



嵌入式系统 应用基础

■ 赵成 主编

—基于S3C2410A的
SKYEYE仿真与实践



国防工业出版社

National Defense Industry Press

嵌入式系统应用基础

——基于 S3C2410A 的 SKYEYE 仿真与实践

赵 成 主编

国防工业出版社

·北京·

内 容 简 介

本书以三星 (samsung) 公司的应用型微处理器 S3C2410A 为基础, 结合虚拟仿真软件 SKYEYE, 全面系统地讲述基于 ARM 公司的 ARM920T 微处理器内核的嵌入式 Linux 系统的开发方法。

本书按照实际的学习与实践流程编排内容结构: 首先介绍嵌入式系统的概念; 然后介绍 ARM 微处理器的体系结构与编程方法; 接着在 VMWARE 虚拟机上建立 Linux 系统; 随后介绍 Linux 系统基础与交叉编译环境; 接下来介绍虚拟仿真软件 SKYEYE, 并介绍了 BootLoader、Linux 内核、根文件系统三个基础内容的应用与仿真; 详细介绍了嵌入式 Linux 设备驱动程序的开发; 最后, 以一个具体的开源游戏模拟器项目为例, 讲述开源软件的应用方法。

本书可作为电子信息、自动化、通信工程、机电一体化、计算机等专业高年级本科生及研究生的教材, 也可供从事 ARM 嵌入式系统开发的工程技术人员参考。

图书在版编目 (CIP) 数据

嵌入式系统应用基础: 基于 S3C2410A 的 SKYEYE 仿真
与实践/赵成主编. —北京: 国防工业出版社, 2012. 2

ISBN 978-7-118-07953-1

I. ①嵌… II. ①赵… III. ①微型计算机 - 系统开发
IV. ①TP360. 21

中国版本图书馆 CIP 数据核字 (2012) 第 012397 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

腾飞印务有限公司

新华书店经售

*

开本 787×1092 1/16 印张 17½ 字数 397 千字

2012 年 2 月第 1 版第 1 次印刷 印数 1—4000 册 定价 35.00 元

(本书如有印装错误, 我社负责调换)

国防书店: (010) 88540777 发行邮购: (010) 88540776

发行传真: (010) 88540755 发行业务: (010) 88540717

前 言

随着 ipad、iphone、MP5、数码相机等数字消费产品市场的成功，带动了业界对智能多媒体电路系统设计与开发的需求，而嵌入式微处理器能够很容易地满足这类系统设计的需求，因此，目前嵌入式系统的开发与应用正在迅速普及，嵌入式系统变得炙手可热。现在，市场上有多种架构的嵌入式微处理器，作为学习嵌入式系统的入门基础，书中基于三星公司的 S3C2410A 微处理器进行讲授，三星公司为该微处理器推出的 SMDK2410 原型评估板应用十分广泛。S3C2410A 微处理器在智能仪器仪表、数字录播设备、娱乐数码产品等领域都有广泛的应用。

现在，众多准备从事嵌入式系统设计与开发的工程技术人员，正徘徊在嵌入式系统的门槛前，其中硬件工程技术人员会感觉嵌入式系统与其他微控制器或微处理器的开发方法有很大的不同，软件设计人员在开发软件时还需要了解电路的配置从而也感觉到无所适从。本书定位于嵌入式系统的入门教程，目标是让读者接触并掌握嵌入式系统的开发与应用基础，为此，在内容编排上不再具体讲解 ARM 微处理器的硬件体系结构，只在应用时在相应章节独立介绍，重点是介绍基于 ARM 嵌入式系统的软件开发，循序渐进地讲解每个知识点；另外，在内容编排上不再具体讲解 ARM 微处理器的汇编语言程序设计，因为在进行应用软件开发时几乎不再使用汇编语言编程，目前，只在 BootLoader 程序的设计中还使用汇编语言进行程序设计。最后，以 S3C2410A 微处理器为核心的嵌入式电路系统的设计原理图可以参阅三星公司推出的 SMDK2410 原型评估板电路图，这里不再给出电路原理图。

本书选用 Linux 2.6.14 内核作为嵌入式 Linux 操作系统，介绍嵌入式 Linux 系统的开发方法，嵌入式系统的软件开发具有类似于计算机软件的开发特点，如果已经具有计算机软件设计与开发的能力，对于本书中的设备驱动部分会很容易理解。

书中将重点内容分为 4 个层次：首先，了解并掌握 Linux 系统的基础应用及嵌入式交叉编译环境的建立，为开发嵌入式系统准备宿主机环境；其次，了解并掌握嵌入式系统中的 BootLoader、Linux 内核、根文件系统等“三个重要基础部分”的应用，这部分是嵌入式系统设计中的基础部分；然后，了解并掌握嵌入式字符型设备驱动程序的设计，这是对微处理器内部的功能模块进行开发的基础；最后，了解并掌握开源软件的应用，现在 Linux 系统环境下有大量成熟的开源软件，可以将这些优秀的软件直接应用在 ARM 嵌入式系统中。

本书按照实际的嵌入式系统实践步骤编排所有内容，只要读者能够跟着书中的内容顺序学习并花时间多进行实践，应该能够很快掌握嵌入式系统开发中的“三个重要基础部分”、字符型设备驱动、开源软件的应用，具有进行嵌入式系统基础部分的设计与开发能力。更进一步地学习嵌入式系统的设计与开发，还需要了解嵌入式系统中的用户图形界面（GUI），限于本书的篇幅，这里没有介绍 GUI 的内容。

为了使广大读者能够简单快速地掌握嵌入式系统的设计与开发，本书以“面向基础、配合图例、注重引导、应用全面”为原则，通过一系列完整的设计与开发过程，包括 VMWARE 虚拟机的安装与设置、Linux 操作系统的安装与服务配置、Linux 系统基础应用的设计、嵌入式

交叉编译环境的建立、BootLoader 程序的应用、Linux 内核的配置、根文件系统的制作、设备驱动程序及应用程序的设计、开源软件的应用实现、软件模拟仿真嵌入式系统的实例等，使读者能够边学习边练习，将理论与实践相结合。

其中，特别要指出的是，为了便于课堂讲授及练习，本书给出了使用 SKYEYE 软件模拟仿真运行 BootLoader 程序、Linux 内核、根文件系统以及设备驱动程序方法，使读者在没有评估板与仿真调试器等硬件设备的情况下也能够学习嵌入式系统的应用开发设计。

作者结合 S3C2410A 嵌入式微处理器相关工程项目的开发经验、教学实例及课程设计实例编写此书。书中主要描述的基于 S3C2410A 的嵌入式 Linux 系统开发部分、基于虚拟仿真软件 SKYEYE 模拟嵌入式 Linux 系统的运行部分、应用字符型设备接口驱动程序的框架开发设备驱动程序部分、片内功能模块的设计实例部分、开源游戏软件的应用实例部分等都有相应的软件及源码。书中讲解的所有程序及实例都在 VMWARE 虚拟机中的 RedHat EL AS4 系统以及 S3C2410A 评估板上作为实践项目得到了实际应用，书中大量的图例可以为用户的学习与开发提供导引及技术参考。如果读者需要相关的硬件及相关的技术咨询可以发送邮件到 ziapc@yahoo.cn。

全书共 10 章。第 1 章介绍嵌入式微处理器的基础知识；第 2 章介绍基于 ARM 架构的嵌入式微处理器的体系结构及编程方法；第 3 章介绍 VMWARE 虚拟机与 RedHat Linux 系统环境的建立，并配置 Samba 服务及共享功能；第 4 章介绍 Linux 系统的开发基础及嵌入式交叉编译工具的安装；第 5 章介绍嵌入式系统的仿真软件 SKYEYE；第 6 章介绍 U-Boot 程序的编译、仿真及扩展；第 7 章介绍 Linux 内核的配置、编译及仿真调试；第 8 章介绍根文件系统的制作与仿真；第 9 章介绍嵌入式 Linux 设备驱动程序的开发；第 10 章介绍开源游戏模拟器 Visual Boy Advance 的应用。本书提供大量的图表作为读者学习与实践的指导，书中每章均配有习题，便于读者掌握和巩固知识。使用本教材的参考学时为 40~50 学时。

本书由赵成主编。第 1 章由华红艳编写；第 2 章由赵建林编写；第 3 章由梁明亮编写；第 4、5 章由张松炜编写；第 6、7、8、9 章由赵成编写；第 10 章由薛文龙编写。全书由赵成统稿。

在本书的撰写过程中，参考了大量的文献资料，在此谨向其作者表示衷心的感谢。

限于编者水平，书中难免存在错误和不当之处，恳请读者批评指正。

编 者

2011 年 12 月

目 录

第 1 章 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 嵌入式系统的结构	1
1.1.3 嵌入式系统的特点	3
1.1.4 嵌入式系统的开发流程	5
1.1.5 嵌入式系统的应用	7
1.2 嵌入式微处理器	8
1.2.1 嵌入式微处理器	8
1.2.2 基于 ARM 架构的嵌入式微处理器	9
1.2.3 基于 MIPS 架构的嵌入式微处理器	10
1.2.4 基于 PowerPC 架构的嵌入式微处理器	12
1.2.5 系统级芯片 SoC	13
1.2.6 嵌入式微处理器的发展趋势	14
1.3 嵌入式系统的软件	14
1.3.1 BootLoader	14
1.3.2 嵌入式操作系统	15
1.3.3 应用程序	16
1.4 嵌入式系统的课程特点	17
本章小结	17
习题	18
第 2 章 ARM 体系结构与编程方法	20
2.1 ARM 微处理器概述	20
2.1.1 ARM 微处理器的特点	20
2.1.2 ARM 微处理器的发展	21
2.1.3 ARM 微处理器 S3C2410A 的应用资源	23
2.2 ARM 体系结构	25
2.2.1 工作模式和指令状态	25
2.2.2 ARM 寄存器	27
2.2.3 ARM 体系的异常中断	29
2.2.4 ARM 微处理器的寻址方式	30
2.3 ARM 编程方法	33
2.3.1 ARM 指令概述	33
2.3.2 ARM 汇编程序设计	34

2.3.3	ARM 混合编程	35
2.4	ARM 系统程序设计基础	37
2.4.1	ADS1.2 集成开发环境简介	37
2.4.2	ADS1.2 集成开发环境下工程的创建	37
	本章小结	43
	习题	44
第 3 章	VMware 虚拟机与 Linux 环境的建立	45
3.1	安装 VMware Workstation 7	45
3.2	安装 Red Hat Enterprise Linux 4	50
3.3	配置 Samba 服务	70
3.4	VMware 虚拟机共享设置	77
	本章小结	82
	习题	83
第 4 章	Linux 开发基础	84
4.1	Linux 系统简介	84
4.1.1	Linux 系统的组成	84
4.1.2	Linux 文件系统的目录结构	85
4.1.3	Linux 系统的特点	86
4.2	Linux 基本命令	87
4.2.1	文件命令	87
4.2.2	目录命令	88
4.2.3	多用户命令与安全性命令	89
4.2.4	其他常用命令	90
4.3	文本编辑器 vi	91
4.3.1	vi 的模式	91
4.3.2	进入与退出 vi	91
4.3.3	编辑模式	92
4.3.4	命令模式	93
4.3.5	末行模式	94
4.4	Linux 函数库	94
4.4.1	静态函数库	94
4.4.2	共享函数库	94
4.4.3	函数库和头文件的保存位置	95
4.5	GCC 及其使用	95
4.5.1	GCC 工具简介	95
4.5.2	可执行文件的格式	96
4.5.3	GCC 的使用	96
4.6	GDB 调试器	99
4.6.1	GDB 的安装与使用	99
4.6.2	基本的 GDB 命令	101

4.6.3	GDB 程序远程调试	101
4.7	GNU make 和 Makefile	104
4.7.1	make 概述	104
4.7.2	make 命令格式	104
4.7.3	Makefile 的基本结构	105
4.7.4	Makefile 的变量与函数	106
4.7.5	Makefile 的隐含规则	108
4.8	Linux Shell 编程	109
4.8.1	Shell 的变量	109
4.8.2	条件语句与循环语句	113
4.8.3	shell 函数的定义与使用	115
4.9	嵌入式交叉编译环境的搭建	116
	本章小结	117
	习题	118
第 5 章	嵌入式系统的仿真软件 SKYEYE	119
5.1	SKYEYE 硬件模拟平台	119
5.1.1	SKYEYE 介绍	119
5.1.2	SKYEYE 系统的原理	120
5.1.3	SKYEYE 的设计实现	122
5.2	SKYEYE 的使用	123
5.2.1	SKYEYE 的安装	123
5.2.2	SKYEYE 的配置文件	125
5.2.3	SKYEYE 程序的执行	129
5.3	SKYEYE 的结构与执行流程	130
5.3.1	SKYEYE 的目录结构	130
5.3.2	SKYEYE 的关键数据结构	132
5.3.3	SKYEYE 的执行流程	137
	本章小结	139
	习题	139
第 6 章	BootLoader 程序	140
6.1	BootLoader 程序的原理	140
6.1.1	BootLoader 的概念	140
6.1.2	BootLoader 程序的原理	140
6.2	U-Boot	142
6.2.1	U-Boot 简介	142
6.2.2	U-Boot 的设计结构	143
6.2.3	U-Boot 的编译	146
6.2.4	U-Boot 的 SKYEYE 仿真	147
6.2.5	U-Boot 的常用命令	148
6.2.6	U-Boot 添加命令的方法	150

6.2.7	U-Boot 添加板级支持包的方法	154
6.2.8	U-Boot 添加驱动的方法	155
6.2.9	U-Boot 的固化	160
6.3	BootLoader 开发工具概览表	162
	本章小结	162
	习题	162
第 7 章	Linux 内核的配置、编译与调试	163
7.1	Linux 2.6 内核特性	163
7.1.1	Linux 2.6 内核简介	163
7.1.2	Linux 2.6 内核的特性	163
7.2	Linux 内核配置	165
7.2.1	下载 Linux 内核	165
7.2.2	修改 Makefile 文件	165
7.2.3	配置内核	166
7.2.4	更改程序运行的物理地址	181
7.3	Linux 内核的编译	182
7.4	基于 SKYEYE 的 Linux 系统仿真	182
7.4.1	复制 vmlinux 到 SKYEYE	182
7.4.2	建立 skyeye.conf 配置文件	183
7.4.3	复制文件系统 initrd.img 到 SKYEYE	183
7.4.4	使用 SKYEYE 仿真 Linux 系统	184
	本章小结	184
	习题	185
第 8 章	根文件系统的制作	186
8.1	根文件系统	186
8.1.1	根文件系统概述	186
8.1.2	常见的文件系统	187
8.2	BusyBox	189
8.2.1	BusyBox 简介	189
8.2.2	配置 BusyBox	189
8.2.3	编译与安装 BusyBox	198
8.3	根文件系统的制作过程	199
8.3.1	根文件系统的结构	199
8.3.2	创建根文件系统的 dev 目录	200
8.3.3	创建根文件系统的 etc 等目录及配置文件	200
8.3.4	创建 EXT2 格式的文件系统映像	202
8.3.5	添加 initrd.img 文件系统映像的内容	202
8.4	基于 SKYEYE 的根文件系统仿真调试	203
	本章小结	204
	习题	204

第 9 章 嵌入式 Linux 设备驱动程序开发	205
9.1 Linux 设备驱动程序简介	205
9.1.1 Linux 设备驱动程序的引入	205
9.1.2 Linux 设备驱动程序的作用	205
9.1.3 Linux 设备驱动程序的基础与特点	206
9.1.4 Linux 的设备驱动开发调试方法	207
9.1.5 设备驱动程序与应用程序的区别	207
9.1.6 查看 Linux 系统中的设备	208
9.1.7 主设备号和次设备号	208
9.1.8 设备文件系统	209
9.1.9 file_operations 结构体	209
9.1.10 file 结构体	211
9.2 设备驱动程序的分类	212
9.2.1 设备驱动程序的分类	212
9.2.2 字符设备驱动的主要内容	213
9.2.3 块设备驱动的主要内容	214
9.2.4 网络设备驱动的主要内容	215
9.3 字符设备驱动程序的实现过程	216
9.3.1 字符设备驱动程序的代码框架	216
9.3.2 字符设备驱动程序中的方法	218
9.4 Linux 字符设备驱动程序源码分析	221
9.4.1 字符型设备驱动程序源代码	221
9.4.2 设备驱动程序的编译与加载	225
9.4.3 用户测试程序	226
9.4.4 驱动程序的测试	229
9.5 添加 LCD 显示驱动程序	232
9.5.1 S3C2410A 的 LCD 控制器	232
9.5.2 添加 LCD 控制器的程序代码	234
9.5.3 添加 Linux 的 LOGO 显示	237
9.6 编译设备驱动到内核	239
9.6.1 建设备驱动程序	239
9.6.2 加载驱动模块到内核	240
9.6.3 内核自动装载驱动模块	242
9.7 S3C2410A 芯片 GPIO 接口的驱动开发	243
9.7.1 GPIO 接口的描述	243
9.7.2 GPIO 接口的驱动程序设计	244
9.7.3 GPIO 接口的应用程序设计	252
本章小结	253
习题	254

第 10 章 开源游戏模拟器的应用	255
10.1 Visual Boy Advance	255
10.1.1 游戏模拟器简介	255
10.1.2 Visual Boy Advance 模拟器	256
10.1.3 Visual Boy Advance 的主要功能	256
10.2 Visual Boy Advance 的配置软件	257
10.2.1 安装交叉编译工具	257
10.2.2 ZLIB 数据压缩库	258
10.2.3 libpng 图像库	259
10.2.4 SDL 多媒体函数库	260
10.2.5 编译与安装 VBA 模拟器	262
10.3 Visual Boy Advance 的调试与运行	263
10.4 Visual Boy Advance 的程序设计	265
10.4.1 VBA 程序设计简介	265
10.4.2 VBA 程序的开发工具	265
10.4.3 一个简单 GBA 应用程序的设计示例	266
本章小结	268
习题	268

第1章 概述

嵌入式系统是体现近年来微电子技术及计算机技术发展的高科技应用系统，它为人们设计了极为简便却仍具有标准 PC（个人计算机）功能的设备，给人们的生产与生活带来了极大的便利与丰富的乐趣，任何人都可能拥有大小不一、形状各异、使用嵌入式技术的电子产品，小到 MP3、PDA 等微型数字化产品，大到网络家电、智能家电、车载电子设备等。在工农业领域中，使用嵌入式技术的数字机床、智能仪表、工业机器人和服务机器人正在逐渐改变着传统的工农业生产方式。目前，嵌入式系统设备在应用数量上已经远远超过了通用计算机。

本章主要介绍有关嵌入式系统的基础知识。首先从嵌入式系统的基本概念开始，介绍其结构、特点、开发流程以及应用。接着介绍嵌入式微处理器，阐述不同的嵌入式微处理器架构。之后简要介绍常见的嵌入式操作系统 Linux、WinCE.net、VxWorks 等。最后介绍嵌入式技术发展现状及趋势。

学习目标：

- 嵌入式系统的概念
- 嵌入式微处理器
- 嵌入式系统的软件
- 嵌入式系统的课程特点

1.1 嵌入式系统的概念

1.1.1 嵌入式系统的定义

英国的电气工程师学会（IEE）对嵌入式系统的定义是用来控制、监视或辅助设备、机器和工厂运行的装置。

从这个概念上来认识嵌入式系统，它应该是一个小型的核心装置，能够进行数据采集，有分析与决策能力，可以通过通信接口或机械部分控制受控的设备、机器或工厂的运行，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中应该具备硬件和软件两个载体，另外还可以涵盖机械等附属装置。但实际上，应用嵌入式系统的产品远远不止这些，类型众多的消费数码产品，例如，MP3、数码相机、游戏机中也应用嵌入式系统。

为了充分体现嵌入式系统的类计算机特性，目前国内普遍认同的一个定义：以应用为中心，以计算机技术为基础，软件硬件可裁剪，满足应用系统对功能、可靠性、成本、体积、功耗等有特殊要求的专用计算机系统。

1.1.2 嵌入式系统的结构

嵌入式系统是面向特定应用的，在客观的需求中，绝大多数嵌入式系统是由用户针对特定任务而定制的，各个系统的功能不同、功耗不同、人机接口也不同，但就其实质而言，它们

一般都是可视为由硬件系统、软件系统两大部分组成的。

在软件系统中，又可以视为由嵌入式操作系统、驱动程序、应用程序、图形用户界面等组成的。

1. 嵌入式系统的软硬件结构

嵌入式系统包括硬件与软件两部分。硬件部分包括嵌入式微处理器，外围硬件设备（存储器、外设器件、I/O 端口及图形控制器等）。软件部分包括操作系统软件（EOS）（要求实时、多任务操作）、驱动程序以及用户的应用程序等。

嵌入式微处理器是整个嵌入式系统的核心，硬件系统的电路配置及软件系统的开发环境都取决于应用系统中的嵌入式微处理器。

(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、存储器(SDRAM、ROM 等)、外设器件、设备 I/O 端口及图形控制器等。

- 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。
- 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 Cache、主存和辅助存储器。常用作主存的存储器有 ROM 类(NOR Flash、EPROM 和 PROM)存储器及 RAM 类(SRAM、DRAM 和 SDRAM)存储器。辅助存储器用来存放大数据量的程序代码或信息，它的容量大，但读取速度与主存相比就慢的很多，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。
- 外设器件。主要是用来扩展片内外设的外围通信接口电路，以及进行电机控制的驱动电路等。
- 设备 I/O 端口。主要提供系统与内部或者外部的硬件接口，如通过 I/O 实现内部 A/D 转换，或者通过 I/O 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) 驱动层：在应用中，嵌入式系统的硬件部分中采用的微处理器多种多样，如何使软件部分的操作系统能简洁有效地应用于不同的硬件环境，是嵌入式系统设计中必须解决的问题。经过不断发展，在硬件部分与软件部分之间，开发出一个中间层，称为“硬件抽象层(HAL)”。

HAL 往往独立于内核，由 CPU 的厂商提供，与 BIOS 很相似。也有的厂商（如 VxWorks 的提供者 WindRiver 公司）把硬件抽象层称为 BSP，即板级支持包（Board Support Package）。

“硬件抽象层”或“板级支持包”是操作系统与目标应用硬件环境的中间接口，将操作系统与目标应用硬件环境的硬件连接在一起，具有硬件相关性，包含了大量与微处理器及设备驱动相关的代码和数据结构。

(3) 系统层：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(Graphic User Interface, GUI)接口、网络系统及通用组件模块组成。其中实时多任务操作系统是整个嵌入式系统开发的软件基础和平台。

(4) 应用软件层：应用软件层则是设计开发人员在系统层与软件层的基础之上，根据具体需要而实现的功能，在嵌入式系统中，用户在应用软件层所开发的软件应该结合系统的硬件环境来实现，通常需要通过驱动层间接地访问硬件资源。

应用软件层是嵌入式系统开发过程中最重要的环节之一。

通过上述介绍，可以用图 1-1 来说明整个嵌入式系统的组成结构，包括硬件层、硬件抽

象层（驱动层）、嵌入式操作系统层（OS 层）及应用层。

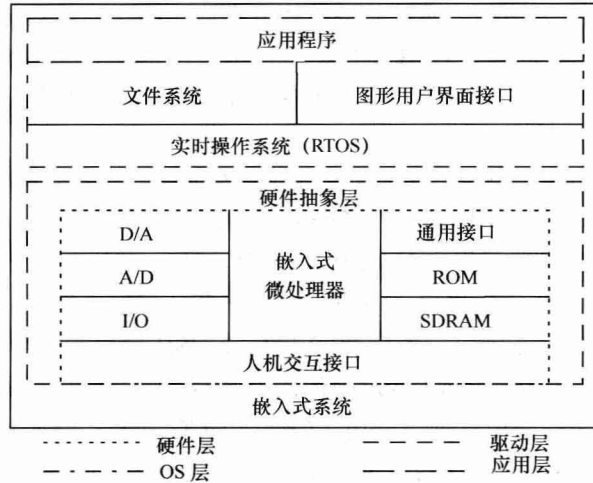


图 1-1 嵌入式系统的结构

2. 嵌入式系统的硬件平台配置

嵌入式微处理器自己是不能独立工作的，需要必要的外围芯片给它提供基本的工作条件。通常情况下，要想使其运行，必须有电源系统对其供电；必须有时钟系统提供时钟信号；必须有复位系统。同时还要设计程序下载调试的接口（JTAG），通过这些基本的外围电路构成 ARM 最小系统。如果芯片内部没有存储器，还要进行一定的存储器扩展，才能固化程序，使系统运行。

人机交互接口界面（包括 LCD 显示屏与键盘电路）、RS-232 通信接口、A/D 转换、直流电机 PWM 驱动等都是重要的组成部分，所以在规划硬件架构时应设计相关的功能电路。如图 1-2 所示。

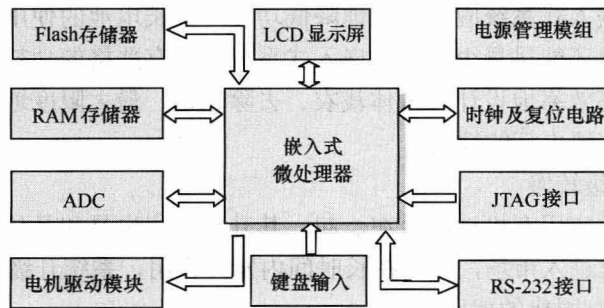


图 1-2 嵌入式系统硬件平台配置

3. 嵌入式系统的开发工具和开发系统

嵌入式系统的硬件和软件处于嵌入式系统产品本身之中，开发工具则独立于嵌入式系统产品之外。开发工具一般用于开发主机，包括编译器、汇编器、链接定位器、调试器等，这些工具一起构成了嵌入式系统的开发系统和开发工具。

1.1.3 嵌入式系统的特点

嵌入式系统是应用于特定环境下，针对特定用途来设计的系统，其硬件和软件都必须高效率地设计。与通用计算机系统相比，嵌入式系统具有以下特点。

1. 嵌入式系统是类计算机系统

标准的 PC 以 CPU 为核心，通过北桥芯片分别连接高速的内存储器、显示设备使用的 PCI-E 接口，通过南桥芯片扩展出 USB 接口、串行接口、并行接口、PS/2 接口、AC'97 音频电路（及接口）、网络电路（及接口）等，一般提供标准的人机接口设备（键盘、鼠标及显示器）。

嵌入式系统与标准的 PC 类似，以嵌入式微处理器为核心，内部集成 Cache 及 RAM 存储器，外部可扩展外存储器，显示设备通过并行 I/O 口、LCD 接口或 SPI 接口连接 LCD 显示屏，嵌入式微处理器的片内外设单元提供 USB 接口、SPI 接口、IIC 接口、IIS 接口、串行接口、通用 I/O 口、摄像头接口等通信接口，可以外扩网络芯片来提供标准的网络接口，可以外扩 SD 卡接口，也可以外扩 PS/2 接口，支持 PS/2 接口或 USB 接口的键盘与鼠标。

标准的 PC 可以安装 Windows 或 Linux 操作系统，嵌入式系统可以固化 WinCE .net 或 Linux 操作系统。标准的 PC 上的应用软件可以用 VC++ 或 GCC 开发，嵌入式系统的应用软件可以用 EVC 或 GCC 开发。嵌入式系统与标准的 PC 系统在软硬件上都有相似之处，因此说嵌入式系统是类 PC 系统。

2. 嵌入式系统通常是面向特定应用的

嵌入式 CPU 与通用计算机的最大不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中，专用于某个或少数几个特定的任务。嵌入式系统处理器通常都具有低功耗、体积小、集成度高等特点，能够把通用 PC 系统中的芯片组及许多板卡完成的任务集成在 CPU 内部，从而有利于嵌入式系统设计趋于小型化，移动能力大大增强，便于嵌入各类的应用设计中。

3. 嵌入式系统更注重低功耗、低成本

与标准的 PC 系统不同，嵌入式系统可能“嵌入”在各类不同的设备上，其中便携式嵌入设备占了很大的比重，比如便携式仪器仪表、随身数码视听设备等，这些设备往往使用电池供电，不可能有充足的电能供应，为了满足便携的舒适性，体积也不会太大，相对来说不会配备散热设备，因此，嵌入式系统应尽可能地降低功耗，延长电池的使用时间，使系统能运行较长的时间并且发热量还能尽量少。整个嵌入式系统设计有严格的功耗预算，其硬件和软件也必须围绕这种特性高效率地设计，量体裁衣、去除冗余，最大限度地降低应用成本，力争在同样的硅片面积上实现更高的性能。

4. 嵌入式系统升级较慢

嵌入式系统和具体应用有机地结合在一起，其升级换代也是和具体的产品同步进行，因此嵌入式系统产品一旦进入市场，便会在长时间内发挥作用，系统升级周期长。

5. 嵌入式系统具有固化的代码

为了提高系统的可靠性和减小系统尺寸，嵌入式系统中的软件一般都固化在 Nor 型或者 Nand 型 Flash ROM 存储芯片中，也可以存储在嵌入式微处理器芯片内部的 Flash ROM 中。嵌入式系统通常并不使用磁盘等存储设备。固化代码是不能被改变的，除非开发人员采用特定的调试设备才能对代码进行升级并重新写入系统。

6. 嵌入式系统开发需要专用开发工具和环境

通用计算机具有完善的人机接口界面，在其中增加一些开发应用程序和环境即可进行对自身的开发；而嵌入式系统本身并不具备自主开发能力，即使设计完成以后用户通常也不能对其中的程序功能进行修改，必须有一套开发工具和环境基于通用计算机的软硬件设备以及各种逻辑分析仪、混合信号示波器等才能进行开发。开发时往往有宿主机和目标机的概念，

宿主机用于程序的开发，目标机作为最后的执行机，开发时需要进行远程调试。

7. 嵌入式系统软件需要 RTOS 开发平台

通用计算机具有完善的操作系统和应用程序接口（API），是计算机基本组成不可分割的一部分，应用程序的开发以及完成后的软件都在操作系统（OS）平台上运行，但一般不是实时的。嵌入式系统则不同，其应用程序可以没有操作系统而直接在芯片上运行；但是为了合理地调度多任务，利用系统资源、系统函数以及专家库函数接口，嵌入式系统一般使用实时操作系统 RTOS，使系统具有实时约束。用户必须自行选配 RTOS 开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，提高软件质量。

8. 嵌入式系统开发人员应具备电路专业知识

通用计算机的开发人员一般是计算机科学或计算机工程方面的专业人士，而嵌入式系统则是要和各个不同行业的应用相结合的，要求更多的计算机以外的专业知识，其开发人员往往是通信工程、电子信息工程、电路与系统、自动控制等专业的人员，因此对非计算机专业人员来说，使用的软件开发环境应该易学、易用、可靠、高效。

9. 嵌入式系统是开放性的知识集成系统

嵌入式系统是将计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物，是一门综合技术学科，是技术密集、资金密集、高度分散、不断创新的知识集成系统。与通用计算机行业不同，嵌入式系统中没有哪一个系列的处理器和操作系统能够像 Wintel（Microsoft 和 Intel 公司联盟）垄断 PC 市场那样垄断嵌入式系统领域。

目前，虽然嵌入式系统的处理器在体系结构上存在着主流，但各不相同的应用领域决定了不可能由其垄断全部市场。嵌入式系统领域的产品和技术应用广泛而灵活，留给各行各业的中、小规模高新技术公司的创新余地很大。应用领域的需求在不断提高，相应的嵌入式处理器架构设计也在飞速发展。现在，嵌入式系统以其开放性的高知识高技术特点，在复杂性、实用性和高效性等方面都达到了一个前所未有的高度。

1.1.4 嵌入式系统的开发流程

嵌入式系统的开发流程在这里是指嵌入式系统整体设计的一般过程和主要步骤。在此将采用任务追踪的方法，从任务分析开始，对系统做最抽象的描述；然后逐步地详细构造系统内部的软硬件模块，进行硬件系统设计、软件系统设计；接着，在设计的软件系统平台上，开发人员进行驱动程序与应用软件开发，然后进行程序的交叉编译并下载程序到硬件电路系统中；在这些工作的基础上，就可以构造出所需要的完整系统了，因此最后进行系统的集成与调试。这一过程与传统的系统开发步骤是一致的。图 1-3 所示为嵌入式系统的开发流程，下面将说明每一步骤应实现的功能。

1. 任务分析阶段

设计应该以客户需求为中心，在开始设计之前，需要充分地了解用户需求、理解用户需求。作为嵌入式系统开发人员，需要将客户需求转换为技术目标，从用户（User）、目标（Goal）、任务（Task）三个方面来理清系统应该实现的各项参数指标，明确系统开发任务。

通过层级分析将任务不断拆解，逐级细化任务，直至用户实际可用的具体操作。随着任务的细化，对系统的理解会越来越清晰，在时间允许的前提下可以调整任务安排。然后再通过任务计划（Plan）将子任务进行重组，描绘出用户实际的操作流程。

嵌入式系统基本上是交互式系统，为了使设计的系统变得更自然、更人性化，需要能够

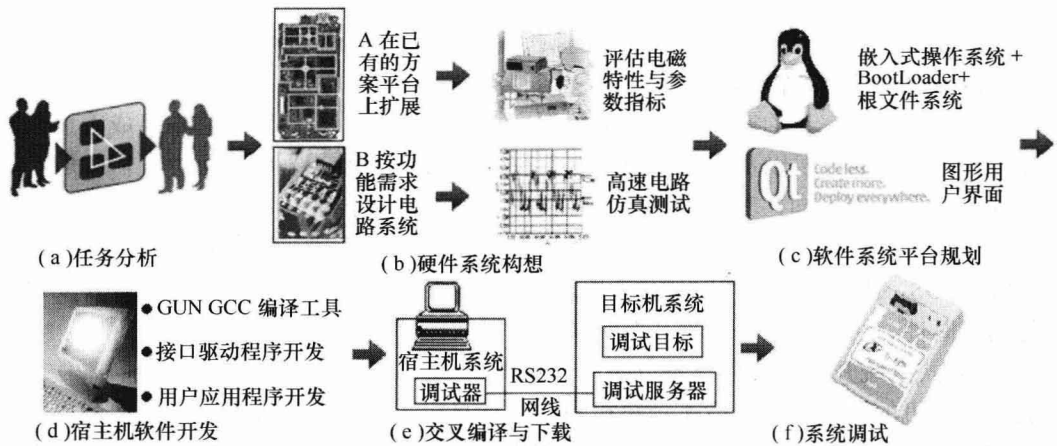


图 1-3 嵌入式系统的开发流程

尝试模拟用户的操作行为路径。

2. 硬件系统构想

客户需求转换为技术目标，技术目标被分配到各个层次的开发任务中，开发任务可归为硬件系统与软件系统开发。软件系统开发比较灵活，工具较多，可实现不同层次的任务构想，但是在预定项目中的硬件平台上却不一定能够支持其运行，因此，可以先设计硬件系统，将硬件平台定型以后，再合理规划软件系统的开发。

硬件系统构想是指按开发任务来设计或配置硬件电路的核心处理器、通信接口外围电路、显示电路、输入设备电路接口、控制机构、存储空间、电源模块等，使其满足系统运行的需求，硬件规模不大不小，性价比高。

如果有可用的评估电路板，或者做过类似项目，那么可在已有的方案平台上进行扩展开发，加快开发周期，使开发人员提高开发效率。同时，为了确保已有的硬件平台能在现在的项目中应用，建议评估其电磁特性，测试其运行参数是否满足系统指标，如是否电磁干扰比较严重，使系统不断地重启；是否其功耗满足系统指标等。

如果没有满足硬件系统的基础设备，就需要从电路原理图、PCB 图、硬件评估板一步一步做出来，如果电路系统中有高速信号，建议在制作 PCB 之前在软件中进行高速电路仿真测试，测试高速信号在 PCB 上传输的完整性，这样可以确保电路系统的可靠性。

3. 软件系统平台规划

嵌入式系统之所以称为“嵌入式”系统，很重要的一点是其系统中运行着“实时的嵌入式操作系统”，因此，在规划软件系统时，首先要选择能在嵌入式微处理器中运行的操作系统。然后，需要确定能支持嵌入式操作系统下载及应用软件下载的 BootLoader 程序软件，即引导程序。接着，如果是本教材中使用的 Linux 操作系统，需要制作其根文件系统。最后，建立图形用户界面，提供方便的交互窗口。

BootLoader、操作系统、根文件系统、图形用户界面等这一系列软件称为嵌入式系统的软件系统平台。

4. 宿主软件开发

嵌入式硬件平台不支持在其上直接进行软件开发，因为嵌入式微处理器及存储空间远远不足以支持编译工具，嵌入式系统的软件目前还只能在 PC 上开发。用户在 PC 上开发基于微处理器体系架构的软件后，再将软件下载到评估电路板上运行，所以这里一般将 PC 称为宿主