



卓越工程师培养计划

■ 嵌入式系统 ■

<http://www.phei.com.cn>

陈志旺 等 编著



STM32

嵌入式微控制器 快速上手 (第2版)



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

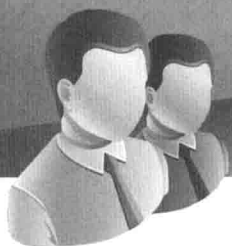


卓越工程师培养计划

■ 嵌入式系统 ■

<http://www.phei.com.cn>

■ 陈志旺 等 编著 ■



STM32

嵌入式微控制器 快速上手 (第2版)

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书介绍了意法半导体 (STMicroelectronics, ST) 公司的 32 位基于 ARM Cortex - M3 内核的 STM32 单片机原理与实践。本书以培养学生的动手能力和增强学生的工程素养为目的, 按照项目驱动的思路展开教学与实践学习, 以自制的开发板上的程序为实例, 将 STM32 单片机的外围引脚特性、内部结构原理、片上外设资源、开发设计方法和应用软件编程、 $\mu\text{C}/\text{OS} - \text{II}$ 操作系统原理及应用等知识传授给读者。

本书适合从事自动控制、智能仪表、电力电子、机电一体化等系统开发的工程技术人员阅读使用, 也可作为高等学校相关专业的“嵌入式系统原理与应用”、“基于 ARM Cortex 内核的单片机系统开发”等课程的教学用书, 还可作为 ARM 相关应用与培训课程的参考用书。

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

STM32 嵌入式微控制器快速上手/陈志旺等编著. —2 版. —北京: 电子工业出版社, 2014. 5
(卓越工程师培养计划)

ISBN 978 - 7 - 121 - 22956 - 5

I. ① S… II. ① 陈… III. ① 微控制器 IV. ① TP332. 3

中国版本图书馆 CIP 数据核字 (2014) 第 074384 号

责任编辑: 张 剑 (zhang@phei.com.cn)

印 刷: 北京京科印刷有限公司

装 订: 三河市皇庄路通装订厂

出版发行: 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: 24 字数: 614 千字

印 次: 2014 年 5 月第 1 次印刷

印 数: 3 000 册 定价: 59.80 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 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

第 2 版前言

1. 本书定位

本书定位为有 51 单片机基础的读者学习 STM32 微控制器的“入门”教材，即本科电类专业，51 单片机课程后的“嵌入式系统原理及应用”课程可选用的教材。本书的目的是深化强化嵌入式系统的基本概念，因此写法中侧重基本原理的阐述，侧重与 51 单片机对比（外设主要选择了 GPIO、EXTI、USART、TIM，分别与 51 课程中的 I/O 接口、外部中断、串口、定时/计数器对应）。在 51 单片机的知识结构中，这 4 个重要外设是按照如下线索组织的：功能→硬件→寄存器，即某个具体的功能是通过硬件实现的，寄存器是硬件与程序员的接口，程序员利用汇编语言或 C 语言通过寄存器控制硬件。而对于 STM32，线索增加了一层，即功能→硬件→寄存器→固件库，程序员调用固件库函数控制硬件。其实细读固件库函数代码后可以发现，还是通过寄存器控制硬件，只是经过固件库封装后，代码更加易读、易用。因此从寄存器到固件库的学习中，读者一定要清晰二者之间的联系。掌握好本书中这 4 个外设接口，就可以结合 Cortex - M3 权威指南、STM32 技术参考手册和固件库手册、STM32 具体型号的数据手册，进行嵌入式开发技能的进一步提升。

2. 嵌入式系统层次模型

与 OSI 网络 7 层模型类似，嵌入式系统也有层次模型，并且在下述具体的知识点中有不同的体现。

第 1 页图 1 - 1；

第 83 页图 4 - 2 和图 4 - 4；

第 85 页图 4 - 5；

第 89 页图 4 - 8；

第 263 页图 11 - 9 和图 11 - 10；

第 269 页图 11 - 12；

第 270 页图 11 - 13。

在上述结构中，第 1 页的图 1 - 1 所示的是一个抽象模型，其他都是它的具体实现。层次模型是研究复杂问题一个有效方法。

3. 章节设置说明

本书篇章布局从理论体系上讲并不十分严。例如，第 2 章 Cortex - M3 体系结构中包含 STM32 存储器映射；NVIC 的优先级，应属于第 2 章，但从应用角度来考虑放到第 5 章 EXTI 中；SysTick 也应属于第 2 章，但从功能角度来考虑放到第 8 章定时器部分。这样设置的目的是从应用的角度来考虑的，这也与嵌入式系统“用中学”的学习理念一致，但读者一定要清晰了解相关概念。

本书在第 1 版的基础上增加了第 13 章，这是一个以 STM32 为核心的嵌入式系统应用实

例。本书第1~2章、第7~13章、附录A~D由燕山大学陈志旺编写，第3章由燕山大学刘宝华编写，第4章由燕山大学王荣彦编写，第5章由燕山大学程淑红编写，第6章由燕山大学吕宏诗编写，参加本书编写的还有李萌、薛佳伟、王敬、吴晨、李晓、李坤、赵春媛、刘砚、赵晓娟和李江艳。全书由陈志旺统稿。书中引用了一些网上文献，无法逐一注明出处，在此向原作者表示感谢！

由于笔者水平有限，书中难免存在错误与不妥之处，欢迎读者朋友不吝赐教！作者的E-mail地址：czwaaron@ysu.edu.cn。

编著者

第 1 版前言

嵌入式系统属于一个交叉学科，它涵盖了微电子技术、电子信息技术、计算机软件和硬件等多项技术领域，覆盖面广。目前嵌入式系统发展很快，很多软/硬件技术出现时间不长，掌握这些新技术的人相对较少。很多高校专业划分过细，难以跟上市场变化的步伐，与实际工程项目脱节严重，仍沿用应试教育的教学方式，理论知识讲授过多，动手环节薄弱，理论联系实际能力较差，学生参与社会实践较少，不了解社会需求。嵌入式教学需要相应的嵌入式开发板和软件，需要有经验的人进行开发流程指导，这在目前的高校中是很难实现的。针对上述问题，我们组织多年在一线授课的教师根据教学经验编写了此书。

1. 对教师的建议

把握“三个统一学单片机”的教学理论。

1) **一般与特殊的统一** 目前电类课程中关于微机方面的课程很多，如《计算机应用基础》、《微机原理》、《单片机原理及应用》、《嵌入式系统原理及应用》、《可编程控制器 PLC》等，这些课程间内容既有重复又有联系，因为以奔腾芯片为核心的 PC、以 8086 为核心的第一代 PC、51 单片机、以 ARM 为核心的嵌入式系统、PLC 都有相同的遗传基因（微机原理）。因此，任课老师应熟悉上述课程，在课堂讲授中将其融会贯通起来，深入挖掘课程的共性，即微机的基本原理，然后引导学生在学习中侧重其差异，这样不仅可以提高学习效率，还可以启发学生思考。

2) **硬件与软件的统一** 通过多年的教学实践我们发现，“软件通过控制寄存器控制硬件”是学生理解的难点，其原因可能是平时同学操作 PC 直接通过 Windows 界面达到驱动硬件的目的，没有用软件驱动过底层硬件。因此在介绍嵌入式系统时，最好从硬件控制寄存器程序讲起，让学生对底层原理有个认识。对于 STM32，本书在第 5 章也揭示了控制寄存器和库函数的相关性。另外，在参考文献 [2] 中，我们将汇编语言依赖硬件的特性深入挖掘，课堂效果较好，推荐采用。

3) **内部结构与外部引脚的统一** 将内部结构、外部引脚、系统功能、指令集统一起来，只有这样才能做到“庖丁解牛，游刃有余”。

2. 对学生的建议

1) **重视实践** 工程师解决问题的能力只有从实践中才能获得。从实践经验中归纳出共性的知识，然后再将这些知识重新应用到新的实践中去，这也是当今的大学生要在未来的实际工作中必须采取的学习和工作方法。只有做到了这一点，才能真正实践以工作为导向的理念：实践、归纳、总结和再实践。学好嵌入式系统，实践必不可少，一定要选一块和微处理器型号对应的开发板，创建一个良好的平台和环境，边实践边学习，尽量弄清其内在原理。硬件开发板的价格不必太高，最好能有自己动手的空间。深入理解 STM32 的硬件最小系统，对 I/O 口、串行通信、键盘、LED、LCD、SPI、I²C、PWM、A/D（包括一些传感器）、D/A 等实验逐个实践，逐步理解，再动手制做一个实际的小系统，底层硬件基础就有了。各个硬

件模块驱动程序的编写是嵌入式系统的必备基础。在编程中，主要针对 main.c 和 stm32f10x_it.c 两个文件，其他定义硬件控制寄存器的文件 stm32f10x_map.h、硬件初始化结构体的文件 stm32f10x_xxx.h 和 stm32f10x_xxx.c 也要仔细研读，将软件和硬件联系起来学习，会加深对硬件的理解。把书上的例程亲手输入到计算机中进行实践是很有必要的，一定要多上机操作。程序是抽象的，有时程序能看懂，但自己却不一定能编出来；而有时虽然程序没看懂，但若经常着手去编，就会非常熟悉该程序应该怎样去处理。要把在书中看到的有意义的例子举一反三；经常回顾自己以前写过的程序，并尝试重写，把自己学到的新知识运用进去，温故而知新；当程序写到一半时却发现自己用的方法烦琐，不要马上停手，应尽快将余下的部分粗略地完成，以保证设计的完整性，然后分析自己的错误并重新设计和编写；遇到问题时，不要马上向其他人求教，以免养成依赖性，要学会自己去解决问题。看到一个编程要求后，首先要在头脑中有一个大体的轮廓，独立构思，不要看参考提示，只有这样才能达到真正的训练目的，才会逐步把思路培养出来；保存好写过的所有的程序——那是最好的积累之一。学习的过程中要经过思考加工，去粗取精，抓本质和精华，仔细研读并尝试修改别人的例程代码，真正将新知识应用到实践中，理解后融入到自己的知识体系中。本书每个知识点后都对应一个开发板实例，推荐以项目开发的方式进行学习，将学习成果变成自己的作品。

2) 重视官方文档 学习时，应关注两个比较重要的文档：《STM32F103 × × × 参考手册》和《STM32 固件库使用手册》。

阅读《STM32F103 × × × 参考手册》时，不需要全部阅读，但是前几章必须重点阅读，包括存储器和总线架构，电源控制，备份寄存器，复位和时钟控制，通用和复用功能 I/O，中断和定时器等。后续章节讲述的是具体的功能模块设计，如果要用到哪个模块，就可以去阅读相应的模块。

阅读《STM32 固件库使用手册》的主要目的是为了简化编程。STM32 提供了一个非常好的固件函数库，只需调用即可。阅读《STM32 固件库使用手册》时，前面几章也是必须阅读的。比如，第 1 章文档和库规范中的命名规则，编码规则，这些都是需要注意的。第 2 章是最关键的，描述了固件库的架构，以及如何使用固件库等。有了这些基础，就可以借助固件库写出自己的代码了。第 4 章以后的章节都是描述某个模块有什么函数，每个函数如何使用等，可以根据需要来阅读。建议对 GPIO 库函数、中断部分库函数、复位和时钟设置的库函数等内容要重点阅读。

无论何时，官方手册是最好的老师和帮手，千万不要因为它的枯燥乏味而将其束之高阁。使用任何外设前，都必须仔细看参考手册和使用手册。

3) 重视交流 在网上建个交流平台，或者开博，或者社区交流，把自己解决的问题和与他人分享，这样通常会让读者个人的研究更具可行性和更深入地被了解；构建或参与技术圈子，一个好的圈子通常会给读者带来钻研和共同提高的激情，或者说是一个良性的竞争环境。

3. 本书特点

在讲述具体内容时，本书各章节均以我们开发的 STM32 开发板为硬件教具，每章均会提供一个设计任务实例，以任务为驱动，通过“学中做、做中学”，即 DIY（Do It Yourself）和 LBD（Learning By Doing）的方式，介绍和讲解所需要用到的新知识、新技能，按照认识

的规律学习和掌握基于 ARM Cortex – M3 内核的 STM32 嵌入式微控制器技术及其应用编程，尽量避免纯理论性描述带给读者的枯燥感。有别于数据手册式的教材，本书并没有面面俱到地谈及 Cortex – M3 的技术细节，各个章节也没有繁冗的寄存器说明（参见 ST 公司网页上的数据手册或本书配套资料），而且涉及微机原理的相关知识都在每章“本章前导知识”中给出（可阅读参考文献 [2]），每章的例程只给出关键代码，这样做的目的旨在突出重点。本书写作过程中，也注意了软、硬件的结合，将 STM32 的内部结构、控制寄存器和库函数对应结构体，还有功能特点和初始化设置等结合起来，揭示软、硬件之间的联系，使读者能够对 STM32 应用“快速上手”。本书不仅是教给大家 STM32 基础知识，更重要的是介绍嵌入式系统的学习方法，启发读者的创意思维。

本书受全国教育科学规划 2010 年度教育部重点课题（课题批准号：GKA103004）的子课题（立项编号：GKA10105）资助。第 1、2、7、8、9、10、11、12 章及附录由燕山大学陈志旺编写，第 3 章由燕山大学刘宝华编写，第 4 章由燕山大学王荣彦编写，第 5 章由燕山大学程淑红编写，第 6 章由燕山大学吕宏诗编写。全书由陈志旺统稿。参加本书编写的还有李萌、薛佳伟、王敬、吴晨、李晓、李坤、赵春媛、刘砚、赵晓娟和李江艳。书中引用了一些网上文献，无法一一注明出处，在此向原作者表示感谢！

由于笔者水平有限，书中难免存在错误与不妥之处，欢迎读者朋友不吝赐教！来信可与如下邮箱联系：czwaaron@ysu.edu.cn。博客：<http://blog.sina.com.cn/gksupermarket>，本书的进一步资料会在博客上更新。

编著者

目 录

第 1 章 嵌入式系统概述	1
1.1 嵌入式系统简介	1
1.1.1 嵌入式系统定义	1
1.1.2 嵌入式系统特点	2
1.1.3 嵌入式系统分类	4
1.1.4 嵌入式系统发展	7
1.2 ARM 体系结构及微处理器系列	10
1.2.1 ARM 公司简介	10
1.2.2 ARM 体系结构简介	13
1.3 Cortex - M 系列处理器简介	17
1.4 STM32 系列微控制器简介	21
1.5 STM32 教学开发板	26
第 2 章 Cortex - M3 体系结构	27
2.1 CM3 微处理器核结构	27
2.2 处理器的工作模式及状态	29
2.3 寄存器	30
2.4 总线接口	34
2.5 存储器的组织与映射	36
2.5.1 存储器格式	36
2.5.2 存储器层次结构	36
2.5.3 CM3 存储器组织	38
2.5.4 STM32 存储器映射	39
2.5.5 位绑定操作	43
2.6 指令集	46
2.6.1 ARM 指令集	46
2.6.2 Thumb 指令集	47
2.6.3 Thumb - 2 指令集	48
2.7 流水线	50
2.8 异常和中断	51
2.9 存储器保护单元 MPU	56
2.10 STM32 微控制器概述	57
2.10.1 STM32 命名	57
2.10.2 STM32 内部资源	58

第 3 章 STM32 最小系统	61
3.1 电源电路	61
3.1.1 供电方案	61
3.1.2 电源管理器	63
3.1.3 低功耗模式	63
3.2 时钟电路	65
3.2.1 HSE 时钟和 HSI 时钟	66
3.2.2 PLL	67
3.2.3 LSE 时钟和 LSI 时钟	67
3.2.4 系统时钟 SYSCLK	68
3.2.5 RCC 寄存器	69
3.3 复位电路	72
3.4 STM32 启动	74
3.5 程序下载电路	79
3.6 STM32 的最小系统	81
第 4 章 STM32 程序设计	82
4.1 嵌入式软件层次结构	82
4.2 Cortex 微控制器软件接口标准	84
4.3 FWLib 固件库	87
4.3.1 STM32 标准外设库	87
4.3.2 固件库命名规则	90
4.3.3 数据类型和结构	91
4.3.4 固件库的应用	94
4.4 嵌入式 C 程序特点	97
4.5 开发环境简介	100
第 5 章 GPIO 原理及应用	102
5.1 GPIO 的硬件结构和功能	102
5.1.1 GPIO 硬件结构	102
5.1.2 复用功能	103
5.1.3 GPIO 输入功能	103
5.1.4 GPIO 输出功能	104
5.1.5 GPIO 速度选择	106
5.1.6 钳位功能	107
5.2 GPIO 寄存器	107
5.3 GPIO 库函数	111
5.4 库函数和寄存器的关系	114
5.5 应用实例	121

第 6 章 EXTI 原理及应用 127

6.1 STM32 中断通道 127

6.2 STM32 中断的过程 131

6.3 NVIC 硬件结构及软件配置 132

6.3.1 NVIC 硬件结构 132

6.3.2 STM32 中断优先级 133

6.3.3 中断向量表 134

6.3.4 NVIC 寄存器 135

6.3.5 NVIC 库结构 140

6.4 EXTI 硬件结构及软件配置 140

6.4.1 EXTI 硬件结构 140

6.4.2 中断及事件 141

6.4.3 EXTI 中断通道和中断源 142

6.4.4 EXTI 寄存器 142

6.4.5 EXTI 库函数 144

6.5 应用实例 145

6.5.1 按键中断 145

6.5.2 中断嵌套案例 1 149

6.5.3 中断嵌套案例 2 151

第 7 章 USART 原理及应用 155

7.1 端口重映射 155

7.2 USART 功能和结构 156

7.2.1 USART 功能 157

7.2.2 USART 结构 158

7.3 USART 帧格式 160

7.4 波特率设置 162

7.5 硬件流控制 163

7.6 USART 中断请求 165

7.7 USART 寄存器 166

7.8 USART 库函数 167

7.9 USART 应用实例 169

7.9.1 直接传送方式 169

7.9.2 中断传送方式 173

7.9.3 串口 Echo 回应程序 174

7.9.4 利用 printf() 的串口编程 174

第 8 章 定时器原理及应用 177

8.1 STM32 定时器概述 177

8.2	通用定时器 TIMx 功能	178
8.3	通用定时器 TIMx 结构	178
8.3.1	时钟源选择	180
8.3.2	时基单元	182
8.3.3	捕获和比较通道	184
8.3.4	计数器模式	186
8.3.5	定时时间的计算	187
8.3.6	定时器中断	188
8.4	通用定时器 TIMx 寄存器	189
8.5	通用定时器 TIMx 库函数	192
8.6	TIM2 应用实例	192
8.6.1	秒表	192
8.6.2	输出比较案例 1	195
8.6.3	输出比较案例 2	199
8.6.4	PWM 输出	202
8.6.5	PWM 输入捕获	204
8.7	RTC 的功能及结构	207
8.7.1	RTC 的基本功能	208
8.7.2	RTC 的内部结构	208
8.8	RTC 控制寄存器	209
8.9	备份寄存器	211
8.10	电源控制寄存器	214
8.11	RTC 相关的 RCC 寄存器	215
8.12	RTC 应用实例	215
8.13	系统时钟 SysTick 简介	220
8.14	SysTick 寄存器	221
8.14.1	控制及状态寄存器 (SYSTICKCSR)	221
8.14.2	重载寄存器 (SYSTICKRVR)	221
8.14.3	当前值寄存器 (SYSTICKCVR)	222
8.14.4	校准值寄存器 (SYSTICKCALVR)	222
8.15	SysTick 应用实例	223

第 9 章 DMA 原理及应用 225

9.1	DMA 简介	225
9.2	DMA 的功能及结构	228
9.2.1	DMA 的功能	228
9.2.2	DMA 结构	229
9.3	DMA 寄存器	230
9.4	DMA 库函数	232

第 10 章	ADC 原理及应用	235
10.1	ADC 的功能及结构	235
10.2	ADC 的工作模式	237
10.3	数据对齐	240
10.4	ADC 中断	241
10.5	ADC 寄存器	241
10.6	ADC 库函数	243
10.7	应用实例	245
第 11 章	$\mu\text{C}/\text{OS}-\text{II}$ 嵌入式操作系统基础	250
11.1	操作系统的功能	250
11.2	操作系统的基本概念	252
11.2.1	进程和线程	252
11.2.2	实时操作系统 RTOS	254
11.2.3	其他概念	256
11.2.4	应用程序在操作系统上的执行过程	262
11.3	操作系统的分类	263
11.3.1	单体结构	263
11.3.2	层次结构	263
11.3.3	微内核结构	264
11.4	$\mu\text{C}/\text{OS}-\text{II}$ 简介	264
11.4.1	$\mu\text{C}/\text{OS}-\text{II}$ 的主要特点	264
11.4.2	$\mu\text{C}/\text{OS}-\text{II}$ 工作原理	265
11.4.3	$\mu\text{C}/\text{OS}-\text{II}$ 的程序设计模式	266
11.5	$\mu\text{C}/\text{OS}-\text{II}$ 移植	268
11.5.1	移植条件	269
11.5.2	移植步骤	270
11.5.3	内核头文件 (OS_CPU.H)	272
11.5.4	与处理器相关的汇编代码 (OS_CPU_A.ASM)	273
11.5.5	与 CPU 相关的 C 函数和钩子函数 (OS_CPU_C.C)	276
第 12 章	$\mu\text{C}/\text{OS}-\text{II}$ 的内核机制	279
12.1	$\mu\text{C}/\text{OS}-\text{II}$ 内核结构	279
12.1.1	$\mu\text{C}/\text{OS}-\text{II}$ 的任务	279
12.1.2	临界代码	281
12.1.3	任务控制块	282
12.1.4	就绪表	283
12.1.5	任务的调度	284
12.1.6	中断处理	288

12.1.7	时钟节拍	289
12.1.8	任务的初始化	291
12.1.9	任务的启动	293
12.2	$\mu\text{C}/\text{OS}-\text{II}$ 的任务管理	294
12.2.1	创建任务	294
12.2.2	删除任务	298
12.2.3	请求删除任务	300
12.2.4	改变任务优先级	302
12.2.5	挂起任务	305
12.2.6	恢复任务	307
12.2.7	任务调度实例	308
12.3	$\mu\text{C}/\text{OS}-\text{II}$ 的时间管理	315
12.3.1	延时函数	315
12.3.2	恢复延时任务	317
12.3.3	系统时间	318
12.4	任务间的通信与同步	319
12.4.1	事件控制块	319
12.4.2	信号量	323
12.4.3	信号量实例	327
第 13 章	嵌入式系统综合设计实例	331
13.1	嵌入式系统开发过程	331
13.2	自平衡小车基本功能	333
13.3	硬件结构	334
13.3.1	电气控制系统整体结构	334
13.3.2	加速度计	335
13.3.3	陀螺仪	335
13.4	控制算法设计	335
13.4.1	角度检测算法设计	335
13.4.2	运动控制算法设计	336
附录 A	嵌入式系统常用缩写和关于端口读/写的缩写表示	339
附录 B	Cortex - M3 指令清单	346
附录 C	51 单片机与 STM 32 微控制器的比较	353
C.1	硬件：寄存器	353
C.2	硬件：存储器空间	353
C.3	硬件：堆栈	354
C.4	硬件：外设	355

C.5 硬件：异常和中断	356
C.6 软件：数据类型	357
C.7 软件：浮点	358
C.8 软件：中断服务程序	359
C.9 软件：非对齐数据	359
C.10 软件：故障异常	360
C.11 软件：设备驱动程序和 CMSIS	361
C.12 软件：混用 C 语言和汇编程序	362
C.13 其他比较	363
附录 D STM32 实验板原理图	364
参考文献	365

第1章 嵌入式系统概述



1.1 嵌入式系统简介

1.1.1 嵌入式系统定义

嵌入式系统通常定义为以应用为中心，以计算机技术为基础，软/硬件可剪裁，对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。嵌入式系统主要由嵌入式微处理器、外围硬件设备、嵌入式操作系统及用户应用软件等部分组成，其分层结构如图 1-1 所示。嵌入式系统因其通常都被嵌入在主要设备之中，故此得名。



图 1-1 嵌入式系统分层结构

通常的嵌入式系统定义中有 4 个基本要点。

【应用中心的特点】 嵌入式系统是嵌入到一个设备或一个过程中的计算机系统，与外部环境密切相关。这些设备或过程对嵌入式系统会有不同的要求。例如，消费电子产品的嵌入式软件与工业控制的嵌入式软件差别非常大，特别是响应时间，它们有些要求时限长，有些要求时限短，有些要求严格，有些要求宽松，这些不同的要求体现了嵌入式系统面向应用的多样化。这个特点可以从用户方和开发方两个方面考虑。

- ⑤ 用户方要求：操作简单，用户打开电源即可直接使用其功能，无需二次开发或仅需少量配置操作；专门完成一个或多个任务；对体积、功耗、价格和开发周期有要求；实时与环境交互；安全可靠，软硬件的错误不能使系统崩溃。
- ⑤ 开发方要求：软件与硬件协同并行开发；多种多样的微处理器；实时操作系统的多样性；与 PC 相比，可利用系统资源很少；应用支持很少；要求特殊的开发工具；调试很容易。

【计算机系统的特点】 嵌入式系统必须是能满足对象系统控制要求的计算机系统, 这里的计算机也包括运算器、控制器、存储器和 I/O 接口。嵌入式系统的最基本支撑技术, 包括集成电路设计技术、系统结构技术、传感与检测技术、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, 以及特定应用领域的数据分析、信号处理和优化技术等。所以本质上嵌入式系统也是各种计算机技术的集大成者。

【软/硬件可裁剪的特点】 嵌入式系统针对的应用场景很多, 因此设计指标要求 (功能、可靠性、成本、体积、功耗等) 差异极大, 实现上很难有一套方案满足所有的系统要求。所以根据需求的不同, 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。

【专用性的特点】 嵌入式系统的应用场合对可靠性、实时性、低功耗要求较高。例如, 它对实时多任务有很强的支持能力, 能完成多任务, 并且有较短的中断响应时间, 从而使内部的代码和实时内核的可执行时间减少到最低限度; 它具有功能很强的存储区保护功能, 这是由于嵌入式系统的软件结构已经模块化, 而为了避免在软件模块之间出现错误的交叉作用, 需要设计强大的存储区保护功能, 同时也有利于软件诊断; 嵌入式微处理器必须功耗很低, 尤其是无线通信设备中靠电池供电的嵌入式系统更是如此。这些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。它并不强调系统通用性 (20 世纪 80 年代的微型计算机技术特性之一即是通用性)。这种专用性通常导致嵌入式系统是一个软、硬件紧密耦合的系统, 因为只有这样才能更有效地提高整个系统的可靠性并降低成本。

因此, 可以说嵌入式系统是计算机技术、微电子技术与行业技术相结合的产物, 是一个技术密集、不断创新的知识集成系统, 也是一个面向特定应用的软、硬件综合体。

嵌入式系统的其他定义如下所述。

(1) IEEE (国际电气和电子工程师协会) 对嵌入式系统定义: 嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置”。

(2) 中国微机学会对嵌入式系统定义: 嵌入式系统是以嵌入式应用为目的的计算机系统, 可以分为芯片级、板卡级、系统级。芯片级嵌入的是含程序或算法的处理器; 板卡级嵌入的是系统中的某个核心模块板; 系统级嵌入的是主计算机系统。

(3) 国内有学者认为, 将一套计算机控制系统嵌入到已具有某种完整的特定功能的 (或者将会具备完整功能的) 系统内 (如各种机械设备), 以实现对原有系统的计算机控制, 这个新系统叫做嵌入式系统。它通常由特定功能模块组成, 主要由嵌入式微处理器、外围硬件设备、嵌入式操作系统及用户应用软件等部分组成。

上述定义 (3) 将通用 PC 系统也囊括进嵌入式系统中, 因为随着嵌入式微控制器性能的提高, 已可以取代 PC 实现其相应功能, 这也是嵌入式系统的发展趋势。施乐公司 Palo Alto 研究中心主任 Mark Weiser 认为: “从长远来看, PC 和计算机工作站将衰落, 因为计算机变得无处不在, 如在墙里、在手腕上、在手写电脑中 (像手写纸一样) 等, 随用随取、伸手可及”。无处不在的计算机就是嵌入式系统。但本书嵌入式系统仅指以微控制器芯片为核心的系统。

1.1.2 嵌入式系统特点

通用 PC 系统 (如图 1-2 所示) 与嵌入式系统 (如图 1-3 所示) 的对比见表 1-1。