,此时出现了众多嵌入式操作系统,它们大多具有跨平台的移植技术,并且在同一个系统之下也可以通过选择开发工具使用Java、C或者汇编语言等开发者熟悉的语言来开发,该阶段比较常用的有WinCE、WM、Linux、VxWorks、 μ C/OS-II、Symbian等。

4. 21世纪到现在

进入21世纪之后,随着相关电子工业技术的发展,嵌入式处理器的相关技术得到了突飞猛进的发展,出现了64位的嵌入式处理器(例如Cortex-A50系列),其处理器内核也已经实现了8核(目前正计划实现16核)。

到目前为止,嵌入式处理器可以分为三个大类:以MTK、高通、三星为代表支持的ARM架构处理器、以Intel为代表支持的x86架构处理器以及其他以FPGA为代表的特殊/专用处理器,如图1-7所示。

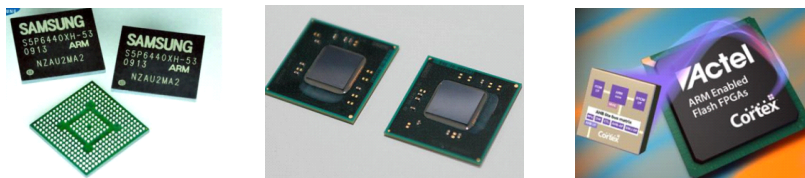


图1-7 进入21世纪之后的嵌入式处理器

随着嵌入式处理器的发展,嵌入式系统的硬件性能得到了极大的提升,此时嵌入式操作系统也开始出现一些新的面孔,Android和iOS则是其中的典型代表,它们从2007年开始(Android于2007年11月正式发布,iOS则是在2007年1月正式发布)就风卷残云般地占领了绝大多数嵌入式消费电子产品(主要是平板电脑、手机和数字播放器)的市场;而微软公司不甘落后,从2010年开始连续发布了WP(Windows Phone)和Windows RT(Run Time)操作系统用于抢

占消费电子产品市场。而在工业控制等领域中，嵌入式操作系统本着稳定可靠的原则，则依然是WinCE、VxWorks和Linux当道。

1.1.4 嵌入式系统的应用前景

嵌入式系统技术有着非常广泛的应用前景，其领域包括如下几项。

1. 工业控制

相对于其他领域，机电产品可以说是嵌入式系统应用最典型最广泛的领域之一。从最初的单片机到现在的工控机，SoC在各种机电产品中均有着巨大的市场。

工业设备是机电产品中最大的一类，在目前的工业控制设备中，工控机的使用非常广泛，这些工控机一般采用的是工业级的处理器和各种设备，其中以x86的微处理器最多。工控的要求往往较高，需要各种各样的设备接口，除了进行实时控制，还需将设备状态、传感器的信息等在显示屏上实时显示。8位的单片机是无法满足这些要求的，以前多数使用16位的处理器。随着处理器的快速发展，目前32位、64位的处理器逐渐替代了16位处理器，进一步提升了系统性能。采用PCI04总线的系统，体积小，稳定可靠，受到了很多用户的青睐。不过这些工控机采用的往往是DOS或者Windows系统，虽然具有嵌入式的特点，却不能称作纯粹的嵌入式系统。另外，在工业控制器和设备控制器方面，则是各种嵌入式处理器的天下。这些控制器往往采用16位以上的处理器，各种MCU、ARM、MIPS、68K系列的处理器在控制器中占据核心地位。这些处理器提供丰富的接口总线资源，可以通过它们实现数据采集、数据处理、通信以及显示(一般是连接LED或者LCD来显示)，现代化的工业控制网络如图1-8所示。

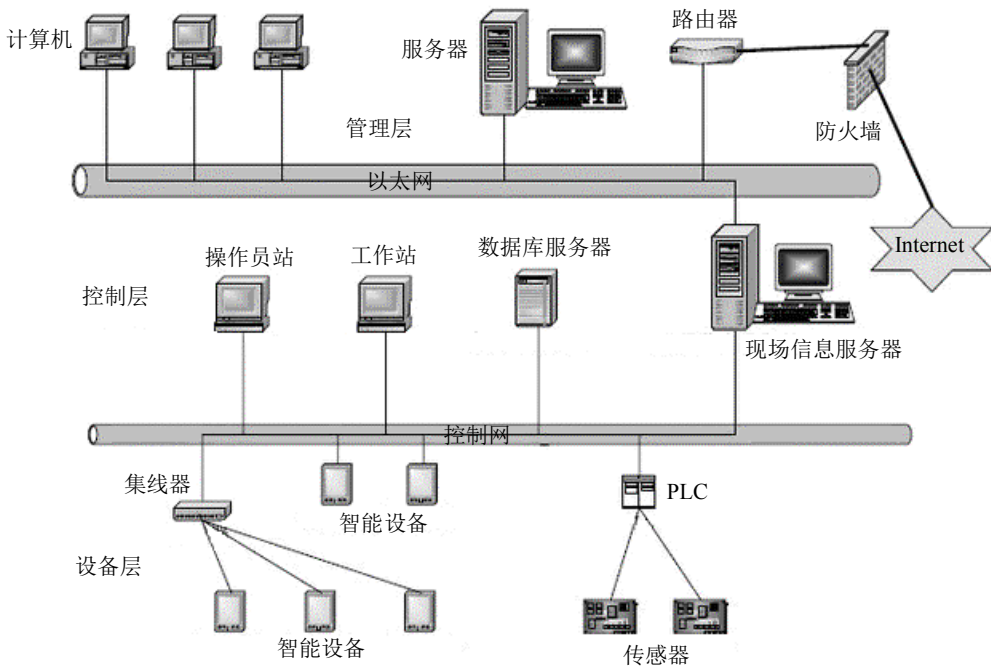


图1-8 工业控制网络

飞利浦公司和ARM公司曾经共同推出了32位RISC嵌入式控制器，适用于工业控制，其

采用最先进的0.18微米CMOS嵌入式闪存处理技术，操作电压可以低至1.2伏，还能降低25%到30%的制造成本，在工业领域中对最终用户而言是一套极具成本效益的解决方案。美国TERN工业控制器基于Am188/186ES、i386EX、NEC V25.Am586(Elan SC520)，采用了SUPERTASK实时多任务内核，可应用于便携设备、无线控制设备、数据采集设备、工业控制与工业自动化设备以及其他需要控制处理的设备。

2. 信息家电

家电行业是嵌入式应用的另一大行业，电视、电冰箱中也嵌有处理器，但是这些处理器只是在控制方面应用。而现在只有按钮、开关的电器显然已经不能满足人们的日常需求。具有用户界面，能远程控制，智能管理的电器是未来的发展趋势，如图1-9所示。

美国国际数据集团(International Data Group, IDG)发布的统计数据表明，未来信息家电将会五至十倍地增长。中国的传统家电厂商向信息家电过渡时，首先面临的挑战是核心操作系统软件开发工作。硬件方面，进行智能信息控制并不是很高的要求，目前绝大多数嵌入式处理器都可以满足硬件要求，真正的难点是如何使软件操作系统容量小、稳定性高且易于开发。在这一方面，Linux核心可以起到很好的桥梁作用，作为一个跨平台的操作系统，它可以支持二三十种CPU，而目前已有众多家电业的芯片都开始做Linux的平台移植工作。



图1-9 智能化家电

20世纪90年代，在各大厂商努力推出适用于新一代家电应用的芯片时，英特尔公司已专门为信息家电业研发了名为StrongARM的ARM CPU系列，这一系列CPU本身不像x86 CPU需要整合不同的芯片组，它在一块芯片中可以包括人们所需要的各项功能，即硬件系统实现了SoC的概念。美商网虎公司已将全球最小的嵌入式操作系统QUARK成功移植到StrongARM系列芯片上，这是第一次把Linux、图形界面和一些程序进行完整移植(QUARK的内核只有143KB)，它将为信息家电提供功能强大的核心操作系统。相信在不久的将来，数字智能家庭必将来到我们身边。

进入21世纪以后，随着ARM体系架构的处理器推陈出新，及Android和iOS操作系统的出现，以智能电视为代表的信息家电逐步走向了成熟化，内置了流媒体播放等应用的小米电视、乐视电视以及各种多媒体播放器(例如Apple TV)现在已经占据了客厅。

3. 机器人

嵌入式芯片的发展将使机器人在微型化、高智能方面的优势体现得更加明显，同时会大幅度降低机器人的价格，使其在工业领域和服务领域获得更广泛的应用。

机器人技术的发展从来就是与嵌入式系统的发展紧密联系在一起。最早的机器人技术是20世纪50年代MIT提出的数控技术，当时使用的还远未达到芯片水平，只是简单的与非门逻辑电路。之后由于处理器和智能控制理论的发展缓慢，从20世纪50年代到70年代初期，机器人技术一直未能获得充分的发展。70年代中期之后，由于智能理论的发展和MCU出现，机器人逐渐成为研究热点，并且获得了长足的发展。

近来由于嵌入式处理器的高度发展，机器人从硬件到软件也呈现了新的发展趋势，如图1-10所示。火星车就是一个典型例子，这个价值10亿美元的技术密集移动机器人，采用的是美国风河公司的VxWorks嵌入式操作系统，可以在不与地球联系的情况下自主工作。如图1-11所示，1997年美国发射的“索杰纳”火星车带有机械手，可以采集火星上的各种地况，并且通过摄像头把火星上的图像发回地面指挥中心。这台火星车在火星上自主工作了3个月，充分体现了VxWorks系统的高可靠性。以索尼的机器狗为代表的智能机器宠物，可以仅仅使用8位的AVR、51单片机或者16位的DSP来控制舵机，进行图像处理，即可制造出那些人见人爱的玩具，这让我们不能不惊叹嵌入式处理器功能的强大。

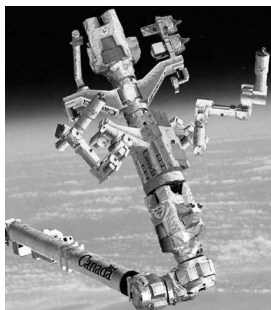


图1-10 美国宇航局太空机器人DEXTRE



图1-11 火星车

近来，32位处理器、Windows CE等32位嵌入式操作系统的盛行，使得操控一个机器人只需要在手持PDA上获取远程机器人的信息，并且通过无线通信控制机器人的运行。与传统的采用工控机相比，要轻巧便捷得多。随着嵌入式控制器越来越微型化、功能化，微型机器人、特种机器人等也将获得更大的发展机遇。

4. POS网络及电子商务

许多公共设施，例如公共交通无接触智能卡(Contactless Smart card, CSC)发行系统、公共电话卡发行系统、自动售货机，各种智能ATM终端将全面走入人们的生活，到时手持一卡就可以走遍天下。

5. 家庭智能管理系统

水、电、煤气表的远程自动抄表、安全防火、防盗系统，其中嵌有的专用控制芯片将代替传统的人工检查，并实现更高、更准确和更安全的性能。目前在服务领域，远程点菜器等

已经体现了嵌入式系统的优势。

6. 交通管理

嵌入式系统技术已经广泛应用于车辆导航、流量控制、信息监测与汽车服务等方面。通过内嵌GPS模块, GSM模块的移动定位终端已经在各种运输行业获得了成功的使用。目前GPS设备已经从尖端产品进入了普通百姓的家庭, 人们可以随时随地找到自己所处的位置。

7. 环境工程与自然

在很多环境恶劣, 地况复杂的地区, 嵌入式系统将实现无人监测。如水文资料实时监测, 防洪体系及水土质量监测、堤坝安全, 地震监测网, 实时气象信息网, 水源和空气污染监测。

在所有这些应用中, 实现网络化控制还需要一个统一的协议体系。就远程家电控制而言, 除了开发出支持TCP/IP的嵌入式系统之外, 家电产品控制协议也需要制订和统一, 这需要家电生产厂家来做。同样的道理, 所有基于网络的远程控制器件都需要通过相关的控制协议与嵌入式系统之间实现接口, 然后再由嵌入式系统来控制并通过网络实现控制。所以, 开发和探讨嵌入式系统的控制协议有着十分重要的意义。

8. 消费电子产品

消费电子产品包括智能手机、平板电脑、手持媒体播放器、家庭/手持游戏机、数码相机等, 嵌入式系统是它们的核心技术, 这些产品目前已经占据了我们生活的方方面面, 成为密不可分的生活必需品。

1.2 嵌入式处理器

嵌入式系统的核心模块就是各种类型的嵌入式处理器。目前几乎每个半导体制造商都生产嵌入式处理器, 越来越多的公司拥有自己的处理器设计部门。嵌入式微处理器的体系结构经历了从CISC到RISC和Compact RISC的转变; 位数由4位、8位、16位、32位到64位; 寻址空间一般为64KB~16MB, 处理速度为0.1MIPS~2000MIPS; 常用的封装为8~144个引脚。根据其现状, 嵌入式处理器可以分为嵌入式微控制器(Embedded Micro Controller Unit, EMCU)、嵌入式微处理器(Embedded Micro Processor Unit, EMPU)、嵌入式数字信号处理器(Embedded Digital Signal Processor, EDSP)和嵌入式片上系统(Embedded System On Chip, ESOC)四类。

1.2.1 嵌入式微控制器(EMCU)

嵌入式微控制器又称单片机, 也就是在一块芯片中集成了整个计算机系统。嵌入式微控制器一般以某种微处理器内核作为核心, 芯片内部集成ROM/EPROM、EEPROM、Flash、RAM、总线、总线逻辑、定时/计数器、WatchDog、I/O口、脉宽调制输出、A/D和D/A等各种必要功能和外设。微控制器由于比微处理器体积小, 功耗和成本低, 可靠性高, 因而是目前嵌入式工业的主流, 品种和数量都很多。其中, 比较具有代表性的通用系列有8051. P51XA、

MCS-251、MCS-96/196/296、MC68HC05/11/12/16和C166/167等。另外还有许多半通用系列，如支持USB接口的MCU 8XC930/931、C540和C541；支持CAN-Bus、LCD的众多专用MCU和兼容系列。

1.2.2 嵌入式微处理器(EMPU)

嵌入式微处理器的基础是通用计算机中的CPU，它一般装配在专门设计的电路板上，只保留与嵌入式应用有关的母板功能，但是电路板上必须包括ROM、RAM、总线接口、各种外设等器件。嵌入式处理器目前主要有ARM、MIPS、Power PC、68K等，如图1-12所示，下面具体介绍。

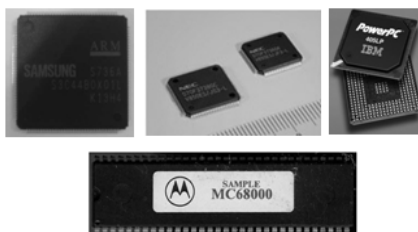


图1-12 各式各样的EMPU

1. ARM

ARM(Advanced RISC Machines)公司是全球领先的16/32位RISC(精简指令集计算机)微处理器知识产权设计供应商。ARM公司通过转让高性能、低成本、低功耗的RISC微处理器、外围和系统芯片设计技术给合作伙伴，使他们能用这些技术来生产各具特色的芯片。ARM已成为移动通信、手持设备和多媒体数字设备嵌入式解决方案的RISC标准。ARM处理器有三大特点：体积小、低功耗、低成本和高性能，16/32位双指令集，全球的合作伙伴众多。

2. MIPS

MIPS(Microprocessor without Interlocked Pipeline Stages)是一种处理器内核标准，它是由MIPS技术公司开发的。MIPS技术公司是一家设计制造高性能、高档次的嵌入式32位和64位处理器的厂商，在RISC处理器方面占有重要地位。2000年，MIPS公司发布了针对MIPS 32 4Kc处理器的新版本以及未来64位MIPS 64 20Kc处理器内核。

MIPS技术公司既开发MIPS处理器结构，又自己生产基于MIPS的32位/64位芯片。为了使用户更加方便地应用MIPS处理器，MIPS公司推出了一套集成的开发工具，称为MIPS IDF(Integrated Development Framework)，特别适用于嵌入式系统的开发。

3. Power PC

Power PC架构的特点是伸缩性好，方便灵活。Power PC处理器品种很多，既有通用的处理器，又有嵌入式控制器和内核，应用范围从高端的工作站、服务器到桌面计算机系统，从消费类电子产品到大型通信设备等各个方面，非常广泛。

目前Power PC独立微处理器与嵌入式微处理器的主频从25MHz到700MHz不等，它们的

能量消耗、大小、整合程度和价格差异悬殊，主要产品模块有主频350~700MHz的Power PC 750CX和750CXe以及主频400MHz的Power PC 440GP等。嵌入式的Power PC 405(主频最高为266MHz)和Power PC 440(主频最高为550MHz)处理器内核可以用于各种集成的系统芯片(SoC)设备上，在电信、金融和其他许多行业具有广泛的应用。

4. Intel ATOM

Intel公司出品的ATOM(凌动处理器)是英特尔历史上体积最小和功耗最小的处理器，其基于新的微处理架构，专门为小型和嵌入式系统设计，和其他嵌入式处理器相比其最大的优势是采用了x86体系结构可以运行Windows操作系统(也可以运行Android操作系统)，能提供更好的通用性，所以在平板电脑等消费电子产品中得到了广泛的应用。

1.2.3 嵌入式数字信号处理器(EDSP)

数字信号处理器对系统结构和指令进行了特殊设计，使其适合于执行DSP算法，编译效率较高，指令执行速度也快。DSP应用正从在通用单片机中以普通指令实现DSP功能，发展到采用嵌入式数字信号处理器。嵌入式数字信号处理器的长处在于能够进行向量运算、指针线性寻址等运算量较大的数据处理。比较有代表性的产品是Motorola的DSP56000系列、Texas Instruments的TMS320系列，以及Philips公司基于可重置嵌入式DSP结构制造的低成本、低功耗的REAL DSP处理器。

1.2.4 嵌入式片上系统(ESOC)

所谓的片上系统SOC是指在一个硅片上实现一个更为复杂的系统。各种处理器内核将作为SOC设计公司的标准库，成为VLSI设计中的一种标准器件，用标准的VHDL语言描述，存储在器件库中。SOC可以分为通用和专用两类。通用系列包括Infineon(Siemens)的TriCore、Motorola的M-Core、某些ARM系列器件等。而专用的SOC专用于某个或者某类系统中，不为一般用户所知。比如Philips的Smart XA，它将XA单片机内核和支持超过2048位复杂RSA算法的CCU单元制作在一块硅片上，形成一个可以加载Java或C语言的专用的片上系统。

1.3 嵌入式操作系统

早期的嵌入式系统很多都不用操作系统，它们只是为了实现某些特定功能，使用一个简单的循环控制对外界的控制请求进行处理，不具备现代操作系统的基本特征(如进程管理、存储管理、设备管理、网络通信等)。不可否认，这对一些简单的系统而言是足够的。但是当系统越来越复杂，利用的范围越来越广泛时，缺少操作系统就成了最大的一个缺点，因为每添加一项新功能都可能需要从头开始设计，否则只能增加开发成本和系统复杂度。

C语言的出现使操作系统的开发变得越来越简单，利用C语言可以很快地写出一个小型的、稳定的操作系统。众所周知，《The C Programming Language》(C程序设计语言)的作者Dennis M. Ritchie 和Brian W. Kernighan利用C语言写出了著名的UNIX操作系统，直接影响了

这30年计算机业的发展。同时，C语言的出现对开发嵌入式系统来说，在效率和速度上都提高了很多。

从20世纪80年代开始，出现了各种各样的商用嵌入式操作系统。这些操作系统大部分都是为专有系统而开发，从而形成了目前多种形式的商用嵌入式操作系统百家争鸣的局面，如VxWorks、Windows CE、Linux和Android等。

现在，网络在人们生活中的应用越来越广泛，在嵌入式系统中使用网络系统也自然成为一项基本的要求。在嵌入式系统中实现网络协议栈，对日常生活中的需要有着广泛的意义。利用嵌入式系统中的网络功能，可以实现下面将要介绍的信息电器这一即将取代PC、从而在后PC时代占据市场主体的商品。我们知道，如果在上面所说的那种采用循环控制的嵌入式系统中加入网络协议栈，其程序复杂度会呈指数级增长。相反，在嵌入式操作系统中增加网络协议模块要方便得多，并且还能方便各种网络应用程序在不同平台之间移植。

嵌入式系统的应用领域日益扩大，提供的应用功能也越来越复杂，当初的控制程序被逐步加入了许多功能，而实际上这些功能大多数是可以由操作系统来提供的。这很自然地会让人联想到应该为嵌入式系统开发一个嵌入式操作系统。由此可见，嵌入式操作系统是由于工程实践的需要而诞生的。而嵌入式操作系统所使用的技术，基本上是从台式计算机操作系统下推而来的。由于应用的需要和硬件条件的限制，嵌入式操作系统一般更加注重占用空间小和效率高等特点。

尽管嵌入式操作系统有功能丰富和稳定性好等优点，但大部分的嵌入式系统仍然继续采用控制程序而没有采用操作系统。除了功能需求和硬件方面的限制因素以外，还主要有以下两条原因：首先，有不少嵌入式系统的控制程序是逐步发展起来的，每一步改动都比较小。这种在原有系统上打补丁的代价，要小于改用操作系统所付出的代价，从而使工程人员很难下决心换用嵌入式操作系统。其次，虽然控制程序在开发成本和可靠性等方面都有缺点，但它最大的好处之一就是没有那些商业化嵌入式操作系统中许多用不着的功能。虽然到目前为止，几乎每一个嵌入式操作系统都号称可以根据应用的需要进行配置，可是大多都是静态配置，也就是用不同的模块编译链接成不同的系统。这种配置使嵌入式操作系统的稳定性大打折扣，因为每一种配置的结果都可以看成是一个全新的系统，其可靠性还需要实践的检验。

虽然上述因素导致许多嵌入式系统仍然沿用控制程序，但控制程序近来在有些应用领域表现得越来越力不从心，需要嵌入式操作系统予以取代。例如，高性能的手持设备、移动设备和复杂的工业控制装置(例如数控机床和机器人等)如果继续采用自己的控制程序，就意味着需要用户自己来做一个专用操作系统，因为设备管理、内存管理和进程管理等都是必不可少的。而精通控制程序的人很难同时又是一个操作系统的专家。

随着嵌入式系统的功能越来越复杂，硬件所提供的条件越来越好，选择嵌入式操作系统也就势在必行。首先，应用开发者的精力通常都集中在自己的应用领域，而没有时间和精力去全面掌握操作系统，所以需要嵌入式操作系统提供服务。其次，嵌入式系统的最大特点就是个性突出，每个具体的嵌入式系统都会有自己独特的地方，当其与某种特殊需要时如果操作系统能给予支持，则往往会有事半功倍的效果。

而且，将嵌入式操作系统引入到嵌入式系统中，能够对嵌入式系统的开发产生极大的推

动作用。在没有操作系统的嵌入式系统下，每当要进行进一步的开发和功能的扩展时，都会带来巨大的劳动力的无谓消耗。而嵌入式操作系统则可以通过提供给用户的各种API，来对嵌入式系统进行有效的管理。

从20世纪80年代起，国际上就开始进行一些商用嵌入式系统和专有操作系统的开发。这些商家开发嵌入式系统已经有20多年的经验，其产品系统目前的应用范围也比较广泛。下面先简要介绍一些嵌入式操作系统。

1.3.1 Linux

随着Linux的迅速发展，嵌入式Linux现在已经有许多的版本，包括强实时的嵌入式Linux(如新墨西哥工学院的RT-Linux和堪萨斯大学的KURT-Linux等)和一般的嵌入式Linux版本(如 μ CLinux和PocketLinux等)。其中，RT-Linux通过把通常的Linux任务优先级设为最低，而所有的实时任务的优先级都高于它，以达到既兼容通常的Linux任务又保证强实时性能的目的。另一种常用的嵌入式Linux是 μ CLinux，它是针对没有MMU(内存管理单元)的处理器而设计的。它不能使用处理器的虚拟内存管理技术，对内存的访问是直接的，所有程序中访问的地址都是实际的物理地址。它专为嵌入式系统做了许多小型化的工作。

1.3.2 VxWorks

VxWorks操作系统是美国风河(WindRiver)公司于1983年设计开发的一种实时操作系统。VxWorks拥有良好的持续发展能力、高性能的内核以及友好的用户开发环境，在实时操作系统领域内占据了一席之地。它以其良好的可靠性和卓越的实时性被广泛应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通信、军事演习、导弹制导、飞机导航等。在美国的F-16、FA-18战斗机、B-2隐形轰炸机和爱国者导弹上，甚至连在火星表面登陆的火星探测器上也使用了VxWorks。VxWorks的很多概念和技术都和Linux很类似，主要是用C语言开发。但VxWorks因价格很高，所以一般应用中很少采用这种操作系统。

VxWorks的主要特点如下。

1. 高性能实时微内核

VxWorks的微内核Wind是一个具有较高性能的、标准的嵌入式实时操作系统内核。它支持抢占式的基于优先级的任务调度，支持任务间同步和通信，还支持中断处理、看门狗(WatchDog)定时器和内存管理。其任务切换时间短、中断延迟小、网络流量大的特点使得VxWorks的性能得到很大提高，与其他嵌入式系统相比具有很大优势。

2. POSIX兼容

POSIX(Portable Operating System Interface)是工作在ISO/IEEE标准下的一系列有关操作系统的软件标准。制定这个标准的目的是为了在源代码层次上支持应用程序的可移植性。这个标准产生了一系列适用于实时操作系统服务的标准集合1003.1b(过去是1003.4)。

3. 自由配置能力

VxWorks提供良好的可配置能力，可配置的组件超过80个，用户可以根据自己系统的功

能需求通过交叉开发环境方便地进行配置。

4. 友好的开发调试环境

VxWorks提供的开发调试环境便于进行操作和配置,开发系统Tornado更是得到了广大嵌入式系统开发人员的欢迎。

5. 广泛的运行环境支持

VxWorks支持多种CPU: 如x86、i960、Sun Sparc、Motorola MC68000、MIPS RX000、Power PC、StrongARM、XScale等。大多数的VxWorks API是专用的。VxWorks提供的板级支持包(Board Support Package, BSP)支持多种硬件板,包括硬件初始化、中断设置、定时器和内存映射等例程。

1.3.3 WinCE

WinCE推出只有几年时间,但目前已占据了很大的市场份额。由于WinCE开发的都是大家熟悉的VC++环境,所以对于一般的开发人员都不会有多大难度,这也是WinCE容易被人们接受的原因。

WinCE具有优先级的多任务操作系统,它允许多重功能、进程在相同时间及系统中运行,WinCE支持最大的32位同步进程。一个进程包括一个或多个线程,每个线程代表进程的一个独立部分,一个线程被指定为进程的基本线程,进程也能创造一个未定数目的额外线程,额外线程的实际数目仅由可利用的系统资源限定。它的模块化设计允许它对从掌上电脑到专用的工业控制器的用户电子设备进行定制。Win CE操作系统的基本内核至少需要200KB的内存。

1.3.4 μ C/OS-II

μ C/OS是“MicroController Operating System”的缩写,它是源码公开的实时嵌入式操作系统, μ C/OS-II的主要特点如下。

- 公开源代码,系统透明,很容易就能把操作系统移植到各个不同的硬件平台上。
- 可移植性强。 μ C/OS-II绝大部分源码是用ANSI C写的,可移植性(Portable)较强。而与微处理器硬件相关的那部分是用汇编语言写的,已经压缩到最低限度,使 μ C/OS-II便于移植到其他微处理器上。
- 可固化。 μ C/OS-II是为嵌入式应用而设计的,这就意味着,只要开发者有固化(ROMable)手段(C编译、连接、下载和固化), μ C/OS-II即可嵌入到开发者的产品中成为产品的一部分。
- 可裁剪。通过条件编译可以只使用 μ C/OS-II中应用程序需要的那些系统服务程序,以减少产品中的 μ C/OS-II所需的存储器空间(RAM和ROM)。
- 占先式。 μ C/OS-II完全是占先式(Preemptive)的实时内核,这意味着 μ C/OS-II总是运行就绪条件下优先级最高的任务。大多数商业内核也是占先式的, μ C/OS-II在性能上和它们类似。

- 实时多任务。 $\mu\text{C}/\text{OS-II}$ 不支持时间片轮转调度法(Round-robin Scheduling)。该调度法适用于调度优先级平等的任务。
- 可确定性。全部 $\mu\text{C}/\text{OS-II}$ 的函数调用与服务的执行时间具有可确定性。

由于 $\mu\text{C}/\text{OS-II}$ 仅是一个实时内核，这就意味着它不像其他实时操作系统那样，它提供给用户的只是一些API函数接口，有很多工作往往需要用户自己去完成。把 $\mu\text{C}/\text{OS-II}$ 移植到目标硬件平台上也只是系统设计工作的开始，后面还需要针对实际的应用需求对 $\mu\text{C}/\text{OS-II}$ 进行功能扩展，包括底层的硬件驱动、文件系统和用户图形接口(GUI)等，从而建立一个实用的RTOS。

1.3.5 eCOS

eCOS(Embedded Configuration Operating System)是由Redhat推出的小型即时操作系统(Real-Time Operating System)，适合用做Bootloader增强和微小型系统。其特点如下：

- 将操作系统做成静态链接(Static Linker)的方式，让应用程序透过链接(Linker)产生出具有操作系统特性的应用程序。这是与嵌入式Linux系统最大的差异。
- 模块化，内核可配置。eCOS具有相当丰富的特性和一个配置工具，后者能够让用户选取自己需要的特性。
- 编译核心小。Linux兼容的嵌入式系统在内核裁减后编译出来的二进制代码大小在500KB以上，这还只包含最简单的内核模块，几乎没有加载任何其他的驱动与协议栈。但是eCOS最小版本只有几百KB，一般一个完整的网路应用，其二进制的代码也就100KB左右。
- 提供了Linux兼容的API，它能使开发人员轻松地将Linux应用移植。
- 具有可组态配置的特性，可针对精确性应用的需求而进行定制化，加上数百种的选项功效，使其能用最少的硬件资源获得最大可能的执行效能。
- 可以在各种硬件平台上执行，包括SUNPLUS、SPCE、ARM、CalmRISC、FR-V、Hitachi H8、IA-32、Motorola 68000、Matsushita AM3x、MIPS、NEC V8xx、PowerPC、SPARC、SuperH以及NiOS II等。

1.3.6 Android

Android(安卓)是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。该操作系统于2007年11月正式发布，2008年10月第一部采用其功能的智能手机(G1)上市，截止到2013年年末，Android的版本号已经发展到了4.4，采用该操作系统的智能设备数量已经超过了10亿台。

Android可以在ARM和x86体系结构的处理器上运行，其采用了分层的架构，可以从高层到低层分为应用程序层、应用程序框架层、系统运行库层和Linux内核层。

Android具有开放性、不受束缚、丰富的硬件、方便开发等优势，当然，其最大的优势还是有Google公司和众多的硬件服务商支持，其缺点是采用了虚拟机制导致效率略低，且碎片化严重。

1.3.7 iOS

iOS是苹果公司于2007年发布的系统，是以Darwin(一种类UNIX操作系统)为基础的操作系统，最开始命名为iPhone OS，后来更名为iOS，是iPod touch、iPad以及Apple TV等产品的操作系统，其版本号目前已经更新到了iOS 7.0.4。

iOS拥有在嵌入式操作系统中最大的应用程序以及最好的App(Application)库，支持ARM体系架构的处理器，具有高安全、支持多语言、流畅、美观等特点，其缺点是封闭性较强。

1.3.8 WP和Windows RT

WP是Windows Phone的简称，是微软在2010年发布的一款智能手机操作系统，将微软旗下的Skype、必应、Xbox Live游戏、Xbox Music音乐与独特的视频体验整合至手机；到2013年年底其已经更新到了8.1版本，其支持ARM体系架构的处理器。

Windows RT(RunTime)则是微软为实时嵌入式系统发布的Windows版本，它采用了Metro风格的用户界面，支持ARM体系架构处理器，但是无法兼容普通x86处理器结构上Windows的软件。“RT”代表“Runtime”，也就是Windows Runtime Library。它是一项非常重要的技术，因为它允许开发人员写一个App，但是却可以同时利用英特尔处理器的Windows 8上运行，还可以在利用ARM处理器的Windows RT上运行。

1.4 嵌入式系统设计

本节主要介绍嵌入式系统设计的一般开发流程和嵌入式系统开发模式。本节将采用自顶向下的方法，从对系统最抽象的描述开始，一步一步地推进到细节内容。

1.4.1 嵌入式系统开发流程

当前，嵌入式开发已经逐步规范化，在遵循一般工程开发流程的基础上，嵌入式开发有其自身的一些特点，如图1-13所示为嵌入式系统开发的一般流程。主要包括系统需求分析(要求有严格规范的技术要求)、体系结构设计、软硬件及机械系统设计、系统集成、系统测试，最后得到最终产品。

1. 系统需求分析

系统需求分析是指确定设计任务和设计目标，并提炼出设计规格说明书，作为正式设计指

导和验收的标准。系统的需求一般分为功能性需求和非功能性需求两方面。功能性需求是系统的基本功能，如输入输出信号、操作方式等；非功能需求包括系统性能、成本、功耗、体

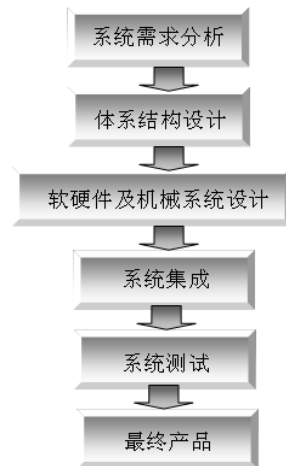


图1-13 嵌入式系统开发流程

积、重量等因素。

2. 体系结构设计

体系结构设计描述了系统如何实现所述的功能需求和非功能需求，包括对硬件、软件和执行装置的功能划分，以及系统的软件、硬件选型等。一个好的体系结构是设计成功与否的关键。

3. 硬件/软件协同设计

硬件/软件协同设计是基于体系结构，对系统的软件、硬件进行详细设计。为了缩短产品开发周期，设计往往是并行的。嵌入式系统设计的工作大部分都集中在软件设计上，采用面向对象技术、软件组件技术、模块化设计是现代软件工程经常采用的方法。

4. 系统集成

系统集成是把系统的软件、硬件和执行装置集成在一起，进行调试，发现并改进单元设计过程中的错误。

5. 系统测试

系统测试是指对设计好的系统进行测试，看其是否满足规格说明书中给定的功能要求。

图1-13所示只是嵌入式系统设计过程的一个大概流程，在实际的系统开发过程中，有一些重要因素是必须要考虑的，包括：功耗、性能(速度与精度达到要求)、成本、用户界面。

另外还必须考虑在系统设计的每一步骤中所要完成的任务，在设计过程的每一步骤中再添加以下细节。

- 必须在设计的每一个阶段对设计进行分析，以决定如何才能满足规格说明要求。
- 必须不断地细化设计，添加细节。
- 必须不断地核实设计，保证它依然满足所有的系统目标，如成本、速度、精度等。

1.4.2 嵌入式系统开发方法

嵌入式系统中的嵌入式处理器，由于其功能复杂、引脚繁多和运行高速，很难用常规目标机处理器的在线仿真(ICE)方式来开发。目前，嵌入式系统的开发有模拟开发调试、嵌入式在线仿真和远程调试等几种方法。

1. 模拟开发调试

有许多调试工具都配有模拟器，它是完全基于主机的软件，在主机上模拟了目标机中处理器的功能和指令。它虽然简单可行，但是缺乏在线调试功能和实时仿真功能。而且，它也只能模拟目标处理器，无法模拟处理器有关I/O的功能。因此，模拟器常作为初步基本调试工具。

ARM公司的开发工具有ARMulator模拟器，可以模拟开发各种嵌入式ARM处理器。它具有指令、周期和定时等3级模拟功能。

- 指令级：可以给出系统状态的精确行为(但没有考虑处理器的真正定时特性)。
- 周期级：可以给出每个周期的处理器精确行为(而实际上每一周期需若干时钟执行周期)。
- 定时级：可以在一周期内的准确时间出现信号(但允许逻辑延迟)。

ARMulator模拟器还可以与C语言和汇编语言配合使用，也可以作为VHDL硬件描述语言的ARM核的行为模型。

2. 嵌入式在线仿真

ARM架构处理器内含嵌入式在线仿真器宏单元，可为JTAG调试提供相应的接口。

JTAG是联合调试行为组织的缩写。由于集成电路的集成度不断提高，芯片的引脚不断增加；此外，为了缩小体积，常常采用表面贴装技术。因此，无法使用常规的在线仿真的方式。JTAG为此制定了边界扫描标准，只需5根引脚就可以实现在线仿真的功能。该标准已被批准为IEEE-1149.1标准。它不但能测试各种集成芯片，也能测试芯片内各类宏单元，还能测试相应的印刷电路板。

同时，为了能达到实时跟踪调试功能，ARM架构处理器还内含嵌入式跟踪宏单元，通过逻辑分析仪来实现实时跟踪调试的功能。所以，ARM架构的Embedded ICE-RT就是指嵌入式在线仿真及实时跟踪调试。

3. 远程调试

高档处理器的调试常常采用这种方式，通过设计一个驻留程序与主机通过串口、Ethernet或USB接口进行连接，然后在驻留程序中编写各个功能模块的测试程序，主机通过收发指令获得目标机的相应信息。

在嵌入式系统中比较流行的驻留程序有Angel、Blob、Red Boot等。

思考与练习

一、填空题

1. 嵌入式系统主要是融合了_____、_____和_____，它将计算机直接嵌入到应用系统中，利用计算机的高速处理能力以实现某些特定的功能。
2. 目前国内对嵌入式系统普遍认同的一个定义是：以_____为中心、以_____为基础、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。
3. 嵌入式系统一般由嵌入式计算机和执行部件组成，其中嵌入式计算机主要由四个部分组成，它们分别是_____、_____、_____和_____。
4. 嵌入式处理器目前主要有_____、_____、_____、68K等，其中_____处理器有三大特点：体积小、低功耗、低成本和高性能，16/32位双指令集，全球的合作伙伴众多。
5. 常见的嵌入式操作系统有_____、_____、_____和Android。

6. 嵌入式系统开发的一般流程主要包括系统需求分析、_____、_____、系统集成、_____，最后得到最终产品。

二、选择题

1. 嵌入式系统中硬件层主要包含了嵌入式系统中必要的硬件设备：()、存储器(SDRAM、ROM等)、设备IO接口等。

- A. 嵌入式微处理器
- B. 嵌入式控制器
- C. 单片机
- D. 集成芯片

2. 20世纪90年代以后，随着系统应用对实时性要求的提高，系统软件规模不断上升，实时核逐渐发展为()，并作为一种软件平台逐步成为目前国际嵌入式系统的主流。

- A. 分时多任务操作系统
- B. 多任务操作系统
- C. 实时操作系统
- D. 实时多任务操作系统

3. 由于其高可靠性，在美国的火星表面登陆的火星探测器上使用的嵌入式操作系统也是()。

- A. Android
- B. VxWorks
- C. Linux
- D. WinCE

4. 嵌入式系统设计过程中一般需要考虑的因素不包括()。

- A. 性能
- B. 功耗
- C. 价格
- D. 大小

5. 在嵌入式系统中比较流行的驻留程序有()。

- A. Angel
- B. Blob
- C. Red Boot
- D. U-Boot

三、简答题

1. 举例说明身边常用的嵌入式系统。
2. 如何理解嵌入式系统，谈谈自己的理解。
3. 简述嵌入式系统的开发方法有哪几种。
4. 结合当前嵌入式系统的发展，想象一下嵌入式系统的应用前景。