



普通高等教育“十一五”国家级规划教材

嵌入式系统及其应用

——基于Cortex-M3内核和STM32F系列微控制器的系统设计与开发
(第2版)

同济大学电子与信息工程学院控制科学与工程系
陈启军 余有灵 张伟 潘登 周伟 编著



同济大学出版社
TONGJI UNIVERSITY PRESS

014036633

TP360.21-43
31-2



普通高等教育“十一五”国家级规划教材

嵌入式系统及其应用

——基于 Cortex-M3 内核和 STM32F 系列微控制器的系统设计与开发
(第 2 版)

同济大学电子与信息工程学院控制科学与工程系

陈启军 余有灵 张 伟 潘 登 周 伟 编著



北航

C1723460

TP360.21-43



同济大学出版社
TONGJI UNIVERSITY PRESS

31-2

内 容 提 要

本书介绍了嵌入式系统领域的基本原理、技术和方法,在内容上偏重自动化和电气类专业的教材选择需求,强调基本原理、硬件设计、软硬件交互,强调在自动化和电气领域的应用,力求能够改变目前高校教学上“软强硬弱”的现状。全书紧跟时代潮流,以 ARM 公司的 Cortex-M3 内核和 ST 公司的 STM32F 系列 MCU 为主要介绍对象,在普适的嵌入式基本原理与具体芯片的结合上较为深入。

本书适合专业教育阶段的自动化电气类本科教学,也适合具有初步经验的嵌入式系统开发者自学或研究生学习,对其他从事电子技术和软件技术开发的人员亦有较大参考价值。

图书在版编目(CIP)数据

嵌入式系统及其应用:基于 Cortex-M3 内核和 STM32F 系列微控制器的系统设计与开发/陈启军等编著.--2 版.

--上海:同济大学出版社,2014.2

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5608-5418-2

I. ①嵌… II. ①陈… III. ①微型计算机—系统设计

—高等学校—教材 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2014)第 024947 号

普通高等教育“十一五”国家级规划教材

嵌入式系统及其应用(第 2 版)

——基于 Cortex-M3 内核和 STM32F 系列微控制器的系统设计与开发

陈启军 余有灵 张 伟 潘 登 周 伟 编著

责任编辑 张平官 责任校对 徐春莲 封面设计 陈益平

出版发行 同济大学出版社 www.tongjipress.com.cn

(地址:上海市四平路 1239 号 邮编:200092 电话:021-65985622)

经 销 全国各地新华书店

印 刷 同济大学印刷厂

开 本 787mm×1092mm 1/16

印 张 32

印 数 1—3100

字 数 798000

版 次 2014 年 2 月第 2 版 2014 年 2 月第 1 次印刷

书 号 ISBN 978-7-5608-5418-2

定 价 62.00 元

本书若有印装质量问题,请向本社发行部调换

版权所有 侵权必究

前 言

自本书第一版出版后,得到了许多学界同行的关注并为多所大学选做嵌入式系统教学的教科书,并收到了许多宝贵的建议和反馈,在此我们深表感谢。这也促使了我们对教材进行一次修订。

本次修订,我们重点更正了原书中存在的一些错误和表达不当之处,增添了部分内容,使之更加合理。希望在嵌入式技术蓬勃发展的今天,能够进一步推动相关人才培养和技术进步。

在本书修订过程中,曾得到同行专家的热情帮助,也得到同济大学出版的全力支持,在此一并表示感谢!

由于本次修订工作时间仓促,书中存在不妥之处,敬请读者进一步批评指正。

编著者

2014年2月10日

初版前言

通用电子计算机自 20 世纪 40 年代诞生之后,一直向着高性能和智能化两个方向发展。但在近 20 年间特别是最近 10 年,借助于微电子技术、通信技术和感知测量技术的发展,一个完整的计算机系统可以在更小的空间内实现,且仍能满足用户的需求,这使得计算机系统的应用范围从传统的科学计算与信息处理进一步拓展到通信、娱乐、视讯、测量、控制、国防、航空航天等各类应用,这种趋势体现为近年嵌入式技术的兴起,大量计算机系统设计的目标,也由传统的以高性能为重偏向更加强调满足用户需求和资源约束的平衡设计。

在这种产业背景下,作为一家在 20 世纪 90 年代初刚刚推出 ARM 内核的小公司,就在 20 年内迅速成长为全球领先的嵌入式与移动领域的旗舰厂商,今天,基于 ARM 内核的芯片年销售量就已超过百亿,ARM 已经成为嵌入式领域的事实标准之一。Cortex 系列内核是 ARM 公司在新的技术条件下推出的全新 ARM 内核产品,并按照市场和应用不同分成高性能、控制和实时应用三个系列,即 Cortex-A, Cortex-M 和 Cortex-R,其中, Cortex-M3 内核旨在面向控制类应用,提供一种高性能、低成本、具有卓越计算能力和出色中断响应速度的 32 位嵌入式平台。由于 Cortex-M3 定位准确,出色地平衡了成本、性能、功耗等各方面的要求,在市场上迅速被 ST, TI, Philips/NXP 等众多厂商所接受,并在实际中替代了传统应用中大量的 16 位和高端 8 位芯片。其中, ST 公司推出的 STM32F103 系列通用微控制器芯片性能优越、成本低廉、资源丰富,尤其适合工业自动化测控应用,自推出伊始就获得了市场的高度认可并在实际中获得了广泛应用,也恰为本书偏重工业市场和自动化类应用的定位提供了完美的硬件支持。

在长期的嵌入式相关类课程的教学过程中,我们也发现许多嵌入式系统教材重点讲述偏软的嵌入式操作系统上的应用程序开发技术,而对硬件方面的基本原理与设计所述较少,特别是许多应用例子以网络通信、手机等为应用背景,不符合自动化、电气、机电等偏电类专业的定位和培养目标。因此,我们在编写本书时,结合自身的需求也重点考虑了电气类专业的教材要求,在陈述内容上更加强调底层、强调硬件原理、强调软硬件交互、强调在自动化和电气领域的应用,而在更高层的软件开发上适当弱化。我们希望,选择本教材的学生,能够更加深刻地从最基本原理和最底层硬件出发,理解嵌入式系统的设计与开发流程,改善过去许多院校教学中软强硬弱的现实情况。

本书从结构上可分为四大部分,第一部分由第 1 章导论构成,讲述嵌入式系统的概念、设计中的一些深层次思考和共性理论基础;第二部分是第 2 章,主要讲述 ARM 公司 Cortex-M3 内核的基本原理,包括 Thumb 2 指令体系、向量中断处理、调试与开发支持等内容;第三部分包含第 3 章至第 12 章,讲述 STM32F103 系列微控制器的内部原理与应用设计技术,其中第 3 章讲述了基于 STM32F103 的最小系统设计,围绕此目标,陈述了与一个基本嵌入式系统设计有关的存储区域管理、中断控制器、时钟树等重要概念,第 4 章至第 12 章则依模块分别陈述,在教学上可适当取舍,这些模块主要由 STM32F103 芯片提供;第四部分包含第 13 章至第 16 章,主要介绍与工程开发有关的基本原理和技能,以及一些具体应用实例,特别是几个大的案例在硬件设计方面介绍得较为详细。其中,第 13 章与嵌入式系统软件开发有关,涉及支撑环境、基本开发流程和原理,限于本书篇幅和定位,内容略少,实际教学中可根据专业定位和学生

情况适当增加。

本书是普通高等教育“十一五”国家级规划教材,其编著工作是在同济大学精品课程建设基金的资助下完成的,其中第1,7,8,11,12,15章由张伟编写,第2章由余有灵、潘登编写(内容以 Joseph Yiu 著、宋岩译的《ARM Cortex-M3 权威指南》一书为蓝本,在此鸣谢),第3章由潘登、张伟编写,第4,5章由潘登编写,第6,9,10,13,14章由周伟编写,陈启军教授负责全书的规划、内容组织、目录制定和统稿。在本书撰写过程中,得到了 ST 公司中国区 MCU 技术中心梁平经理、ST 公司蒋建国工程师、ARM 公司原中国区总裁谭军博士、ARM 公司姜宁和秦好亮工程师、上海庆科(MXCHIP)公司王永虹总经理和徐炜工程师等专家的大力支持,在此一并致谢。本书编写过程中,还参考和吸收了互联网上大量关于 Cortex 技术发展的文章和资料,以期能够博采众家之长更好地服务教学,在此也对这些网络上大量 Cortex 爱好者和工程师朋友的奉献表示真挚的感谢。本书在编写过程中,一直得到同济大学出版社张平官编审的热情关心与大力支持;同济大学电信学院控制系的程微宏、马杏宇、厉鹏飞三位同学绘制了本书许多插图,同济大学出版社的工作人员也付出了辛勤的劳动,在此谨向各位支持过关心过本书编著的同仁和朋友一并致谢!

由于嵌入式领域发展迅速,加之作者对有关具体技术细节把握不够,成书较为仓促,书中难免存在差错,包含中外文档规范不一致引起符号表达不统一,以及一些需要进一步商榷甚至是错误的观点,敬请广大专家和读者指正,我们力争未来再版,并吸收更多工业实际应用与教学实践中的经验教训,使之能够成为适合自动化、电气、机电等专业的首选嵌入式系统教材。

编 者

于同济大学

2010年9月10日

目 录

前言

初版前言

第 1 章 嵌入式系统导论	(1)
1.1 嵌入式系统——从部件到系统的集成	(1)
1.1.1 什么是嵌入式系统	(1)
1.1.2 嵌入式系统——从部件到系统的集成	(2)
1.2 计算的基本原理和历史演变	(4)
1.2.1 计算的概念——从数值计算到通用信息处理和智能计算	(4)
1.2.2 计算的基本模型:图灵机理论模型	(5)
1.2.3 计算的发展规律	(7)
1.3 计算机的基本原理和历史演变	(8)
1.3.1 计算机的诞生	(8)
1.3.2 计算机的发展	(9)
1.3.3 面向嵌入式应用的架构改进	(12)
1.4 嵌入式系统的历史沿革	(14)
1.5 ARM, Cortex 和 STM32 简介	(15)
1.5.1 ARM 系列内核	(15)
1.5.2 Cortex 系列内核	(19)
1.5.3 STM32F103 系列微控制器	(20)
1.6 嵌入式系统工程设计 with 开发	(23)
1.6.1 需求分析	(23)
1.6.2 架构和概要设计	(24)
1.6.3 详细设计与开发	(25)
1.6.4 测试反馈	(25)
1.7 本课程学习内容和目标	(26)
习题	(26)
第 2 章 Cortex-M3 微处理器	(27)
2.1 Cortex-M3 微处理器内核	(27)
2.1.1 内核体系结构	(28)
2.1.2 系统总线结构	(30)
2.1.3 寄存器	(32)
2.1.4 存储器管理	(37)
2.1.5 工作模式	(50)
2.1.6 异常与中断	(51)

2.1.7 堆栈.....	(65)
2.1.8 CoreSight 调试与跟踪系统	(68)
2.1.9 Cortex-M3 内核的其他特性	(70)
2.2 指令系统.....	(73)
2.2.1 Thumb-2 指令分类	(75)
2.2.2 统一汇编语言.....	(76)
2.2.3 16-bit Thumb-2 指令集编码格式	(77)
2.2.4 32-bit Thumb-2 指令集编码格式	(83)
2.2.5 条件执行.....	(94)
2.2.6 未定义及不可预测指令.....	(96)
2.2.7 寄存器域编码 0b1111 的用途	(97)
2.2.8 寄存器域编码 0b1101 的用途	(98)
2.2.9 Cortex-M3 常用的 Thumb-2 指令	(99)
2.2.10 Thumb-2 指令与 ARM 体系架构下的指令比较	(114)
2.2.11 基于 Cortex-M3 的 Thumb-2 指令集	(118)
习题.....	(129)
第 3 章 STM32 基础及最小系统设计	(130)
3.1 从 Cortex-M3 到 STM32F103	(130)
3.1.1 微处理器、微控制器和系统.....	(130)
3.1.2 STM32F103 微控制器	(131)
3.2 存储器与总线架构	(139)
3.2.1 存储子系统基本构架	(139)
3.2.2 存储器映像	(140)
3.2.3 位带绑定(Bit-Banding)	(144)
3.2.4 嵌入式闪存	(145)
3.2.5 寄存器说明	(155)
3.2.6 启动配置	(158)
3.2.7 应用实例	(159)
3.3 中断和事件	(161)
3.3.1 嵌套向量中断控制器(NVIC)及其特性	(162)
3.3.2 外部中断/事件控制器(EXTI)	(164)
3.3.3 中断寄存器描述	(166)
3.3.4 应用实例	(169)
3.4 STM32F103x 的时钟系统	(173)
3.5 基于 STM32 的最小系统参考设计	(174)
习题.....	(178)
第 4 章 DMA 控制器	(179)
4.1 主要特性	(179)
4.2 功能描述	(180)

4.2.1 DMA 处理	(182)
4.2.2 仲裁器	(182)
4.2.3 DMA 通道	(183)
4.2.4 可编程的数据传输宽度、对齐方式和数据大小端	(185)
4.2.5 错误管理	(187)
4.2.6 DMA 请求映像	(187)
4.3 DMA 寄存器	(188)
4.4 DMA 应用实例	(192)
习题	(196)
第 5 章 通用和复用功能 I/O	(198)
5.1 GPIO 功能描述	(198)
5.1.1 通用目标 I/O(GPIO)	(200)
5.1.2 原子位设置或位清除	(200)
5.1.3 外部中断/唤醒线	(200)
5.1.4 复用功能(AF)	(200)
5.1.5 I/O 复用功能的软件重新映射	(201)
5.1.6 GPIO 锁定机制	(201)
5.1.7 输入配置与输出配置	(201)
5.1.8 复用功能配置	(203)
5.1.9 模拟输入配置	(204)
5.2 GPIO 寄存器描述	(205)
5.3 复用功能 I/O 和调试配置(AFIO)	(207)
5.3.1 引脚功能选择	(207)
5.3.2 BxCAN 复用功能重映射	(207)
5.3.3 JTAG/SWD 复用功能重映射	(208)
5.3.4 定时器复用功能重映射	(209)
5.3.5 USART 复用功能重映射	(210)
5.3.6 I2C 复用功能重映射	(211)
5.3.7 SPI 复用功能重映射	(211)
5.4 AFIO 寄存器描述	(211)
5.5 GPIO 和 AFIO 寄存器地址映像	(214)
5.5.1 GPIO 寄存器地址映像	(214)
5.5.2 AFIO 寄存器地址映像	(215)
5.6 应用实例	(215)
习题	(217)
第 6 章 STM32 高级定时器原理与应用	(218)
6.1 定时/计数器的基本原理与实现方法	(218)
6.1.1 完全硬件实现	(218)
6.1.2 纯软件方式	(219)

6.1.3 微控制器中的可编程定时/计数器	(219)
6.2 STM32 高级定时/计数器	(220)
6.2.1 STM32 高级定时器的主要特点	(220)
6.2.2 高级定时器概述	(221)
6.3 STM32 高级定时器寄存器描述	(222)
6.4 STM32 高级定时器工作原理及应用	(239)
6.4.1 定时器的时基信号	(239)
6.4.2 重复计数器	(245)
6.4.3 定时器定时应用实例	(246)
6.4.4 输入捕获	(247)
6.4.5 输出比较模式	(248)
6.4.6 STM32 高级定时器捕获/比较应用实例	(252)
6.4.7 STM32 高级定时器触发工作模式	(256)
习题	(258)
第 7 章 STM32 的 USART 模块	(259)
7.1 串行通信概述	(259)
7.2 串行通信的基本原理	(259)
7.2.1 USART 的扩展——RS232C 接口和标准	(259)
7.2.2 RS232C 的连接	(261)
7.2.3 流控和握手	(262)
7.2.4 分帧和组帧	(263)
7.2.5 错误检测和 CRC 校验	(263)
7.2.6 RS485	(264)
7.3 STM32F103 的串行通信模块	(264)
7.3.1 基本结构和连接	(265)
7.3.2 单字节传输	(266)
7.3.3 分频设置和波特率选择	(267)
7.3.4 基于 RTS 和 CTS 硬件握手协议的流控过程	(268)
7.3.5 常用全双工异步通信的发送配置	(270)
7.3.6 全双工异步通信的接收配置	(271)
7.3.7 关于传输错误	(272)
7.3.8 多处理器通信	(273)
7.3.9 校验控制	(273)
7.3.10 LIN 模式	(273)
7.3.11 USART 同步模式	(273)
7.3.12 单线半双工通信	(274)
7.3.13 智能卡	(274)
7.3.14 IrDA SIR ENDEC 功能块	(275)
7.3.15 利用 DMA 实现连续通信	(276)

7.3.16 中断请求	(277)
7.4 USART 寄存器描述	(278)
7.5 USART 应用实例分析	(286)
习题	(289)
第 8 章 STM32 的 SPI 模块	(290)
8.1 串行外设接口概述	(290)
8.2 串行外设接口 SPI 的基本原理	(291)
8.2.1 主从式连接架构	(291)
8.2.2 接口信号线介绍	(292)
8.2.3 数据传输的时序模式	(293)
8.2.4 多个从机的连接	(294)
8.3 STM32F103 的串行外设接口模块	(294)
8.3.1 基本结构和连接	(295)
8.3.2 时钟信号的相位和极性	(296)
8.3.3 数据帧格式	(297)
8.3.4 SPI 从模式	(297)
8.3.5 SPI 主模式	(298)
8.3.6 状态标志	(299)
8.3.7 CRC 计算	(299)
8.3.8 利用 DMA 的 SPI 通信	(300)
8.3.9 错误标志	(300)
8.3.10 中断	(301)
8.4 SPI 寄存器描述	(301)
8.5 SPI 应用实例分析	(306)
习题	(311)
第 9 章 I2C 总线原理及其应用	(312)
9.1 I2C 总线概述	(312)
9.1.1 I2C 总线特点	(312)
9.1.2 I2C 总线标准的发展历史	(313)
9.1.3 I2C 总线术语	(313)
9.2 I2C 总线原理	(314)
9.2.1 I2C 硬件构成	(314)
9.2.2 位传输	(315)
9.2.3 数据传输格式	(316)
9.3 STM32 I2C 模块原理	(319)
9.3.1 STM32 I2C 模块特点	(319)
9.3.2 I2C 寄存器描述	(320)
9.3.3 STM32 I2C 模块的通信实现	(330)
9.4 STM32 I2C 扩展 EEPROM 应用	(335)

9.4.1	概述	(336)
9.4.2	管脚描述	(336)
9.4.3	串行 EEPROM 芯片的寻址	(336)
9.4.4	写操作方式	(337)
9.4.5	读操作方式	(338)
9.4.6	STM32 I2C 模块扩展 24C64 应用	(339)
	习题	(345)
第 10 章	CAN 总线原理及其应用	(346)
10.1	CAN 总线概述	(346)
10.1.1	CAN 总线通信概述	(346)
10.1.2	CAN 报文传输	(348)
10.2	STM32 的 CAN 通信模块	(350)
10.2.1	STM32 bxCAN 通信模块概述	(350)
10.2.2	控制寄存器描述	(351)
10.2.3	邮箱寄存器描述	(359)
10.2.4	CAN 过滤器寄存器	(363)
10.3	STM32 bxCAN 模块工作过程	(364)
10.3.1	bxCAN 模块工作模式	(364)
10.3.2	bxCAN 模块数据发送管理	(365)
10.3.3	bxCAN 模块数据接收管理	(367)
10.3.4	bxCAN 模块标识符过滤器	(368)
10.3.5	bxCAN 模块出错管理	(370)
10.3.6	bxCAN 模块位时间特性	(370)
10.3.7	bxCAN 通信与出错中断管理	(371)
10.4	STM32 CAN 应用实例	(373)
10.4.1	CAN 总线硬件设计	(373)
10.4.2	STM32 CAN 通信软件示例	(373)
	习题	(378)
第 11 章	STM32 的模拟数字转换模块	(379)
11.1	A/D 变换的基本原理	(379)
11.1.1	采样	(380)
11.1.2	量化	(380)
11.1.3	编码	(381)
11.2	ADC 模块的主要技术指标和选型考虑	(383)
11.2.1	位数	(383)
11.2.2	采样速率	(383)
11.2.3	分辨率	(383)
11.2.4	量化误差	(384)
11.2.5	绝对精度	(385)

11.2.6	相对精度	(385)
11.2.7	偏移误差	(385)
11.2.8	增益误差	(385)
11.2.9	AD 线性误差	(386)
11.2.10	微分非线性	(387)
11.2.11	积分非线性	(387)
11.2.12	输入失调电压	(388)
11.2.13	输入失调电流	(388)
11.2.14	输入阻抗	(388)
11.2.15	增益带宽积 GBP	(389)
11.2.16	运放的单位增益带宽	(389)
11.2.17	运放建立时间	(389)
11.2.18	压摆率	(390)
11.3	ADC 模块的外围软硬件设计	(390)
11.3.1	电压测量与峰值测量	(391)
11.3.2	单路测量与多路测量	(391)
11.3.3	异步测量和同步测量	(391)
11.3.4	关于电压基准	(391)
11.3.5	查询式 A/D	(392)
11.3.6	中断式 A/D	(393)
11.3.7	Timer 驱动的周期采样	(393)
11.3.8	外部触发式启动 A/D	(394)
11.3.9	同步 A/D	(395)
11.3.10	DMA 数据传输	(395)
11.3.11	STM32F103 的 A/D 变换模块	(395)
11.4	STM32F103 ADC 寄存器介绍	(397)
11.5	STM32F103 的 ADC 模块的使用	(405)
11.5.1	ADC 的使能	(405)
11.5.2	ADC 时钟	(405)
11.5.3	通道选择	(405)
11.5.4	转换模式	(406)
11.5.5	模拟看门狗	(406)
11.5.6	扫描模式	(407)
11.5.7	注入通道管理	(407)
11.5.8	间断模式	(408)
11.5.9	校准	(408)
11.5.10	数据对齐	(408)
11.5.11	可编程的通道采样时间	(409)
11.5.12	外部触发转换	(409)

11.5.13	DMA 请求	(410)
11.5.14	双 ADC 模式	(410)
11.5.15	温度传感器/VREFINT 内部通道	(413)
11.5.16	中断	(414)
11.6	基于 STM32F103 的 A/D 变换示例	(414)
	习题	(417)
第 12 章	STM32 支撑开发环境	(418)
12.1	嵌入式系统开发的流程	(418)
12.1.1	嵌入式项目的生命周期	(418)
12.1.2	嵌入式软件的开发环节	(419)
12.1.3	交叉编译与软件调试	(419)
12.2	基于 Keil MDK 的 STM32 开发支撑环境	(421)
12.2.1	ARM 开发工具	(421)
12.2.2	基于 Keil MDK 的 STM32 开发环境	(422)
12.2.3	开发环境硬件连接	(423)
12.2.4	基于 Keil 的软件开发流程	(423)
12.3	STM32 启动文件解析	(433)
12.4	ARTX 嵌入式操作系统使用初步	(440)
12.5	嵌入式系统软件开发的高级主题	(442)
12.5.1	形式化规范与证实技术	(442)
12.5.2	设计架构与模式	(442)
12.5.3	低功耗软件设计	(444)
	习题	(444)
第 13 章	基于 STM32 的多功能综合实验板设计	(445)
13.1	综合实验板介绍	(445)
13.2	MDVSTM32-107 实验板模块设计	(446)
13.2.1	电源电路设计	(446)
13.2.2	通用 I/O 口电路设计	(447)
13.2.3	基于 I2C 总线扩展	(447)
13.2.4	CAN 总线扩展	(449)
13.2.5	USB 总线扩展	(450)
13.2.6	智能卡接口电路	(450)
13.2.7	I2S 扩展音频电路	(450)
13.2.8	RS232 扩展和 IrDA 扩展	(453)
13.2.9	SD 卡扩展电路	(453)
13.2.10	TFT 液晶扩展电路	(453)
13.2.11	电机扩展接口	(454)
13.2.12	以太网扩展	(457)
13.2.13	AD 电路扩展	(457)

13.2.14 MCU 电路设计	(457)
13.2.15 调试电路设计	(458)
13.2.16 扩展接口	(460)
习题	(462)
第 14 章 基于 STM32 的电动自行车控制器设计	(463)
14.1 直流无刷电机的基本原理	(463)
14.1.1 直流无刷电机结构	(463)
14.1.2 直流无刷电机工作原理	(464)
14.2 直流无刷电机应用系统设计	(467)
14.2.1 硬件电路实现	(469)
14.2.2 软件电路实现	(475)
习题	(581)
第 15 章 AMR 单相电能表的参考设计	(482)
15.1 需求和目标系统特性	(482)
15.2 硬件设计方案	(483)
15.2.1 层次化硬件架构和接口设计	(483)
15.2.2 测量电路	(483)
15.2.3 MCU 和外设电路	(484)
15.2.4 供电电路	(485)
15.2.5 磁保持继电器	(486)
15.3 STPM10 测量集成芯片	(486)
15.3.1 STPM10 测量芯片介绍	(486)
15.3.2 STPM10 与 MCU 的接口	(486)
15.3.3 使用 DMA 的 SPI 读过程	(490)
15.3.4 STPM10 校准	(490)
15.4 账户管理	(491)
15.5 目标机的测试与评估	(492)
15.5.1 目标机	(492)
15.5.2 AMR 的评估	(493)
习题	(494)
主要参考文献	(495)

第1章 嵌入式系统导论

本章首先陈述嵌入式系统的基本概念,然后通过对计算和计算机发展历史的凝练以及相应时代社会需求的特点来挖掘嵌入式领域发展的规律,使读者得以把握这个领域的发展方向。最后介绍嵌入式系统领域的典型供应商 ARM 和 ST 的产品,以及嵌入式系统工程开发的流程,并提出了本课程的学习目标。

1.1 嵌入式系统——从部件到系统的集成

1.1.1 什么是嵌入式系统

通用电子计算机自 20 世纪 40 年代诞生之后,一直向着高性能和智能化两个方向发展。但是在近 20 年间特别是最近 10 年,借助于微电子技术、通信技术和感知测量技术的发展,一个完整的计算机系统可以在更小的空间内实现,且仍能满足用户的需求,这使得计算机系统的应用范围从传统的科学计算与信息处理进一步拓展到通信、娱乐、视讯、测量、控制、国防、航空航天等各个领域,如图 1-1 常见嵌入式系统示例,计算机系统设计的目标,也由传统的以高性能为重偏向更加强调整满足用户需求和资源约束的平衡设计。总结计算机领域过去 60 年的发

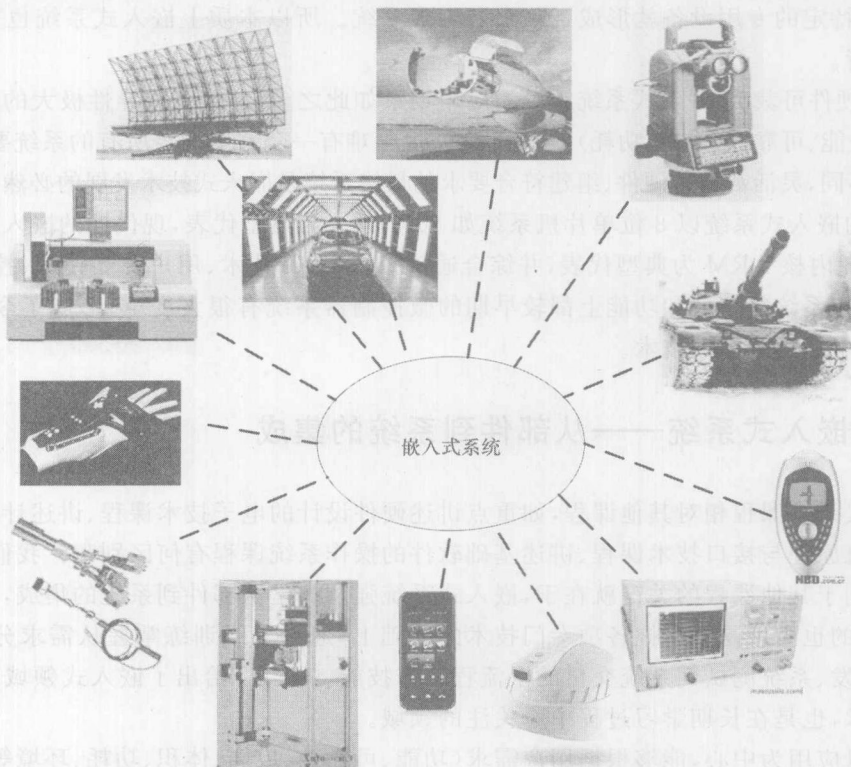


图 1-1 常见嵌入式系统示例

展历史可以发现,计算机系统经过了专用—通用—再专用的螺旋式发展道路,其中,今天嵌入式系统的广泛应用就是其“再专用”阶段的直接体现。

但是,由于嵌入式系统技术与非计算机学科如电子、通信、传感与测量、控制等学科的结合非常紧密,应用范围非常广泛,以至于很难给出一个严格的、公认的嵌入式系统定义,这里我们给出一个比较全面合理的说明:

嵌入式系统是以应用为中心,以现代计算机技术为基础,能够根据用户需求(功能、可靠性、成本、体积、功耗、环境等)灵活裁剪软硬件模块的专用计算机系统。

在上述关于什么是嵌入式系统的说明中,有这样几个要点:

- 以应用为中心:强调嵌入式系统的目标是满足用户的特定需求,而不是像目前的PC机那样定位在通用信息处理。就绝大多数完整的嵌入式系统而言,用户打开电源即可直接享用其功能,无需二次开发或仅需少量配置操作。

- 专用性:嵌入式系统的应用场合大多对可靠性、实时性有较高要求,这就决定了服务于特定应用的专用系统是嵌入式系统的主流模式,它并不强调系统的通用性和可扩展性,这与20世纪80年代强调通用化的微型计算机技术在出发点上是根本不同的。这种专用性通常也导致嵌入式系统是一个软硬件紧密集成的最终系统,因为这样才能更有效地提高整个系统的可靠性并降低成本,并使之具有更好的用户体验。

- 以现代计算机技术为核心:嵌入式系统的最基本支撑技术,大致上包括集成电路设计技术、系统结构技术、传感与检测技术、嵌入式操作系统(Embedded OS/EOS)和实时操作系统(RTOS)技术、资源受限系统的高可靠软件开发技术、系统形式化规范与验证技术、通信技术、低功耗技术、特定应用领域的数据分析、信号处理和控制优化技术等,它们围绕计算机基本原理,集成进特定的专用设备就形成了一个嵌入式系统。所以本质上嵌入式系统也是各种技术的集大成者。

- 软硬件可裁剪:嵌入式系统针对的应用场景如此之多,并带来差异性极大的设计指标要求(功能、性能、可靠性、成本、功耗),以至于现实上很难有一套方案满足所有的系统要求,因此根据需求的不同,灵活裁剪软硬件,组建符合要求的最终系统是嵌入式技术发展的必然技术路线。

早期的嵌入式系统以8位单片机系统如8051系统为典型代表,现代期的嵌入式系统则以32位低功耗内核ARM为典型代表,并综合通信技术、感知技术、用户交互技术、控制技术、低功耗技术,在系统复杂性和功能上都较早期的微控制器系统有很大变革,形成了今天以“嵌入式系统”为主题的一系列技术。

1.1.2 嵌入式系统——从部件到系统的集成

嵌入式系统课程相对其他课程,如重点讲述硬件设计的电子技术课程、讲述计算机设计与实践的微机原理与接口技术课程、讲述基础软件的操作系统课程有何区别呢?我们认为,嵌入式系统有别于其他课程的关键就在于,嵌入式系统强调的是从部件到系统的集成,课程学习的目标,相应的也应该是在掌握各项专门技术的基础上,通过实践训练掌握从需求分析、系统设计、系统开发、系统测试到系统交付的全流程工作技能。图1-2给出了嵌入式领域的核心技术和外围技术,也是在长期学习过程中应关注的领域。

作为以应用为中心,能够根据用户需求(功能、可靠性、成本、体积、功耗、环境等)灵活裁剪软硬件模块的嵌入式系统的出现与发展,是与20世纪80年代开始,计算机成本大幅度降低、