

高等院校嵌入式人才培养规划教材
Gaodeng Yuanxiao Qianrushu Rencai Peiyang Guihua Jiaocai

ARM嵌入式 体系结构与接口技术 (Cortex-A8版)

华清远见嵌入式学院 杨胜利 刘洪涛 主编



ARM Embedded Architecture and Interface Technology

理实一体化

提供实训项目拓展

配备教学实验平台 经典教学视频



DVD-ROM

 人民邮电出版社
POSTS & TELECOM PRESS

013071077

高等院校嵌入式人才培养规划教材

Gaodeng Yuanxiao Qianrushu Rencai Peiyang G

TP332-43

60

ARM嵌入式 体系结构与接口技术 (Cortex-A8版)

华清远见嵌入式学院 杨胜利 刘洪涛 主编



ARM Embedded Architecture
and Interface Technology



北航

C1680207

人民邮电出版社

北京

TP 332-43

60

图书在版编目 (CIP) 数据

ARM嵌入式体系结构与接口技术 : Cortex-A8版 / 华清远见嵌入式学院, 杨胜利, 刘洪涛主编. — 北京 : 人民邮电出版社, 2013.9

高等院校嵌入式人才培养规划教材
ISBN 978-7-115-31741-4

I. ①A… II. ①华… ②杨… ③刘… III. ①微处理器—系统结构—高等学校—教材②微处理器—接口技术—高等学校—教材 IV. ①TP332

中国版本图书馆CIP数据核字 (2013) 第150443号

内 容 提 要

本书在全面介绍 CORTEX-A8 处理器的体系结构、编程模型、指令系统及 EclipseFor ARM 开发环境的同时, 以基于 CORTEX-A8 的应用处理器 S5PC100 为核心, 详细介绍了系统的设计及相关接口技术。接口技术中涵盖了 I/O、中断、串口、存储器、PWM、A/D、DMA、LCD、CamIF。书中提供了大量实验内容。

本书可作为高等院校或高等职业院校电子、通信、自动化、计算机等专业的 ARM 体系结构、接口技术课程的教材, 也可作为相关嵌入式开发人员的参考书。

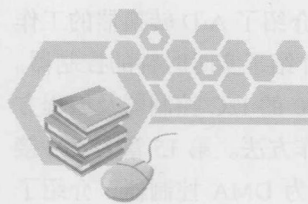
-
- ◆ 主 编 华清远见嵌入式学院 杨胜利 刘洪涛
责任编辑 王 威
责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 16.75 2013 年 9 月第 1 版
字数: 424 千字 2013 年 9 月北京第 1 次印刷
-

定价: 49.80 元 (附光盘)

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号



随着消费群体对产品要求的日益提高,嵌入式技术在机械器具制造、电子产品制造、通信、信息服务等行业领域得到了大显身手的机会,应用日益广泛,相应地企业对嵌入式技术人才的需求也越来越多。因此近几年来,各高职高专院校开始纷纷开设嵌入式专业或方向。但是,各院校在嵌入式专业教学建设的过程中几乎都面临教材难觅的困境。虽然目前市场上的嵌入式技术开发相关书籍比较多,但几乎都是针对有一定基础的行业内研发人员而编写的,并不完全符合学校的教学要求。学校教学需要一套充分考虑学生现有知识基础和接受度的,明确各门课程教学目标的,便于学校安排课时的嵌入式专业教材。

针对教材缺乏的问题,我们以多年来在嵌入式工程技术领域内人才培养、项目研发的经验为基础,汇总了近几年积累的数百家企业对嵌入式技术研发相关岗位的真实需求,调研了数十所开设“嵌入式工程技术”专业的高职院校的课程设置情况、学生特点和教学用书现状。通过细致的整理和分析,对专业技能和基本知识进行合理划分,我们于2009年编写了这套高等院校嵌入式人才培养规划教材,包括以下5本:

《嵌入式技术基础》

《ARM 嵌入式体系结构与接口技术》

《嵌入式 Linux 操作系统》

《嵌入式 Linux C 语言开发》

《嵌入式应用程序设计》

时隔4年,嵌入式技术在飞速发展,一方面相关产品已经更新换代,另一方面嵌入式专业的教学方式也发生了改变。所以这次我们组织老师对原系列书进行修改和扩充。

本书是其中之一。ARM 作为一种32位的高性能、低成本的嵌入式 RISC 微处理器,已得到广泛的应用。目前,ARM Cortex 处理器已经占据了大部分嵌入式处理器的中高端产品市场。本书以 Cortex-A8 处理器为平台,介绍了嵌入式系统开发的各个主要环节。本书侧重实践,辅以代码加以讲解,从分析的角度来学习嵌入式开发的各种技术。本书使用的工具是 Eclipse+GNU 工具集实现的开源开发环境。

本书将嵌入式软硬件理论讲解和技能实践融合在一起,全书共18章。第1章为嵌入式系统基础知识,介绍了嵌入式系统的组成及嵌入式开发概述。第2章为 ARM 技术概述,讲解了 ARM 体系结构、应用选型及编程模型等。第3章为 ARM 的指令系统,重点介绍了 ARM 指令集。第4章为 GNU 汇编伪指令集,主要介绍了 ARM 程序中常用的伪指令、伪操作及 ARM 汇编语句的格式和结构。第5章为 ARM 集成开发环境搭建,主要介绍了 Realview MDK 环境的安装和使用。第6章为 GPIO 编程,介绍了 GPIO 的概念及 S5PC100 的 GPIO 操作方法。第7章为 ARM 系统时钟及编程,介绍了 S5PC100 处理器核的时钟原理及工作机制。第8章为 ARM 异常处理及编程,介绍了 ARM 处理器的异常处理,及 S5PC100 的中断控制器工作原理。第9章为串行通信 UART 接口,介绍了串行通信的概念及 S5PC100 串口的操作方法。第10章为定时器,介绍了定时器的



工作原理及 S5PC100 定时器接口的操作方法。第 11 章为 A/D 转换器,介绍了 A/D 转换器的工作原理及 S5PC100 A/D 控制器的操作方法。第 12 章为实时时钟 (RTC),介绍了实时时钟的控制器。第 13 章为 I2C 总线,介绍了 I2C 总线通信的概念,总线规范, S5PC100 的 I2C 控制器及编程方法。第 14 章为存储器接口,介绍了 Nor Flash、Nand Flash 存储器的操作方法。第 15 章为 SPI 接口,介绍了 SPI 总线协议及 SPI 总线控制器的基本编程方法。第 16 章为 DMA 控制器,介绍了 S5PC100 下 DMA 控制器的操作方式和编程模型。第 17 章为 LCD 接口技术,介绍了基于 S5PC100 的 LCD 控制器,详细介绍了 LCD 控制器相关的寄存器及配置方法、编程模型等。第 18 章为 CAMIF 接口技术,介绍了基于 S5PC100 的 CAMIF 控制器以及 OV9650 芯片、SCCB 的操作方式。

在学习本书时,读者要具有一定的数字电路和 C 语言的基础知识。

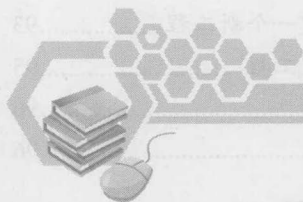
本书由刘洪涛主编并统校全稿。本书的完成需要感谢华清远见嵌入式学院,教材内容参考了学院与嵌入式企业需求无缝对接的、科学的专业人才培养体系。同时,嵌入式学院从业或执教多年的行业专家团队也对教材的编写工作做出了贡献,冯利美、贾燕枫等在书稿的编写过程中认真阅读了所有章节,提供了大量在实际教学中积累的重要素材,对教材结构、内容提出了中肯的建议,并在后期审校工作中提供了很多帮助,在此表示衷心的感谢。

本书所有源代码、PPT 课件、教学素材等辅助教学资料,请到人民邮电出版社教学服务与资源网 (www.ptpedu.com.cn) 下载。

由于作者水平所限,书中不妥之处在所难免,恳请读者批评指正。对于本书的批评和建议,可以发到 www.embedu.org 技术论坛。

编 者

2013 年 4 月



目 录

第1章 嵌入式系统基础知识 1

1.1 嵌入式系统的概述 1

1.1.1 嵌入式系统简介 1

1.1.2 嵌入式系统的特点 2

1.1.3 嵌入式系统的发展 3

1.2 嵌入式系统的组成 5

1.2.1 嵌入式系统的硬件组成 5

1.2.2 嵌入式系统的软件组成 6

1.3 嵌入式操作系统举例 6

1.3.1 商业版嵌入式操作系统 6

1.3.2 开源版嵌入式操作系统 7

1.4 嵌入式系统开发概述 8

小结 13

思考与练习 13

第2章 ARM 技术概述 14

2.1 ARM 体系结构的技术

特征及发展 14

2.1.1 ARM 公司简介 14

2.1.2 ARM 技术特征 15

2.1.3 ARM 体系结构的发展 16

2.2 ARM 微处理器简介 18

2.2.1 ARM7 处理器系列 19

2.2.2 ARM9 处理器系列 19

2.2.3 ARM9E 处理器系列 20

2.2.4 ARM11 处理器系列 20

2.2.5 SecurCore 处理器系列 20

2.2.6 StrongARM 和 Xscale 处理器系列 21

2.2.7 MPCore 处理器系列 21

2.2.8 Cortex 处理器系列 21

2.3 ARM 微处理器结构 23

2.4 ARM 微处理器的应用选型 23

2.4.1 ARM 芯片选择的

一般原则 24

2.4.2 选择一款适合教学的

ARM 芯片 24

2.5 Cortex-A8 内部功能及特点 28

2.6 数据类型 28

2.6.1 ARM 的基本数据类型 28

2.6.2 浮点数据类型 29

2.6.3 存储器大/小端 29

2.7 Cortex-A8 内核工作模式 30

2.8 Cortex-A8 存储系统 31

2.8.1 协处理器 32

2.8.2 存储管理单元 33

2.8.3 高速缓冲存储器 33

2.8.4 NEON 技术 34

2.8.5 安全域(TrustZone) 34

2.9 流水线 34

2.9.1 流水线的概念与原理 34

2.9.2 流水线的分类 35

2.9.3 影响流水线性能的因素 36

2.10 寄存器组织 37

2.11 程序状态寄存器 39

2.12 SAMSUNG S5PC100 处理器

介绍 41

小结 43

思考与练习 43

第3章 ARM 的指令集合 44

3.1 ARM 指令集 44

3.1.1 数据操作指令 44

3.1.2 乘法指令 51

3.1.3 Load/Store 指令 53



3.1.4 跳转指令.....	59	5.2.2 创建一个新工程.....	93
3.1.5 状态操作指令.....	62	5.2.3 调试工程.....	95
3.1.6 协处理器指令.....	63	小结.....	96
3.1.7 异常产生指令.....	65	思考与练习.....	96
3.2 ARM 指令的寻址方式.....	67		
小结.....	71	第 6 章 GPIO 编程.....	97
思考与练习.....	71	6.1 GPIO 功能介绍.....	97
第 4 章 GNU 汇编伪指令集.....	72	6.2 S5PC100 芯片的 GPIO 控制器 介绍.....	98
4.1 GNU 汇编器的平台无关 伪指令.....	72	6.2.1 特性.....	98
4.1.1 伪指令概念.....	72	6.2.2 GPIO 分组预览.....	98
4.1.2 符号定义伪指令.....	73	6.2.3 S5PC100 的 GPIO 常用 寄存器分类.....	98
4.1.3 数据定义伪指令.....	74	6.2.4 S5PC100 I/O 接口常用 寄存器详解.....	99
4.1.4 汇编控制伪操作.....	76	6.3 S5PC100 的 GPIO 的实例.....	100
4.1.5 杂项伪操作.....	78	6.3.1 电路原理.....	101
4.2 GNU 汇编器支持的 ARM 伪指令.....	80	6.3.2 寄存器设置.....	101
4.2.1 ADR 伪指令.....	80	6.3.3 程序编写.....	101
4.2.2 ADRL 伪指令.....	80	小结.....	102
4.2.3 LDR 伪指令.....	81	思考与练习.....	102
4.3 ARM 汇编语言的程序结构.....	82	第 7 章 ARM 系统时钟及编程.....	103
4.3.1 汇编语言的程序格式.....	82	7.1 S5PC100 时钟域的划分.....	103
4.3.2 汇编语言的子程序调用.....	82	7.2 S5PC100 时钟的产生 过程分析.....	104
4.3.3 过程调用标准 AAPCS/ATPCS.....	83	7.2.1 时钟的产生.....	104
4.3.4 汇编语言程序设计举例.....	84	7.2.2 模块对应的时钟域.....	105
4.4 汇编语言与 C 语言的 混合编程.....	84	7.3 S5PC100 时钟源的选择.....	105
4.4.1 GNU 内联汇编.....	85	7.3.1 时钟来自于外部引脚.....	105
4.4.2 C 和汇编的混合编程.....	87	7.3.2 时钟来自于时钟 管理单元 (CMU).....	106
小结.....	88	7.4 S5PC100 时钟的配置.....	106
思考与练习.....	89	7.5 S5PC100 时钟配置寄存器 描述.....	108
第 5 章 ARM 集成开发环境搭建.....	90	7.6 S5PC100 时钟源配置示例.....	109
5.1 FS-JTAG 仿真器介绍.....	90	小结.....	110
5.2 开发环境搭建.....	91	思考与练习.....	110
5.2.1 开发工具的安装.....	91		



第8章 ARM异常处理及编程 111

- 8.1 ARM异常中断处理概述 111
- 8.2 ARM体系异常种类 112
- 8.3 ARM异常的优先级 116
- 8.4 ARM处理器模式和异常 117
- 8.5 ARM异常响应和处理程序
返回 117
 - 8.5.1 中断响应的概念 117
 - 8.5.2 ARM异常响应流程 118
 - 8.5.3 从异常处理程序中返回 119
- 8.6 ARM系统中异常中断处理程序的
安装 121
- 8.7 ARM的SWI异常中断处理程序
设计 121
- 8.8 FIQ和IRQ异常中断程序
设计 124
- 8.9 基于Cortex-A8内核的
S5PC100异常程序设计 125
 - 8.9.1 S5PC100中断机制分析 125
 - 8.9.2 S5PC100中断处理
程序实例 128
- 小结 132
- 思考与练习 132

第9章 串行通信接口 133

- 9.1 串行通信 133
 - 9.1.1 串行通信与并行通信的
概念 133
 - 9.1.2 异步串行方式的特点 134
 - 9.1.3 异步串行方式的数据
格式 134
 - 9.1.4 同步串行方式的特点 134
 - 9.1.5 同步串行方式的数据
格式 134
 - 9.1.6 比特率、比特率因子与
位周期 135
 - 9.1.7 RS-232C串口规范 135
 - 9.1.8 RS-232C接线方式 137

9.2 S5PC100异步串行通信 137

- 9.2.1 S5PC100串口控制器
概述 137
- 9.2.2 S5PC100串口控制器
寄存器 138
- 9.3 串口发送接收程序示例 142
 - 9.3.1 电路连接 142
 - 9.3.2 程序的编写 142
 - 9.3.3 调试与运行程序 145
- 小结 145
- 思考与练习 145

第10章 PWM定时器 146

- 10.1 S5PC100 PWM定时器 146
 - 10.1.1 PWM定时器概述 146
 - 10.1.2 PWM定时器特点 147
 - 10.1.3 PWM定时器的寄存器 148
 - 10.1.4 PWM定时器示例 152
- 10.2 S5PC100看门狗定时器 154
 - 10.2.1 S5PC100看门狗定时器
概述 154
 - 10.2.2 看门狗定时器寄存器 155
 - 10.2.3 看门狗定时器程序编写 156
- 小结 157
- 思考与练习 157

第11章 A/D转换器 158

- 11.1 A/D转换器原理 158
 - 11.1.1 A/D转换基础 158
 - 11.1.2 A/D转换的技术指标 159
 - 11.1.3 A/D转换器类型 160
 - 11.1.4 A/D转换的一般步骤 164
- 11.2 S5PC100 A/D转换器 164
 - 11.2.1 S5PC100 A/D转换器
概述 164
 - 11.2.2 S5PC100 A/D控制器
寄存器 165
- 11.3 A/D转换器示例 167
 - 11.3.1 电路连接 167



11.3.2 程序的编写	167	14.2.3 AM29LV160D 扇区/块 擦除操作	188
11.3.3 调试与运行结果	168	14.2.4 AM29LV160D 芯片 擦除操作	188
小结	168	14.2.5 AM29LV160D 与 S5PC100 的接口电路	189
思考与练习	168	14.2.6 AM29LV160D 存储器的 程序设计	190
第 12 章 实时时钟 RTC	169	14.3 NAND Flash 操作	192
12.1 RTC 介绍	169	14.3.1 芯片介绍	192
12.2 RTC 控制器	170	14.3.2 读操作过程	193
12.3 RTC 控制器寄存器详解	170	14.3.3 擦除操作过程	194
12.4 RTC 应用示例	173	14.3.4 写操作过程	195
小结	174	14.4 S5PC100 中 NAND Flash 控制器的操作	195
思考与练习	174	14.4.1 S5PC100 中 NAND Flash 控制器概述	195
第 13 章 I2C 总线	175	14.4.2 S5PC100 中 NAND Flash 控制器寄存器详解	196
13.1 I2C 总线	175	14.5 S5PC100 NAND Flash 接口 电路与程序设计	198
13.1.1 I2C 总线介绍	175	14.5.1 K9F2G080U 和 S5PC100 的接口电路	198
13.1.2 I2C 总线术语	176	14.5.2 S5PC100 控制 K9F2G080U 的程序设计	198
13.1.3 I2C 总线位传输	176	小结	202
13.1.4 I2C 总线数据传输	177	思考与练习	202
13.1.5 I2C 总线寻址方式	177	第 15 章 SPI 接口	203
13.1.6 快速和高速模式	178	15.1 SPI 总线协议理论	203
13.2 I2C 总线控制器	179	15.1.1 协议简介	203
13.2.1 S5PC100 下的 I2C 控制器介绍	179	15.1.2 协议内容	203
13.2.2 I2C 总线控制寄存器 详解	179	15.2 SPI 控制器详解	205
13.3 I2C 总线应用示例	180	15.2.1 S5PC100 的 SPI 控制器 简介	205
13.3.1 电路原理分析	180	15.2.2 时钟源控制	206
13.3.2 代码实现	182	15.2.3 寄存器详解	206
小结	183	15.3 SPI 接口应用示例	208
思考与练习	183	小结	214
第 14 章 存储器接口	184		
14.1 Flash ROM 介绍	184		
14.2 NOR Flash 操作	186		
14.2.1 AM29LV160D 芯片介绍	186		
14.2.2 AM29LV160D 字 编程操作	187		



思考与练习.....	214	17.2 LCD 控制器应用示例.....	237
第 16 章 DMA 控制器.....	215	小结.....	241
16.1 PL330 原理概述.....	215	思考与练习.....	241
16.1.1 DMAC 简述.....	215	第 18 章 CAMIF 接口技术.....	242
16.1.2 S5PC100 下的 DMAC 模型.....	216	18.1 OV9650 介绍.....	242
16.1.3 PL330 简述.....	217	18.1.1 芯片功能描述.....	242
16.2 PL330 详解.....	219	18.1.2 OV9650 物理参数.....	243
16.2.1 PL330 指令集.....	219	18.1.3 OV9650 寄存器详解.....	244
16.2.2 相关寄存器详解.....	223	18.2 SCCB 总线.....	246
16.3 S5PC100 PL330 示例.....	224	18.2.1 SCCB 协议介绍.....	246
小结.....	228	18.2.2 SCCB 的总线编程.....	247
思考与练习.....	228	18.3 CAMIF 接口详解.....	248
第 17 章 LCD 接口技术.....	229	18.3.1 基于 S5PC100 的 CAMIF 接口介绍.....	248
17.1 LCD 控制器原理概述.....	229	18.3.2 S5PC100 CAMIF 寄存器 详解.....	250
17.1.1 LCD 控制器介绍.....	229	18.3.3 CAMIF 应用示例.....	253
17.1.2 S5PC100 的 LCD 控制器 介绍.....	230	小结.....	256
17.1.3 S5PC100 的 LCD 控制器 操作.....	231	思考与练习.....	256
17.1.4 LCD 控制器寄存器.....	233	参考文献.....	257

第1章

嵌入式系统基础知识

嵌入式系统已成为当前较为热门的领域之一，受到了社会各方面的广泛关注，越来越多的人开始学习嵌入式系统开发。本章将向读者介绍嵌入式系统的基本知识。

本章主要内容：

- 嵌入式系统的概述；
- 嵌入式系统的组成；
- 嵌入式操作系统举例；
- 嵌入式系统开发概述。

1.1 嵌入式系统的概述

1.1.1 嵌入式系统简介

经过 30 多年的发展，嵌入式系统已经广泛地渗透到人们的学习、工作、生活中，可以看到，嵌入式系统已经应用到科学研究、工程设计、军事技术、各类产业、商业文化艺术、娱乐业以及人们的日常生活等方方面面。表 1-1 列举了嵌入式系统应用的部分领域。随着数字信息技术和网络技术的飞速发展，计算机、通信、消费电子的一体化趋势将日益明显，而这必将培育出一个庞大的嵌入式应用市场。嵌入式系统技术也成了当前关注、学习研究的热点之一。大家可能会问究竟什么是嵌入式系统呢？嵌入式系统本身是一个相对模糊的定义，不同的组织对其定义也略有不同，但大意是相同的。下面来介绍一下嵌入式系统的相关定义。

按照电气和电子工程师协会（IEEE）的定义，嵌入式系统是用来控制、监控或者辅助操作机器、装置、工厂等大规模系统的设备（devices used to control, monitor, or assist the operation of equipment, machinery or plants）。这个主要是根据嵌入式系统



的用途来进行定义的。

表 1-1

嵌入式系统应用领域举例

领 域	应 用
消费电子	信息家电 智能玩具 通信设备 移动存储 视频监控
工业控制	工控设备 智能仪表 汽车电子 电子农业
网 络	网络设备 电子商务 无线传感器
医务医疗	医疗电子
军事国防	军事电子
航空航天	各类飞行设备、卫星等

更具一般性的,也是在多数书籍资料中使用的嵌入式系统的定义:嵌入式系统是指以应用为中心,以计算机技术为基础,软件、硬件可剪裁,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。

根据以上嵌入式系统的定义,可以看出,嵌入式系统是由硬件和软件相结合组成的具有特定功能、用于特定场合的独立系统。其硬件主要由嵌入式微处理器、外围硬件设备组成;其软件主要由底层系统软件 and 用户应用软件组成。

1.1.2 嵌入式系统的特点

(1) 专用,软、硬件可剪裁、可配置。

从嵌入式系统的定义可以看出,嵌入式系统是面向应用的,和通用系统最大的区别在于嵌入式系统功能专一。根据这个特性,嵌入式系统的软、硬件可以根据需要进行精心设计、量体裁衣、去除冗余,以实现低成本、高性能。也正因如此,嵌入式系统采用的微处理器和外围设备种类繁多,系统不具通用性。

(2) 低功耗、高可靠性、高稳定性。

嵌入式系统大多用在特定的场合,要么是环境条件恶劣,要么要求其长时间连续运转,因此嵌入式系统应具有高可靠性、高稳定性、低功耗等性能。

(3) 软件代码短小精悍。

由于成本和应用场合的特殊性,嵌入式系统的硬件资源(如内存等)通常都比较少,因此对嵌入式系统的设计也提出了较高的要求。嵌入式系统的软件设计尤其要求高质量,要在有限资源上实现高可靠性、高性能的系统。虽然随着硬件技术的发展和成本的降低,在高端嵌入式产品上也开始采用嵌入式操作系统,但和 PC 资源比起来还是少得可怜,因此嵌入式系统的软件代码依然要在保证性能的情况下,占用尽量少的资源,保证产品的高性价比,使其具有更强的竞争力。

(4) 代码可固化。

为了提高系统的执行速度和可靠性,嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中,而不是存储于磁盘中。

(5) 实时性。

很多采用嵌入式系统的应用具有实时性要求,因此大多嵌入式系统采用实时性系统。但



需要注意的是嵌入式系统不等于实时系统。

(6) 交互性。

嵌入式系统不仅功能强大,而且使用灵活方便,一般不需要键盘、鼠标等。人机交互以简单方便为主。

(7) 嵌入式系统软件开发通常需要专门的开发工具和开发环境。

(8) 要求开发、设计人员有较高的技能。

嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合的产物。这一点就决定了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,从事嵌入式系统开发的人才也必须是复合型人才。

1.1.3 嵌入式系统的发展

1. 嵌入式系统的发展阶段

在过去的30多年中,嵌入式系统主要经历了4个发展阶段。

第1阶段是以单芯片为核心的可编程控制器形式的嵌入式系统。这类系统大部分应用于一些专业性强的工业控制系统中,一般没有操作系统的支持,软件通过汇编语言编写。这一阶段系统的主要特点是系统结构和功能相对单一,处理效率较低,存储容量较小,几乎没有用户接口。由于这种嵌入式系统使用简单、价格低,因此以前在国内工业领域应用较为普遍,但是目前已经远不能满足高效的、需要大容量存储的现代工业控制和新兴信息家电等领域的需求。

第2阶段是以嵌入式CPU为基础、以简单操作系统为核心的嵌入式系统。其主要特点是:CPU种类繁多,通用性比较弱;系统开销小,效率高;操作系统达到一定的兼容性和扩展性;应用软件较专业化,用户界面不够友好。

第3阶段是以嵌入式操作系统为标志的嵌入式系统。其主要特点是:嵌入式操作系统能运行于各种不同类型的微处理器上,兼容性好;操作系统内核小、效率高,并且具有高度的模块化和扩展性;具备文件和目录管理、支持多任务、支持网络应用、具备图形窗口和用户界面;具有大量的应用程序接口API,开发应用程序较简单;嵌入式应用软件丰富。

第4阶段是以Internet为标志的嵌入式系统。这是一个正在迅速发展的阶段。目前大多数嵌入式系统还孤立于Internet之外,但随着Internet的发展以及Internet技术与信息家电、工业控制技术结合的日益密切,嵌入式设备与Internet的结合将代表着嵌入式系统的未来。

2. 嵌入式系统的发展趋势

(1) 小型化、智能化、网络化、可视化。

随着技术水平的提高和人们生活的需要,嵌入式设备(尤其是消费类产品)正朝着小型化、便携式和智能化的方向发展。如果携带笔记本电脑外出办事,肯定希望它轻薄小巧,甚至可能希望有一种更便携的设备来替代它,目前的上网本、MID(移动互联网设备)、便携投影仪等都是因类似的需求而出现的。对嵌入式系统而言,可以说是已经进入了嵌入式互联网时代(有线网、无线网、广域网、局域网的组合),而嵌入式设备和互联网的紧密结合,更为日常生活带来了极大的方便和无限的想象空间。随着嵌入式设备的功能越来越强大,未来冰箱、洗衣机



等家用电器都将实现网上控制；异地通信、协同工作、无人操控场所、安全监控场所等的可视化已经成为了现实，随着网络运载能力的提升，可视化程度将得到进一步完善。人工智能、模式识别技术也将在嵌入式系统中得到应用，这会使得嵌入式系统更具人性化、智能化。

(2) 多核技术的应用。

人们需要处理的信息越来越多，这就要求嵌入式设备的运算能力越来越强，因此需要设计出更强大的嵌入式处理器，多核技术处理器在嵌入式系统中的应用将更为普遍。

(3) 低功耗（节能）、绿色环保。

在嵌入式系统的硬件和软件设计中都在追求更低的功耗，以求嵌入式系统能获得更长的可靠工作时间。如手机的通话和待机时间、MP3 听音乐的时间等。同时，绿色环保型嵌入式产品将更受到人们的青睐，在嵌入式系统设计中也会更多地考虑辐射和静电等问题。

(4) 云计算、可重构、虚拟化等技术被进一步应用到嵌入式系统中。

简单地讲，云计算是将计算分布在大量的分布式计算机上，这样只需要一个终端，就可以通过网络服务来实现需要的计算任务，甚至是超级计算任务。云计算 (Cloud Computing) 是分布式处理 (Distributed Computing)、并行处理 (Parallel Computing) 和网格计算 (Grid Computing) 的发展，或者说是这些计算机科学概念的商业实现。在未来几年里，云计算将得到进一步发展与应用。

可重构性是指在一个系统中，其硬件模块或（和）软件模块均能根据变化的数据流或控制流对系统结构和算法进行重新配置（或重新设置）。可重构系统最突出的优点就是能够根据不同的应用需求，改变自身的体系结构，以便与具体的应用需求相匹配。

虚拟化是指计算机软件在一个虚拟的平台上而不是真实的硬件上运行。虚拟化技术可以简化软件的重新配置过程，易于实现软件标准化。其中 CPU 的虚拟化可以实现单 CPU 模拟多 CPU 并行运行，允许一个平台同时运行多个操作系统，并且都可以在相互独立的空间内运行而互不影响，从而提高工作效率和安全性。虚拟化技术是降低多内核处理器系统开发成本的关键，也是未来几年最值得期待和关注的关键技术之一。

各种技术的成熟与其在嵌入式系统中的应用，将不断为嵌入式系统增添新的魅力和发展空间。

(5) 嵌入式软件开发平台化、标准化，系统可升级，代码可复用技术将更受重视。

嵌入式操作系统将进一步走向开放、开源、标准化、组件化。嵌入式软件开发平台化也将是今后的一个趋势，越来越多的嵌入式软硬件行业标准将出现，最终的目标是使嵌入式软件开发简单化，这也是一个必然规律。同时随着系统复杂性的提高，系统可升级和代码复用技术在嵌入式系统中将得到更多的应用。另外，因为嵌入式系统采用的微处理器种类多，不够标准，所以在嵌入式软件开发中将更多地使用跨平台的软件开发语言与工具，目前，Java 语言正在被越来越多地使用到嵌入式软件开发中。

(6) 嵌入式系统软件将逐渐 PC 化。

需求的增加和网络技术的发展是嵌入式系统发展的一个源动力。移动互联网的发展，将进一步促进嵌入式系统软件 PC 化。如前所述，结合跨平台开发语言的广泛应用，未来嵌入式软件开发的观念将逐渐被淡化，也就是说嵌入式软件开发和非嵌入式软件开发的区别将逐渐减小。

(7) 融合趋势。

嵌入式系统软硬件融合、产品功能融合、嵌入式设备和互联网的融合趋势加剧。嵌入式系统设计中软硬件结合将更加紧密，软件将是其核心。消费类产品将在运算能力和便携方面



进一步融合。传感器网络的迅速发展，将极大地促进嵌入式技术和互联网技术的融合。

(8) 安全性。

随着嵌入式技术和互联网技术相结合的发展，嵌入式系统的信息安全问题日益凸显，保证信息安全也成为了嵌入式系统开发的重点和难点。

1.2 嵌入式系统的组成

从前面的介绍可以知道，嵌入式系统总体上是由硬件和软件组成的，硬件是其基础，软件是其核心与灵魂。它们之间的关系如图 1-1 所示。

1.2.1 嵌入式系统的硬件组成

嵌入式系统的硬件设备包括嵌入式处理器和外围设备。其中的嵌入式处理器（CPU）是嵌入式系统的核心部分，它与通用处理器最大的区别在于嵌入式处理器大多工作在为特定用户群专门设计的系统中，它将通用处理器中许多由板卡完成的任务集成到芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时使系统具有很高的效率和可靠性。如今，全世界嵌入式处理器已经超过 1000 多种，流行的体系结构有 30 多个系列，其中以 ARM、PowerPC、MC 68000、MIPS 等的使用最为广泛。

外围设备是嵌入式系统中用于完成存储、通信、调试、显示等辅助功能的其他部件。目前常用的嵌入式外围设备按功能可以分为存储设备（如 RAM、SRAM、Flash 等）、通信设备（如 RS-232 接口、SPI 接口、以太网接口等）和显示设备（如显示屏等）3 类。

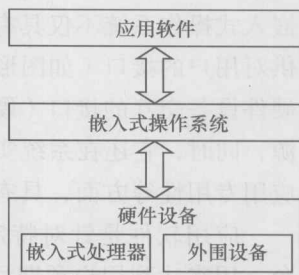


图 1-1 嵌入式系统结构简图



常见存储器 RAM、SRAM、SDRAM、ROM、EPROM、EEPROM、Flash 概念辨析

存储器可以分为很多种类，其中根据掉电后数据是否丢失可以分为 RAM（随机存储器）和 ROM（只读存储器），其中 RAM 的访问速度比较快，但掉电后数据会丢失，而 ROM 掉电后数据不会丢失。人们通常所说的内存即指系统中的 RAM。

RAM 又可分为 SRAM（静态存储器）和 DRAM（动态存储器）。SRAM 是利用双稳态触发器来保存信息的，只要不掉电，信息是不会丢失的。DRAM 是利用 MOS（金属氧化物半导体）电容存储电荷来储存信息，必须通过不停地给电容充电来维持信息，因此 DRAM 的成本、集成度、功耗等明显优于 SRAM。

而通常人们所说的 SDRAM 是 DRAM 的一种，它是同步动态存储器，利用一个单一的系统时钟同步所有的地址数据和控制信号。使用 SDRAM 不但能提高系统表现，还能简化设计、提供高速的数据传输。在嵌入式系统中经常使用。

EPROM、EEPROM 都是 ROM 的一种，分别为可擦除可编程 ROM 和电可擦除 ROM，但使用不是很方便。

Flash 也是一种非易失性存储器（掉电不会丢失），它擦写方便，访问速度快，已大大取代了传统的 EPROM 的地位。由于它具有和 ROM 一样掉电后数据不会丢失的特性，因此很多人称其为 Flash ROM。



1.2.2 嵌入式系统的软件组成

在嵌入式系统不同的应用领域和不同的发展阶段,嵌入式系统的软件组成也不完全相同。其基本组成如图 1-2 所示。

如图 1-2 左侧所示,在某些特殊领域中,嵌入式系统软件没有使用通用计算机系统那样的结构。嵌入式操作系统从嵌入式发展的第 3 阶段起开始引入。

嵌入式操作系统不仅具有通用操作系统的一般功能,如向上提供对用户的接口(如图形界面、库函数 API 等),向下提供与硬件设备交互的接口(硬件驱动程序等),管理复杂的系统资源,同时,它还在系统实时性、硬件依赖性、软件固化性以及应用专用性等方面,具有更加鲜明的特点。

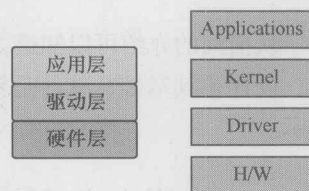


图 1-2 嵌入式系统软件组成图

应用软件是针对特定应用领域,基于某一固定的硬件平台,用来达到用户预期目标的计算机软件。嵌入式系统自身的特点,决定了嵌入式应用软件不仅要求达到准确性、安全性和稳定性等方面的要求,而且还要尽可能地进行代码优化,以减少对系统资源的消耗,降低硬件成本。

1.3 嵌入式操作系统举例

嵌入式操作系统主要有商业版和开源版两大阵营,从长远看,嵌入式系统开源、开发将是其发展趋势。

1.3.1 商业版嵌入式操作系统

下面简单地列举 VxWorks、Windows CE 两种商业版嵌入式操作系统。

1. VxWorks

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统(RTOS),它是当前市场占有率最高的嵌入式实时操作系统。VxWorks 的实时性做得非常好,其系统本身的开销很小,进程调度、进程间通信、中断处理等系统公用程序精练而有效,使得它们造成的延迟很短。另外,VxWorks 提供的多任务机制,对任务的控制采用了优先级抢占(Linux 2.6 内核也采用了优先级抢占的机制)和轮转调度机制,充分保证了可靠的实时性,并使同样的硬件配置能满足更强的实时性要求。另外,VxWorks 具有高度的可靠性,从而保证了用户工作环境的稳定。同时,VxWorks 还有很完备强大的集成开发环境,这也大大方便了用户的使用。

但是,由于 VxWorks 的开发和使用都需要交高额的专利费,因此大大增加了用户的开发成本。同时,由于 VxWorks 的源码不公开,造成它部分功能的更新(如网络功能模块)滞后。

2. Windows CE

Windows CE 是微软开发的一个开放的、可升级的 32 位嵌入式操作系统,是基于掌上型



电脑类的电子设备操作系统。它是精简的 Windows 95。Windows CE 的图形用户界面相当出色,而且具有模块化、结构化和基于 Win32 应用程序接口以及与处理器无关等特点。它继承了传统的 Windows 图形界面,用户在 Windows CE 平台上不仅可以使 Windows 95/98 上的编程工具(如 Visual Basic、Visual C++等),也可以使用同样的函数、同样的界面风格,使绝大多数 Windows 上的应用软件只需简单地修改和移植就可以在 Windows CE 平台上继续使用。但与 VxWorks 相同,Windows CE 也是要收费的,但相对于 VxWorks 要经济很多。

1.3.2 开源版嵌入式操作系统

下面简单地介绍嵌入式 Linux 开源版嵌入式操作系统。

嵌入式 Linux (Embedded Linux) 是指对标准 Linux 经过小型化裁剪处理之后,能够固化在容量只有几 KB 或者几 MB 字节的存储器芯片或者单片机中,是适合于特定嵌入式应用场合的专用 Linux。在目前已经开发成功的嵌入式系统中,大约有一半使用的是 Linux,这与它自身的优良特性是分不开的。

嵌入式 Linux 同 Linux 一样,具有低成本、多种硬件平台支持、优异的性能和良好的网络支持等优点。另外,为了更好地适应嵌入式领域的开发,嵌入式 Linux 还在 Linux 基础上做了部分改进,如下所示。

(1) 改善了内核结构。

Linux 内核采用的是整体式结构 (Monolithic),整个内核是一个单独的、非常大的程序,这样虽然能够使系统的各个部分直接沟通,提高系统响应速度,但与嵌入式系统存储容量小、资源有限的特点不相符合。因此,嵌入式系统经常采用的是另一种称为微内核 (Microkernel) 的体系结构,即内核本身只提供一些最基本的操作系统功能,如任务调度、内存管理、中断处理等,而类似于文件系统和网络协议等附加功能则运行在用户空间中,并且可以根据实际需要取舍。这样就大大减小了内核的体积,便于维护和移植。

(2) 提高了系统实时性。

现有的 Linux 是一个通用的操作系统,虽然它也采用了许多技术来加快系统的运行和响应速度,但从本质上来说并不是一个嵌入式实时操作系统。因此,利用 Linux 作为底层操作系统,在其上层进行实时化改造,从而构建出一个具有实时处理能力的嵌入式系统,如 RT-Linux 已经成功地应用于航天飞机的空间数据采集、科学仪器测控和电影特技图像处理等各种领域。

嵌入式 Linux 同 Linux 一样,也有众多的版本,其中不同的版本分别针对不同的需要在内核等方面加入了特定的机制。嵌入式 Linux 的主要版本见表 1-2。

表 1-2

嵌入式 Linux 的主要版本

版 本	简 单 介 绍
μCLinux	开放源码的嵌入式 Linux 的典范之作。它主要针对目标处理器没有存储管理单元 MMU。其运行稳定,具有良好的移植性和优秀的网络功能,对各种文件系统有完备的支持,并提供标准丰富的 API
RT-Linux	由美国墨西哥理工学院开发的嵌入式 Linux 实时操作系统
Embedix	根据嵌入式应用系统的特点重新设计的 Linux 发行版本。它提供了超过 25 种的 Linux 系统服务,包括 Web 服务器等。此外还推出了 Embedix 的开发调试工具包、基于图形界面的浏览器等。Embedix 是一种较完整的嵌入式 Linux 解决方案