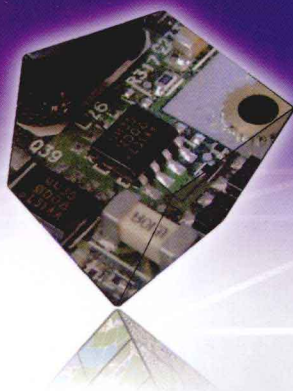


嵌入式技术与应用丛书
苏州大学精品教材建设资助项目



嵌入式系统 原理与实践

——ARM Cortex-M4 Kinetis 微控制器

王宜怀 吴 瑾 蒋银珍 编著

- ◎ 基于ARM Cortex-M4 Kinetis微控制器讲述嵌入式系统
- ◎ 深入分析学习嵌入式系统的误区，给出正确的学习方法
- ◎ 配有网上光盘，内含底层驱动构件源程序、测试实例、辅助阅读材料、教学课件、相关芯片资料及常用工具



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

嵌入式技术与应用丛书
苏州大学精品教材建设资助项目

嵌入式系统原理与实践

——ARM Cortex-M4 Kinetis 微控制器

王宜怀 吴 瑾 蒋银珍 编著

电子工业出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书是国内第一本以 ARM Cortex-M4 内核的 Kinetis 微控制器为蓝本来讲述嵌入式系统的图书。Kinetis 系列微控制器将高效的 ARM Cortex-M4 内核与先进的低功耗设计技术相结合,是业内功耗最低的基于 Cortex-M4 的 MCU 解决方案。全书共 15 章,其中前 4 章简要阐述了嵌入式系统的知识体系、学习误区、学习建议和基于硬件构件的嵌入式系统开发方法,给出了 ARM Cortex-M4 简介及 K60 硬件最小系统,示例了第一个样例程序及开发环境下的工程组织方法,完成了第一个 K60 工程的入门任务,并讲解了第一个带中断的实例,前 4 章囊括了学习一个新 MCU 完整要素的入门环节;第 5 章到 14 章分别给出了 GPIO 的应用实例(键盘、LED 与 LCD)、定时器、A/D、D/A、比较器、TSI、SPI、I2C、I2S、Flash、CAN、SDHC、USB、以太网及 K60 其他模块等;最后一章给出了进一步学习的指导。

本书适用于有关高校嵌入式系统的教学或技术培训资料,也可供 ARM Cortex-M4 应用工程师作为技术研发参考。

本书提供了网上光盘,内含所有底层驱动构件源程序、测试实例、辅助阅读材料、教学课件、相关芯片资料及常用软件工具,网上光盘下载地址 <http://sumcu.suda.edu.cn>;读者也可以登录华信教育资源网(www.hxedu.com.cn)免费注册后下载。为了更好地服务读者,作者为本书开通了 QQ(1601873607,昵称为 MQXandKinetis),同时也开通了新浪微博,名称为 MQXandKinetis(<http://weibo.com/u/2579021437>),欢迎广大读者访问交流,并提出宝贵建议。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式系统原理与实践:ARM Cortex-M4 Kinetis 微控制器/王宜怀等编著. —北京:电子工业出版社,2012.3
(嵌入式技术与应用丛书)

ISBN 978-7-121-15822-3

I. ①嵌… II. ①王… III. ①微处理器, ARM IV. ①TP332

中国版本图书馆 CIP 数据核字(2012)第 016533 号

责任编辑:田宏峰

印 刷:北京市李史山胶印厂

装 订:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:26.75 字数:680 千字

印 次:2012 年 3 月第 1 次印刷

印 数:4 000 册 定价:49.90 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

嵌入式计算机系统简称为嵌入式系统，其概念最初源于传统测控系统对计算机的需求。随着以微处理器（MPU）为内核的微控制器（MCU）制造技术的不断进步，计算机领域在通用计算机系统与嵌入式计算机系统这两大分支分别得以发展。通用计算机已经在科学计算、通信、日常生活等各个领域产生了重要的影响。在后 PC 时代，嵌入式系统的广阔应用是计算机发展的重要特征。一般来说，嵌入式系统的应用范围可以粗略分为两大类：一类是电子系统的智能化（如工业控制、汽车电子、数据采集、测控系统、家用电器、现代农业、传感网应用等），这类应用也被称为微控制器领域；另一类是计算机应用的延伸（如平板电脑、手机、电子图书等），这类应用也被称为应用处理器领域。在 ARM 产品系列中，ARM Cortex-M 系列与 ARM Cortex-R 系列适用于电子系统的智能化类应用，即微控制器领域；ARM Cortex-A 系列适用于计算机应用的延伸，即应用处理器领域。不论如何分类，嵌入式系统的技术基础是不变的，即要完成一个嵌入式系统产品的设计，需要有硬件、软件及行业领域相关知识。但是，随着嵌入式系统中软件规模日益增大，对嵌入式底层驱动软件的封装提出了更高的要求，可复用性与可移植性受到特别的关注，嵌入式软、硬件构件化开发方法逐步被业界所重视。

本书读者对象

本书以嵌入式硬件构件与底层软件构件设计为主线，按照嵌入式软件工程的要求，以飞思卡尔（Freescale）的 Kinetis 系列微控制器（ARM Cortex-M4 内核）中的 K60 为蓝本阐述嵌入式系统的软件、硬件设计及基本开发方法。包含通用知识、底层驱动构件基本思想、底层驱动构件源程序、底层驱动构件测试与应用实例等。主要面向高年级本科生或研究生的教学。为适合于教学及篇幅所限，注意以下几点：①从应用角度阐述嵌入式系统通用基础知识，并分散到相应章节中；②本书不阐述汇编指令，一开始就使用 C 语言编程；③不详细介绍芯片内部寄存器，而是从编程方法角度给出编程要点及底层驱动构件封装与应用方法；④为方便教学与实际应用，书中列出所有底层驱动构件对外接口函数原型；⑤给出所有底层驱动构件的测试用例；⑥提供网上光盘，给出丰富的教学与开发资料，也扩展了纸质版的内容。与本书随后出版的还有我们撰写的另外两本书，一本是《Kinetis 微控制器体系结构与开发实践——基于 ARM Cortex-M4 内核》，给出 ARM Cortex-M4 的汇编指令分析、Kinetis 微控制器的内部寄存器、底层驱动构件封装原理的技术细节及应用注意点等，主要面向工程师，也可供教师及研究生参考，该书不再阐述嵌入式系统的通用基础知识。另一本是《嵌入式实时操作系统——MQX 应用开发技术》，阐述飞思卡尔的开源嵌入式实时操作系统 MQX 的应用方法，该书也以 Kinetis 微控制器为蓝本进行撰写。这三本书形成一个 ARM Cortex-M4、Kinetis 微控制器、MQX 的体系。教育部支持的高校科技竞赛之一“全国大学生飞思卡尔杯智能汽车竞赛”，从 2012 年开始可以使用 Kinetis 微控制器，一些学校也以此为蓝本进行嵌入式系统及应用教学，本书的主要目的是配

合这两项工作。一些学校举行智能车比赛所使用的是 K10 芯片，它是 K60 芯片的子集，K10 芯片没有 USB、以太网等较为复杂的模块。

本书基本思想

从 1991 年开始进行单片机与嵌入式系统应用开发至今，作者从事该方向研究与教学工作已有 20 多年时间。其中 1991—1999 年期间，使用 MCS-51 系列 MCU。2000 年以来，陆续以飞思卡尔（2004 年以前是摩托罗拉半导体部）的 HC08/S08、S12/S12X、ColdFire 等系列 MCU 为蓝本先后写了一些嵌入式应用技术入门方面的书，得到了许多读者的肯定，深受感动。在写书方面，多年来一直在探索如何能够使读者不误入歧途，如何能够快速入门，如何能够规范编程，如何能够由浅入深、循序渐进，如何能够使读者打好嵌入式硬件与软件基础。为此从以下几点把握写作。

（1）把与芯片无关的通用知识分离出来，从涉及底层编程角度对基本原理进行简明扼要的阐述，分别放入相应章节的前面或网上光盘中。这些知识主要包括通用 I/O、串行通信、键盘编码原理、LED 扫描原理、SPI、PWM、USB、I2C、CAN、A/D、D/A、嵌入式以太网等，并在各书中基本保持不变。经过多年不断修改完善，可把通用部分斟酌得更好一些。也使得 8 位、16 位、32 位的书风格保持一致。新的芯片出来后，书的修改只要更新与芯片相关的部分即可。

（2）与硬件相关的部分采用了硬件构件思想，制定了一些基本规范，对底层驱动进行构件化封装，既提高了可复用性与可移植性，又使程序结构更加清晰，从而让初学者可以“先使用、后理解”。

（3）不论是 8 位、16 位、32 位，也不论是哪个芯片，从编程角度来看，把与硬件相关的共性和与硬件无关的共性分别抽象出来，力求做到：硬件相关部分风格一致，硬件无关部分程序一致。这样便于融会贯通，不再纠结于芯片位数、操作系统等问题。

从 2010 年 6 月拿到飞思卡尔 Kinetis 系列参考手册 Rev.0 开始，至今已近一年半时间，参考手册也升级到 Rev.6。期间，苏州大学飞思卡尔嵌入式与物联网研发团队学习消化了 1 800 多页的参考手册；制作了 K60 核心板、Kinetis 系列写入器；为了与 RTOS 无缝衔接而修订原来的底层驱动构件规范；按统一规范逐个模块编写底层驱动构件，并设计测试用例；与 SD-ExtBoard-D 型扩展板对接并编写实验教学程序；在实时操作系统 MQX 下编写用例，以检验底层驱动构件的无缝衔接。本书就是在这些工作的基础上，结合多年来撰写嵌入式书籍的经验写成的。

关于飞思卡尔微控制器

飞思卡尔是全球最大的半导体公司之一，在微控制器领域长期居全球市场领先地位，以高可靠性获得业界的一致赞誉。该公司的微控制器产品系列齐全，由不同位数（如 8 位、16 位、32 位等）、不同封装形式（如 DIP、SOIC、QFP、LQFP、BGA 等）、不同温度范围（0~70℃、-40~85℃、-40~105℃、-40~125℃等）、所含模块不同等构成了庞大的产品系列。飞思卡尔的 S08（8 位）、S12/S12X（16 位）、ColdFire（32 位）、ARM Cortex（32 位）等系列 MCU，广泛地应用于汽车电子、消费电子、工业控制、网络和无线市场等嵌入式系统各个领域。系列齐全的微控制器产品，为嵌入式系统各种应用提供了选择与解决方案，使得用户可以各取所需。不论是电子系统智能化还是计算机应用延伸的嵌入式应用设计，无论需要怎样的系统功能和集成度，总能从飞思卡尔庞大产品系列中选取一款合适的芯片进行应用开发。这正是嵌入式系统产品设计者所期望的，也节省了嵌入式学习者的时间，可以加快开发进度，提高开发质量。

本书特点

2009 年以来, 作者撰写的《基于 32 位 ColdFire 构建嵌入式系统》、《嵌入式技术基础与实践(第 2 版)》与《嵌入式系统设计实战——基于飞思卡尔 S12X 微控制器》等书, 系统阐述和应用了嵌入式构件开发思想, 本书秉承这些理念, 按照“通用知识—芯片编程结构概要—基本编程方法—底层驱动构件封装—应用方法与举例”的线条, 逐步阐述电子系统智能化嵌入式应用的软件与硬件设计。

(1) 把握通用知识与芯片相关知识之间的平衡。书中对于嵌入式“通用知识”的基本原理, 以应用为立足点, 进行语言简洁、逻辑清晰的阐述, 同时注意与芯片相关知识之间的衔接, 使读者在更好地理解基本原理的基础上, 理解芯片应用的设计, 反过来又加深对通用知识的理解。

(2) 把握硬件与软件的关系。嵌入式系统是软件与硬件的综合体, 嵌入式系统设计是一个软件、硬件协同设计的工程, 不能像通用计算机那样把软件、硬件完全分开。特别是对电子系统智能化嵌入式应用来说, 没有对硬件的理解就不可能写好嵌入式软件, 同样没有对软件的理解也不可能设计好嵌入式硬件。因此, 本书注重把握硬件知识与软件知识之间的关系。

(3) 对底层驱动进行构件化封装。书中对每个模块均给出根据嵌入式软件工程基本原则并按照构件化封装要求编制底层驱动程序, 同时给出详细、规范的注释及对外接口, 为实际应用提供底层构件, 方便移植与复用, 可以为读者进行实际项目开发节省大量时间。

(4) 设计合理的测试用例。书中所有源程序均经测试通过, 并保留测试用例在本书的网上光盘中, 避免了因例程的书写或固有错误给读者带来烦恼。这些测试用例, 也将为读者验证与理解带来方便。

(5) 网上光盘提供了所有模块完整的底层驱动构件化封装程序与测试用例(包括 CW-IDE 下的程序、IAR-IDE 下的程序)。需要使用 PC 的程序测试用例, 还提供了 PC 的 C#源程序。网上光盘中还提供了分章阅读资料(分章英文资料及中文参考翻译)、CW10.1 简明使用方法、IAR 简明使用方法、写入器驱动与使用方法、部分工具软件、有关硬件原理图及其他技术资料。网上光盘的版本将会适时更新。

(6) 提供硬件核心板、写入调试器, 方便读者进行实践与应用。同时提供了核心板与苏州大学飞思卡尔嵌入式中心设计制作的 SD-ExtBoard-D 型扩展板的对接, 以满足教学实验需要。

本书主要内容

本书以飞思卡尔的 ARM Cortex-M4 内核的 Kinetis 系列微控制器为蓝本阐述嵌入式系统的软件与硬件设计。全书共 15 章, 其中第 1 章为概述, 简要阐述嵌入式系统的知识体系、学习误区、学习建议和基于硬件构件的嵌入式系统开发方法。第 2 章给出 ARM Cortex-M4 简介及 K60 硬件最小系统。第 3 章给出第一个样例程序及开发环境下的工程组织方法, 完成第一个 K60 工程的入门任务。第 4 章阐述串行通信接口 UART, 并给出第一个带中断的实例。第 1~4 章囊括了学习一个新 MCU 完整要素的入门环节。第 5~14 章分别给出 GPIO 的应用实例(键盘、LED 与 LCD)、定时器、A/D、D/A、比较器、TSI、SPI、I2C、I2S、Flash、CAN、SDHC、USB、以太网及 K60 其他模块等。第 15 章给出进一步学习指导。

本书提供了网上光盘, 内含所有底层驱动构建源程序、测试实例、辅助阅读材料、教学课件、相关芯片资料及常用工具, 下载网址为 <http://sumcu.suda.edu.cn>, 读者也可以登录华信教育资源网(www.hxedu.com.cn)免费注册后下载。为了更好地服务读者, 作者为本书开通了 QQ(1601873607, 昵称为 MQXandKinetis), 同时也开通了新浪微博, 名称为 MQXandKinetis (<http://weibo.com/u/2579021437>), 欢迎广大读者访问交流, 并提出宝贵建议。

作者分工与致谢

本书由王宜怀负责编制提纲和统稿工作,并撰写第1~6章、第15章。吴瑾撰写第7~10章、蒋银珍撰写第11~14章。苏州大学嵌入式系统与物联网方向研究生朱乐乐、苏勇、石晶、姚丹丹、李翠霞、周杰、秦保波、陈爱兵等协助书稿整理及程序调试工作,他们卓有成效的工作,使本书更加实用。飞思卡尔半导体有限公司的 Andy Mastronardi 先生、殷刚先生、马莉女士一直关心支持苏州大学飞思卡尔嵌入式中心的建设,为本书的撰写提供了硬件及软件资料,并提出了许多宝贵建议。飞思卡尔半导体有限公司的许多技术人员提供了技术支持。电子工业出版社的田宏峰编辑为本书的出版做了大量细致的工作,在此一并表示诚挚的谢意。

鉴于作者水平有限,书中难免存在不足和错误之处,恳望读者提出宝贵意见和建议,以便再版时改进。

王宜怀

2012年2月于苏州大学

目 录

第 1 章 概述	1	第 2 章 Kinetis 微控制器概述与 K60 硬件 最小系统	24
1.1 嵌入式系统定义、由来及特点	1	2.1 学习一个新 MCU 芯片的基本要素	24
1.1.1 嵌入式系统的定义	1	2.2 ARM 背景知识简介	25
1.1.2 嵌入式系统的由来及其与 微控制器的关系	2	2.2.1 ARM 简介	25
1.1.3 嵌入式系统的特点	3	2.2.2 Cortex-M4 处理器特性简介	28
1.2 嵌入式系统的知识体系、学习误区及 学习建议	4	2.3 Kinetis 系列微控制器概述及型号标识	29
1.2.1 嵌入式系统的知识体系	4	2.3.1 Kinetis 系列微控制器概述	29
1.2.2 嵌入式系统的学习误区	5	2.3.2 Kinetis 系列微控制器型号标识	31
1.2.3 基础阶段的学习建议	7	2.4 K60 系列微控制器的存储器映像与 编程结构	31
1.2.4 嵌入式系统开发所遇到的 若干问题	8	2.4.1 K60 系列 MCU 性能概述与 内部结构简图	31
1.3 嵌入式硬件构件的基本思想与 应用方法	9	2.4.2 K60 系列存储器映像	33
1.4 基于硬件构件的嵌入式系统 硬件电路设计	9	2.5 K60 的引脚功能与硬件最小系统	34
1.4.1 设计时需要考虑的基本问题	9	2.5.1 K60 的引脚功能	35
1.4.2 硬件构件化电路原理图绘制的 简明规则	11	2.5.2 K60 的硬件最小系统原理图	37
1.4.3 实验 PCB 板设计的简明规则	13	2.5.3 Kinetis 写入器与 K60 核心板	39
1.5 基于硬件构件的嵌入式底层软件构件的 编程方法	16	2.5.4 硬件最小系统测试方法	40
1.5.1 嵌入式硬件构件和软件构件的 层次模型	16	2.6 ARM Cortex-M4 的寄存器及指令简介	41
1.5.2 底层构件的实现方法与 编程思想	17	2.6.1 ARM Cortex-M4 的寄存器简介	41
1.5.3 硬件构件及底层软件构件的 重用与移植方法	18	2.6.2 ARM Cortex-M4 的指令 系统简介	43
1.6 嵌入式系统的常用术语	19	2.7 本章小结	44
1.6.1 与硬件相关的术语	19	习题	44
1.6.2 与通信相关的术语	21	第 3 章 第一个样例程序及工程组织	45
1.6.3 与功能模块及软件相关的术语	21	3.1 GPIO 模块的驱动构件设计	45
1.7 本章小结	22	3.1.1 GPIO 的基础知识	45
习题	23	3.1.2 GPIO 模块概要与编程要点	46
		3.1.3 GPIO 驱动构件设计	48
		3.2 CodeWarrior 开发环境简介	54
		3.3 嵌入式设计编码基本规范	55
		3.3.1 硬件驱动构件文件	56

3.3.2 数据类型	56	5.1.3 键盘驱动构件测试实例	107
3.3.3 函数	57	5.2 LED 模块的驱动构件设计	109
3.3.4 源码文件夹结构	58	5.2.1 LED 的基础知识	109
3.4 第一个 C 语言工程: 控制小灯闪烁	59	5.2.2 LED 驱动构件设计	110
3.4.1 Light 构件设计	59	5.2.3 LED 驱动构件测试实例	113
3.4.2 Light 构件测试工程主程序	62	5.3 LCD 模块的驱动构件设计	115
3.4.3 在 CW 环境下导入样例工程	63	5.3.1 LCD 的基础知识	115
3.5 理解第一个 C 工程	63	5.3.2 LCD 驱动构件设计	120
3.5.1 CW 开发环境下工程文件 组织框架	64	5.3.3 LCD 驱动构件测试实例	125
3.5.2 文件说明	65	5.4 本章小结	126
3.5.3 芯片上电启动执行过程	72	习题	127
3.6 在 CW 环境下创建一个新的工程	72	第 6 章 定时器相关模块	128
3.7 本章小结	73	6.1 计数器/定时器的基本工作原理	128
习题	74	6.2 可编程延迟模块 PDB	129
第 4 章 异步串行通信	75	6.2.1 PDB 的基础知识	129
4.1 异步串行通信的基础知识	75	6.2.2 PDB 模块概要与编程要点	129
4.1.1 基本概念	75	6.2.3 PDB 构件设计及测试实例	130
4.1.2 RS-232C 总线标准	77	6.3 Flex 定时器 FTM	135
4.1.3 电平转换电路原理	78	6.3.1 FTM 的基础知识	135
4.2 UART 模块功能概述	79	6.3.2 FTM 模块概要与编程要点	135
4.3 UART 模块的编程结构	80	6.3.3 FTM 构件设计及测试实例	138
4.4 UART 模块的底层驱动构件设计	85	6.4 周期中断定时器 PIT	142
4.4.1 UART 构件的函数原型设计	85	6.4.1 PIT 的基础知识	142
4.4.2 UART 构件的头文件 (hw_uart.h)	85	6.4.2 PIT 模块概要与编程要点	143
4.4.3 UART 构件的源程序文件 (hw_uart.c)	88	6.4.3 PIT 构件设计及测试实例	144
4.5 以查询方式接收的 UART 模块 测试实例	92	6.5 低功耗定时器 LPTMR	148
4.6 以中断方式接收的 UART 模块 测试实例	94	6.5.1 LPTMR 的基础知识	148
4.7 本章小结	96	6.5.2 LPTMR 模块概要与编程要点	148
习题	97	6.5.3 LPTMR 构件设计及测试实例	148
第 5 章 GPIO 的应用实例: 键盘、LED 与 LCD	98	6.6 载波调制发射器 (CMT)	155
5.1 键盘模块的驱动构件设计	98	6.6.1 CMT 的基础知识	155
5.1.1 键盘模型及接口	98	6.6.2 CMT 模块概要与编程要点	156
5.1.2 键盘驱动构件设计	100	6.6.3 CMT 构件设计及测试实例	157
		6.7 实时时钟	160
		6.7.1 RTC 基础知识	160
		6.7.2 RTC 模块概要与编程要点	161
		6.7.3 RTC 构件设计及测试实例	162
		6.8 本章小结	166
		习题	167

第7章 A/D、D/A、CMP 和 TSI 模块..... 168

7.1	16 位 A/D 转换模块的驱动构件设计.....	168
7.1.1	A/D 转换的基础知识.....	168
7.1.2	A/D 转换常用传感器简介.....	169
7.1.3	A/D 转换模块概要与编程要点.....	170
7.1.4	A/D 转换驱动构件设计.....	171
7.1.5	A/D 转换驱动构件测试实例.....	175
7.2	12 位 D/A 转换模块的驱动构件设计.....	177
7.2.1	D/A 转换的基础知识.....	177
7.2.2	D/A 转换模块概要.....	178
7.2.3	D/A 转换驱动构件设计.....	179
7.2.4	D/A 转换驱动设计构件 测试实例.....	180
7.3	K60 的比较器模块的驱动构件设计.....	182
7.3.1	CMP 基础知识.....	182
7.3.2	CMP 模块概要与编程要点.....	182
7.3.3	CMP 驱动构件设计.....	184
7.3.4	CMP 驱动构件测试实例.....	187
7.4	触摸感应接口.....	189
7.4.1	TSI 基础知识.....	189
7.4.2	TSI 模块概要与编程要点.....	190
7.4.3	TSI 驱动构件设计.....	190
7.4.4	TSI 驱动构件测试实例.....	194
7.5	本章小结.....	196
	习题.....	197

第8章 SPI、I2C 与 I2S..... 198

8.1	SPI 模块的驱动构件设计.....	198
8.1.1	SPI 基础知识.....	198
8.1.2	K60 的 SPI 模块概要与 编程要点.....	202
8.1.3	SPI 驱动构件设计.....	204
8.1.4	SPI 驱动构件测试实例.....	210
8.2	I2C 模块的驱动构件设计.....	211
8.2.1	I2C 基础知识.....	211
8.2.2	K60 的 I2C 模块概要与 编程要点.....	215
8.2.3	I2C 驱动构件设计.....	217
8.2.4	I2C 驱动构件测试实例.....	224
8.3	I2S 模块的驱动构件设计.....	227

8.3.1	I2S 基础知识.....	227
8.3.2	K60 的 I2S 模块概要与 编程要点.....	229
8.3.3	I2S 驱动构件设计.....	232
8.3.4	I2S 驱动构件测试实例.....	235
8.4	本章小结.....	236
	习题.....	237

第9章 Flash 在线编程..... 238

9.1	Flash 基础知识.....	238
9.2	Flash 模块概要与编程要点.....	239
9.2.1	K60 的 Flash 模块.....	239
9.2.2	Flash 存储器主要寄存器说明.....	241
9.2.3	Flash 存储器编程要素.....	243
9.3	Flash 在线编程驱动构件设计.....	247
9.3.1	Flash 在线编程驱动构件头 文件.....	247
9.3.2	Flash 模块驱动构件源文件.....	248
9.4	Flash 在线编程驱动构件测试实例.....	251
9.5	Flash 模块保护与加密功能.....	255
9.5.1	Flash 模块的保护功能.....	255
9.5.2	Flash 模块的加密功能.....	255
9.6	本章小结.....	257
	习题.....	257

第10章 K60 的 CAN 总线开发方法..... 258

10.1	CAN 总线基础知识.....	258
10.1.1	CAN 总线协议的历史概况.....	258
10.1.2	CAN 硬件系统的典型电路.....	258
10.1.3	CAN 总线的有关基本概念.....	260
10.1.4	帧结构.....	263
10.1.5	位时间.....	266
10.2	FlexCAN 模块概述与编程要点.....	267
10.2.1	FlexCAN 模块特征.....	267
10.2.2	操作模式.....	268
10.2.3	内存映射/寄存器简介.....	269
10.2.4	报文缓冲区结构.....	270
10.2.5	仲裁处理、匹配处理及 报文缓冲区管理.....	273
10.2.6	FlexCAN 驱动构件设计.....	274
10.3	CAN 驱动构件测试实例.....	283

10.3.1 回环测试	283	12.4.3 K60 作为 USB 从机的 构件化设计	341
10.3.2 非回环测试	286	12.4.4 K60 的 USB 模块测试实例	348
10.4 本章小结	287	12.5 本章小结	350
习题	288	习题	350
第 11 章 大容量 SD 存储卡 SDHC	289	第 13 章 基于 K60 的嵌入式以太网	351
11.1 SDHC 基本概念及硬件特性	289	13.1 嵌入式以太网相关基础知识	351
11.1.1 SD 概述	289	13.1.1 以太网的由来与协议模型	351
11.1.2 SD 历史与发展	289	13.1.2 以太网中主要物理设备	353
11.1.3 SD 的物理特性	290	13.1.3 相关名词解释	355
11.2 SD 存储卡系统概念	292	13.2 K60 以太网模块概要与编程要点	358
11.2.1 SD 总线拓扑结构	292	13.2.1 以太网模块引脚及硬件连接	358
11.2.2 总线协议	293	13.2.2 以太网物理层收发器 EPHY 的 编程基础	359
11.3 SDHC 模块基本编程方法	296	13.3 以太网驱动构件设计	360
11.3.1 SDHC 模块功能简介	296	13.3.1 物理层构件	360
11.3.2 SD 模块编程概要	298	13.3.2 链路层构件	363
11.4 SD 读写驱动构件设计	298	13.4 以太网测试实例	373
11.4.1 SDHC 构件驱动设计	299	13.4.1 第一个构件实例：网络连接	373
11.4.2 DiskIO 驱动构件设计	304	13.4.2 第二个测试实例：以太网帧的发送 和接收	375
11.4.3 SD 卡读写测试主程序	313	13.4.3 第三个测试实例：链路层 ARP 协议编程	378
11.5 本章小结	315	13.4.4 第四个测试实例：使用 ICMP 协议响应 Ping 请求	380
习题	316	13.4.5 第五个测试实例：UDP 报文的 发送和接收	383
第 12 章 USB 2.0 编程	317	13.4.6 第六个测试实例：TCP 报文的 发送和接收	384
12.1 USB 基本概念及硬件特性	317	13.4.7 第七个测试实例：HTTP 协议静 态页面的实现	388
12.1.1 USB 概述	317	13.5 本章小结	390
12.1.2 USB 相关基本概念	319	习题	391
12.1.3 USB 的物理特性	323	第 14 章 系统时钟与其他功能模块	392
12.2 USB 的通信协议	325	14.1 时钟系统	392
12.2.1 USB 基本通信单元：包	325	14.1.1 时钟系统概述	392
12.2.2 USB 通信中的事务处理	327	14.1.2 时钟设置方法及实例代码 分析	395
12.2.3 从设备的枚举看 USB 数据传输	328		
12.3 USB 模块功能简介	332		
12.3.1 K60 的 USB 模块功能简介	332		
12.3.2 K60 的 USB 模块主要寄存器 介绍	333		
12.4 K60 作为 USB 从机的开发方法	337		
12.4.1 PC 方 USB 设备驱动程序的 选择及基本原理	337		
12.4.2 PC 作为 USB 主机的程序 设计	340		

14.2 芯片配置模块.....	397	14.6.2 交叉开关	404
14.3 电源管理模块.....	398	14.7 看门狗	405
14.3.1 电源模式.....	399	14.7.1 看门狗定时器	405
14.3.2 低功耗唤醒单元.....	400	14.7.2 外部看门狗	411
14.4 端口控制与中断模块.....	401	14.8 本章小结	411
14.4.1 端口控制与中断模块简介.....	401	习题	412
14.4.2 功能说明.....	402	第 15 章 进一步学习指导	413
14.5 复位与启动模块.....	403	15.1 关于更为详细的技术资料	413
14.5.1 复位.....	403	15.2 关于实时操作系统 RTOS.....	413
14.5.2 启动模块.....	404	15.3 关于嵌入式系统稳定性问题	414
14.6 杂项控制模块与交叉开关.....	404	参考文献	416
14.6.1 杂项控制模块.....	404		

本章导读：作为全书导引，本章主要知识点有①简要给出嵌入式系统定义、由来及特点；②讨论嵌入式开发中的硬件、软件的可复用与可移植性问题；③提出嵌入式硬件构件的基本思想，阐述基于硬件构件的嵌入式系统开发方法；④给出基于硬件构件的嵌入式系统硬件电路设计与嵌入式底层软件构件的编程方法；⑤给出嵌入式系统常用术语。

◆ 1.1 嵌入式系统定义、由来及特点

1.1.1 嵌入式系统的定义

嵌入式系统 (Embedded System) 有多种多样的定义，但本质是相同的。本书关于嵌入式系统的定义取自于美国 CMP Books 出版的 Jack Ganssle 和 Michael Barr 著作《Embedded System Dictionary》^①。

嵌入式系统的定义：一种计算机硬件和软件的组合，也许还有机械装置，用于实现一个特定功能。在某些特定情况下，嵌入式系统是一个大系统或产品的一部分。世界上第一个嵌入式系统是 1971 年 Busicom 公司用 Intel 单芯片 4004 微处理器完成的商用计算器系列。该词典^①还给出了嵌入式系统的一些示例，如微波炉、手持电话、计算器、数字手表、录像机、巡航导弹、GPS 接收机、数码相机、传真机、跑步机、遥控器和谷物分析仪等，难以尽数。通过与通用计算机的对比可以更形象地理解嵌入式系统的定义。该词典给出的通用计算机定义是：计算机硬件和软件的组合，用于通用计算平台。PC、MAC 和 UNIX 工作站是最流行的现代计算机。

我国国家标准 GB/T 5271《信息技术词汇—嵌入式系统与单片机》部分，给出的嵌入式系统定义是：置入应用对象内部起操作控制作用的专用计算机系统。

国内对嵌入式系统定义曾进行过广泛讨论，有许多不同说法，其中嵌入式系统定义的涵盖面问题是主要争论焦点之一。例如，有的学者认为不能把手持电话叫嵌入式系统，而只能把其中起控制作用的部分叫嵌入式系统，而手持电话可以称为嵌入式系统的应用产品。其实，这些并不妨碍人们对嵌入式系统的理解，所以不必对定义感到困惑。有些国内学者特别指出，在理解嵌入式系统定义时，不要把嵌入式系统与嵌入式系统产品相混淆。实际上，从口语或书面语言角度，不区分“嵌入式系统”与“嵌入式系统产品”，只要不妨碍对嵌入式系统的理解就没有关系。

为了更清楚地阐述嵌入式系统特点，首先介绍大多数嵌入式系统的核心部件——MCU（微控制器）的基本概念。

① 中译本：Jack Ganssle 等著，马广云等译，《英汉双解嵌入式系统词典》，北京航空航天大学出版社，2006 年。

1.1.2 嵌入式系统的由来及其与微控制器的关系

1. MCU（微控制器）的基本含义

MCU 是单片微型计算机（单片机）的简称，早期的英文名是 Single-chip Microcomputer，后来大多数称之为微控制器（Microcontroller）或嵌入式计算机（Embedded Computer）。现在 Microcontroller 已经是计算机中一个常用术语，但在 1990 年之前，大部分英文词典并没有这个词。我国学者一般使用中文“单片机”一词，而缩写使用“MCU”^①。所以本书后面的简写一律以 MCU 为准。MCU 的基本含义是：在一块芯片上集成了中央处理单元（CPU）、存储器（RAM/ROM 等）、定时器/计数器及多种输入/输出（I/O）接口的比较完整的数字处理系统。图 1-1 给出了典型的 MCU 组成框图。

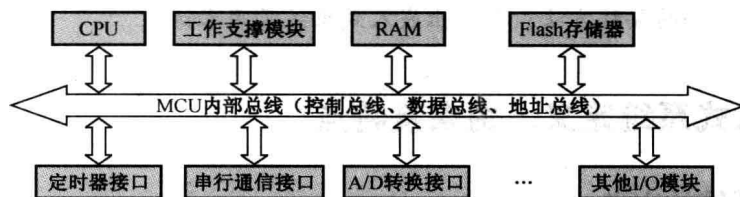


图 1-1 一个典型的 MCU 内部框图

MCU 是在计算机制造技术发展一定阶段的背景下出现的，它使计算机技术从科学计算领域进入到智能化控制领域。从此，计算机技术在两个重要领域——通用计算机领域和嵌入式（Embedded）计算机领域都获得了极其重要的发展，为计算机的应用开辟了更广阔的空间。

就 MCU 组成而言，虽然它只是一块芯片，但包含了计算机的基本组成单元，仍由运算器、控制器、存储器、输入设备、输出设备五部分组成，只不过这些都集成在一块芯片上，这种结构使得 MCU 成为具有独特功能的计算机。

2. 嵌入式系统的由来

通俗地说，计算机是因科学家需要一个高速的计算工具而产生的。直到 20 世纪 70 年代，电子计算机在数字计算、逻辑推理及信息处理等方面表现出非凡的能力。在通信、测控与数据传输等领域，人们对计算机技术给予了更大的期待。这些领域的应用与单纯的高速计算要求不同，主要表现在：直接面向控制对象；嵌入到具体的应用体中，而非计算机的面貌出现；能在现场连续可靠地运行；体积小，应用灵活；突出控制功能，特别是对外部信息的捕捉与丰富的输入/输出功能等。由此可以看出，满足这些要求的计算机与满足高速数值计算的计算机是不同的。因此，一种称之为 MCU 或微控制器的技术得以产生并发展。为了区分这两种计算机类型，通常把满足海量高速数值计算的计算机称为通用计算机系统，而把嵌入到实际应用系统中，实现嵌入式应用的计算机称之为嵌入式计算机系统，简称嵌入式系统。

3. 嵌入式系统与 MCU 的关系

何立民先生说：“有些人搞了十多年的 MCU 应用，不知道 MCU 就是一个最典型的嵌入式系统”^②。实际上，MCU 是在通用 CPU 基础上发展起来的，MCU 具有体积小、价格低、稳定可靠等优点，它的出现和迅猛发展，是控制系统领域的一场技术革命。MCU 以其较高的性能价格比、灵活性等特点，在现代控制系统中具有十分重要的地位。大部分嵌入式系统以 MCU 为核心进行

① MCU 的英文全称是 Microcontroller Unit。

② 详见《单片机与嵌入式系统应用》，2004 年第 1 期。

设计。MCU 从体系结构到指令系统都是按照嵌入式系统的应用特点专门设计的, 它能很好地满足应用系统的嵌入、面向测控对象、现场可靠运行等方面的要求。因此以 MCU 为核心的系统是应用最广的嵌入式系统。在实际应用时, 开发者可以根据具体要求与应用场合, 选用最佳型号的 MCU 嵌入到实际应用系统中。

在 MCU 出现之前, 人们必须用模拟电路、数字电路实现大部分计算与控制功能, 这样使得控制系统体积庞大, 易出故障。MCU 出现以后, 情况发生了变化, 系统中的大部分计算与控制功能由 MCU 的软件实现。其他电子线路成为 MCU 的外围接口电路, 承担着输入/输出与执行动作等功能, 而计算、比较与判断等原来必须用电路实现的功能, 可以用软件取代, 大大提高了系统的性能与稳定性, 这种控制技术称之为嵌入式控制技术。在嵌入式控制技术中, 核心是 MCU, 其他部分依此而展开。

1.1.3 嵌入式系统的特点

对于嵌入式系统特点, 不同学者也许有不同的说法。这里从与通用计算机对比的角度谈嵌入式系统的特点。

1) 嵌入式系统属于计算机系统, 但不单独以通用计算机的面目出现

嵌入式系统的本名叫做嵌入式计算机系统 (Embedded Computer System), 它不仅具有通用计算机的主要特点, 又具有自身特点。嵌入式系统也必须要有软件才能运行, 但其隐含在种类众多的具体产品中。同时, 通用计算机种类屈指可数, 而嵌入式系统不仅芯片种类繁多, 而且由于应用对象大小各异, 嵌入式系统作为控制核心, 已经融入到各个行业的产品之中。

2) 嵌入式系统开发需要专用工具和特殊方法

嵌入式系统不像通用计算机那样有了计算机系统就可以进行应用开发。在一般情况下, MCU 芯片本身不具备开发功能, 必须要有一套与相应芯片配套的开发工具和开发环境。这些工具和环境一般基于通用计算机上的软硬件设备以及各种逻辑分析仪、混合信号示波器等。开发时往往有主机和目标机的概念, 主机用于程序的开发, 目标机作为程序的执行机, 开发时需要交替结合进行。

3) 使用 MCU 设计嵌入式系统, 数据与程序空间采用不同存储介质

在通用计算机系统中, 程序存储在硬盘上。在实际运行时, 通过操作系统将要运行的程序从硬盘调入内存 (RAM), 运行中的程序、常数、变量均在 RAM 中; 而以 MCU 为核心的嵌入式系统, 其程序被固化到非易失性存储器中^①, 变量及堆栈使用 RAM 存储器。

4) 开发嵌入式系统涉及软件、硬件及应用领域的知识

嵌入式系统与硬件紧密相关, 嵌入式系统的开发需要硬件、软件协同设计、协同测试。同时, 由于嵌入式系统专用性很强, 通常是用在特定应用领域, 如嵌入在手机、冰箱、空调、各种机械设备、智能仪器仪表中起核心控制作用, 功能专用。因此, 进行嵌入式系统的开发, 还需要对领域知识有一定的理解。当然, 一个团队协作开发一个嵌入式产品, 其中各个成员可以扮演不同角色, 但对系统的整体理解与把握并相互协作, 有助于一个稳定可靠嵌入式产品的诞生。

5) 嵌入式系统的其他特点

除了以上特点之外, 嵌入式系统还具有其他方面的特点。

在资源方面: 嵌入式系统通常专用于某一特定应用领域, 其硬件资源不会像通用计算机那样

^① 目前, 非易失性存储器通常为 Flash 存储器, 特点见有关“Flash 存储器在线编程”章节。

丰富；在可靠性方面：嵌入式系统一般要求更高可靠性和稳定性；在实时性方面：相当多嵌入式系统有实时性要求；在成本方面：嵌入式系统通常极其关注成本；在功耗要求方面：一些嵌入式系统要求低功耗；在生命周期方面：嵌入式系统通常比通用计算机系统生命周期长，升级换代比通用计算机慢；在知识综合方面：嵌入式系统是将先进的计算机技术、半导体技术及电子技术与各个行业的具体应用相结合的产物，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。嵌入式系统的构成既有硬件又有软件，不仅包括应用软件，也可能包括系统软件；它既有数字电路又有模拟电路，其产品技术含量高，涉及多种学科，不容易开发，因此也不容易形成技术垄断。

这些特点决定了嵌入式系统的开发方法、开发难度、开发手段等，均不同于通用计算机，也不同于常规的电子产品。

◆ 1.2 嵌入式系统的知识体系、学习误区及学习建议

1.2.1 嵌入式系统的知识体系

嵌入式系统的应用范围可以粗略分为两大类：①电子系统的智能化（如工业控制、现代农业、家用电器、汽车电子、测控系统、数据采集等），这类应用也被称为微控制器领域；②计算机应用的延伸（如平板电脑、手机、电子图书等），这类应用也被称为应用处理器领域。在 ARM 产品系列中，ARM Cortex M 系列与 ARM Cortex R 系列适用于电子系统的智能化类应用，即微控制器领域；ARM Cortex A 系列适用于计算机应用的延伸，即应用处理器领域，本书以 ARM Cortex-M4 内核 MCU 为蓝本阐述嵌入式应用与设计方法。不论如何分类，嵌入式系统的技术基础是不变的，即要完成一个嵌入式系统产品设计，需要有硬件、软件及行业领域相关知识。

硬件主要有 MCU 的硬件最小系统、输入/输出外围电路、人机接口设计。软件设计有固化软件的设计，也可能含 PC 软件的设计。行业知识需要通过协作、交流与总结获得。

概括地说，学习以 MCU 为核心的嵌入式系统，需要以下软件硬件基础知识与实践训练：

- 硬件最小系统（包括电源、晶振、复位、写入调试器接口等）；
- 通用 I/O（开关量输入/输出，涉及各种二值量检测与控制）；
- A/D 转换（各种传感器信号的采集与处理，如红外、温度、光敏、超声波、方向等）；
- D/A 转换（对模拟量设备利用数字进行控制）；
- 通信（SCI、SPI、I2C、CAN、USB、嵌入式以太网、无线传感器网络等）；
- 显示（LED、LCD、触摸屏等）；
- 控制（控制各种设备，包括 PWM 等控制技术）；
- 数据处理（图形、图像、语音、视频等处理或识别）；
- 各种具体的应用。

事实上，万变不离其宗，任何应用都可以归入这几类。而应用中的硬件设计、软件设计、测试等都必须遵循嵌入式软件工程的方法、原理与基本原则。所以，嵌入式软件工程也是嵌入式系统知识体系的有机组成部分，只不过，它融于具体项目的开发过程之中。

以上实践训练涉及硬件基础、软件基础及相关领域知识。计算机语言、操作系统、开发环境等均是完成这些目的的工具。有些初学者，容易把工具的使用与所要达到的真正目的相混淆。例如，有的初学者，学了很长时间的嵌入式操作系统移植，而不进行实际嵌入式系统产品的开发，

到了最后,做不好一个嵌入式系统小产品,偏离了学习目标,甚至放弃了嵌入式系统领域。这就是进入了嵌入式系统学习误区的情况,下面对此做一些分析。

1.2.2 嵌入式系统的学习误区

关于嵌入式系统的学习方法,因学习经历、学习环境、学习目的、已有的知识基础等不同,可能在学习顺序、内容选择、实践方式等方面有所不同。但是,应该明确哪些是必备的基础知识,哪些应该先学,哪些应该后学;哪些必须通过实践才能获得的;哪些是与具体芯片无关的通用知识,哪些是与具体芯片或开发环境相关的知识。

由于微处理器与微控制器种类繁多,也可能由于不同公司、不同机构出于自身的利益,给出一些误导性宣传,特别是我国嵌入式微控制器制造技术的落后及其他相关情况,使得人们对微控制器的发展,在认识与理解上存在差异,导致一些初学者,进入了嵌入式系统的学习误区,浪费了宝贵的学习时间。下面分析初学者可能存在的几个误区。

1. 嵌入式系统学习误区 1——操作系统的困惑

如果说,学习嵌入式系统不是为了开发嵌入式应用产品,那就没有具体目标了,许多诸如学习方法问题也就不必谈了。实际上,这正是许多人想学,又不知从何开始学习的关键问题所在,不知道自己学习的具体目标。于是,看了一些培训广告,看了书店中书架上种类繁多的嵌入式系统的书籍,或上网以“嵌入式系统”为关键词进行查询,然后参加培训或看书,开始“学习起来”。对于有计算机阅历的人,往往选择一个嵌入式操作系统就开始学习了。不好听的比喻,有点像“瞎子摸大象”,只了解其一个侧面。这样如何能对嵌入式产品的开发过程有个全面了解呢?针对许多初学者选择“xxx 嵌入式操作系统+xxx 处理器”的嵌入式系统入门学习模式,我认为是不合适的。我的建议是:首先把嵌入式系统软件与硬件基础打好了,再根据实际应用需要,选择一种实时操作系统(RTOS)进行实践。要记住:RTOS 是开发某些嵌入式产品的辅助工具,是手段,不是目的。况且许多嵌入式产品并不需要 RTOS。所以,一开始就学习 RTOS,并不符合“由浅入深、循序渐进”的学习规律。

另外,即使学习 RTOS,也要认识到 RTOS 种类繁多,实际使用何种 RTOS,一般需要工作单位确定。基础阶段主要学习 RTOS 的基本原理与在 RTOS 之上的软件开发方法,而不是学习如何设计 RTOS。以开发实际嵌入式产品为目标的学习者,不要把过多的精力花在设计或移植 RTOS 上面。正如很多人使用 Windows 操作系统,而设计 Windows 操作系统只有 Microsoft。许多人“研究”Linux,但不使用它,浪费时间了,人的精力是有限的,所以学习必须有所选择。

2. 嵌入式系统学习误区 2——硬件与软件的困惑

以 MCU 为核心的嵌入式技术的知识体系必须通过具体的 MCU 来体现、实践与训练。但是,选择任何型号的 MCU,其芯片相关的知识只占知识体系的 20%左右,80%左右是通用知识。但是这 80%的通用知识,必须通过具体实践才能掌握,所以学习嵌入式技术要选择一个系列的 MCU。但不论如何,嵌入式系统均含有硬件与软件两大部分,它们之间的关系如何呢?

有些学者仅从电子角度认识嵌入式系统。认为“嵌入式系统=MCU 硬件系统+小程序”。这些学者,大多具有良好的电子技术基础知识。而实际情况是,早期 MCU 内部 RAM 小、程序存储器外接,需要外扩各种 I/O,没有像现在这样 USB、嵌入式以太网等较复杂的接口,因此,程序占总设计量小于 50%,使人们认为嵌入式系统(MCU)是“电子系统”,以硬件为主、程序为辅。但是,随着 MCU 制造技术的发展,不仅 MCU 内部 RAM 越来越大,Flash 进入 MCU 内部改变了传统的嵌入式系统开发与调试方式,固件程序可以被更方便地调试与在线升级,许多情况与开