

1CD
附赠
本书程序源代码

Cortex-M3 之STM32 嵌入式系统设计

廖义奎 编著

- STM32最小系统设计
- TFT驱动与显示设计
- 触摸屏驱动设计
- SD卡驱动与FAT文件系统设计
- μ C/OS-II在STM32上的移植设计
- 汉字与图形图像显示设计
- 摄像头驱动与图像采集设计
- 以太网及Web远程控制系统设计



 中国电力出版社
CHINA ELECTRIC POWER PRESS

Cortex-M3 之 STM32 嵌入式系统设计

廖义奎 编著



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 摘 要

本书介绍了以 ARM Cortex-M3 为内核的 STM32F103 增强型微控制器的特点, 深入讲解其硬件和软件设计方法。本书分为 STM32 基础开发、STM32 深入开发以及 STM32 高级开发三部分。基础开发部分介绍了嵌入式系统概述、STM32 最小系统设计、STM32 程序设计入门、GPIO 应用、GCC 编译器的安装与应用、STM32 外部中断、面向对象设计的本质、USART 通信等内容; 深入开发部分介绍了深入 STM32 的工作原理、定时器与日历、ADC 应用、DMA 应用、备份寄存器与看门狗程序、TFT 驱动与显示、触摸屏驱动、SD 卡驱动与 FAT 文件系统等; 在高级开发部分介绍 $\mu\text{C}/\text{OS-II}$ 在 STM32 上的移植、汉字与图形图像显示、摄像头驱动与图像采集、以太网及 Web 远程控制系统设计等内容。

本书配套光盘中附有所有章节的源程序。本书适合于嵌入式开发人员作为开发参考资料, 也适合于高校师生作为单片机、嵌入式系统课程的教材和教学参考书。

图书在版编目 (CIP) 数据

Cortex-M3 之 STM32 嵌入式系统设计 / 廖义奎编著. —北京: 中国电力出版社, 2012.2

ISBN 978-7-5123-2671-2

I. ①C… II. ①廖… III. ①微处理器—系统设计
IV. ①TP332

中国版本图书馆 CIP 数据核字 (2012) 第 017190 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

航远印刷有限公司印刷

各地新华书店经售

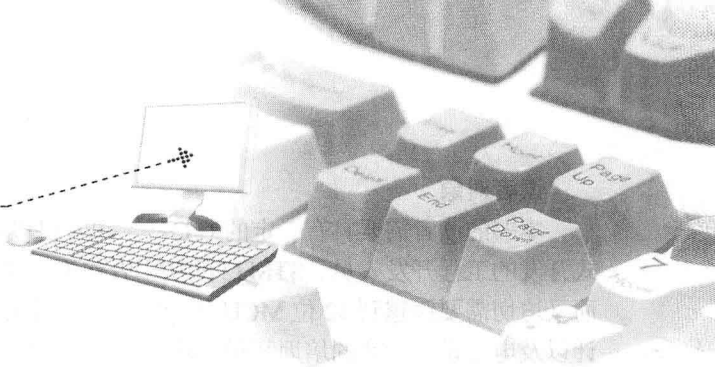
*

2012 年 4 月第一版 2012 年 4 月北京第一次印刷
787 毫米×1092 毫米 16 开本 27 印张 722 千字
印数 0001—3000 册 定价 58.00 元 (含 1CD)

敬告读者

本书封面贴有防伪标签, 加热后中心图案消失
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究



前 言

在科研项目研究、产品开发、毕业设计以及电子竞技等活动中，经常遇到 8 位单片机速度、I/O 口、内部 RAM 以及内部 Flash 不够用等问题。随着 32 位微控制器成本的降低，采用 32 位微控制器作为 8 位单片机系统的升级与更新换代已成为最佳选择，特别是内部带 Flash 的低成本 ARM 微控制器的使用，以接近 8 位单片机的成本即可获得更高性能。

目前许多 IC 厂商都推出了内部带 Flash 的低成本 32 位 ARM 微控制器，例如 ARM Cortex-M3 系列微控制器。它具有两个很重要的特点，一是低成本，二是高性能。在成本方面，价格与 8 位/16 位微控制器相差不多；内带 Flash，不需要外接 ROM，简化了设计，电路更简洁。在高性能方面，运算速度快，例如以 Cortex-M3 为内核的 STM32F2 系列微控制器，内核主频高达 120MHz，内部带有硬件乘法器、硬件除法器、以太网控制器、支持 USB 2.0 接口等。由此可见，32 位微控制器在性能上是 8 位、16 位微控制器无法比拟的。

在代码的大小方面，ARM Cortex-M3 微控制器提供优于 8 位和 16 位体系结构的代码密度。在减少对内存的需求和最大限度地提高片上闪存的使用率方面，都具有很大的优势。

STM32F103 微控制器构建于高性能的 ARM Cortex-M3 内核，工作频率为 72MHz，内置高速存储器（最高可达 1M 字节的闪存和 128K 字节的 SRAM），丰富的增强型 I/O 端口和连接到两条 APB 总线的外设。增强型器件都包含 2~3 个 12 位的 ADC、4 个通用 16 位定时器和 2 个 PWM 定时器。

成本低，该系列微控制器与常见的 8 位、16 位单片机在价格上基本接近。既有 32 位单片机的性能，又与 8 位、16 位单片机价格相当，可直接代替 8 位/16 位单片机应用于一些小型控制系统中。

体积小，可把该应用系统的 PCB 面积压缩到最小，以便应用到小体积的产品中，例如智能继电器、微型水位控制器、恒温控制器等。

性能高，包含标准和先进的通信接口：5 个 USART 接口、3 个 SPI 接口、2 个 I²C 接口、2 个 I²S 接口、1 个 SDIO 接口、一个 USB 接口和一个 CAN 接口。STM32F103 是一个完整的系列，其成员之间引脚对引脚完全兼容，软件和功能也兼容。

GCC 编译器是一套以 GPL 及 LGPL 许可证发行的开源、自由软件。GCC 编译器是移植到中央微控制器架构以及操作系统最多的编译器。由于 GCC 已成为 GNU 系统的官方编译器（包括 GNU/Linux），它也成为编译与建立其他操作系统的主要编译器，包括 Linux 系列、BSD 系列、Mac OS X、NeXTSTEP 与 BeOS 等。

GCC 通常是跨平台软件首选的编译器。有别于一般局限于特定系统与执行环境的编译器，GCC 在所有平台上都使用同一个前端处理程序，产生一样的中间代码，此中间代码在各个不同的平台上都一致，并可输出正确无误的最终代码。

GCC 功能强大、性能优越，并且开放源代码，用户可以免费使用，从而降低了开发成本。

读者对象

本书的读者需要具有一定的 C/C++、单片机以及电子线路设计基础，适合于从事 ARM 嵌入式开发的工程开发人员、STM32 的初学者作为参考资料，更适合于从事 8 位、16 位 MCU 开发，而又迫切需要跨越到 32 位 MCU 平台的工程开发人员。也适合于高校师生作为课程设计、毕业设计以及电子设计竞赛的培训和指导教材，以及作为本、专科单片机、嵌入式系统相关课程的教材。

光盘使用

本书 ARM 程序的编译环境都是 GCC，Obtain_Studio 集成开发环境 (IDE) 软件已带有 GCC，并自动配置 GCC 的运行环境，因此可以在 Obtain_Studio 中直接编译本书的程序。

配套光盘中包括了所有章节的程序代码，读者可以直接拷贝下来使用，并仿照这些程序源代码去快速开发新的应用程序。

联系作者

如果读者在使用本书时遇到相关的技术问题，或者对本书介绍的 ARM 开发板感兴趣或有疑问，可以通过电子邮件与作者联系 (javawebstudio@163.com)，作者将尽最大努力与广大读者一起解决开发过程中遇到的问题，共同进步。

致谢

在本书的编写过程中，陈钧做了大量的准备工作，并参与了前几章部分内容的资料收集与整理工作，在此表示衷心的感谢。感谢陆才志、苏宇、梁创英、许金、王继平、韦运忠、徐卫怡、蓝艺峥、苏金秀分别审阅了本书的章节内容。

本书在编写过程中参考了大量的文献资料，一些资料来自互联网和一些非正式出版物，书后的参考文献无法一一列举，在此对原作者表示诚挚的谢意。

限于作者水平，并且编写时间比较仓促，书中难免存在错误和疏漏之处，敬请读者批评指正。

编 者

目 录

前言

第1章 概述

1

1.1 嵌入式系统定义	1
1.2 嵌入式系统的发展	1
1.2.1 从单片机到嵌入式系统	1
1.2.2 从芯片级设计到系统级设计	2
1.2.3 从面向器件到面向任务的设计	2
1.2.4 从单处理器设计到多处理器设计	2
1.3 嵌入式系统的应用	3
1.4 ARM 系列嵌入式系统处理器	4
1.4.1 ARM 处理器分类	4
1.4.2 ARM Cortex 处理器	5
1.4.3 ARM Cortex-M3 处理器	6
1.4.4 ARM Cortex-A8 处理器	7
1.5 从 8/16 位处理器到 ARM Cortex-M3/M0	7
1.6 常见的 Cortex-M0/M3 系列 MCU	9

第2章 STM32 最小系统设计

14

2.1 STM32F103C 最小系统设计方案	14
2.2 最小系统设计的要素	16
2.2.1 STM32 外部晶振	16
2.2.2 复位电路	18
2.2.3 LED、Key 及 BOOT 跳线	19
2.2.4 稳压电源及 ISP 下载口	20
2.2.5 IO 端口	23
2.3 PCB 图设计	24

第3章 STM32 程序设计入门

26

3.1 STM32 入门之 Hello World 程序	26
3.1.1 开发环境	26
3.1.2 编写 STM32 的 C 程序	27
3.1.3 用 GCC 编译 stm32 程序	29

3.1.4	STM32 程序下载	30
3.1.5	在 Obtain_Studio 中编译 Hello World 程序	32
3.2	不同开发板的 Hello World 程序	32
3.3	基于 STM32 固件库的入门程序	33
3.3.1	STM32 固件库	33
3.3.2	STM32 固件库外设的初始化和设置	35
3.3.3	基于 STM32 固件库的程序设计	36
3.4	基于 STM32 固件库的 Hello World 程序代码分析	37

第4章 GPIO 应用

40

4.1	认识 STM32 GPIO	40
4.1.1	GPIO 功能特点	40
4.1.2	STM32 IO 口的优点	41
4.1.3	STM32 固件库中提供的 GPIO 库函数	42
4.2	KEY_LED 程序	43
4.2.1	创建 stm32_C++KEY_LED 项目	43
4.2.2	stm32_C++KEY_LED 项目程序分析	44
4.3	低层代码分析	47
4.3.1	GPIO 端口的定义	47
4.3.2	AHB/APB 桥的配置	49
4.3.3	GPIO 引脚的配置	50
4.3.4	GPIO 的读写	53

第5章 GCC 编译器的安装与应用

54

5.1	GCC 介绍	54
5.1.1	GCC 概述	54
5.1.2	MinGW 简介	54
5.1.3	MinGW 的安装	55
5.1.4	MinGW 测试	56
5.1.5	常见 GCC 用法	58
5.2	ARM GCC 编译器	61
5.2.1	WinARM 编译器	61
5.2.2	Sourcery G++ Lite for ARM EABI 编译器	62
5.3	Obtain_Studio 集成开发系统	64
5.3.1	Obtain_Studio 集成开发系统介绍	64
5.3.2	Obtain_Studio 集成开发系统常用技巧	67
5.4	GCC Make 编译文件设计	69
5.4.1	GCC Make 常用命令	69
5.4.2	makefile 文件规则	71
5.4.3	makefile 文件函数	75
5.5	GCC 编译器 LD 脚本	79

5.5.1 C/C++程序内存空间	79
5.5.2 GCC LD 脚本基础	82
5.5.3 STM32 程序中的 LD 脚本程序	86

第6章 STM32 外部中断

89

6.1 STM32 外部中断	89
6.2 STM32 外部中断实例	90
6.3 STM32 中断配置	92
6.3.1 STM32 外部中断程序分析	92
6.3.2 中断通道配置	94
6.3.3 中断优先级配置	94
6.3.4 外部中断模式配置	94
6.3.5 外部中断响应函数配置	97

第7章 面向对象程序设计

99

7.1 程序风格	99
7.1.1 程序风格的比较	99
7.1.2 编程风格在程序设计中的作用	100
7.2 跨越开发板	100
7.2.1 端口映射的方法	100
7.2.2 模式设置的方法	101
7.3 分类与封装	101
7.3.1 什么是分类与封装	101
7.3.2 封装的实现	102
7.4 隐藏与权限	103
7.4.1 隐藏	103
7.4.2 权限	103
7.5 继承	104
7.5.1 CGpio 类的继承	104
7.5.2 测试 CLed 和 CKey 类	104
7.6 组装	105
7.6.1 GPIO 的组装	105
7.6.2 GPIO 组装的测试	106
7.7 C++在嵌入式系统中的应用	108
7.7.1 C++介绍	108
7.7.2 兼容 C 语言	108
7.7.3 在 C++程序中调用 C 函数	110
7.7.4 面向对象程序设计语言	110
7.7.5 泛型编程语言	111
7.7.6 STL 编程	113
7.7.7 接口编程	114

8.1	从 51 单片机到 STM32 的串口通信	121
8.2	USART 通用串口通信设计	124
8.2.1	USART 通用串口	124
8.2.2	USART 通用串口通信设计方案	125
8.3	USART 通用串口程序设计入门	125
8.3.1	USART 数据发送程序设计	125
8.3.2	USART 数据接收程序设计	126
8.4	中断方式的数据接收	127
8.4.1	中断方式的数据接收程序设计	127
8.4.2	多个串口驱动对象的协同工作	128
8.5	USART 驱动程序的设计	129
8.5.1	USART 驱动程序	129
8.5.2	printf 与 cout 的实现	133
8.6	深入 STM32 USART 的工作原理	136
8.6.1	USART 工作原理	136
8.6.2	发送器	137
8.6.3	接收器	139
8.6.4	USART 初始化函数 USART_Init	141
8.6.5	USART 波特率的计算方法	142

9.1	STM32 启动原理	144
9.1.1	STM32 启动过程分析	144
9.1.2	STM32 软件复位与功耗控制	145
9.2	系统时钟分析	147
9.2.1	系统时钟种类	147
9.2.2	STM32 固件库设置时钟	149
9.2.3	系统时钟配置	150
9.3	存储器以及存储器映射	160
9.4	NVIC 嵌套中断向量控制器	162
9.4.1	NVIC 嵌套中断向量控制器	162
9.4.2	STM32 的 NVIC 优先级	166
9.5	STM32 向量表及配置	170
9.5.1	STM32 复位后从哪个地址开始执行	170
9.5.2	STM32 向量表	170
9.5.3	用户程序中的向量表	171

10.1	SysTick 定时器	179
10.1.1	关于 SysTick	179

10.1.2	SysTick 测试程序	180
10.1.3	SysTick 程序分析	181
10.2	RTC 定时器	184
10.2.1	RTC 定时器介绍	184
10.2.2	RTC 的本质与测试程序	186
10.2.3	日历算法	187
10.2.4	STM32 的 RTC 日历测试程序	190
10.2.5	STM32 RTC 程序分析	191
10.2.6	RTC 秒中断	193
10.2.7	RTC 闹钟	195
10.2.8	RTC 校准	197
10.3	通用定时器	198
10.3.1	STM32 定时器的种类	198
10.3.2	通用定时器介绍	200
10.3.3	通用定时器基本应用程序设计	200
10.3.4	通用定时器常用模式	203
10.3.5	输出模式测试实例	205
10.3.6	输入捕获模式测试实例	206

第 11 章 ADC 应用

209

11.1	ADC 与数字信号处理系统设计	209
11.1.1	数字信号处理系统设计	209
11.1.2	STM32 简单的 ADC 应用实例	212
11.1.3	过采样技术	213
11.1.4	欠采样技术	215
11.2	STM32 的 ADC 简介	216
11.3	STM32 ADC 入门实例	220
11.3.1	STM32 ADC 入门测试程序	220
11.3.2	STM32 ADC 程序分析	221
11.3.3	STM32 内部温度测量	228
11.4	STM32 ADC 注入方式	229
11.4.1	STM32 ADC 注入方式简介	229
11.4.2	STM32 双 ADC 模式	230
11.4.3	STM32 ADC 注入方式实例	230

第 12 章 DMA 应用

235

12.1	STM32 的 DMA 简介	235
12.1.1	任务转移策略	235
12.1.2	STM32 的 DMA 功能	236
12.2	DMA 在 ADC 中的应用	238
12.2.1	任务转移策略的 DMA ADC 应用实例	238

12.2.2	DMA_ADC 程序分析	240
12.3	DMA 在 USART 中的应用	245
12.3.1	任务转移策略的 USART DMA 数据发送	245
12.3.2	任务转移策略的 USART DMA 数据接收	248
12.3.3	任务队列策略的 USART DMA 发送中断应用	251
12.3.4	任务循环策略的 USART DMA 接收中断应用	254

第 13 章 备份寄存器与看门狗程序

258

13.1	STM32 备份寄存器	258
13.1.1	备份寄存器特点	258
13.1.2	BKP 应用实例	259
13.2	STM32 看门狗	261
13.2.1	STM32 看门狗介绍	261
13.2.2	独立看门狗介绍	262
13.2.3	独立看门狗程序设计	263
13.2.4	窗口看门狗介绍	265
13.2.5	窗口看门狗测试程序	267

第 14 章 TFT 驱动与显示

269

14.1	LCD 概述	269
14.1.1	LCD 简介	269
14.1.2	LCD 接口	270
14.2	Ili9xx 系列 TFT 驱动芯片	271
14.3	TFT 测试程序	275
14.3.1	TFT 测试程序准备工作	275
14.3.2	TFT 测试主程序	275
14.3.3	字符的显示	277
14.4	基于 FSMC 的 TFT 驱动程序设计	279
14.4.1	STM32 的 FSMC 功能	279
14.4.2	FSMC 与 TFT 端口连接与端口映射	279
14.4.3	FSMC 与 TFT 的内存空间映射与操作	281
14.4.4	FSMC 初始化	282
14.4.5	TFT 初始化	287
14.4.5	TFT 驱动程序统一接口函数的实现	290
14.5	基于 GPIO 的 TFT 驱动程序设计	292

第 15 章 触摸屏驱动

297

15.1	触摸屏介绍	297
15.2	触摸屏驱动 IC	300
15.3	触摸屏测试项目	302
15.4	触摸屏驱动程序分析	303

15.5 触摸屏校准	311
15.5.1 触摸屏校准算法	311
15.5.2 触摸屏校准的实现	312

第16章 SD卡驱动与FAT文件系统 **316**

16.1 STM32的SDIO接口	316
16.1.1 常见存储卡种类	316
16.1.2 SD卡结构	317
16.1.3 STM32的SDIO接口	318
16.2 FAT文件系统	320
16.2.1 FAT文件系统概述	320
16.2.2 FatFs介绍	321
16.3 STM32 SDIO接口	324
16.4 SD卡文件读写实例	326
16.4.1 准备工作	326
16.4.2 SD卡文件读写实例	327
16.4.3 SD卡文件操作类CFile的设计	328
16.4.4 目录操作	329

第17章 μ C/OS-II在STM32上的移植 **331**

17.1 μ C/OS-II概述	331
17.1.1 μ C/OS-II简介	331
17.1.2 μ C/OS-II的组成部分	331
17.2 μ C/OS-II移植到STM32	332
17.3 μ C/OS-II工作原理	337
17.3.1 μ C/OS-II启动过程	337
17.3.2 任务切换的相关函数解析	338

第18章 汉字与图形图像显示 **343**

18.1 汉字显示	343
18.1.1 汉字库	343
18.1.2 程序中加入汉字库实现汉字显示	345
18.1.3 使用SD卡上的汉字库实现汉字显示	346
18.2 图形绘制	350
18.3 图像显示	353
18.3.1 位图与bmp文件格式	353
18.3.2 bmp文件操作	356
18.3.3 bmp图像显示测试程序	359

第19章 摄像头驱动与图像采集 **362**

19.1 摄像头接口	362
------------	-----

19.1.1	图像传感器	362
19.1.2	OV7670 摄像头	362
19.1.3	CMOS 摄像头接口	364
19.2	CMOS 摄像头测试程序	366
19.3	深入 CMOS 摄像头驱动程序原理	368
19.3.1	SCCB 协议	368
19.3.2	SCCB 协议驱动程序设计	370
19.3.3	CMOS 摄像头驱动程序设计	373

第 20 章 以太网及 Web 远程控制系统设计

378

20.1	ENC28J60 以太网控制器	378
20.2	网络测试程序	382
20.2.1	Web Server 测试	382
20.2.2	UDP 通信测试	385
20.3	IP/ICMP 协议与 Ping 命令的实现	386
20.3.1	以太网数据包结构	386
20.3.2	IP 协议	387
20.3.3	ICMP 协议	389
20.3.4	Ping 命令	392
20.3.5	Ping 命令的实现	393
20.4	UDP 通信原理	393
20.4.1	UDP 协议	393
20.4.2	UDP 通信的实现	394
20.5	Web Server 程序设计	399
20.5.1	Web Server 原理	399
20.5.2	TCP 设计	402
20.5.3	Web Server 设计	403
20.6	ENC28J60 驱动程序设计	405
20.6.1	STM32 SPI 接口	405
20.6.2	STM32 SPI 驱动程序	407
20.6.3	ENC28J60 驱动程序	409

概 述

1.1 嵌入式系统定义

嵌入式系统即控制、监视或者辅助装置、机器和设备运行的装置。这主要是从应用上加以定义的,从中可以看出嵌入式系统是软件和硬件的综合体,还涵盖了机械等附属装置。

目前国内相关业者一个普遍认同的定义是: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。

在构成上,嵌入式系统以微处理器及软件为核心部件,这两者缺一不可。在特征上,嵌入式系统具有方便、灵活地嵌入到其他应用系统的特征,即具有很强的可嵌入性。嵌入式系统本身是一个可独立执行的系统,但更重要的是它可作为一个部件嵌入到其他应用系统中。

按嵌入式微处理器类型划分,嵌入式系统可分为以单片机为核心组成的嵌入式单片机系统;以工业计算机板为核心组成的嵌入式计算机系统;以 DSP 为核心组成的嵌入式数字信号处理器系统;以 FPGA 及软 CPU (SOPC) 为核心组成的嵌入式 SOPC 系统等。

由于嵌入式系统的内容很广,在含义上与传统的单片机系统和计算机系统有很多重叠部分。为了方便区分,在实际应用中通常给嵌入式系统加了一些不成文的约定:

(1) 嵌入式系统的微处理器通常是由 32 位及以上的 RISC 处理器组成,例如 ARM、MIPS 等。

(2) 嵌入式系统的软件系统通常是以嵌入式操作系统为核心,外加用户应用程序。

(3) 嵌入式系统在特征上具有明显的可嵌入性。

从狭义上说,嵌入式系统都应该具备上述三个特征。从广义上说,只要具备上述三个特征中的部分特征,也可以看作是嵌入式系统。

1.2 嵌入式系统的发展

如今手机向智能化发展,电视向网络化发展,这些都得益于嵌入式系统日新月异的发展。

1.2.1 从单片机到嵌入式系统

单片机在近二三十年里,已经渗透到各个领域,并且与人们的日常生活密不可分,给人民生活 and 工业生产带来极大方便。从信号采集、处理到传输都能由单片机来完成。

随着互联网时代的来临,许多电子设备需要联网和更智能化、更强的计算能力,比如音频、视频的数据采集、处理和传输,丰富的图形界面等。单片机越来越不能满足应用对象的需求,开发工作也变得越来越复杂和庞大。



随着微电子技术的进步，芯片的制造成本大大降低，而功能却大大增强，16 位和 32 位的嵌入式微处理器逐渐成为嵌入式系统开发设计的主流。不过，只有嵌入式微处理器是不够的，OEM（原始设备制造商）还需要有一个运行于嵌入式微处理器上的操作系统。

嵌入式操作系统具有良好的可移植性，能够用在根据应用要求选择的微处理器中；软件开发工作变得规范，容易测试，可实现模块化编程，同时由多个人共同完成一个任务；解决以往开发产品存在的诸多不安全隐患。很多软件厂商根据嵌入式系统发展的需要，推出了多种不同特点的嵌入式操作系统。例如，Microsoft 公司的 Windows CE、3COM 公司的 Palm OS、Symbian 公司的 EPOC 等。

1.2.2 从芯片级设计到系统级设计

嵌入式系统的设计需要具有能跨越传统硬件和软件设计界限的复合型设计人员，这是嵌入式系统技术与传统单片机和微处理器系统技术的重要区别。嵌入式系统具有系统级应用开发技术的基本特征，而以往的微处理器、单片机和 DSP 处理器应用设计，则属于器件或芯片级应用开发技术。

嵌入式系统的设计以操作系统和硬件系统提供的接口作为应用系统设计的基础，其开发技术与 PC 机的应用开发类似。因此，嵌入式系统设计工作大部分集中在系统一级的应用上。

在嵌入式系统应用开发中，设计人员不必了解很多底层知识和技术，只需要在操作系统之上完成应用系统设计即可。

嵌入式系统设计根据实际需要来选择不同的硬件平台、不同的软件平台，以及不同的软硬件模块，实现成本、性能、功率和体积的可选性与可控性。

1.2.3 从面向器件到面向任务的设计

随着各种各样的微处理器、微控制器、可编程逻辑器件功能的增强以及成本的降低，电子线路设计与十年前相比，已经发生了本质的变化，即设计思想的变化。

过去电子线路的设计，由于器件成本较高，种类也比较单一，许多设计都是先看器件能实现什么功能，再决定去做什么任务，即面向器件的设计。现在，由于器件品种非常丰富，器件性能从低端到高端一应俱全，不管是从性能还是从成本方面考虑，都有非常大的选择余地。因此，所需要实现的任务才是核心问题，可以根据任务目标去选择最恰当的器件，这就是面向任务的设计。

1.2.4 从单处理器设计到多处理器设计

多微处理器和多核心微处理器在并行工作的能力方面要比单处理器或单核强。过去电子线路的设计，多以单一微处理器为核心，目前已发展为双处理器以及多处理器或多核心处理器的设计。任务的执行也由顺序执行发展为并行执行。

通常情况下，一个应用系统都以一个微处理器为核心，外加键盘驱动、显示驱动以及控制驱动，以及一些 ADC、DAC 芯片进行数据采集及数据输出。目前，在大多数情况下，这样安排不一定是合理的。

例如，ADC 转换，采用一片专用的芯片，有时不如采用一片带 ADC 的 AVR 单片机简单，或者不如采用一片 ARM 处理实现效果好（用 ARM 实现的优点是同时可以把 ADC 转换、数字滤波、异常处理、输出通信协议等都放在一片 ARM 中实现，这是专用 ADC 芯片所不能实现的）。键盘以及 LED/LCD 驱动电路，采用专用芯片也不如采用一片 MCS-51 或 AVR 单片机来实现的成本更低。做快速事务处理及流程控制，采用一片 FPGA（SOPC）远比用一片 MCS-51、AVR 以及

ARM 处理器有效,即相当于以可编程逻辑器件为核心,而把 AVR、ARM 等作为 FPGA (SOPC) 的外围器件,实现通信、ADC 转换等外设功能,这一点与常规的以 AVR、ARM 微处理器为核心的应用系统正好相反。这些情况都是电子技术、电子器件的不断进步所带来的设计思想上的突破。

1.3 嵌入式系统的应用

1. 在智能消费设备中的应用

目前,嵌入式系统最为成功的应用是在智能消费设备中的应用,例如,智能手机、PDA、GPS 导航仪以及多媒体播放器等。

智能手机的诞生,是由掌上电脑(Pocket PC)演变而来的。最早的掌上电脑不具备手机的通话功能,但是随着用户对于掌上电脑的个人信息服务方面功能依赖的提升,但又不习惯于随时都携带手机和 PPC 两个设备,所以厂商将掌上电脑的系统移植到了手机中,于是就有了智能手机这个概念。

从广义上说,智能手机除了具备手机的通话功能外,还具备了 PDA 的大部分功能,特别是个人信息管理以及基于无线数据通信的浏览器、GPS 和电子邮件功能。智能手机为用户提供了足够的屏幕尺寸和带宽,既方便随身携带,又为软件运行和内容服务提供了广阔的舞台,很多增值业务可以就此展开,例如,股票、新闻、天气、交通、商品、应用程序下载、音乐和图片下载等。结合 3G 通信网络的支持,智能手机必将成为一种功能强大,集通话、短信、网络接入、影视娱乐为一体的综合性个人手持终端设备。

2. 在工业控制中的应用

目前已经有大量的 8 位、16 位、32 位嵌入式微控制器应用在工业设备中,网络化是提高生产效率和产品质量、减少人力资源主要途径,如工业过程控制、数字机床、电网设备监测、石油化工系统。

传统的可编程逻辑控制器(Programmable Logic Control, PLC)经过十几年的发展,技术已经非常成熟,凭其高速度、高稳定性、高性能的特点在工业控制领域得到了广泛应用。与此同时,随着现代计算机技术和电子技术的发展,在工程方面逐渐表现出其缺点:传统的 PLC 产品互不兼容,缺乏统一的标准,难以构建开放的硬件体系结构;另外,各厂商产品的编程方法差别很大,技术专有成性较强,工作人员培训时间过长,造成传统 PLC 应用成本居高难下,性价比增长很缓慢。

20 世纪 90 年代中期,计算机和微电子技术的迅猛发展以及 PLC 的国际标准 IEC61131 的制定,产生了软 PLC 技术。所谓“软 PLC 技术”,就是用 PC 机作为硬件支撑平台,利用软件实现标准硬件 PLC 的基本功能,也就是将 PLC 的控制功能封装在软件内,运行于 PC 机的环境中,以 PC 机为基础的系统,同时具备了传统 PLC 和 PC 机的双重功能和多重优点。

软 PLC 技术一般由开发系统和运行系统两个部分组成。开发系统运行在 PC 平台上,而运行系统则运行在嵌入式硬件平台上。一般而言,嵌入式硬件平台上都要运行某种嵌入式操作系统,比如 uC/OSII、uClinux、Windows CE 等。

现代开发系统的特点是:具有集成化开发环境,界面友好,易于使用。软 PLC 嵌入式系统是基于 IEC61131 国际标准的,有比较统一的支持语言(C、C++、汇编等)。在嵌入式硬件平台上运行的系统主要负责解释执行由开发系统编译、链接后产生的目标文件,运行系统一般由运行内核、通信接口和系统管理三大部分组成,其设计和实现精度很高,它的执行效率将直接影响到系统在现场控制中的反应速度。

3. 在航天与交通方面的应用

在宇航方面的应用，往往会成为民用的先导。例如执行火星探路者飞行任务的飞行器，和定居者登陆舱一样，都安装有嵌入式的 PowerPC，执行控制和其他运行功能。

基于嵌入式系统的 GPS 接收机（或导航仪）目前已经获得非常广泛的应用。例如，飞机、轮船、地面车辆以及步行者都可以利用 GPS 导航器进行导航。汽车导航系统是在全球定位系统 GPS 基础上发展起来的一门新型技术，由 GPS 导航、车速传感器、陀螺传感器、CD-ROM 驱动器、LCD 显示器组成。GPS 导航系统与电子地图、无线电通信网络、计算机车辆管理信息系统相结合，可以实现车辆跟踪和交通管理等许多功能。

嵌入式系统在交通运输方面目前也有非常好的应用潜力，例如，汽车的多媒体播放系统、无线上网系统等。

4. 信息家电及家庭智能管理系统

信息家电及家庭智能管理系统方面将是嵌入式系统未来最大的应用领域，例如冰箱、空调等的网络化、智能化将引领人们的生活步入一个崭新的空间，即使用户不在家，也可以通过电话线、网络进行远程控制。又例如水、电、煤气表的远程自动抄表，以及安全防火、防盗系统，其中嵌入式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。目前在餐饮服务领域，例如远程点菜器等，已经体现了嵌入式系统的优势。

5. 在网络与通信系统中的应用

嵌入式系统将广泛用于网络与通信系统之中。例如，ARM 把针对移动互联网市场的产品分为两类，一类是 Smartphone（智能手机），一类是笔记本电脑。Netbook（上网本）是介于笔记本和 Smartphone 中间的一个产品。ARM 过去在 PC 机上的业务很少，但现在市场对更低功耗的移动计算平台的需求带来了新的机会，因此 ARM 在不断推出性能更高的 CPU 来拓展市场。ARM 新推出的 Cortex-A8、Cortex-A9/MPCore 处理器可以用于高端 Smartphone，也可用于 Netbook。现在已经有众多半导体厂商在采用 ARM 的 Cortex-A8 和 Cortex-A9 开发产品并应用于 Netbook，今后市场上将可以买到很多基于 Cortex-A8 的 Netbook。

公共交通无接触智能卡（Contactless Smartcard, CSC）发行系统、公共电话卡发行系统和自动售货机，各种智能 ATM 终端将全面走入人们的生活。

6. 环境工程

嵌入式系统在环境工程中的应用也很广泛，例如，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。在很多环境恶劣、地况复杂的地区，依靠嵌入式系统将能够实现无人监测。

7. 机器人

嵌入式芯片的发展将使机器人在微型化、高智能方面优势更加明显，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。

1.4 ARM 系列嵌入式系统处理器

1.4.1 ARM 处理器分类

目前的嵌入式系统，绝大多数都采用了以 ARM 为内核架构的微处理器。ARM 公司自 1990 年成立以来，在 32 位 RISC（Reduced Instruction Set Computer）CPU 的开发设计上不断突破，其设计的微处理器结构已经从 v1 发展到了 v7，ARM 处理器的核心及构架如表 1-1 所示。